AD NUMBER

AD849017

LIMITATION CHANGES

TO:

Approved for public release; distribution is unlimited.

FROM:

Distribution authorized to U.S. Gov't. agencies only; Test and Evaluation; JAN 1969. Other requests shall be referred to Air Force Flight Dynamics Laboratory, ATTN: FDD, Wright-Patterson AFB, OH 45433.

AUTHORITY

AFFDL ltr dtd 29 Oct 1973

AFFDL-TR-68-43
PART II

RANDOM-VIBRATION ANALYSIS SYSTEM FOR COMPLEX STRUCTURES

PART II. COMPUTER PROGRAM DESCRIPTION

K. TSURUSAKI
F. S. WALLACE
The Boeing Company

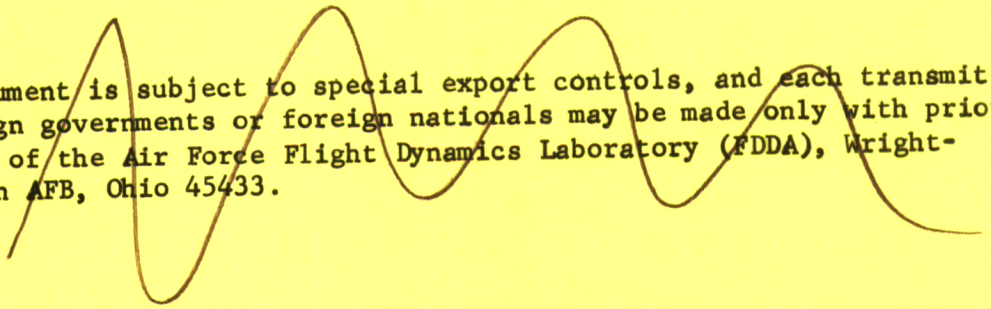
TECHNICAL REPORT AFFDL-TR-68-43, PART II

JANUARY 1969

AIR FORCE FLIGHT DYNAMICS LABORATORY
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.



This document is subject to special export controls, and each transmittal to foreign governments or foreign nationals may be made only with prior approval of the Air Force Flight Dynamics Laboratory (FDDA), Wright-Patterson AFB, Ohio 45433.

Approved for public release; distribution unlimited.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

100 - March 1969 - CO455- 67-1499

AFFDL-TR-68-43
PART II

RANDOM-VIBRATION ANALYSIS SYSTEM FOR COMPLEX STRUCTURES

PART II. COMPUTER PROGRAM DESCRIPTION

K. TSURUSAKI
F. S. WALLACE

FOREWORD

The computer program reported herein was prepared by The Boeing Company for the Aero-Acoustics Branch, Vehicle Dynamics Division, Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio, under contract AF 33(615)-5155. The study demonstrates the application of finite-element matrix methods in determining the responses and fatigue life of complex panels excited by random pressure fluctuations. The program is part of a continuing effort to establish tolerance levels and design criteria for sonic fatigue prevention under the exploratory development program of the Air Force Systems Command. The effort was conducted under project 1471 "Aero-Acoustic Problems," task 147101 "Sonic Fatigue." Mr. D. L. Smith and Mr. M. C. Eifert of the Aero-Acoustics Branch were the task engineers.

The period covered by this effort is July 1966 through June 1968. This report is AFFDL-TR-68-43 "Random-Vibration Analysis System for Complex Structures," Part II "Computer Program Description" and is one of four documents prepared under contract AF 33(615)-5155. One document is Part I of this report entitled, "Engineering User's Guide." The other reports are AFFDL-TR-67-81, "A Finite-Element Analysis of Simple Panel Response to Turbulent Boundary Layers," and AFFDL-TR-68-44 "Finite-Element Analysis of Complex Panel Response to Random Loads." The Boeing Company's document number for this report is D6-23145 Part II.

The research was conducted by Loyd D. Jacobs and Dr. Dennis R. Lagerquist of the Structural Dynamics Staff of The Boeing Company's Commercial Airplane Division in Renton, Washington. The principal programming effort was conducted by Frank S. Wallace and Kiyoharu Tsurusaki of The Boeing Company's computing department.

Many other Boeing personnel contributed significantly to the project by providing major modifications to existing computer programs. They are R. D. Palm for extensive modifications to the matrix structural generation program, H. B. Noonchester for major modification and improvement in the eigenvalue/eigenvector program, and L. Anderson for modifications to the matrix manipulation module.

This report was submitted by the authors in July 1968.

This report has been reviewed and is approved.

Walter J. Mykytow

WALTER J. MYKYTOW
Asst. for Research & Technology
Vehicle Dynamics Division
AF Flight Dynamics Laboratory

ABSTRACT

A programming description is presented for a computer program developed to aid in the design of sonic-fatigue-resistant aircraft structures. The computer program is written in FORTRAN IV and MAP for the IBM 7094 Mod II. The program employs matrix structural analysis methods to calculate statistical measurements of response (deflection and stress) for complex structure subjected to pressure loads random in both time and space. The program is organized into two phases, each performed separately. The phases are further organized in modular form for ease of maintenance and/or modification.

CONTENTS

CONTENTS (Concluded)

ILLUSTRATIONS

NOMENCLATURE

$[C]$	damping matrix
$[C_F(\omega)]$	force co-PSD matrix ($lb^2 \cdot sec$)
$[F]$	flexibility matrix
r	eigenvalue
$[F_r]$	reduced flexibility matrix
F_x, F_y, F_z	forces in the x , y , and z directions, respectively (lb)
g	structural damping coefficient
$[H(i\omega)]$	admittance matrix
i	$\sqrt{-1}$
$[K]$	stiffness matrix
$[K_{cc}]$	partition of stiffness matrix corresponding to constrained freedoms
$[K_{fc}]$, $[K_{cf}]$	partition of stiffness matrix relating free and constrained nodes
$[K_{ff}]$	partition of stiffness matrix corresponding to unrestrained freedoms
$[K_r]$	reduced-stiffness matrix
$[K_{11}]$, $[K_{12}]$, $[K_{21}]$, $[K_{22}]$	partitions of stiffness matrix
K	number of modal cross products in analysis
$[M]$	mass matrix
$\{M_j\}$	generalized mass
M_{xy}	plate-element twisting moment

M_x, M_y, M_z	bending moments about x , y , and z axes, respectively (in. -lb)
m	number of normal modes
N	number of kinematic degrees of freedom
$[Q_F(\omega)]$	force quad-PSD matrix ($lb^2 \cdot sec$)
[S]	matrix of stress-deflection relationships
$[S_1], [S_2]$	partitions of stress matrix
$[\overline{\delta_q \delta_r}]$	deflection covariance matrix ($in.^2$)
Δx, Δy, Δz	deflections in x , y , and z directions , respectively (in.)
λ, μ	proportionality factors for stiffness and mass-proportional damping, respectively
$[\overline{\sigma_s \sigma_t}]$	stress covariance matrix
τ_x, τ_y, τ_{xy}	plate element shearing forces (lb/in.)
$[\Phi_\sigma(i\omega)]$	stress cross-PSD matrix
$\{\phi_{(j)}\}$	eigenvector
Θ_x, Θ_y, Θ_z	rotations about x , y , and z axes, respectively (rad)
ω	angular frequency (rad/sec)
$\lceil \rfloor$	diagonal matrix
$[\ ]^T$	transpose of matrix
$[\Phi_\delta(i\omega)]$	deflection cross-PSD matrix ($in.^2$)

I

INTRODUCTION

RANVIB is a computer program developed to aid in sonic fatigue analysis. The program employs matrix methods to calculate statistical measurements of response (deflection and stress) for complex structure subjected to random sound fields. The RANVIB system is written in FORTRAN IV and MAP languages for use on an IBM 7094 Mod II under the IBSYS Version 13 operating system.

The computer program report is in two Parts:

(1) Part I—Engineering User's Guide

(2) Part II—Computer Program Description

Part I is a guide for an engineer's use of RANVIB. Part II describes the RANVIB computer program and is intended primarily for the programmer/analyst responsible for the implementation and subsequent maintenance of the system.

Development of the theory of this program and its application to specific problems are presented in document AFFDL-TR-68-44, reference 1. An earlier study that uses portions of this program is reported in document AFFDL-TR-67-81, reference 2.

This volume is divided into four sections. Section II describes the logical and system-oriented organization of the RANVIB program. Sections III and IV describe the programming details of the FORTRAN modules of the system. Section III is a description of the modules in the phase I structural and vibration programs. Section IV describes phase II programs: the random loading module and the random response solution modules. Appendixes I and II contain descriptions and listings of the matrix manipulative scheme TL01 used throughout RANVIB, and general-purpose subroutines. Appendixes III and IV contain the listings of phase I and II programs, respectively.

II

GENERAL PROGRAM DESCRIPTION

1. PROGRAM ORGANIZATION

The RANVIB system is divided into phases I and II. Phase I (figure 1) is an integrated set of computer programs for determining the static and dynamic characteristics of the structure. Phase II (figure 2) uses results of phase I and determines sonic loads and random structural response. This division permits the analysts to assess the results of phase I before proceeding to phase II. The matrix interpretive scheme TL01 written in MAP is used to perform matrix operations in both phases.

a. Phase I—Structural and Vibration Programs

There are two major modules in phase I.

The first one MAST (matrix structural generator) generates and merges element stiffness and stress matrices and reduces unwanted freedoms to form the reduced structural stiffness, flexibility, and stress-deflection matrices. These matrices are then merged and stored on tape for later use by the FREMOD routine and phase II programs.

The second module FREMOD (frequencies and modes generator) generates natural frequencies, normal modes, and generalized masses using the flexibility matrix (MAST output tape) and mass matrix (card input).

An optional path in phase I executes only the FREMOD module using a previously calculated flexibility matrix. This option is a convenient feature when an analyst wants to change mass distribution and/or change the number of modes. A new diagonal-mass matrix is required; input is on cards.

b. Phase II—Random Load and Response Programs

This section generates the excitation cross-power spectral density (cross PSD) and the response solution options (deflection and stress cross PSD) and statistical moments.

The random pressure loads (excitations) are stored on an intermediate output tape used in the response programs. The results from phase I output tape are also used in phase II.

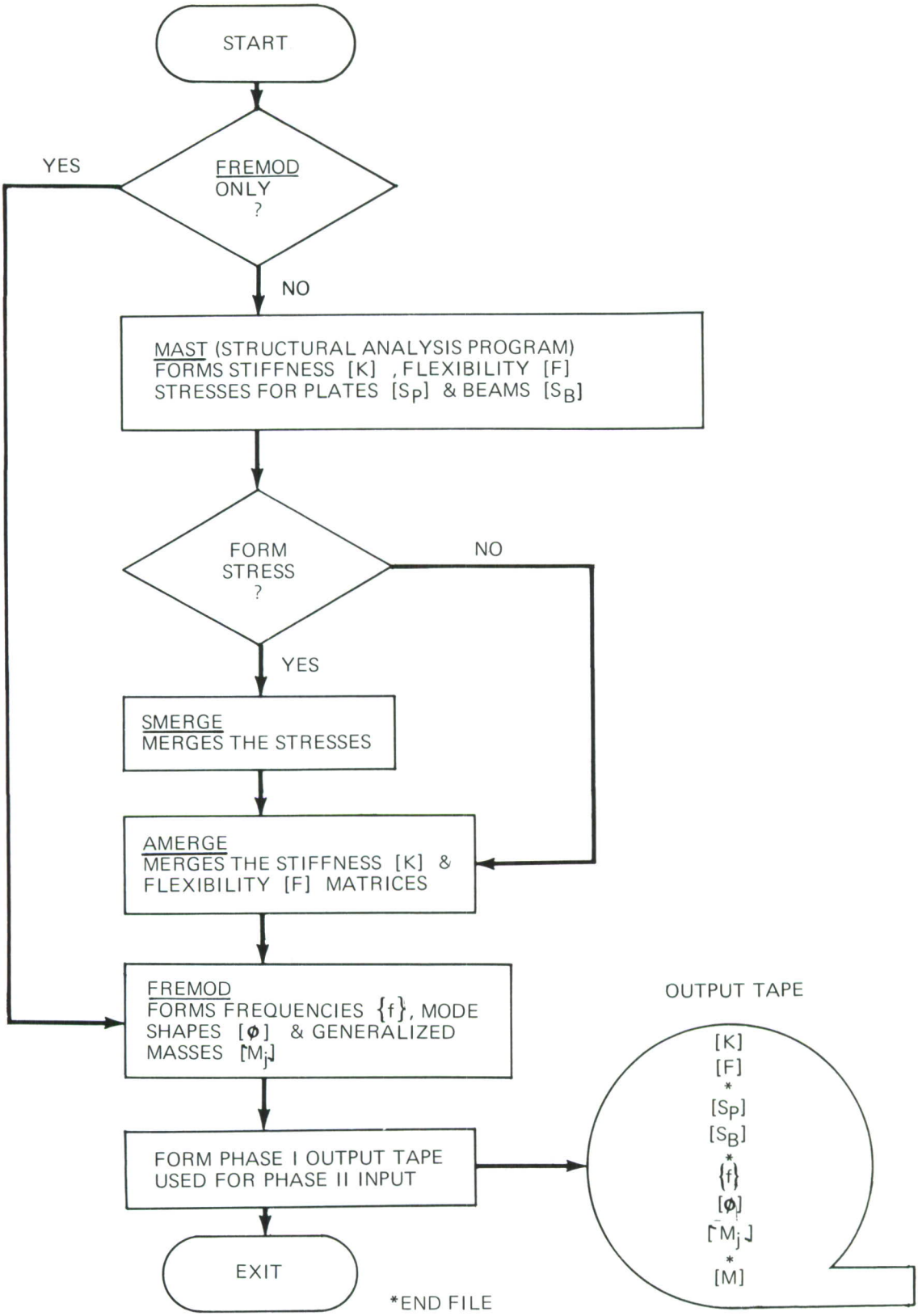


Figure 1. Phase I Organization

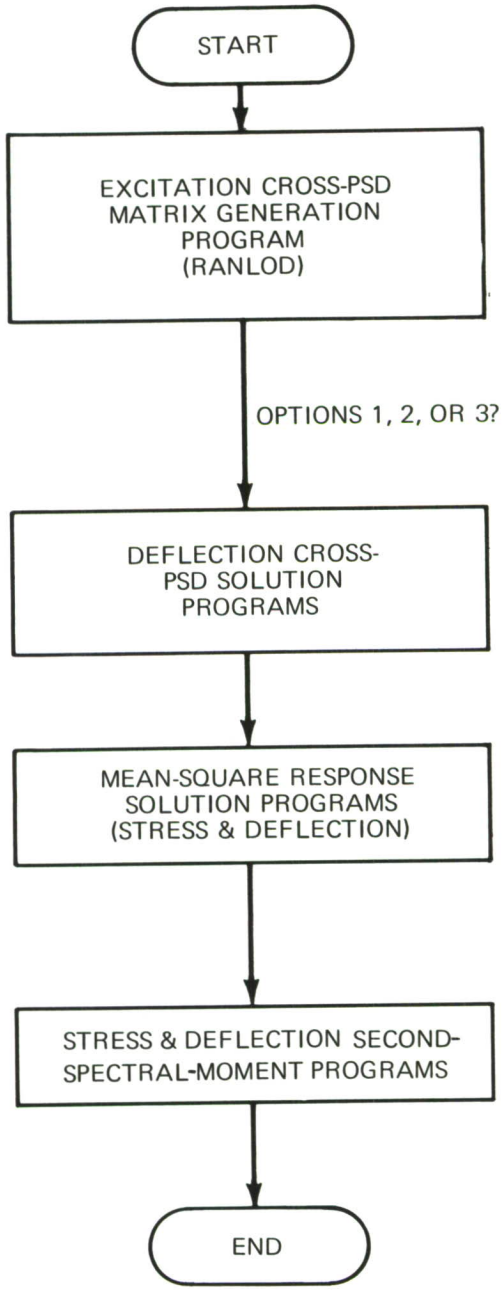


Figure 2. Phase II Organization

The response solution modules are divided into three options:

(1) Option 1—General viscous damping

(2) Option 2—Normal modes

(3) Option 3—Normal modes without cross terms

Reference 3 describes the options in more detail.

Cross PSD and joint moments are formed for each option. It is also possible to generate the stress second spectral moments used in predicting fatigue life.

2. SYSTEM ORGANIZATION

a. Phase I and II Overlay Structures

The overlay structure on a subroutine basis is illustrated in figure 3. The detailed overlay structure for MAST and FREMOD are shown in figures 17 and 32, respectively (pages 36 and 67).

b. Core and Tape Requirements

The RANVIB program requires a 32K core when operating on the IBM 7094 Mod II computer under the IBSYS Version 13 system.

Tape requirements in phase I and phase II are shown in table I. Core maps of phases I and II are shown in tables II and III, respectively.

c. Deck and Master Tape Setups

(1) Operating Procedure

The following is the procedure a system analyst should follow in initiating phase I and phase II operations for the direct-coupled system (DCS).

The nine files on the phase I master tape are generated by the card-to-tape process. The first file contains binary decks, and the remaining eight files contain TL01 data decks (figure 4). Use the appropriate control cards for the DCS for phases I and II. Mount the phase I master tape on logical unit 9. The output tape from phase I will contain four files on logical unit 10.

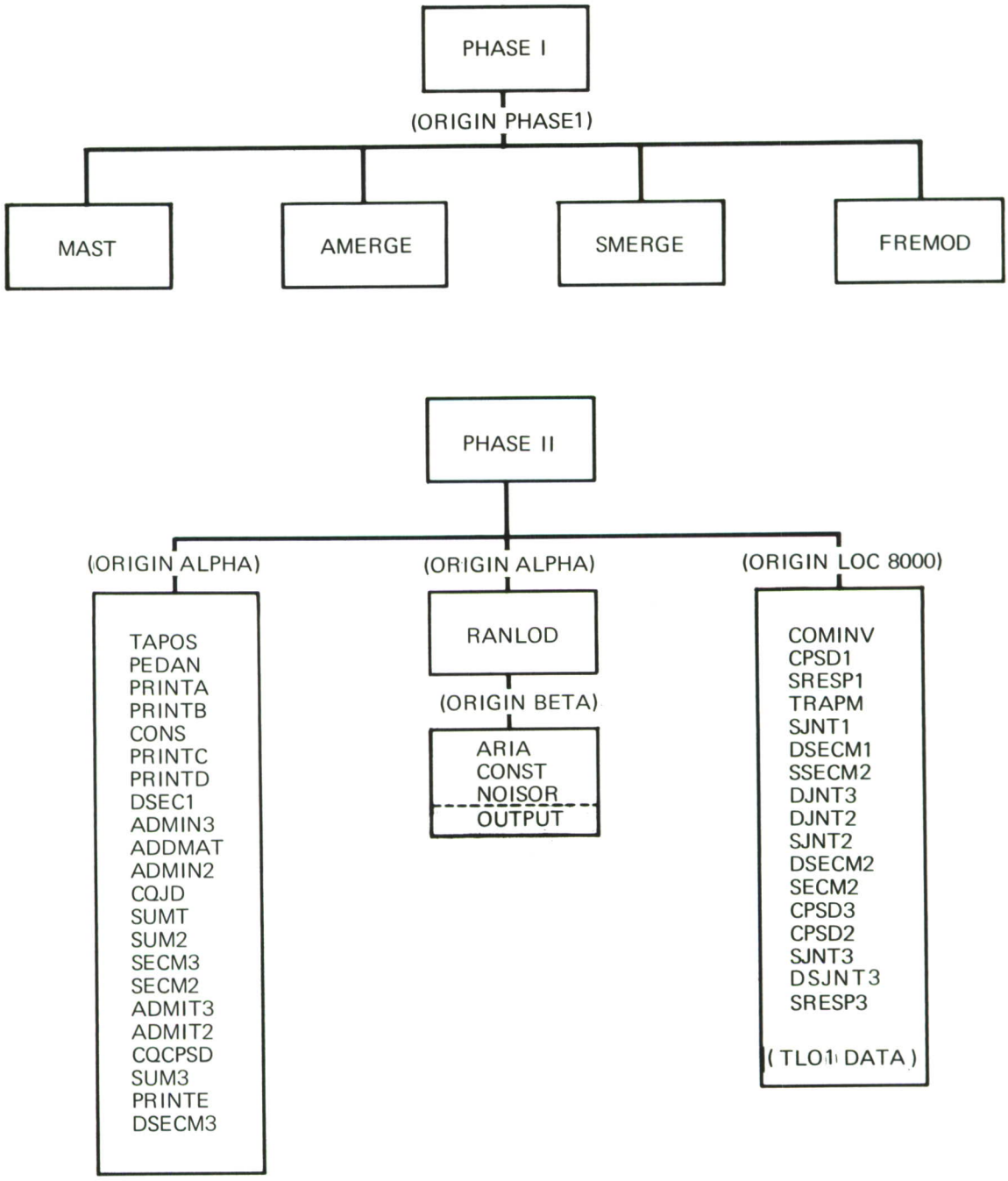


Figure 3. Phase I and II Overlay Structures

Table I. Tape Use

Logical unit	Function
PHASE I*	
2	The parameter matrix and stiffness matrix are stored on this tape (output from MAST). This tape is also used as a scratch unit.
3	Scratch
4	Scratch
5	Standard system input tape
6	Standard system output tape
8	Stress matrices for plates from MAST
9	Program master tape
10	Phase I output tape
12	Stress matrices for beams from MAST
PHASE II**	
9	Program master tape
10	Input tape from phase I
1-4, 7-8, 11-17	Intermediate scratch tapes
5	Standard system input tape
6	Standard system output tape

*See figure 19 for tape use of logical units 1 through 16 for the MAST routine (page 43).

**Figure 43, page 84, illustrates the tape use in the phase II program.

The phase II master tape consists of seven files and is again generated by the card-to-tape process. The first file contains binary decks, and the remaining six files contain TL01 data decks (figure 5). In this operation, mount the phase II master tape on logical unit 9.

The master tapes for phases I and II are assigned to system unit SYSLB4 (logical unit 9). The system overlay is assigned to system unit SYSCK2.

Table II. Phase I Core Maps

```
SYSTEM                                  00000 THRU  02717
FILE BLOCK ORIGIN                       02720
FILES        1.     UNIT07
             2.     UNIT08
             3.     UNIT09
             4.     UNIT10
             5.     UNIT11
             6.     UNIT12
             7.     UNIT13
             8.     UNIT14
             9.     UNIT15
            10.     UNIT16
            11.     UNIT17
            12.     UNIT01
            13.     UNIT02
            14.     UNIT03
            15.     UNIT04
            16.     UNIT05
            17.     UNIT06
FILE LIST ORIGIN                        03234
PRE-EXECUTION INITIALIZATION            03262
CALL ON OBJECT PROGRAM                  03321
OBJECT PROGRAM                          03326 THRU  71152
```

LINK	DECK	ORIGIN	CONTROL SECTIONS	(/NAME/=NON 0 LENGTH, (LOC)=DELETED, *=NOT REFERENCED)
0	BLK	03326	/TAPES / 03326	/REDUC / 03347
	PHASE1	03351	/PRNT / 03352	..... 03636 *
	WRTETP	03650	WRTETP (03650)	
	READTP	04324	READTP (04324)	
	FVIO	04677	.FVIO. 04677	
	FILE	05014	.UN07. 05014	.UN08. 05015 .UN09. 05016 .UN10. 05017 .UN11. 05020
			.UN12. 05021	.UN13. 05022 .UN14. 05023 .UN15. 05024 .UN16. 05025
			.UN17. 05026	
	.LINK	05027	/.LDT / 05027	/.LRECT/ 05062 /.LVEC / 05156
	.LXCON	05244	.LXSTR 05244	.LXSTP 05247 .LXOUT 05315 .LXERR 05324 .LXCAL 05327 *
			.LXRTN 05327	IBEXIT 05327 * .DBCLS 05511 * .LXARG 05663 .LO 05706 *
			.CLSE 05714	.LFBL 05715 * .LUNB 05716 .DFCUT 05717
	.IODEF	05723	.DEFIN 05723	.ATTAC 05727 * .CLOSE 05731 .OPEN 05733 .READ 05735
			.WRITE 05737	.BSR 05747 .READR 05757 * .RELES 05761 * .LAREA 05772 *
			.LFBLK 06010	.LTSX 06013 * .AREA1 06025 .LUNBL 06033 * .ENTRY 06037
			.GOA 06072	.GO 06076 .DERR 06112 .NOPXI 06113 .COMXI 06115
			.EX34 06137	

Table II–Continued

	.IOCSF	06144											
	.LOVRY	10446	.LOVRY	(10446)	.LDT	(05027)	.LRECT	(05062)	.LVEC	(05156)			
	.LXSL	11154	.LXSEL	11154	.LXSL1	11155	.LXTST	11160 *	.LXOVL	11220 *	.LXMOD	11262 *	
			.LXIND	11306	.LXDIS	11311	.LXFLG	11312	.LTCH	11313			
	.FPTRP	11321	.FFPT.	11321 *	.FPOUT	11456	.FPARG	11464	/.COUNT/	11466 *	OVFLOW	11532	
	XIT	11541	EXIT	11541	.EXIT.	11541							
	FXEM	11542	.FXEM.	11542	.FXOUT	12077	.FXARG	12105	/.OPTN./	12161 *			
	FOUT	12172	.FOUT.	12172									
	FCNV	12561	.FCON.	12561	.FCNV.	12604	.ENDFS	12617	.CNVSW	12621	.FDX1	12625	
			.FDX2	12626	.DBC	12630	.DBC10	12766	.DBC20	13014	.DDSW	13024	
			.DDFIX	13033	.FIXSW	13041	.DDBC	13116	.DDRS1	13361	.DDRS2	13363 *	
			.D1	13366	.D2	13370	.FERR2	13455 *	.ANPT	13511	.ONPT	13526	
			.LNTP	13611	.AOUT	13660	.DFLT	13676	.FLT	14032	.DEXPN	14122	
			.FXD	14123	.HOUT	14253	.INTG	14323	.LOUT	14441	.DOUT	14460	
			.XCF	14511	.TEST	15217	.KOUNT	15222	.LIST	15225	.DONE	15236	
			.OUTBF	15302	.BUF	15332	.QSTO	15333	.WIDTH	15334	.GAIN	15335	
			.GAIN1	15336	.FBDBF	15346	.DDDFL	15372	.DDFLG	15373	.MQD	15374	
			.PEX	15375	.FEXP	15376	.DIG	15377					
	FIOB	15415	.FIOB.	15415	.FCNT	15520	.FBLT.	15616	.FBDT.	15636	.FRLR.	15662	
			.FRLR.	(15662)	.FWLR.	15726	.FWLR.	(15726)	.FBIBF	15766	.FRITE	16060	
	FIOS	16066	.FIOS.	16066	.FSEL.	16226	.FILR.	16232 *	.FRTB.	16241	.FRTO.	16246	
			.FILL.	16251	.FCLS	16253 *	.FOPN	16257 *	REOF	16263 *	.TOUT.	16426	
			.REED	16434 *	.BIN	16435 *	.FCT	16436	.FCKSZ	16440			
	FIOH	16522	.FIOH.	16522	.FFIL.	17307	.FRTN.	17334					
	FWRD	17532	.FWRD.	17532									
	FWRB	17556	.FWRB.	17556									
	FRDD	17602	.FRDD.	17602									
	FRDB	17630	.FRDB.	17630									
	FPRN	17654	.FPRN.	17654									
	UNO1	20013	.UNO1.	20013									
	UNO2	20014	.UNO2.	20014									
	UNO3	20015	.UNO3.	20015									
	UNO4	20016	.UNO4.	20016									
	UNO5	20017	.UNO5.	20017									
	UNO6	20020	.UNO6.	20020	.BUFSZ	20021							
	FSQR	20024	SQRT	20024									
	FBST	20077	.FBST.	20077									
	FEFT	20314	.FEFT.	20314									
	FRWT	20414	.FRWT.	20414									
	FSLDI	20521	.FSLI.	20537	.FSDI.	20545 *							
	FSLBI	20556	.FBLI.	20574	.FBDI.	20602 *							
	FSLI	20614	.SLI.	20614	.SLI1.	20621	.SDI.	20627	.SDI1.	20635			
	FSLDO	20650	.FSLO.	20666	.FSDO.	20674 *							
	FSLBO	20705	.FBLO.	20723	.FBDO.	20731 *							
	FSLO	20743	.SLO.	20743	.SLO2.	20751	.SDO.	20756	.SDO2.	20765			
	FDVCHK	20777	DVCHK	20777									
	FOVERF	21016	OVERFL	21016									

Table II–Continued

	FASNCS 21040	ASIN 21040	ACOS 21042 *	/ERASE / 21171			
	FBSF 21215	BSF 21217	/ERASE /(21171)				
	FFSF 21334	FSF 21336	/ERASE /(21171)				
	FFSR 21420	FSPR 21422 *	FSR 21424	/ERASE /(21171)			
	FKRD 21505	KRD 21507	KRDG 21511 *	CART 21512 *			
	FRUN 22326	UNLOAD 22330	/ERASE /(21171)				
1	MAST* 22355	/RENT / 22356	/MAPSTR/ 22360	/CONT1 / 22362	/LASTND/ 24022	/COMS / 24332	
		/TERMS / 24642	/CONTRL/ 24655	/TAPES /(03326)	/TITL / 24665	/PAGE / 24702	
		/REDUC /(03347)	MAST 26002				
	PAGH* 26016	/TITL /(24665)	/TAPES /(03326)	/PAGE /(24702)	PAGHED 26066		
	UNPAC* 26100	UNPACK 26210					
	PRINT* 26246	PRINT 26601					
2	SUBM1* 26636	/CONT1 /(22362)	/CONT2 / 26637	/CONT3 / 30277	/RENT /(22356)	/LASTND/(24022)	
		/COMS /(24332)	/TERMS /(24642)	/CONTRL/(24655)	/SKIP / 31737	/NOMERG/ 31743	
		SUBM1 31774					
3	GENRA* 32006	/CONT1 /(22362)	/CONT2 /(26637)	/CONT3 /(30277)	/LASTND/(24022)	/CORD / 32007	
		/CONTRL/(24655)	/TERMS /(24642)	/ADPRO / 45567	/SSTR / 45575	/CHECK / 45577	
		/TAPES /(03326)	/FLAG / 45601	/TITL /(24665)	GENRAT 45727		
	REDUC* 45741	REDUCE 46136					
4	INFO* 46171	/CONT1 /(22362)	/CONT2 /(26637)	/CONT3 /(30277)	/LASTND/(24022)	/CORD /(32007)	
		/CONTRL/(24655)	/TERMS /(24642)	/TAPES /(03326)	/TITL /(24665)	/REDUC /(03347)	
		INFO 47266					
5	PLATE* 46171	/RENT /(22356)	/CONT1 /(22362)	/CONT3 /(30277)	/LASTND/(24022)	/TAPES /(03326)	
		/CORD /(32007)	/TERMS /(24642)	/ADPRO /(45567)	/SSTR /(45575)	/PSTIF2/ 46172	
		/PSTIF1/ 46174	/PSTIF8/ 47274	/PSTIF9/ 51100	/PSTIFC/ 51102	/PSTIFD/ 51105	
		/PSTIFG/ 51110	/PMTR1 / 51124	/STRAN / 51135	/FLAG /(45601)	/PSTIFL/ 51435	
		/SWAP / 51436	/PSTIFH/ 51437	/TITL /(24665)	/BUCK / 51447	/THRST / 51450	
		PLATE 55026					
	MUL1* 55047	MUL1 55235					
	MUL2* 55274	MUL2 55457 *					
	PSTIF* 55515	/RENT /(22356)	/TPLN1 / 55516	/PSTIFL/(51435)	/PSTIF9/(51100)	PSTIF 55562	
	QUAD* 55576	/PSTIF2/(46172)	/ADPRO /(45567)	/PSTIF3/ 55577	/PSTIF5/ 55605 *	/PSTIF8/(47274)	
		/PSTIFH/(51437)	/PSTIFK/ 55651	/PSTIFL/(51435)	/TEMP2 / 55652 *	/TPLN1 /(55516)	
		/RENT /(22356)	/SWAP /(51436)	/LOAD / 56422 *	QUAD 56651		
	LAMK* 56667	LAMK 56757					
	KLAMT* 57013	KLAMT 57117					
	TRI* 57157	/PSTIF1/(46174)	/PSTIF3/(55577)	/PSTIF7/ 57160	/PSTIF9/(51100)	/PSTIFH/(51437)	
		/PSTIFD/(51105)	/BUCK /(51447)	/THRST /(51450)	TRI 57473		
	INP* 57511	/PSTIFA/ 57512	/PSTIFB/ 57556	INP 57721			
	INPM* 57733	/PSTIF2/(46172)	/ADPRO /(45567)	/PSTIF7/(57160)	/PSTIF9/(51100)	/PSTIFB/(57556)	

Table II–Continued

			/PSTIFD/(51105)	INPM 60744			
	INPST*	60762	/PSTIF2/(46172)	/ADPRO /(45567)	/PSTIF7/(57160)	/PSTIF9/(51100)	/PSTIFA/(57512)
			/PSTIFC/(51102)	/PSTIFD/(51105)	INPST 61534		
	OUTP*	61550	/PSTIFA/(57512)	/PSTIFB/(57556)	OUTP 61573		
	OUTPM*	61605	/PSTIFA/(57512)	/PSTIFC/(51102)	/PSTIFD/(51105)	OUTPM 61725	
	OUTPS*	61741	/PSTIF2/(46172)	/ADPRO /(45567)	/PSTIF7/(57160)	/PSTIFB/(57556)	OUTPSH 62561
	COMBI*	62573	/PSTIFA/(57512)	/PSTIFB/(57556)	COMBIN 62763		
	STORE*	63013	/PSTIF1/(46174)	/PSTIF8/(47274)	/PSTIF9/(51100)	/PSTIFB/(57556)	/PSTIFK/(55651)
			STORE 63640				
	MOVE*	63667	MOVE 64001				
	PMTR*	64025	/PMTR1 /(51124)	/PSTIF9/(51100)	/PSTIFG/(51110)	PMTR 64373	
	SMTR*	64407	/PSTIF9/(51100)	/PSTIFH/(51437)	/STRAN /(51135)	SMTR 65634	
	LOCAL*	65655	/RENT /(22356)	/PSTIF9/(51100)	/PSTIFG/(51110)	/PSTIFH/(51437)	/PSTIFL/(51435)
			/TAPES /(03326)	LOCAL 66421			
	COPLA*	66433	/PSTIF9/(51100)	/PSTIFG/(51110)	/PSTIFH/(51437)	/NOMERG/(31743)	/CHECK /(45577)
			/TAPES /(03326)	COPLAN 66712			
6	FSCN	46171	COS 46171	SIN 46172			
	FXP2	46365	.XP2. 46365				
	BEAM*	46503	/TITL /(24665)	/CONT1 /(22362)	/CONT2 /(26637)	/LASTND/(24022)	/CORD /(32007)
			/TERMS /(24642)	/TAPES /(03326)	/PROPT / 46504	/ADPRO /(45567)	/SSTR /(45575)
			/VAR / 47634	/N3N3 / 47636	/FLAG /(45601)	BEAM 56015	
	TINVR*	56040	TINVR 56642				
	SMULT*	56746	SMULT 57126				
	MULT*	57161	MULT 57513				
	SBMTR*	57570	/TAPES /(03326)	/VAR /(47634)	/ADPRO /(45567)	/N3N3 /(47636)	SBMTR 60361
	SSTIF*	60501	/VAR /(47634)	/ADPRO /(45567)	/PROPT /(46504)	SSTIF 61242	
	MAD*	61410	MAD 61426				
	SBGS*	61453	SBGS 61642				
	OFST*	61772	OFST 62115				
	CSTIF*	62232	/TAPES /(03326)	/VAR /(47634)	/ADPRO /(45567)	/PROPT /(46504)	CSTIF 63511
	CBMTR*	63610	/TAPES /(03326)	/VAR /(47634)	/ADPRO /(45567)	/LTRAN / 63611 *	CBMTR 64775
7	MERGE*	32006	/CONT1 /(22362)	/CONT2 /(26637)	/CONT3 /(30277)	/LASTND/(24022)	/CONTRL/(24655)
			/TERMS /(24642)	/COMS /(24332)	/SKIP /(31737)	/TAPES /(03326)	/REDUC /(03347)
			MERGE 32336				
	SOR*	32352	SOR 32501				
8	MERGB*	32550	/CONT1 /(22362)	/CONT2 /(26637)	/CONT3 /(30277)	/COMS /(24332)	/LASTND/(24022)
			/TERMS /(24642)	/CONTRL/(24655)	/TAPES /(03326)	MERGBC 71124	
9	STRES*	32550	/TERMS /(24642)	/CONTRL/(24655)	/SKIP /(31737)	/PBSIZE/ 32551	/TAPES /(03326)
			/LOADS / 32556	STRESS 46116			
	MSTRE*	46130	/CONTRL/(24655)	/LASTND/(24022)	/TERMS /(24642)	/SKIP /(31737)	/PBSIZE/(32551)
			/TAPES /(03326)	/MAPSTR/(22360)	/LOADS /(32556)	MSTRES 55741	
10	SORCON	26636	/SORT / 26637	/TERMS /(24642)	/TAPES /(03326)	/COMS /(24332)	/CONT1 /(22362)
			/RSIZE / 45517	/MAPSTR/(22360)	/CONTEM/ 47014	/REDUC /(03347)	SOLN 47132
	TEST*	47144	TEST 47226				

Table II–Concluded

11	FKSRT	47263	/LASTND/(24022)	/TERMS /(24642)	/TAPES /(03326)	/SORT /(26637)	/CONT1 /(22362)				
			/CONTEM/(47014)	/COMS /(24332)	FKSORT 52065						
12	KFSRT	47263	/SORT /(26637)	/TAPES /(03326)	/TERMS /(24642)	/COMS /(24332)	/CONT1 /(22362)				
			/CONTEM/(47014)	KFFSRT 67356							
13	CNCT*	47263	/SORT /(26637)	/TERMS /(24642)	/TAPES /(03326)	/COMS /(24332)	CONECT 50667				
14	EXPND*	5C703	EXPAND 67465								
15	EXTRN*	5C703	EXTRAN 70134								
16	DELET*	47263	/TAPES /(03326)	/TERMS /(24642)	/RSIZE /(45517)	/LASTND/(24022)	/MAPSTR/(22360)				
			/TEMPO / 47264	DELETE 64251							
	SSORT*	64315	/SORT /(26637)	/TAPES /(03326)	/TERMS /(24642)	/RSIZE /(45517)	/COMS /(24332)				
			/MAPSTR/(22360)	/TEMPO /(47264)	SSORT 65724						
17	HELP	26636	TL01 26636								
	YTLOSB	26664	YTL01 26664 *	YTL01 26665	MMM 27122 *	ERRCDE 34437 *	TAPE 34441 *				
	INVERT	35420	INV4DS 35420								
	DATASB	36540	DATA 36540								
18	FREMO*	22355	FREMOD 57453								
	WDVALV	57474	VALVCT 60645								
	INRPRD	6C724	INRPRD (60724)	PRDSUM 60744 *							
19	WDHESS	61020	HESSEN 61517								
20	WDQRIT	61020	QRITER 62210								
21	WDSORT	61020	SORTRT 61310								
22	WDVECT	61020	VECTOR 61453								
23	WDTRN1	61020	TRANS1 61423								
24	WDTRN2	61020	TRANS2 61376								
25	AMERG*	22355	/PRNT /(03352)	AMERGE 55533							
26	SMERG*	22355	/PRNT /(03352)	SMERGE 63005							

I/O BUFFERS	71153 THRU 77772
UNUSED CORE	77773 THRU 77777

Table III. Phase II Core Maps

```
SYSTEM                                  00000 THRU  02717
FILE BLOCK ORIGIN                       02720
FILES          1.     UNIT07
               2.     UNIT08
               3.     UNIT09
               4.     UNIT10
               5.     UNIT11
               6.     UNIT12
               7.     UNIT13
               8.     UNIT14
               9.     UNIT15
              10.     UNIT16
              11.     UNIT17
              12.     UNIT01
              13.     UNIT02
              14.     UNIT03
              15.     UNIT04
              16.     UNIT05
              17.     UNIT06
FILE LIST ORIGIN                        03234
PRE-EXECUTION INITIALIZATION            03276
CALL ON OBJECT PROGRAM                  03327
OBJECT PROGRAM                          03334 THRU  65757
```

LINK	DECK	ORIGIN	CONTROL SECTIONS	(/NAME/=NON 0 LENGTH, (LOC)=DELETED, *=NOT REFERENCED)
0	PHASE2	03334	/BLK1 / 03335	/BLK2 / 03563 /BLK3 / 03575 ..... 06111 *
	SCALE*	06125	SCALE 06203	
	FVIO	06232	.FVIO. 06232	
	FILE	06347	.UN07. 06347	.UN08. 06350 .UN09. 06351 .UN10. 06352 .UN11. 06353
			.UN12. 06354	.UN13. 06355 .UN14. 06356 .UN15. 06357 .UN16. 06360
			.UN17. 06361	
	READTP	06362	READTP (06362)	
	WRTETP	06735	WRTETP (06735)	
	.LINK	07411	/.LDT / 07411	/.LRECT/ 07445 /.LVEC / 07533
	.LXCON	07621	.LXSTR 07621	.LXSTP 07624 .LXOUT 07672 .LXERR 07701 .LXCAL 07704 *
			.LXRTN 07704	IBEXIT 07704 * .DBCLS 10066 * .LXARG 10240 .LO 10263 *
			.CLSE 10271	.LFBL 10272 * .LUNB 10273 .DFOUT 10274
	.IODEF	10300	.DEFIN 10300	.ATTAC 10304 * .CLOSE 10306 .OPEN 10310 .READ 10312
			.WRITE 10314	.BSR 10324 * .READR 10334 * .RELES 10336 * .LAREA 10347 *
			.LFBLK 10366	.LTSX 10370 * .AREA1 10402 .LUNBL 10410 * .ENTRY 10414
			.GOA 10447	.GO 10453 .DERR 10467 .NOPXI 10470 .COMXI 10472
			.EX34 10514	
	.IOCSF	10521		
	.LOVRY	13023	.LOVRY (13023)	.LDT (07411) .LRECT (07445) .LVEC (07533)
	.LXSL	13531	.LXSEL 13531	.LXSL1 13532 .LXTST 13535 * .LXOVL 13575 * .LXMOD 13637 *
			.LXIND 13663	.LXDIS 13666 .LXFLG 13667 .LTCH 13670

Table III–Continued

	.FPTRP	13676	.FFPT.	13676 *	.FPOUT	14033	.FPARG	14041	/.COUNT/	14043 *	OVFLOW	14107 *	
	.ERAS.	14116	E.1	14116	E.2	14117	E.3	14120	E.4	14121			
	.XCC.	14122	CC.1	14122	CC.2	14123	CC.3	14124	CC.4	14125			
	XIT	14126	EXIT	14126	.EXIT.	14126 *							
	FXEM	14127	.FXEM.	14127	.FXOUT	14464	.FXARG	14472	/.OPTW./	14546 *			
	FOUT	14557	.FOUT.	14557									
	FCNV	15146	.FCON.	15146	.FCNV.	15171	.ENDFS	15204	.CNVSW	15206	.FDX1	15212	
			.FDX2	15213	.DBC	15215	.DBC10	15353	.DBC20	15401	.DDSW	15411	
			.DDFIX	15420	.FIXSW	15426	.DDBC	15503	.DDRS1	15746	.DDRS2	15750 *	
			.D1	15753	.D2	15755	.FERR2	16042 *	.ANPT	16076	.ONPT	16113	
			.LNTP	16176	.AOUT	16245	.DFLT	16263	.FLT	16417	.DEXPN	16507	
			.FXD	16510	.HOUT	16640	.INTG	16710	.LOUT	17026	.OOUT	17045	
			.XCF	17076	.TEST	17604	.KOUNT	17607	.LIST	17612	.DONE	17623	
			.OUTBF	17667	.BUF	17717	.QSTO	17720	.WIDTH	17721	.GAIN	17722	
			.GAIN1	17723	.FBDBF	17733	.DDDFL	17757	.DDFLG	17760	.MOD	17761	
			.PEX	17762	.FEXP	17763	.DIG	17764					
	FIOB	20002	.FIOB.	20002	.FCNT	20105	.FBLT.	20203	.FBDT.	20223	.FRLR.	20247	
			.FRLR.	(20247)	.FWLR.	20313	.FWLR.	(20313)	.FBIBF	20353	.FRITE	20445	
	FIOS	20453	.FIOS.	20453	.FSEL.	20613	.FILR.	20617 *	.FRTB.	20626	.FRTD.	20633	
			.FILL.	20636	.FCLS	20640 *	.FOPN	20644 *	REOF	20650 *	.TOUT.	21013	
			.REED	21021 *	.BIN	21022 *	.FCT	21023	.FCKSZ	21025			
	FIOH	21107	.FIOH.	21107	.FFIL.	21674	.FRTN.	21721					
	FWRD	22117	.FWRD.	22117									
	FWRB	22143	.FWRB.	22143									
	FRDD	22167	.FRDD.	22167									
	FRDB	22215	.FRDB.	22215									
	FPRN	22241	.FPRN.	22241									
	UN01	22400	.UN01.	22400									
	UN02	22401	.UN02.	22401									
	UN03	22402	.UN03.	22402									
	UN04	22403	.UN04.	22403									
	UN05	22404	.UN05.	22404									
	UN06	22405	.UN06.	22405	.BUFSZ	22406							
	FLOG	22411	ALOG10	22411 *	ALOG	22412							
	FXPF	22615	EXP	22615									
	FSCN	22736	COS	22736	SIN	22737							
	FSQR	23132	SQRT	23132									
	FXP1	23205	.XP1.	23205									
	FEFT	23321	.FEFT.	23321									
	FRWT	23421	.FRWT.	23421									
	FSLDI	23526	.FSLI.	23544	.FSDI.	23552 *							
	FSLBI	23563	.FBLI.	23601	.FBDI.	23607 *							
	FSLI	23621	.SLI.	23621	.SLI1.	23626	.SDI.	23634	.SDI1.	23642			
	FSLDO	23655	.FSLO.	23673	.FSDO.	23701 *							
	FSLBO	23712	.FBLO.	23730	.FBDO.	23736 *							
	FSLO	23750	.SLO.	23750	.SLO2.	23756	.SDO.	23763	.SDO2.	23772			
	FASNCS	24004	ASIN	24004	ACOS	24006 *	/ERASE /	24135					
	FBSF	24161	BSF	24163	/ERASE /	(24135)							

Table III–Continued

	FFSF	24300	FSF	24302	/ERASE	/(24135)													
	FFSR	24364	FSPR	24366 *	FSR	24370	/ERASE	/(24135)											
	FKRD	24451	KRD	24453	KRDG	24455 *	CART	24456 *											
	FRUN	25272	UNLOAD	25274	/ERASE	/(24135)													
1	HELP	25321	TLO1	25321															
	YTLOSB	25347	YTLO1	25347 *	YTLO1	25350	MMM	25605 *	ERRCDE	33122 *	TAPE	33124 *							
	INVERT	34103	INV4DS	34103															
	DATASB	35223	DATA	35223															
2	TAPOS*	25321	/BLK3	/(03575)	TAPOS	25435													
3	RANLO*	25321	/BLK1	/(03335)	/BLK2	/(03563)	/BLK3	/(03575)	/BLK4	/ 25322	/BLK5	/ 25347							
			/BLK6	/ 26031	RANLOD	64306													
4	ARIA**	64322	/BLK1	/(03335)	/BLK2	/(03563)	/BLK3	/(03575)	/BLK4	/(25322)	/BLK5	/(25347)							
			/BLK6	/(26031)	ARIA	64405													
5	CONST*	64322	CONST	64423															
6	NOISO*	64322	/BLK1	/(03335)	/BLK2	/(03563)	/BLK3	/(03575)	/BLK4	/(25322)	/BLK5	/(25347)							
			/BLK6	/(26031)	NOISOR	65050													
	OUTPUT	65123	/BLK1	/(03335)	/BLK2	/(03563)	/BLK3	/(03575)	/BLK4	/(25322)	/BLK5	/(25347)							
			/BLK6	/(26031)	OUTPT	65665 *													
7	PEDAN*	25321	/BLK1	/(03335)	/BLK2	/(03563)	PEDAN	65735											
8	PRNTA*	25321	/BLK1	/(03335)	/BLK2	/(03563)	PRINTA	45551											
9	PRNTB*	25321	/BLK1	/(03335)	/BLK2	/(03563)	/BLK3	/(03575)	PRINTB	26273									
10	CONS*	25321	/BLK1	/(03335)	/BLK2	/(03563)	CONS	26003											
11	PRNTC*	25321	/BLK2	/(03563)	PRINTC	45465													
12	PRNTD*	25321	/BLK1	/(03335)	/BLK2	/(03563)	/BLK3	/(03575)	PRINTD	26117									
13	DSEC1*	25321	/BLK1	/(03335)	/BLK2	/(03563)	DSECM1	65731											
14	ADMN3*	25321	/BLK1	/(03335)	/BLK2	/(03563)	/BLK3	/(03575)	ADMIN3	32774									
15	ADDMA*	25321	/BLK1	/(03335)	/BLK2	/(03563)	/BLK3	/(03575)	ADDMAT	65604									
16	ADMN2*	25321	/BLK1	/(03335)	/BLK2	/(03563)	ADMIN2	42552											
17	CQJD*	25321	/BLK2	/(03563)	CQJD	65565													
18	SUMT*	25321	/BLK2	/(03563)	SUMT	65436													

Table III–Concluded

19	SUM2*	25321	/BLK2	/(03563)	SUM2	65565					
20	SECM3*	25321	/BLK1	/(03335)	/BLK2	/(03563)	SECM3	32657			
21	SECM2*	25321	/BLK1	/(03335)	/BLK2	/(03563)	SECM2	42426			
22	ADMT3*	25321	/BLK1	/(03335)	/BLK2	/(03563)	/BLK3	/(03575)	ADMIT3	33316	
23	ADMT2*	25321	/BLK1	/(03335)	/BLK2	/(03563)	ADMIT2	42053			
24	CQCPS*	25321	/BLK2	/(03563)	CQCPSD	65644					
25	SUM3*	25321	/BLK1	/(03335)	/BLK2	/(03563)	SUM3	65535			
26	PRNTE*	25321	/BLK1	/(03335)	/BLK2	/(03563)	/BLK3	/(03575)	PRINTE	26161	
27	DSECM*	25321	/BLK1	/(03335)	/BLK2	/(03563)	DSECM3	65414			

I/O BUFFERS	65760 THRU	77767
UNUSED CORE	77770 THRU	77777

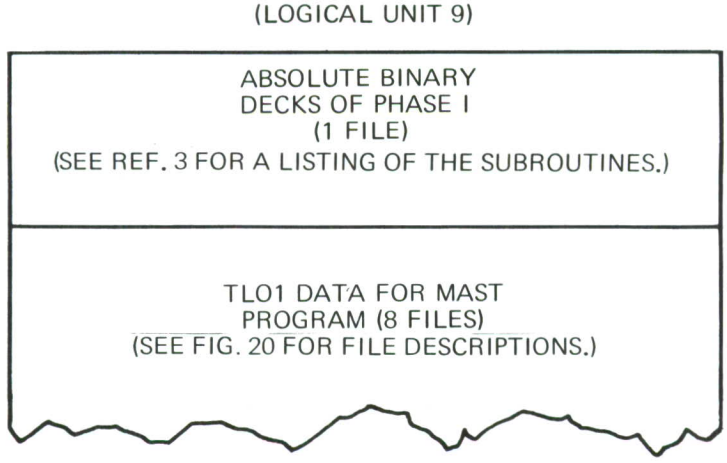


Figure 4. Phase I Master Tape

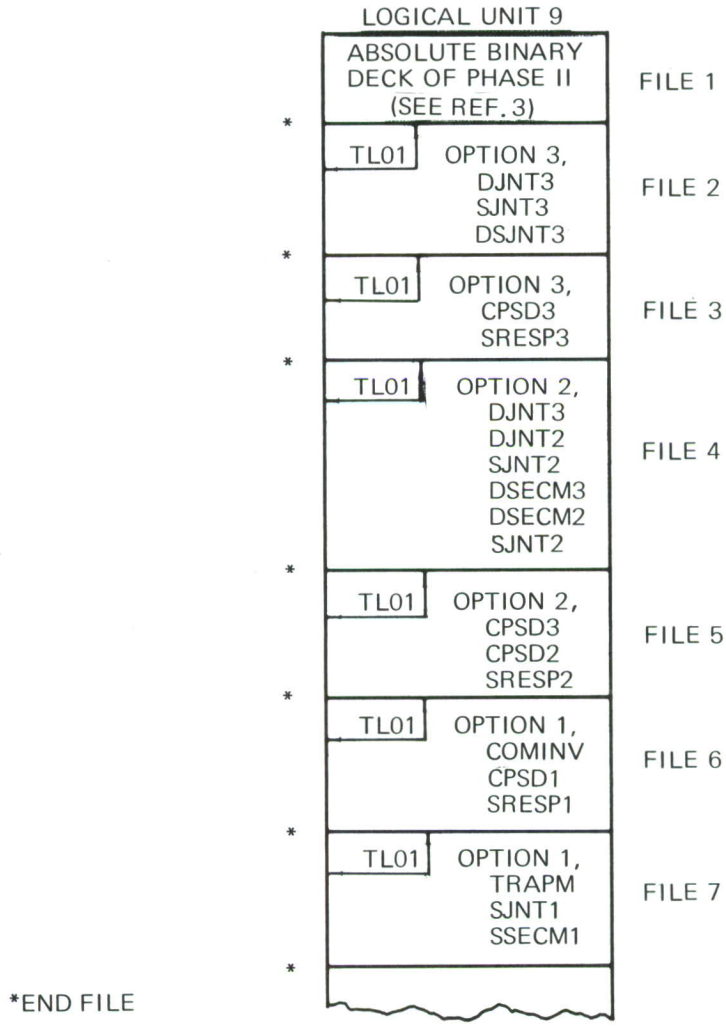


Figure 5. Phase II Master Tape

(2) Input/Output Editor

The input/output editor ($IEDIT) is used for two functions: (1) to read information off the master tape, and (2) to assist the programmer in modifying existing subroutines.

When used to read information off the master tape, the $IEDIT card precedes the component control card of the deck that is affected. The specifications on the control card remain in effect until the end of the application or until another $IEDIT card changes the specifications. The format of the $IEDIT card with optional instructions starting in column 16 for the RANVIB system is:

```
Column:  2  4  6  —  —  —  16 18 20 22 24 26 —  —  —  80
         1  3  5  —  —  —  17 19 21 23 25 —  —  —
         $IEDIT—  —  —  SYSLB4, SRCH —  —  —
```

The following procedure should be followed when an analyst/programmer wants to change any source or binary subroutines.

(a) Pull the appropriate $IBLDR card out of the phase I/phase II control deck.

(b) Insert a $IEDIT card with no optional instructions (columns 16 and on are blank).

(c) Insert the modified subroutine source or binary deck.

Insert a $IEDIT card with optional instructions SYSLB4, SRCH (see the example above).

```
[Col 1        Col 16          ]
 $IEDIT       SYSLB4,SRCH
```

III

PHASE I—STRUCTURAL AND VIBRATION PROGRAMS

Phase I is an integrated set of computer programs for determining the static and dynamic characteristics of the structure. The execution of this phase is controlled by the PHASE1 subroutine. The MAST module, FREMOD module, and intermediate matrix merge programs are called from the control program. The program listings for phase I are included in appendix III.

1. MATRIX STRUCTURAL GENERATOR PROGRAM (MAST)

a. General Description

The purpose of the MAST module is to generate and merge element stiffness and stress matrices and to reduce out unwanted freedoms.

The input is in card form and describes the physical structure in terms of nodes connected by beams and plates. Other inputs are boundary conditions that fix the structure in space and retained-freedom information that specifies the freedoms to be retained in the final matrices. The output is via tape to the other modules and includes the reduced stiffness matrix, flexibility matrix, and stress matrices for both beams and plates.

The module is divided into four segments. The first segment (generation) generates elemental stiffness and stress matrices for beam and plate elements. The second segment (merge) merges these elemental matrices to form the structural stiffness and stress matrices and deletes the constrained freedoms from the stiffness matrix. The third segment (sorting) sorts the stiffness and stress matrices into retained and reduced partitions. The final segment (reduction) then performs the actual equation solving necessary for reduction.

The module is restricted to a maximum of 2,000 nodes and 7,000 retained freedoms. The number of beam and plate elements is unrestricted. These limits are set by core storage limitations, and any attempt to run problems in this size range may be restricted by machine reliability and the peripheral storage size of the computer being used.

b. Major Program Functions

(1) Elemental-Matrix Generation

(a) Beam Matrices

The elemental matrices for beams are generated one at a time in the order that the input data are read. This generation includes the various transformation matrices and the stiffness matrix in local coordinates. The first transformation matrix generated (if offsets are present) is the offset transformation from the beam's neutral axis location to node point location. Next, the stress-transformation matrix and local-stiffness matrix are generated. The offset transformation is then applied to the stress transformation and this premultiplies the local-stiffness matrix to obtain the beam-stress matrix. This beam-stress matrix is finally premultiplied by the transpose of the combined offset and stress-transformation matrix to obtain the stiffness matrix in structural coordinates.

These matrices are then written on tape for later use by the merge segment. The structure of these elemental matrices is as shown in figures 6 and 7.

Each beam is also checked to determine which partitions of the merged stiffness matrix it will contribute to. A list of partitions is created that contains partition identification numbers for the partitions having non-null elements. This list is updated when elements are found that will contribute to partitions not already in the list.

(b) Plate Matrices

The plate-element matrices are also generated one plate at a time in the order that the input data are read. The elemental stiffness matrix in local coordinates is first generated and then the coordinate and stress-transformation matrices are generated. In generating the stiffness for quadrilateral plates, the program subdivides the quadrilateral into four triangles (figure 8) with a fifth "dummy" node placed at the centroid. The four triangles are then merged to form the local-stiffness matrix for the quadrilateral plate, and the terms for the fifth node are reduced out. Next, the local-stiffness matrix is post multiplied by the transpose of the coordinate-transformation matrix and this result is saved.

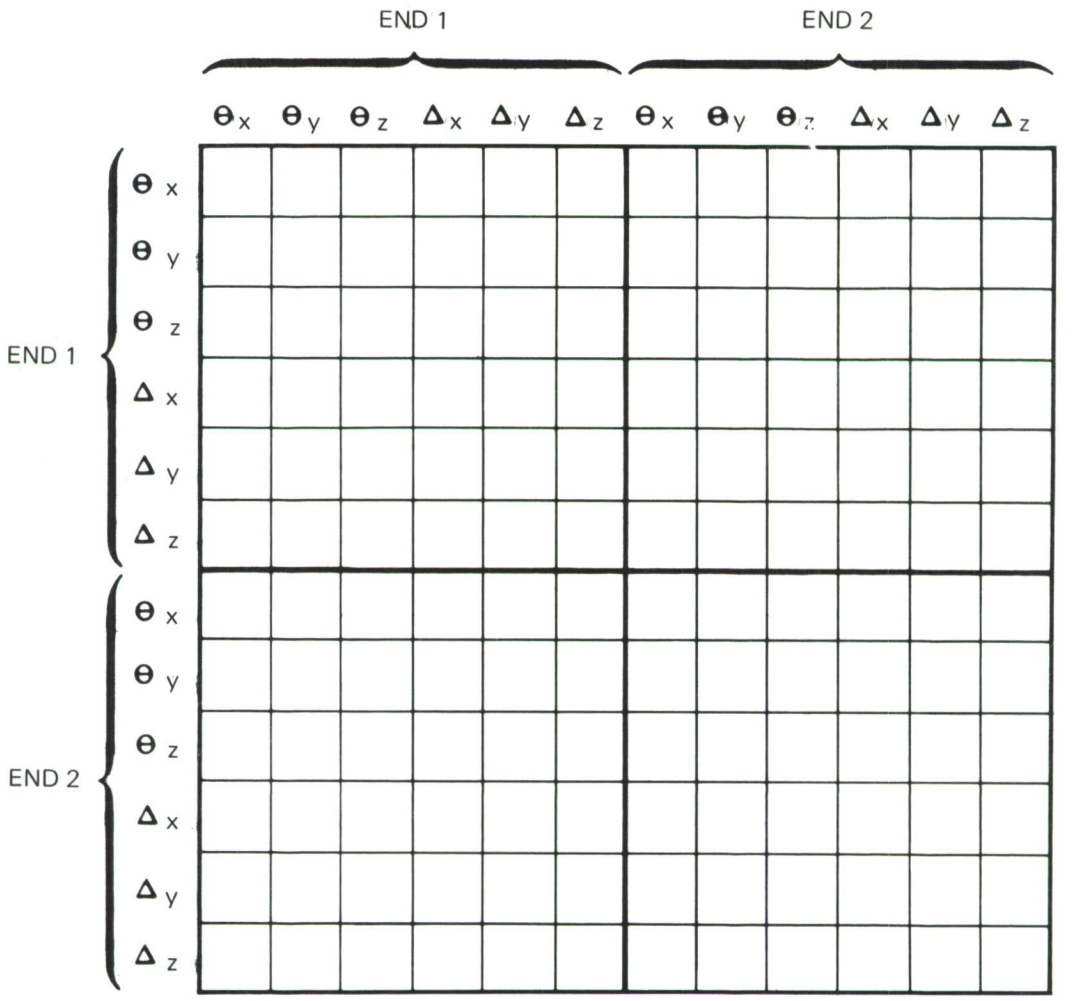


Figure 6. Elemental Beam Stiffness Matrix Layout

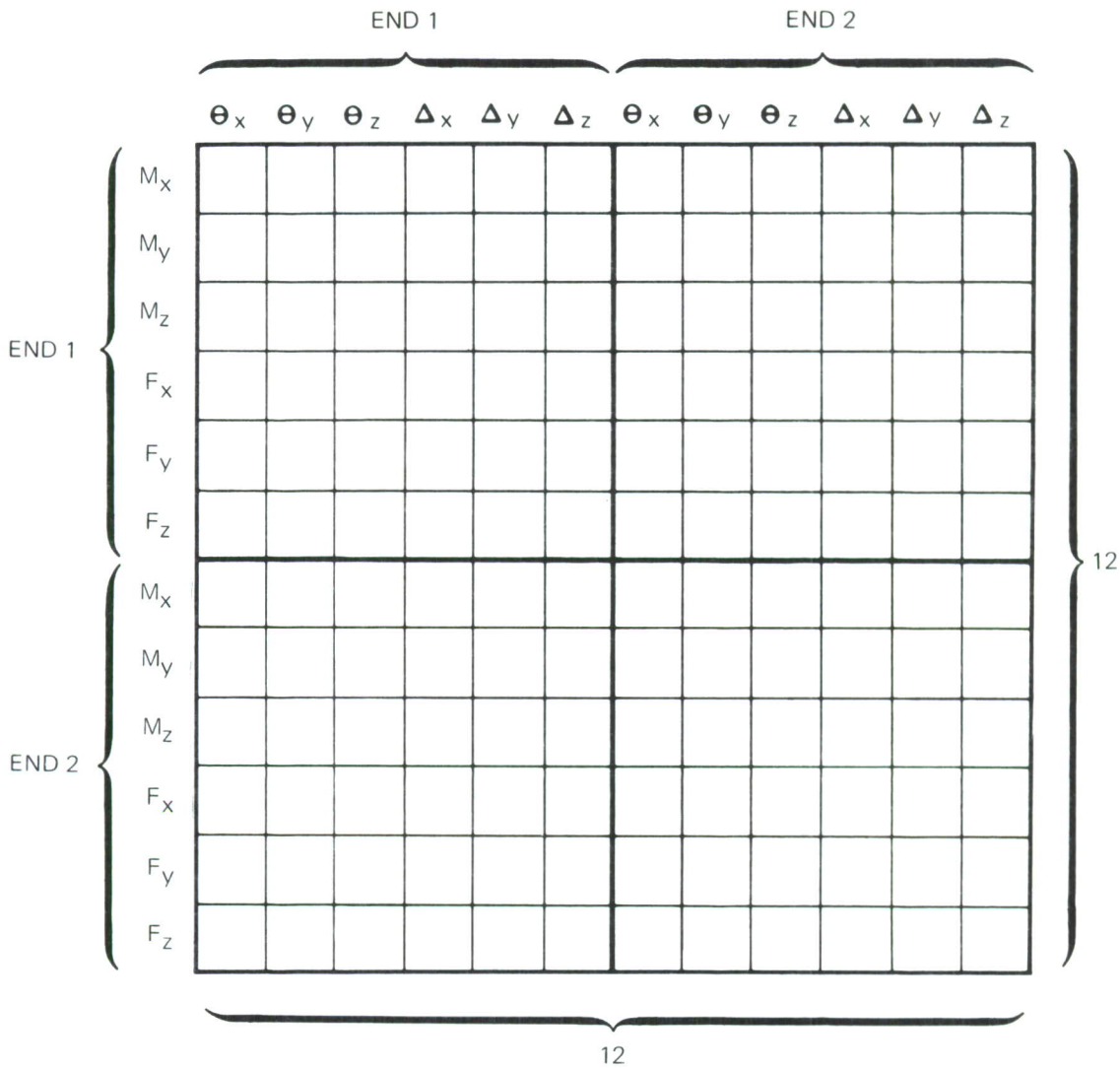
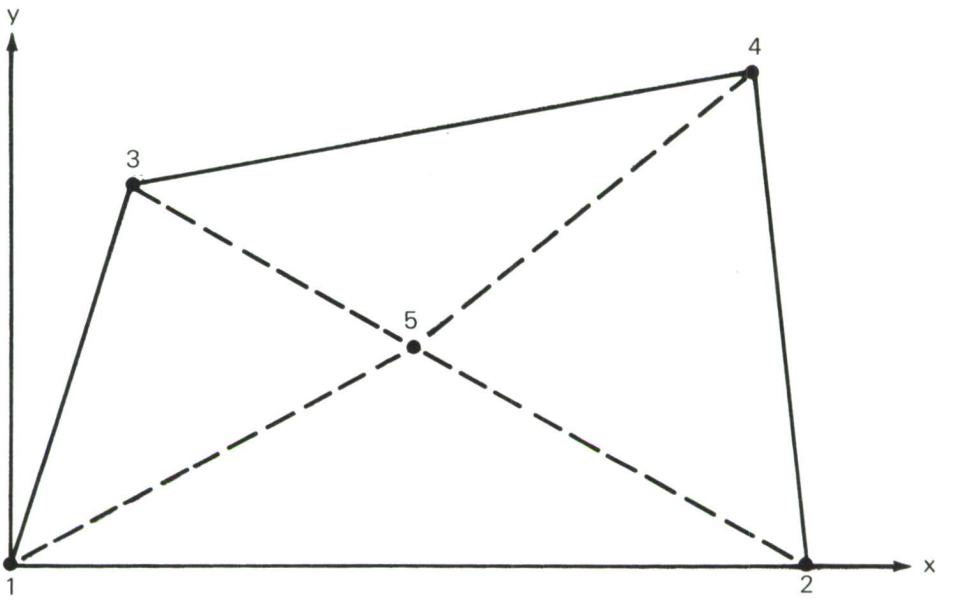


Figure 7. Elemental Beam Stress Matrix Layout



NOTE: NODE 5 IS AT CENTROID OF PLATE.

Figure 8. Quadrilateral Plate Layout

This intermediate result is then premultiplied by the stress-transformation matrix producing the elemental stress matrix (figure 9) that is saved on tape. The coordinate transformation of the stiffness matrix is then completed by premultiplying the product of the local-stiffness matrix and the transpose of the coordinate-transformation matrix by the coordinate-transformation matrix, resulting in an elemental stiffness matrix in structural coordinates (figure 10). This result is then saved on tape. As with the beams, each plate is checked to see which partitions of the merged matrix it will contribute to, and the connectivity data are updated accordingly.

(2) Structural Matrix Formation

(a) Matrix Partitioning and Identification

The structural stiffness and stress matrices are handled in partitioned form. The maximum size of these partitions is determined by the core storage limitations of the computer. The partition size for the stiffness matrix is 60 by 60, and the partition size of the stress matrix is 96 by 60. Each stiffness-matrix partition corresponds to ten nodes with six freedoms per node (figure 11), unless the option for specifying smaller partition sizes is used. The beam-stress matrix partitions contain ten nodes with six freedoms per node in the column

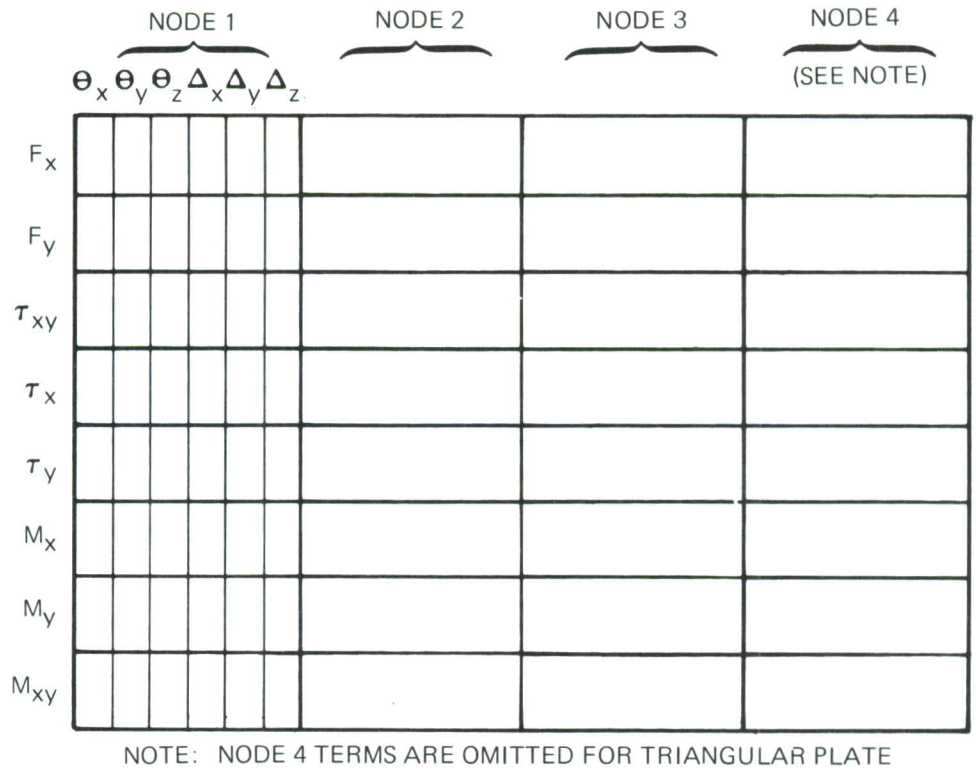


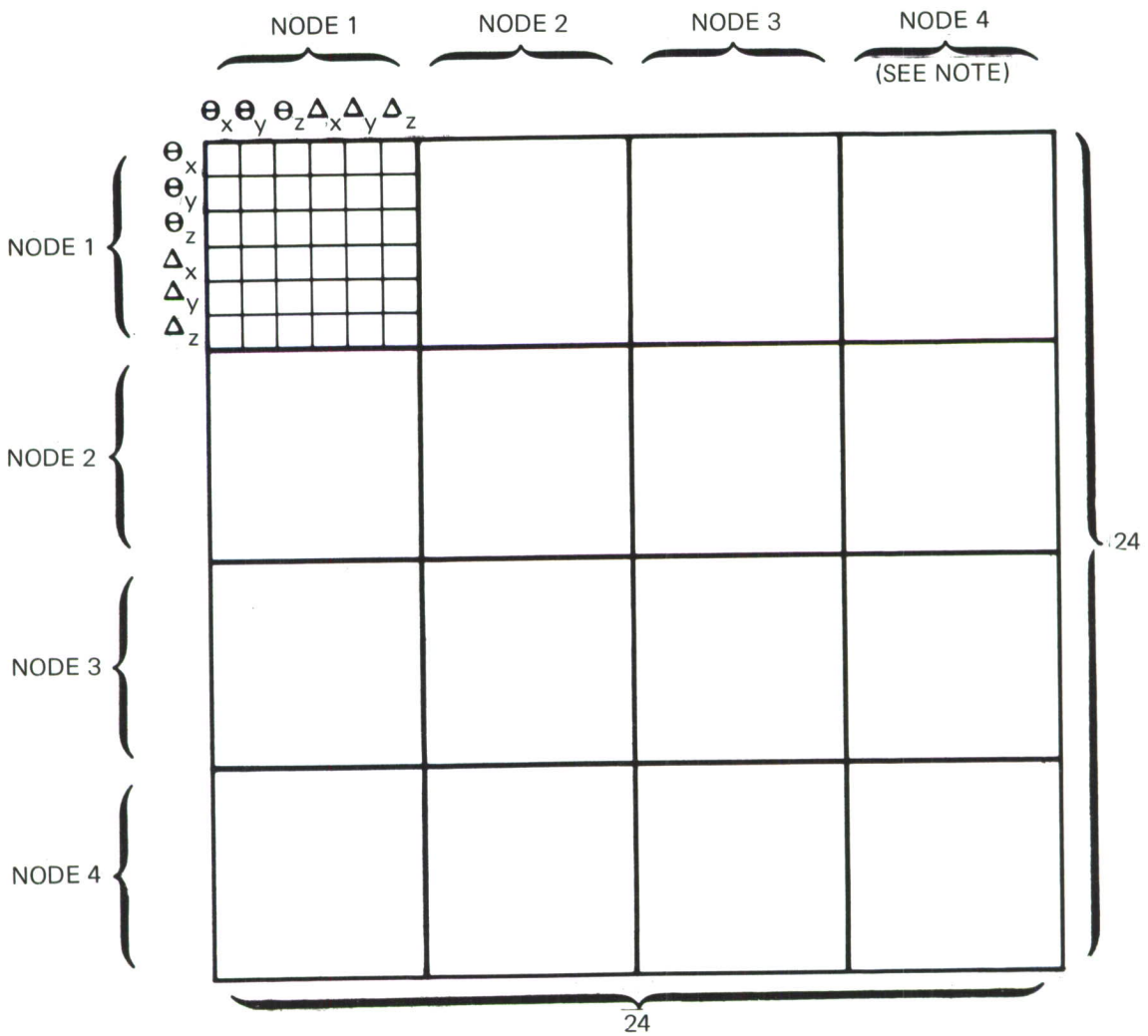
Figure 9. Elemental Plate Stress Matrix Layout

direction and eight beams with twelve stresses per beam in the row direction (figure 12). The plate-stress matrix partitions contain ten nodes in the column direction and twelve plates with eight stresses per plate in the row direction.

Each matrix partition is given a partition identification number that indicates its location in the overall matrix. The number consists of two parts: the row position and the column position (figures 11 and 12). Therefore, partition 1001 is the first partition in row 1, 1002 is the second partition of row 1, and 2001 is the first partition in row 2. Both stiffness and stress partitions are identified in this manner.

(b) Matrix Merge Procedure

The stiffness matrix is merged by partition with three partitions being merged simultaneously. The connectivity data created in the generation phase contain a list of the non-null partitions; the merge procedure is controlled by this array. The merge segment first sorts this array into ascending order of partition identification number and then begins merge by taking the first three identification numbers and merging all of the beam and plate elements that contribute to these three partitions.



NOTE: ROWS AND COLUMNS FOR NODE 4 ARE OMITTED FOR TRIANGULAR PLATES YIELDING AN (18 x 18) STIFFNESS MATRIX.

Figure 10. Elemental Plate Stiffness Matrix Layout

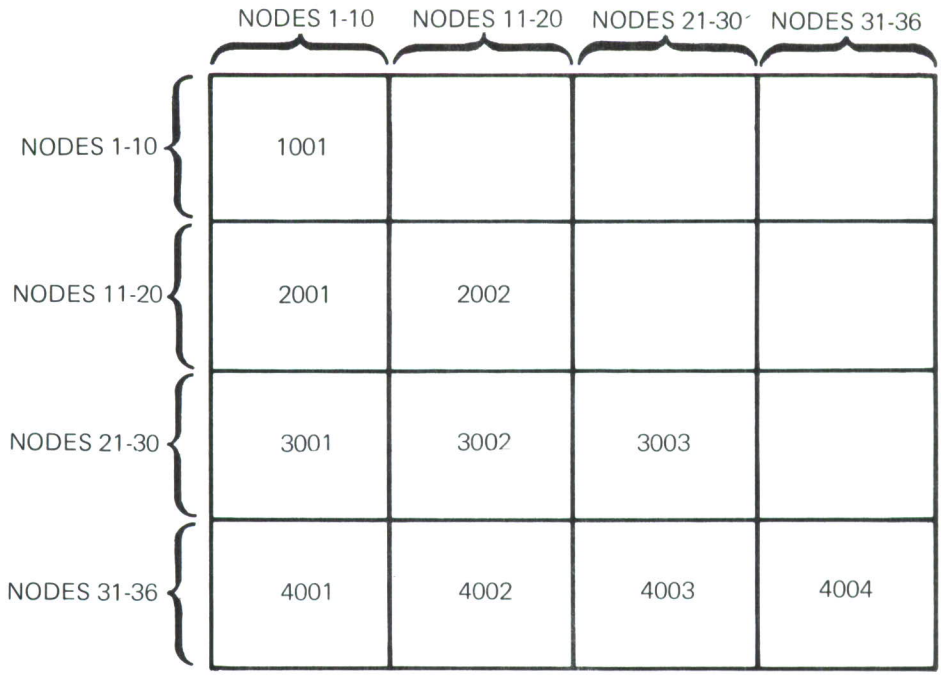


Figure 11. Structural Matrix Partitioning

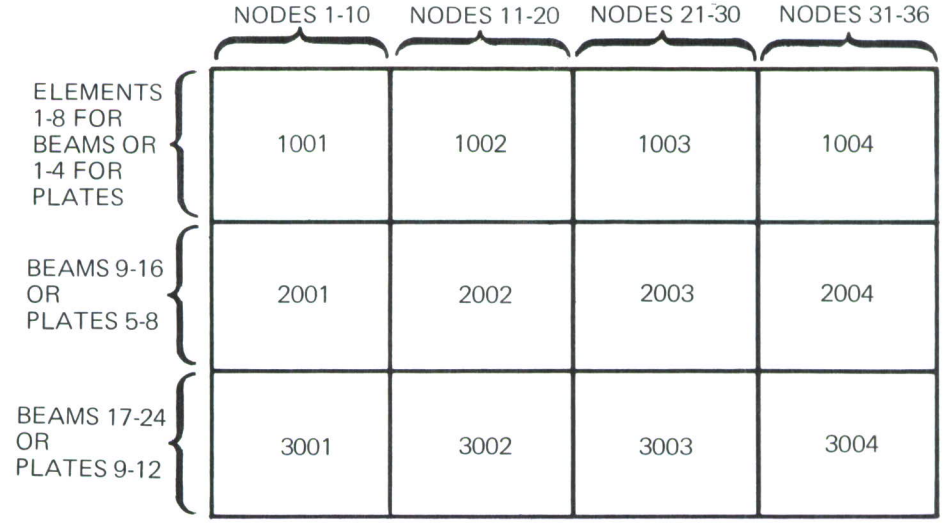


Figure 12. Merged Stress Matrix

After the merging of each group of three stiffness matrix partitions, the three partitions are sorted by applying the constraint conditions specified in the input data. The elements relating the unrestrained freedoms are sorted into the upper left-hand corner of each partition, whereas the constrained freedoms are sorted into the lower right-hand corner of the partition. The upper right-hand and lower left-hand portions of the partition contain the crosscoupling terms between the two types of freedoms (figure 13).

Only the upper left-hand portion (hereafter referred to as $[K_{ff}]$ for K-free-free) is saved on tape for later use. The other parts are discarded.

The program merges only the non-null partitions. The matrix is in lower triangular form since it is symmetrical about the diagonal.

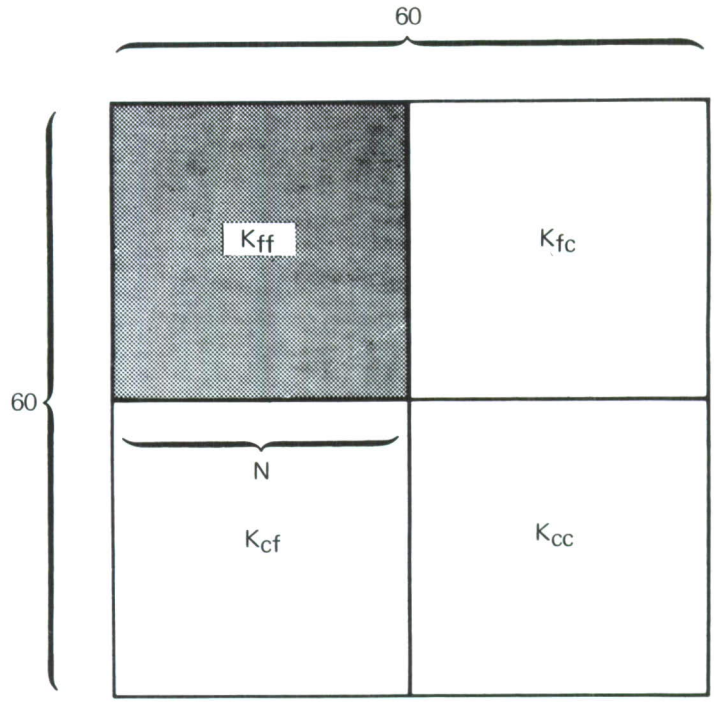
The stress matrices are merged in much the same way as the stiffness matrices with the main difference being that the stress terms do not add to each other as they do in the stiffness matrix. Again, only the non-null partitions are formed. The beam and plate stresses are merged separately, resulting in two structural stress matrices: one for beams and one for plates.

(3) Matrix Sorting and Expansion

To reduce unwanted freedoms, the $[K_{ff}]$ matrix must be sorted into four parts. The $[K_{11}]$ part relates freedoms that are to be retained. The $[K_{22}]$ part relates freedoms that are to be reduced. The $[K_{12}]$ and $[K_{21}]$ parts contain the terms of crosscoupling between the two types of freedoms. Also, since the TL01 reduction phase requires matrices in full form rather than in lower triangular, the four parts $[K_{11}]$, $[K_{12}]$, $[K_{21}]$, and $[K_{22}]$ must be expanded from lower triangular to full form.

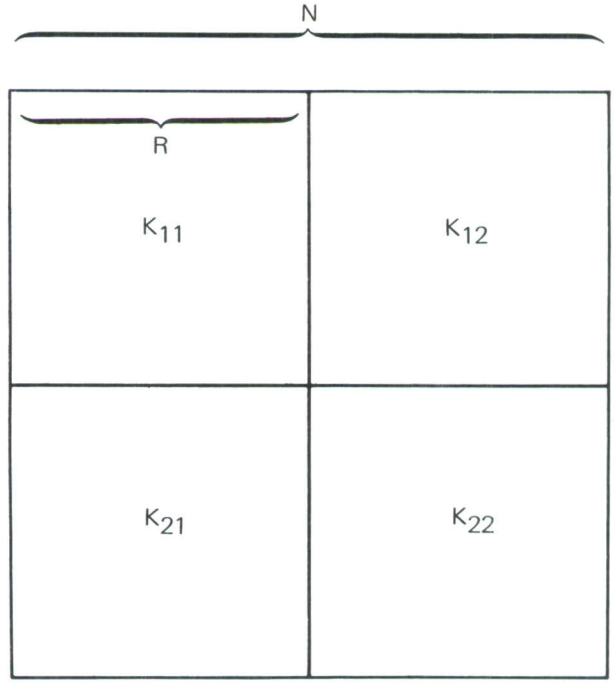
The stress matrix must be sorted into two parts in the column direction with only the first part $[S_1]$ corresponding to the $[K_{11}]$ freedoms and with the second part $[S_2]$ corresponding to the $[K_{22}]$ freedoms (figure 14).

The sorting of the $[K_{ff}]$ partitions is done one partition at a time, and the four parts are written on separate tapes. Since $[K_{ff}]$ does not contain null partitions, these must also be supplied at this time with the result being the four parts $[K_{11}]$, $[K_{12}]$, $[K_{21}]$, and $[K_{22}]$ on tapes in lower triangular form with null partitions inserted. The expansion process takes place



NOTE: $[K_{fc}]$, $[K_{cf}]$, AND $[K_{cc}]$ TERMS ARE DELETED AND ONLY $[K_{ff}]$ IS WRITTEN ON TAPE.

(a) Merged



(b) Sorted

Figure 13. Stiffness Matrix Partitions

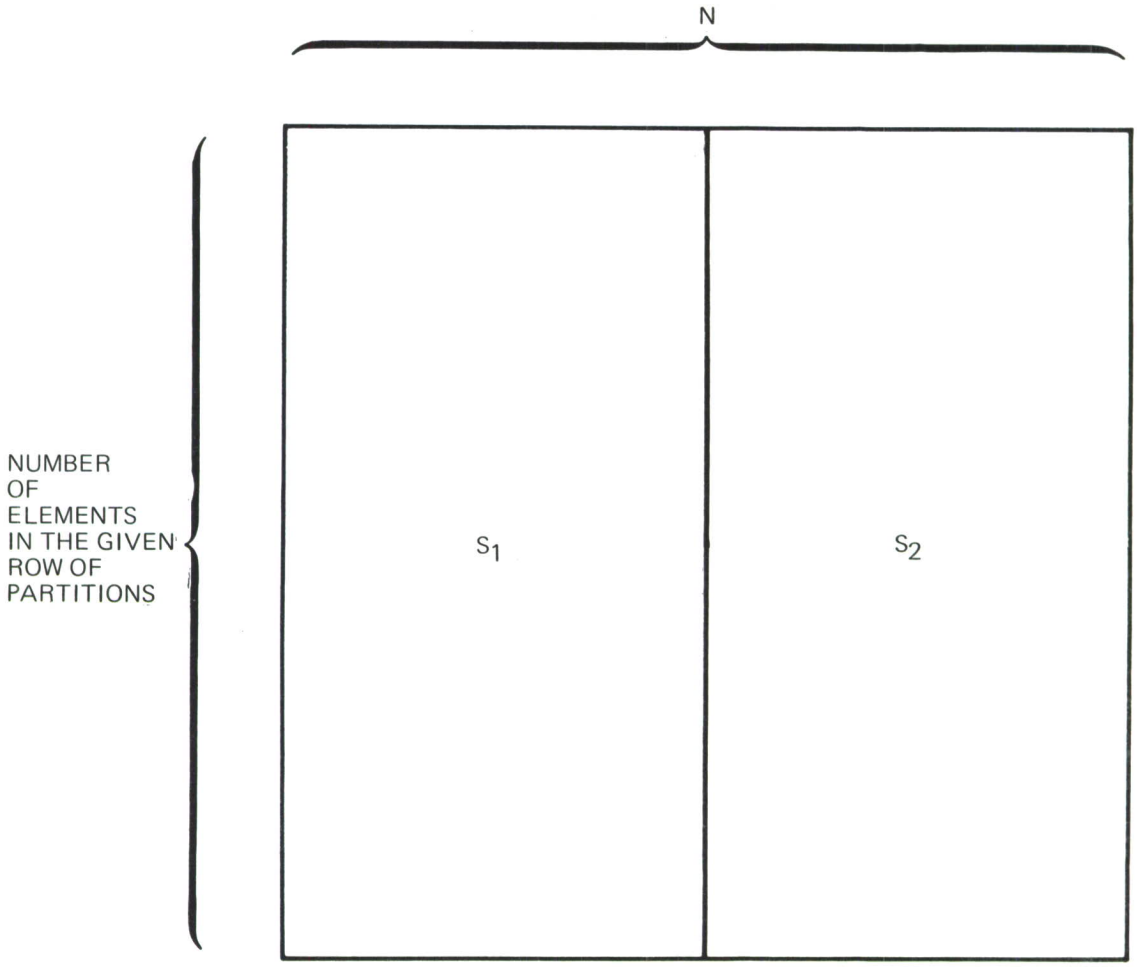


Figure 14. Sorted Stress Matrix Partition

next with $[K_{11}]$ and $[K_{22}]$ being expanded by the same subroutine (EXPAND). The expansion of $[K_{12}]$ and $[K_{21}]$, however, is complicated by the fact that a given partition in $[K_{ff}]$ need not have any retained freedoms. This results in $[K_{21}]$ and $[K_{12}]$ matrices that are not exactly lower triangular in form (figure 15). Therefore, a separate subroutine is required to perform the expansion of these two matrices (EXTRAN) with $[K_{21}]$ being formed first and $[K_{12}]$ then being written using the transpose of the appropriate $[K_{21}]$ partitions.

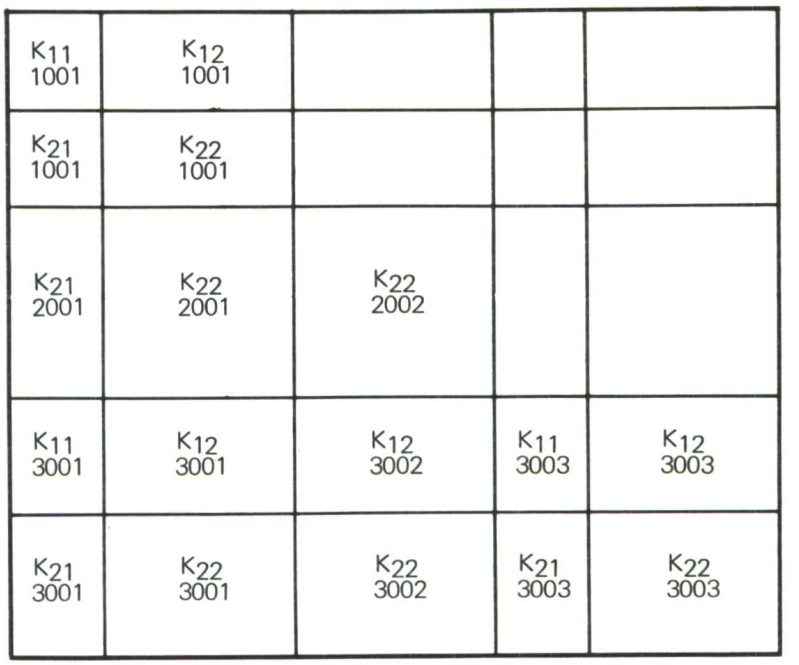
The result of this sorting and expansion is two tapes, one containing a parameter matrix in the first file (giving the number of partitions in each of the following files) and giving $[K_{11}]$ in the second file, $[K_{12}]$ in the third file, and $[K_{21}]$ in the fourth file. The $[K_{22}]$ matrix is written on a separate tape.

The stress sorting is somewhat simpler, because expansion is not required. There is, however, one complication. The stress matrices contain columns for all freedoms in the structure (both fixed and constrained). This requires that the columns representing constrained freedoms be deleted before sorting. Once this is accomplished, the two stress matrices may be sorted in a manner similar to the stiffness matrix but in the column direction only. The two parts $[S_1]$ and $[S_2]$ are written on one tape with a parameter matrix (similar to stiffness) in the first file, $[S_1]$ in the second file, and $[S_2]$ in the third file.

(4) Reduction Procedure

When the sorted stiffness and stress matrices have been written on tape in the proper form, the MAST module then calls the TL01 matrix package to form the reduced stiffness and stress matrices. The TL01 "data phases" are located on tape, and the MAST module executes them as requested by the specific program options being used (figure 16).

For general use, the program will execute data phases 1 through 3 in sequence with data phases P and B being executed only if the stress option is used. For program checkout, the reduced stiffness, reduced flexibility, and reduced stress matrices may be printed out. Refer to the listing of the MAST subroutine in paragraph 1.b.(3)(d).



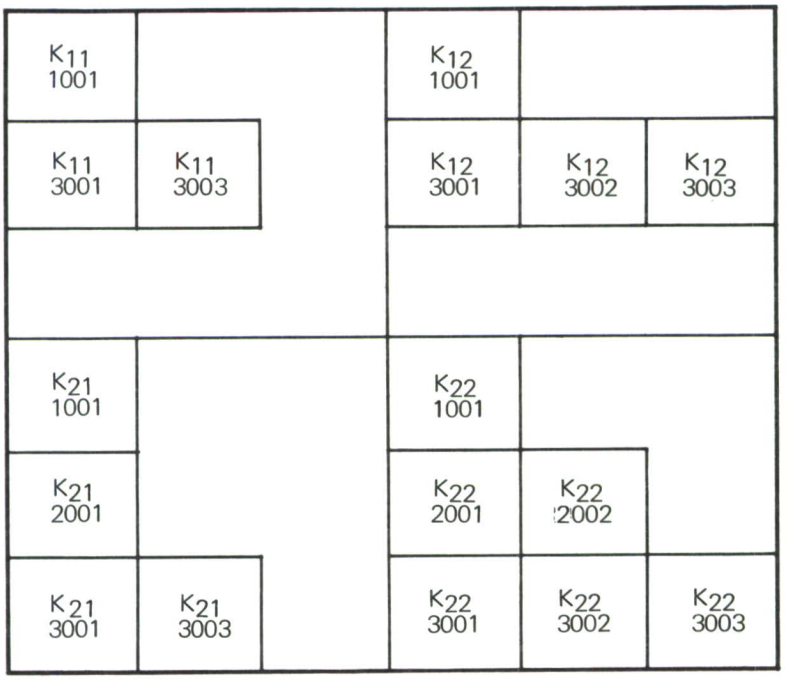


Figure 15. Sorted Stiffness Matrices

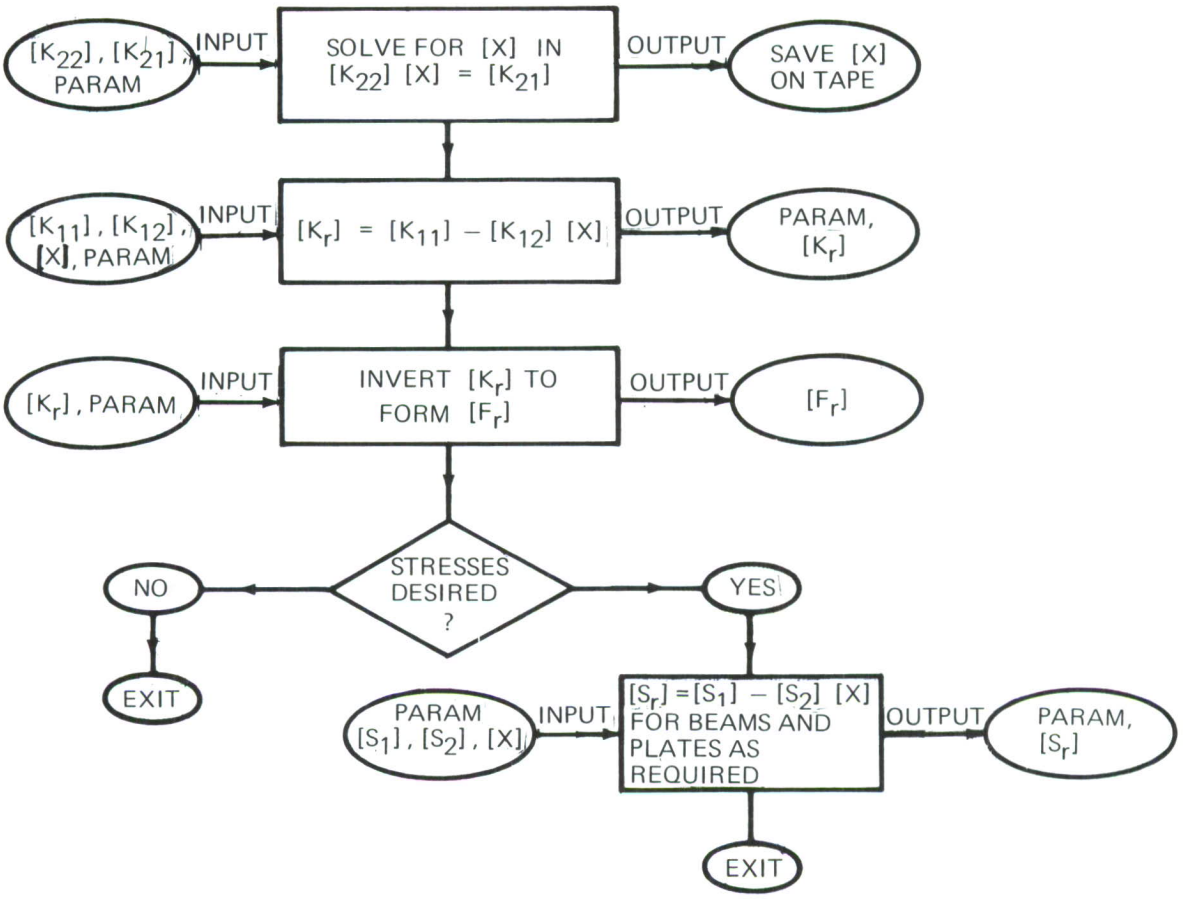


Figure 16. TL01 Phase Layout

c. Programming Organization

(1) Flow Diagrams

On the following pages are the overlay structure (figure 17) and flow diagrams (figure 18) for the MAST module.

(2) Tape Use

Figures 19 and 20 are the tape-use charts and tape format for the MAST module for the RANVIB system.

d. MAST Subroutine Listing

The following list identifies the MAST module subroutines and their functions. The subroutines are listed in the order they occur on the phase I master tape.

Subroutine	Function
MAST	Reads control cards and controls execution of MAST module
PAGHED	Prints page heading
UNPACK	Unpacks constraint-condition data
PRINT	Prints matrices or vectors
SUBM1	Controls generation of merge
GENRAT	Controls element generation
REDUCE	Reduces freedoms from elemental matrices
INFO	Reads in nodal data
PLATE	Reads plate data and controls
MUL1	Matrix multiplication, $[C] = [A][B]$
MUL2	Matrix multiplication, $[C] = [A]^T [B]$
PSTIF	Controls local-coordinate generation and elemental stiffness matrix generation for plates
QUAD	Controls elemental stiffness generation for quadrilateral plates
LAMK	Performs coordinate transformation from local-stiffness to structural stiffness matrices

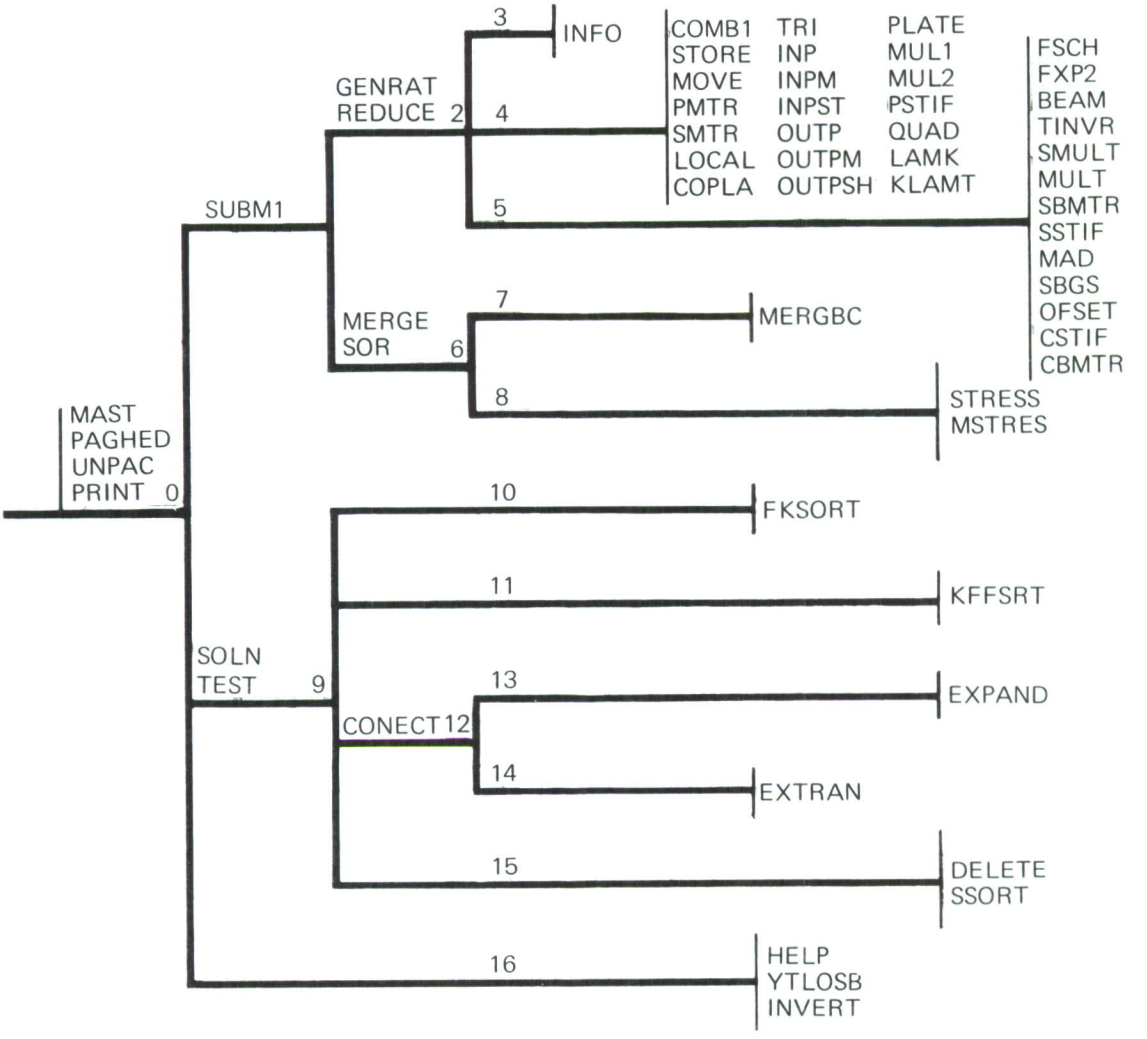


Figure 17. Mast Overlay Structure

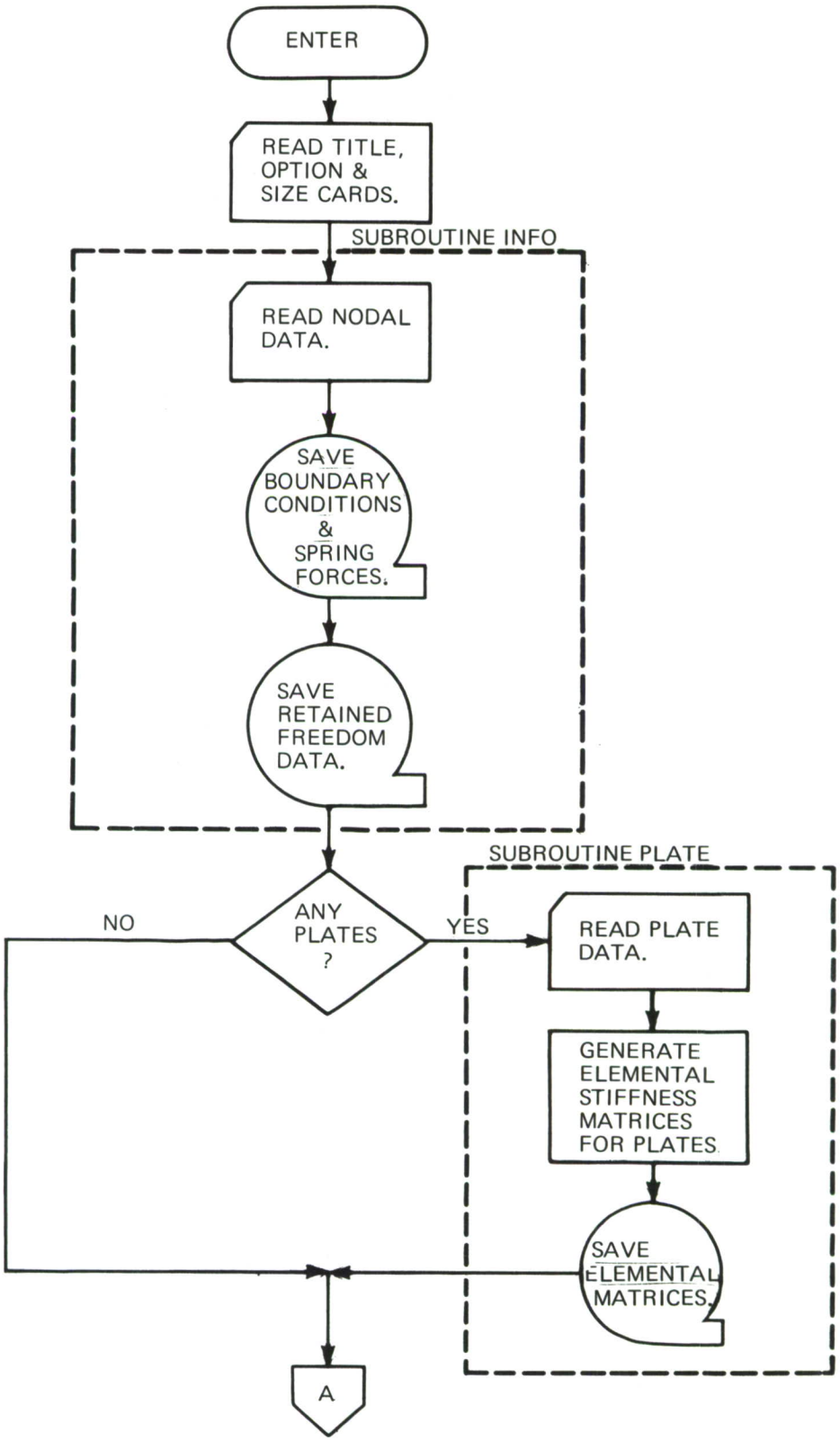


Figure 18. MAST Flow Chart

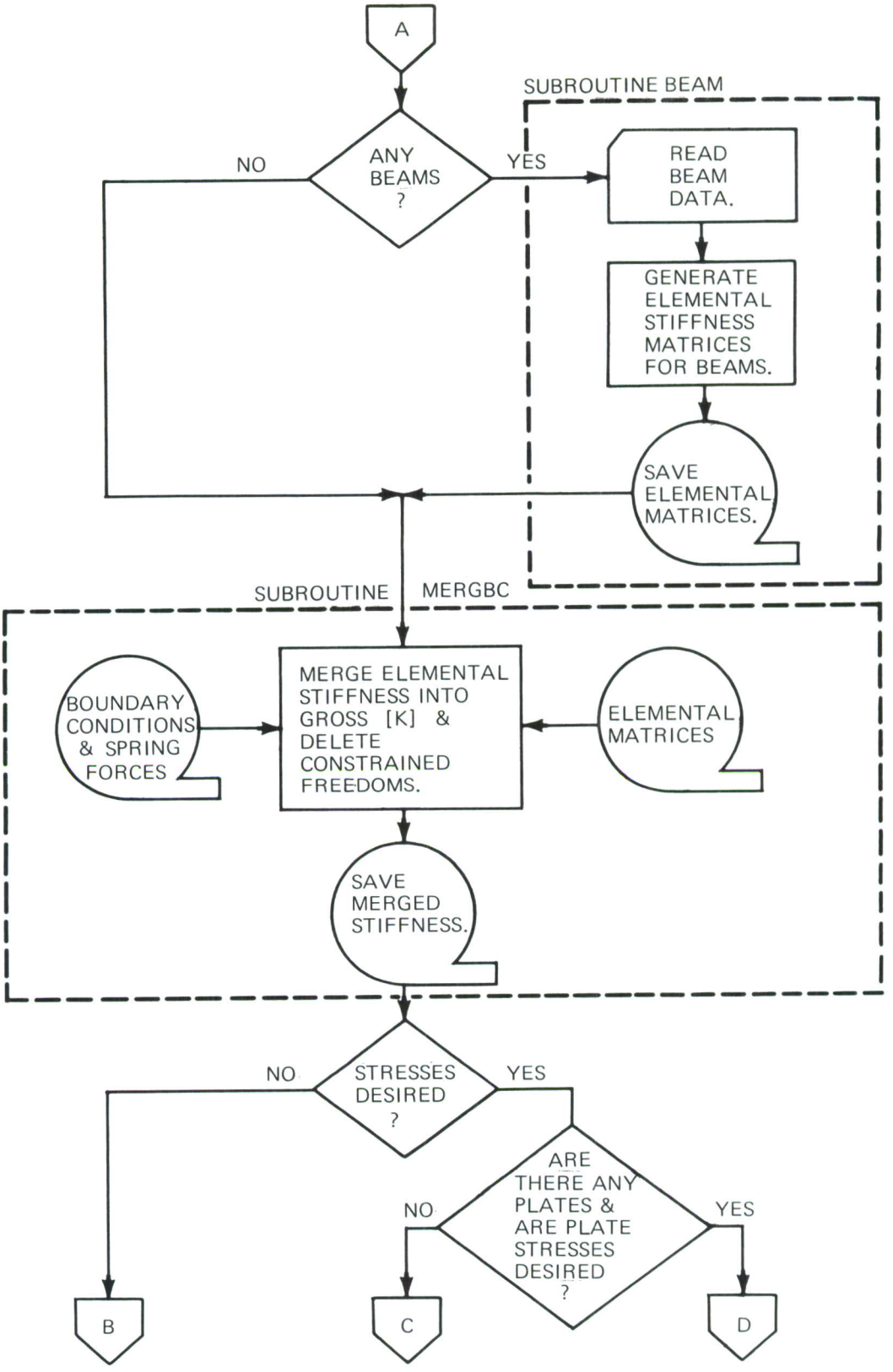


Figure 18.—Continued

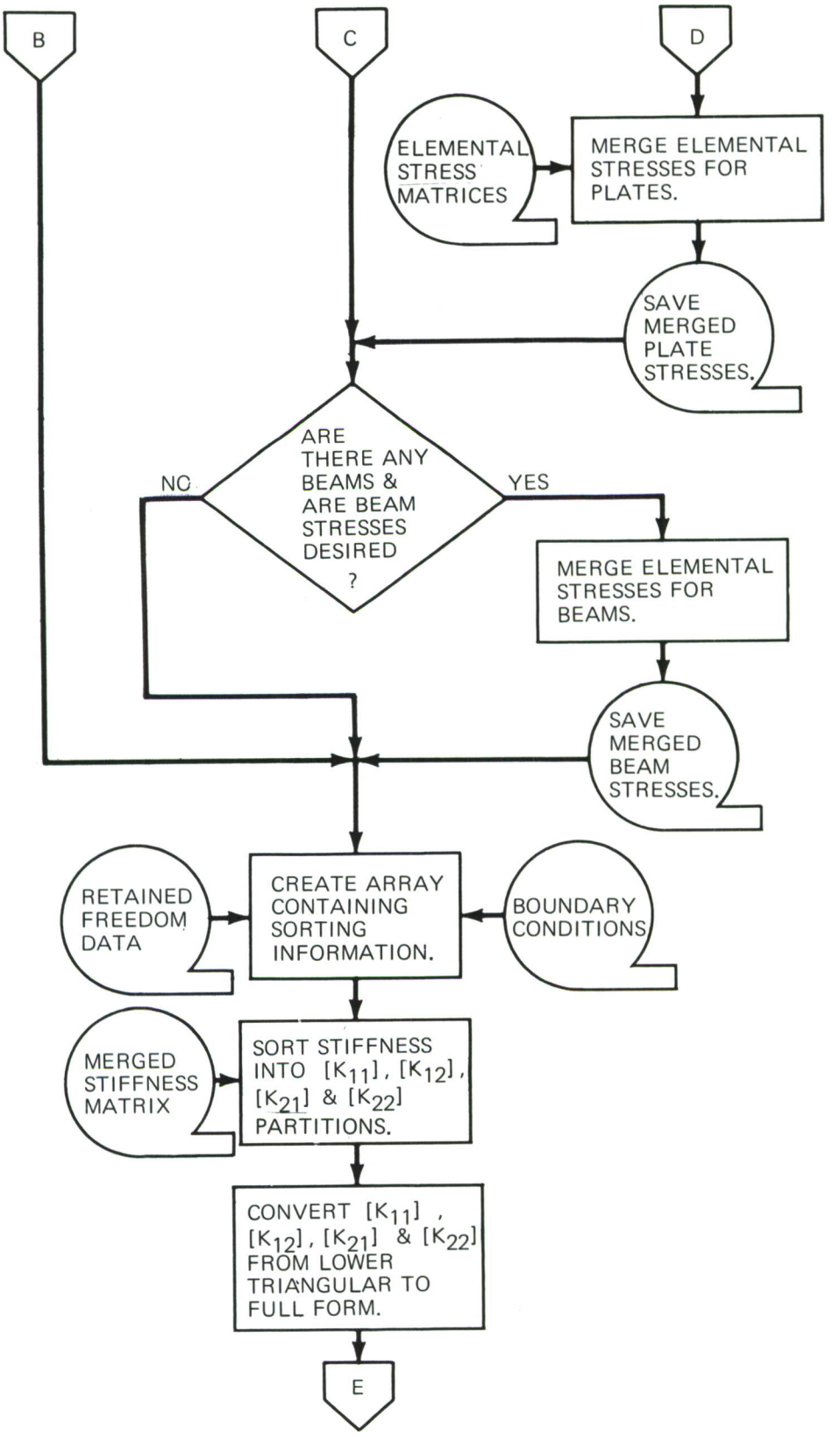


Figure 18—Continued

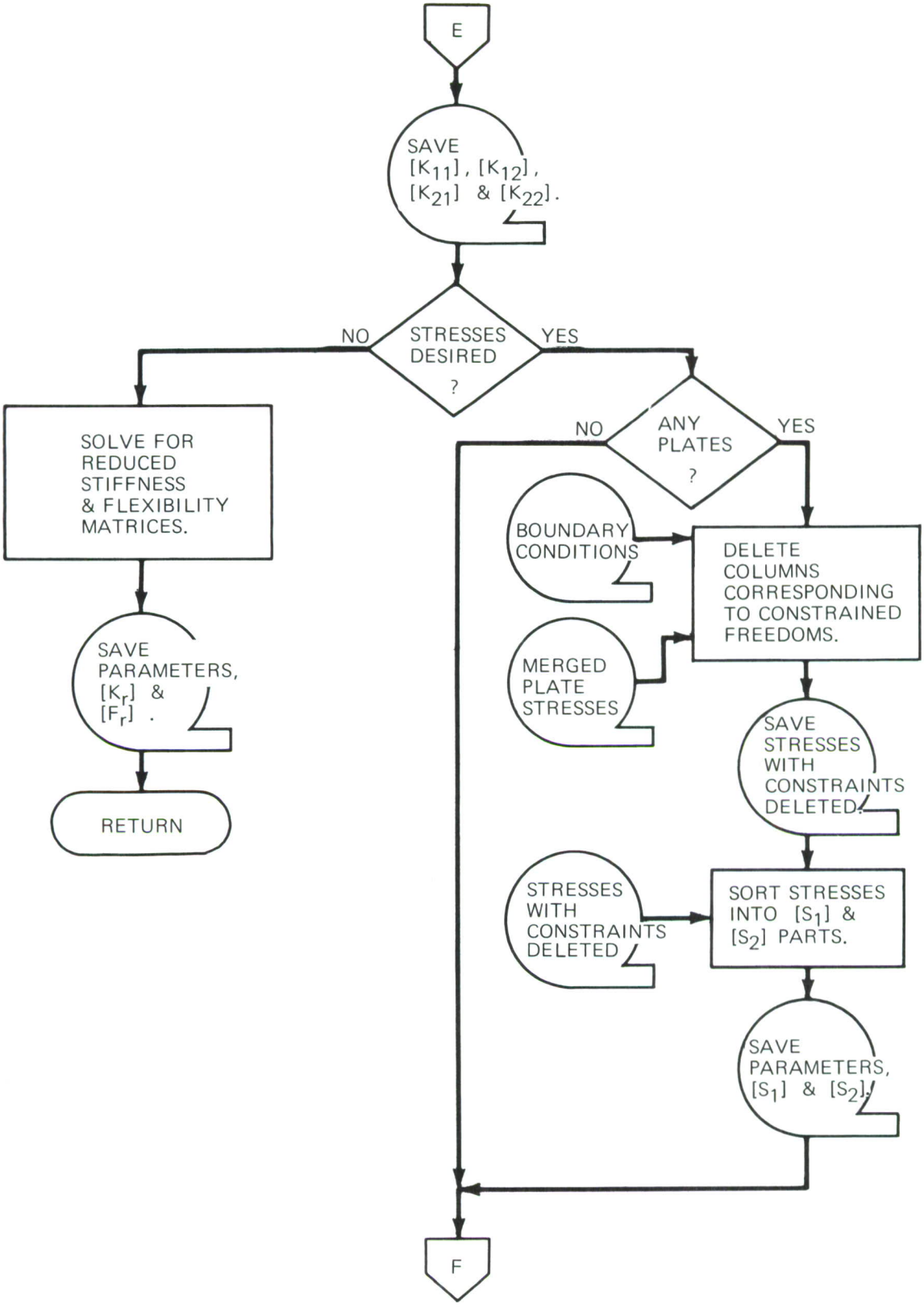


Figure 18–Continued

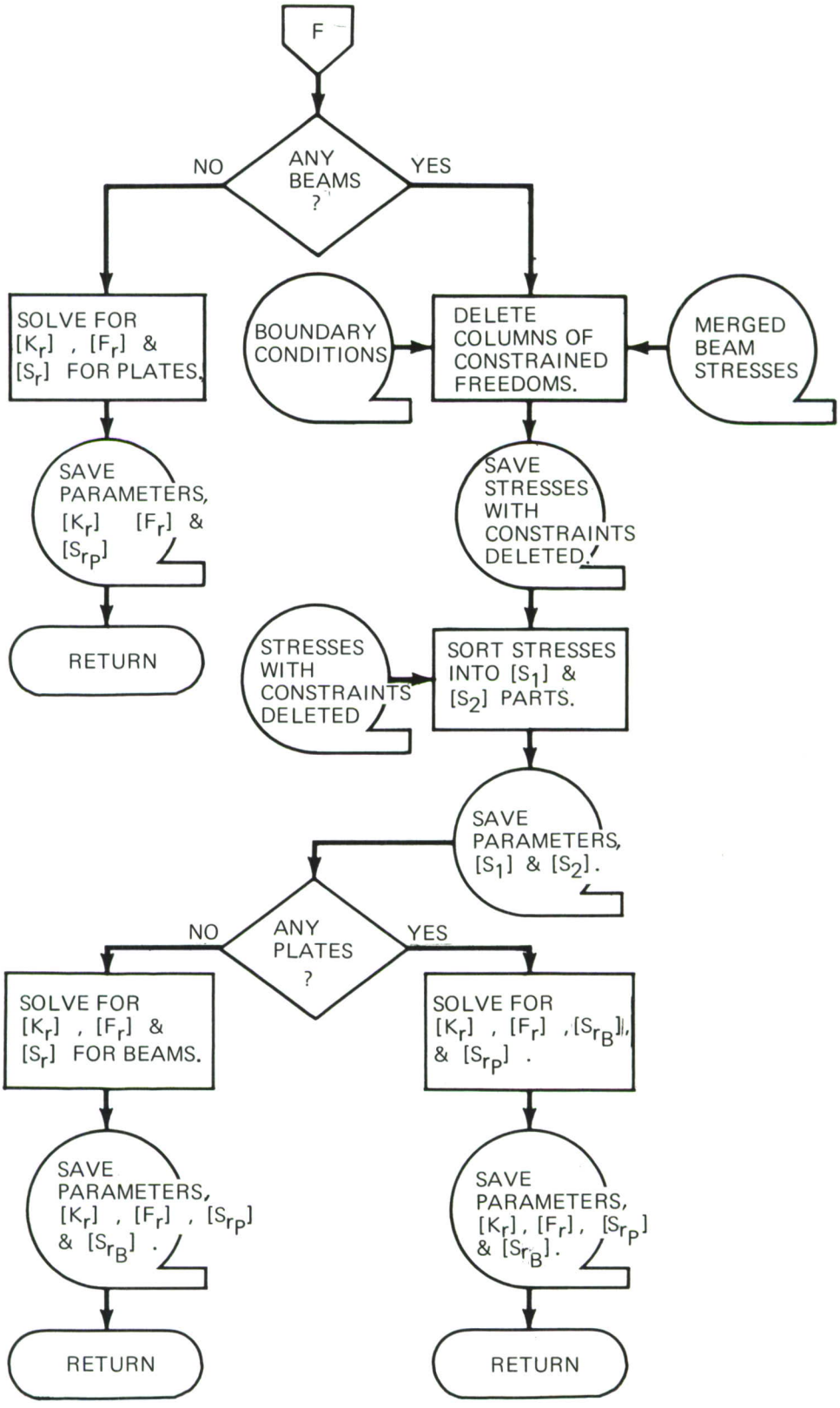


Figure 18.—Concluded

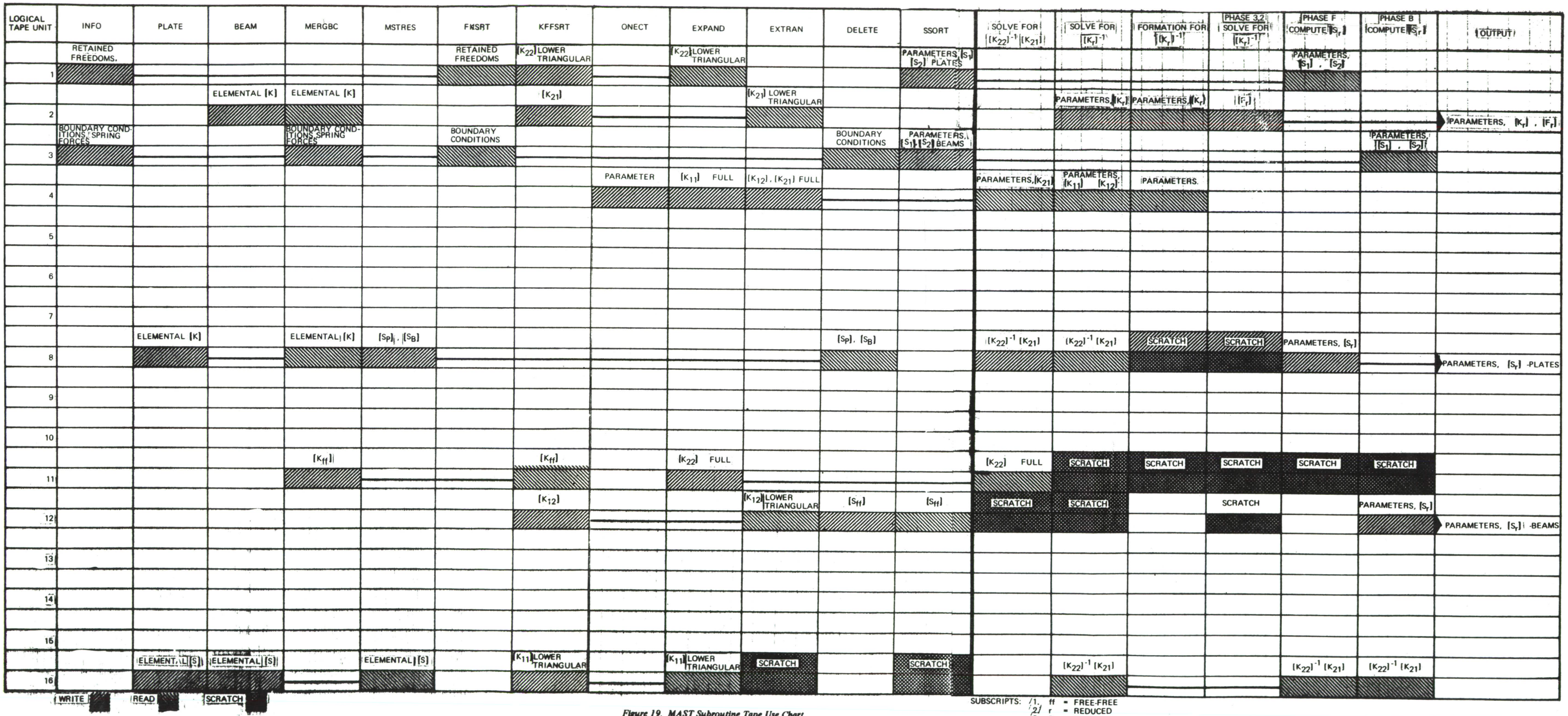


Figure 19. MAST Subroutine Tape Use Chart

SUBSCRIPTS: /1, ff = FREE-FREE
/2/ r = REDUCED
3. B = BEAMS
4. P = PLATES

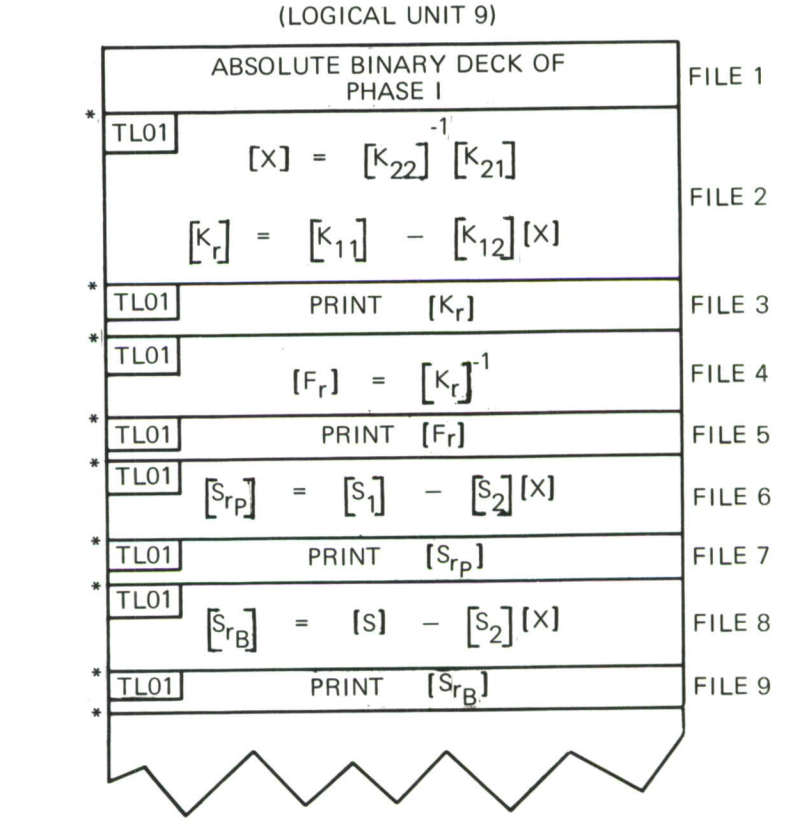


Figure 20. Phase I Master Tape Detail

Subroutine	Function
KLAMT	Performs coordinate transformation from local-stiffness to structural stiffness matrices
TRI	Controls generation of elemental stiffness matrices for triangular plates
INP	Controls generation of in-plane plate stiffness terms
INPM	Generates in-plane moment plate terms
INPST	Generates in-plane plate stretching terms
OUTP	Controls generation of out-of-plane plate stiffness terms
OUTPM	Generates out-of-plane plate moment terms
OUTPSH	Generates out-of-plane plate shear terms
COMBIN	Combines in-plane terms or out-of-plane terms
STORE	Stores in-plane and out-of-plane terms in elemental stiffness matrix
MOVE	Matrix addition, $[B] = [A] + [B]$
PMTR	Generates coordinate-transformation matrix for triangular or quadrilateral plates
SMTR	Generates stress-transformation matrix for triangular or quadrilateral plates
LOCAL	Generates local coordinates for plates and checks for re-entrant corner and node sequencing
COPLAN	Checks quadrilateral plates for coplanarity
BEAM	Reads beam data and controls the generation of beam-element stiffness matrices
TINVR	Inverts beam flexibility matrix to get beam stiffness
SMULT	Matrix multiplication-subtraction, $[C] = [C] - [A][B]$
MULT	Matrix multiplication-addition, $[C] = [C] + [A][B]$ or $[C] = [C][A]^T[B]$ or may store results in $[B]$

Subroutine	Function
SBMTR	Generates transformation matrix for straight beams
SSTIF	Generates stiffness matrix for straight beams
MAD	Matrix addition for 12 by 12 only, $[A] = [A] + [B]$
SBGS	Generates geometric stiffness matrices for straight beams
OFST	Generates offset-transformation matrix for beams
CSTIF	Generates elemental stiffness matrix for curved beam
CBMTR	Generates transformation matrix for curved beams
MERGE	Controls merge of structural stiffness and stress matrices
SOR	Sorts the partition identification number list prior to merge of stiffness
MERGBC	Merges the structural stiffness matrix
STRESS	Controls merge of the structural stress matrices
MSTRES	Merges the structural stress matrices
SOLN	Controls the sorting of the structural stiffness and stress matrices
TEST	Tests the list of partition identifications to determine if a given partition number is present
FKSORT	Creates the control array used to sort the stiffness and stress matrices
KFFSRT	Sorts the structural stiffness matrix into retained and reduced freedoms
CONECT	Controls the expansion of the sorted stiffness matrix into full form
EXPAND	Expands the $[K_{11}]$ and $[K_{22}]$ matrices into full form
EXTRAN	Expands the $[K_{12}]$ and $[K_{21}]$ matrices into full form
DELETE	Deletes the columns of constrained freedoms from the structural stress matrices

Subroutine	Function
SSORT	Sorts the structural stress matrices into retained and reduced parts
DATA PHASE I	Solves the equation $[K_{22}][X] = [K_{21}]$
DATA PHASE II	Calculates $[K_r] = [K_{11}] - [K_{12}][X]$
DATA PHASE III	Calculates $[S_r] = [S_1] - [S_2][X]$ (beams)
DATA PHASE IV	Calculates $[S_r] = [S_1] - [S_2][X]$ (plates)

2. INTERMEDIATE MATRIX MERGE

The stiffness and flexibility matrices are merged in AMERGE to an N-by-N matrix from the partitioned matrices output on tape from the MAST module.

The stresses are merged and re-partitioned in SMERGE to 8-by-8 matrices for plates and 6-by-6 matrices for beams.

The merged flexibility matrix is used in the FREMOD module. The stiffness matrix is only used when a phase II option 1 solution is desired. The partitioned stresses are used when the stresses in phase II are wanted.

a. Subroutine AMERGE (ITAPE, NTAPE, NF1)

This subroutine merges the stiffness [K] or flexibility [F] matrices.

Method: The matrices [K] or [F] that are formed in subroutine MAST and stored in row-partitioned form are merged to form a N-by-N matrix and stored on phase I output tape.

Input: Stiffness/flexibility matrix on tape 2

Output: Merged stiffness/flexibility matrix on tape 10

Error: READTP/WRTETP error messages

Subroutines required: READTP/WRTETP

Argument list: ITAPE—Input tape of stiffness/flexibility
NTAPE—Output tape of stiffness/flexibility
NF1—Number of file marks to skip past before reading starts. NF1 = 1 for the stiffness matrix and NF1 = 2 for the flexibility matrix.

Length: 33215_8

Flow chart: See figure 21.

b. Subroutine SMERGE

Method: This subroutine merges the stresses from the MAST subroutine and re-partitions and stores the matrices on the phase I output tape. The stresses for plates and beams are re-partitioned to 8-by-8 and 6-by-6 matrices, respectively.

Input: Parameter matrix and stress matrices for plates and beams from ITAPE

Output: The re-partitioned stress matrices are stored on NTAPE. The matrices for plates are stored first, then the beam stresses are stored. Either or both stresses may be stored on tape.

Error: READTP/WRTETP error messages

Subroutines required: READTP/WRTETP

Argument list: ITAPE—Input stress matrices are on this tape.
NTAPE—Output of the re-partitioned stress matrices are stored on this tape.
ITEST—This variable is set to 8 if the plate stresses are to be re-partitioned; 6 if the beam stresses are to be re-partitioned.

Length: 40472_8

Flow chart: See figure 22.

3. VIBRATION PROGRAM (FREMOD)

a. General Description

The purpose of the FREMOD module is to calculate the natural frequencies and normal mode shapes, given the flexibility matrix and mass matrix. This is done by solving the dynamic matrix for eigenvalues and eigenvectors using the QR algorithm.

See the macro flow chart (figure 23) and organization chart (figure 24).

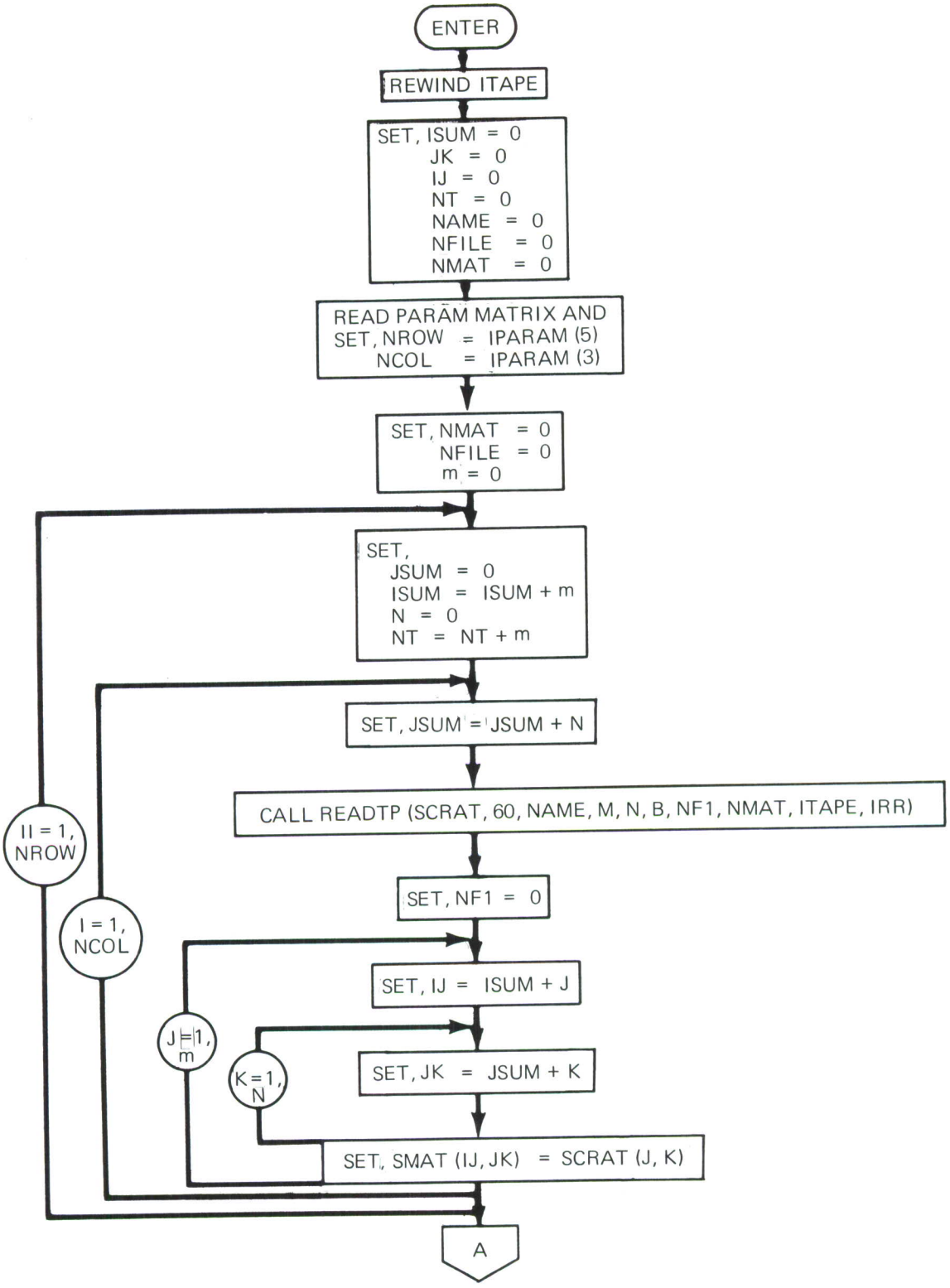


Figure 21. AMERGE Flow Chart

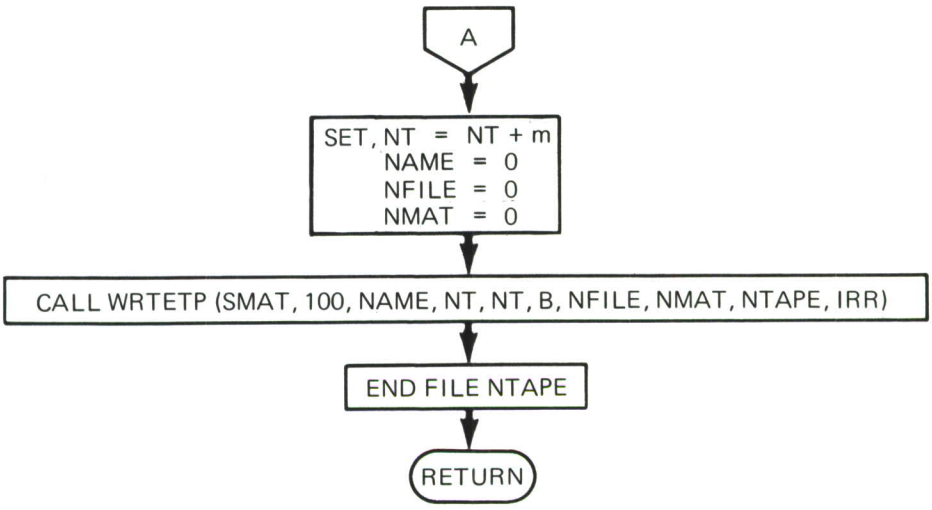


Figure 21—Concluded

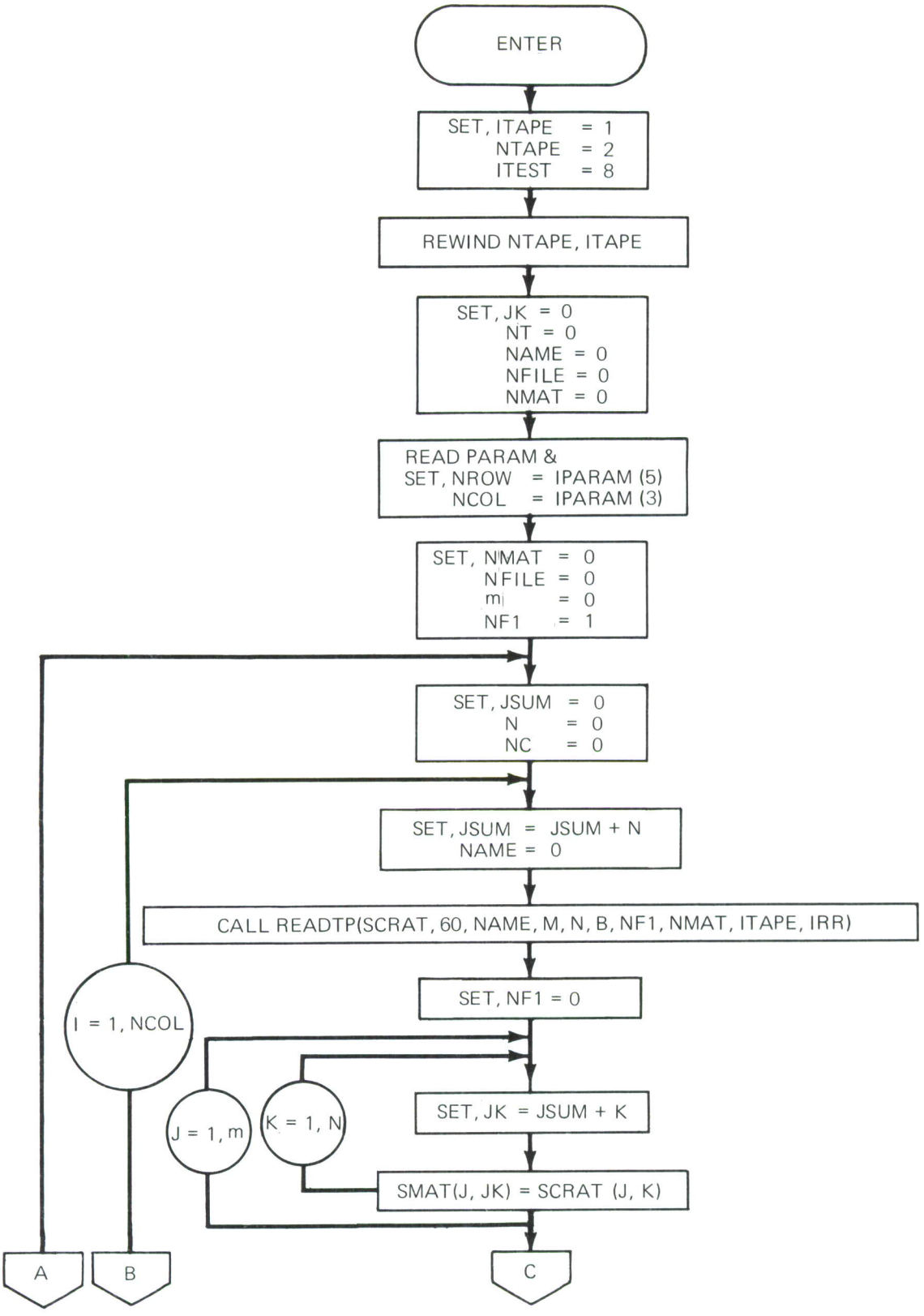


Figure 22. SMERGE Flow Chart

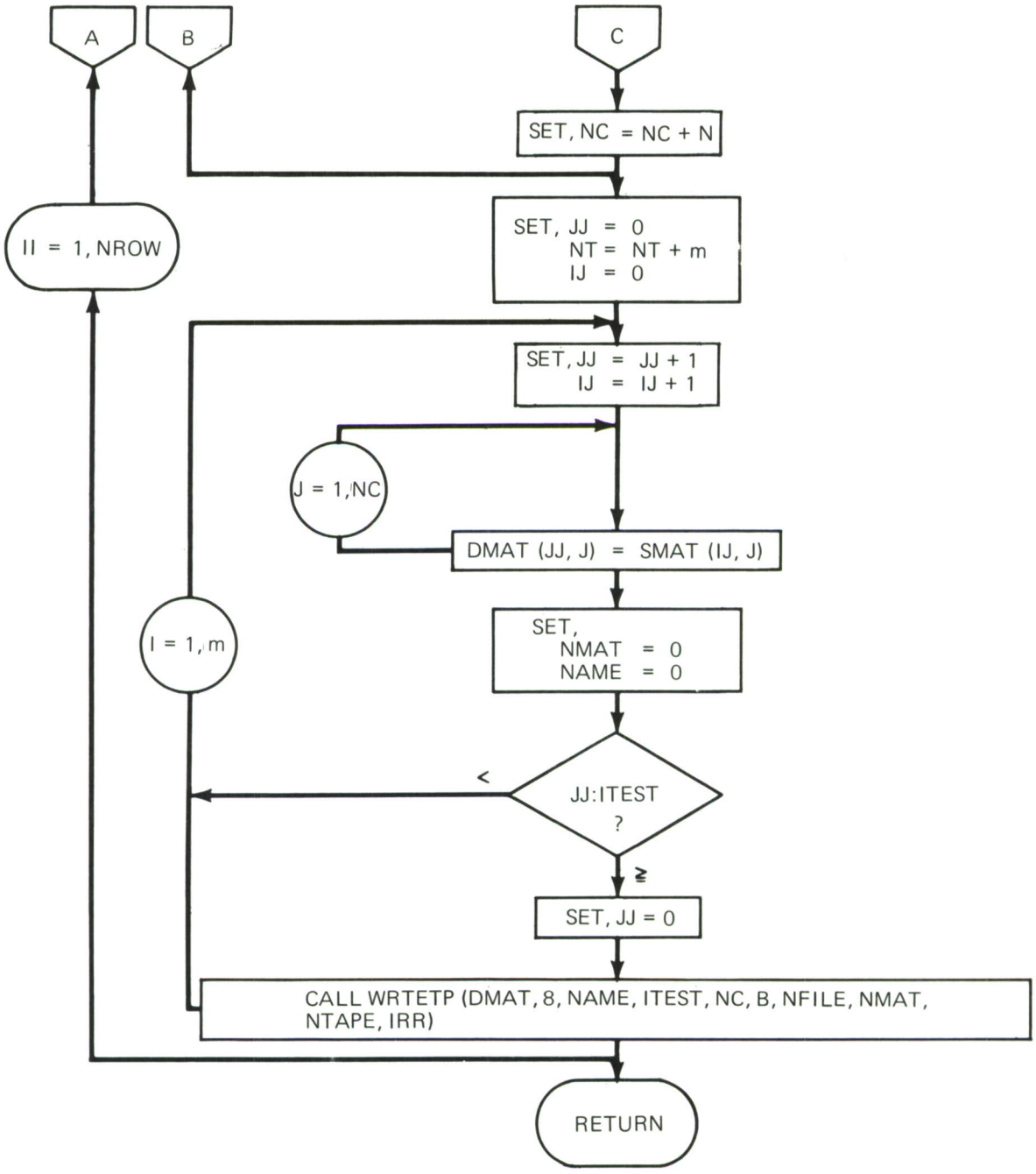


Figure 22—Concluded

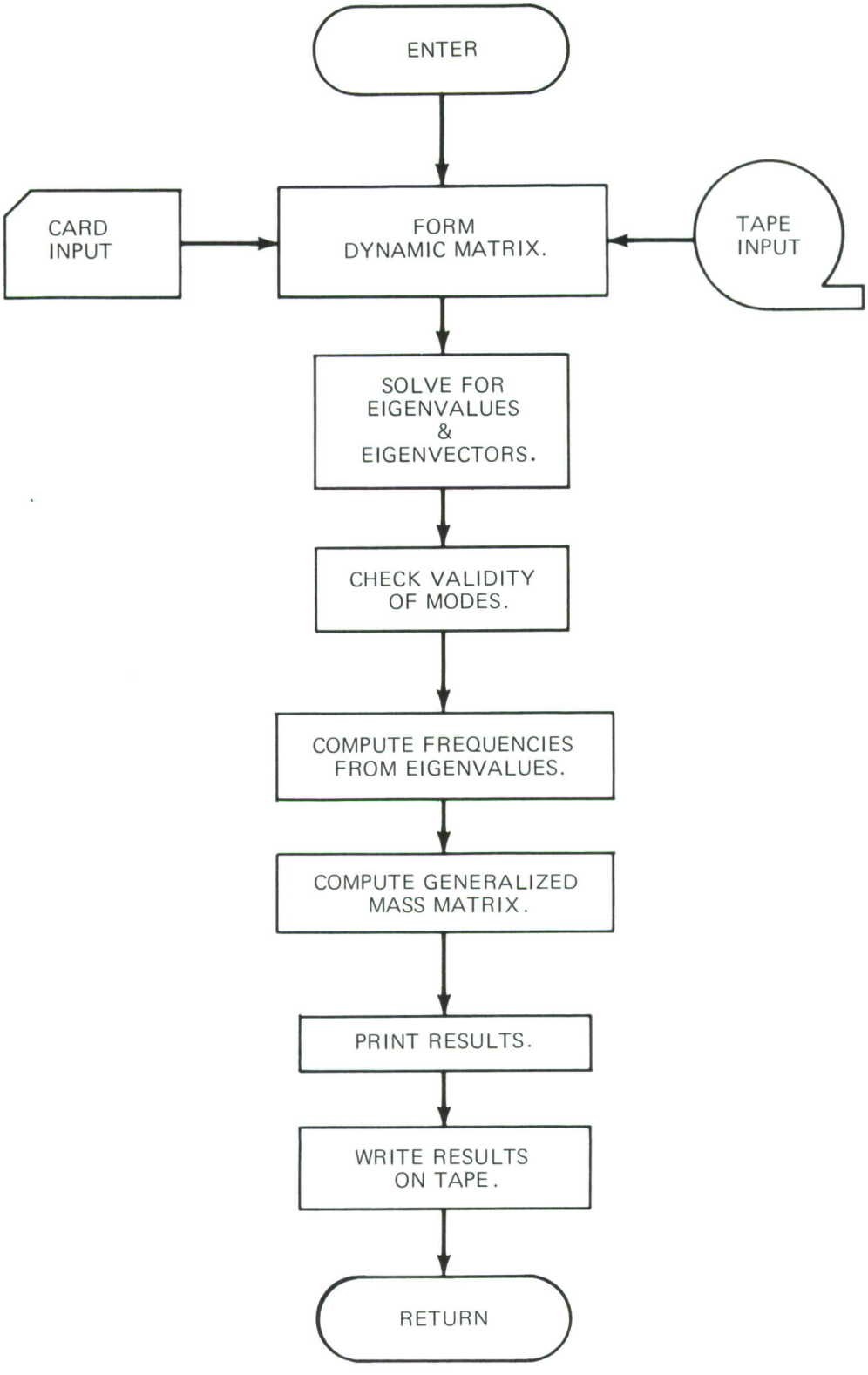


Figure 23. FREMOD Macro Flow Chart

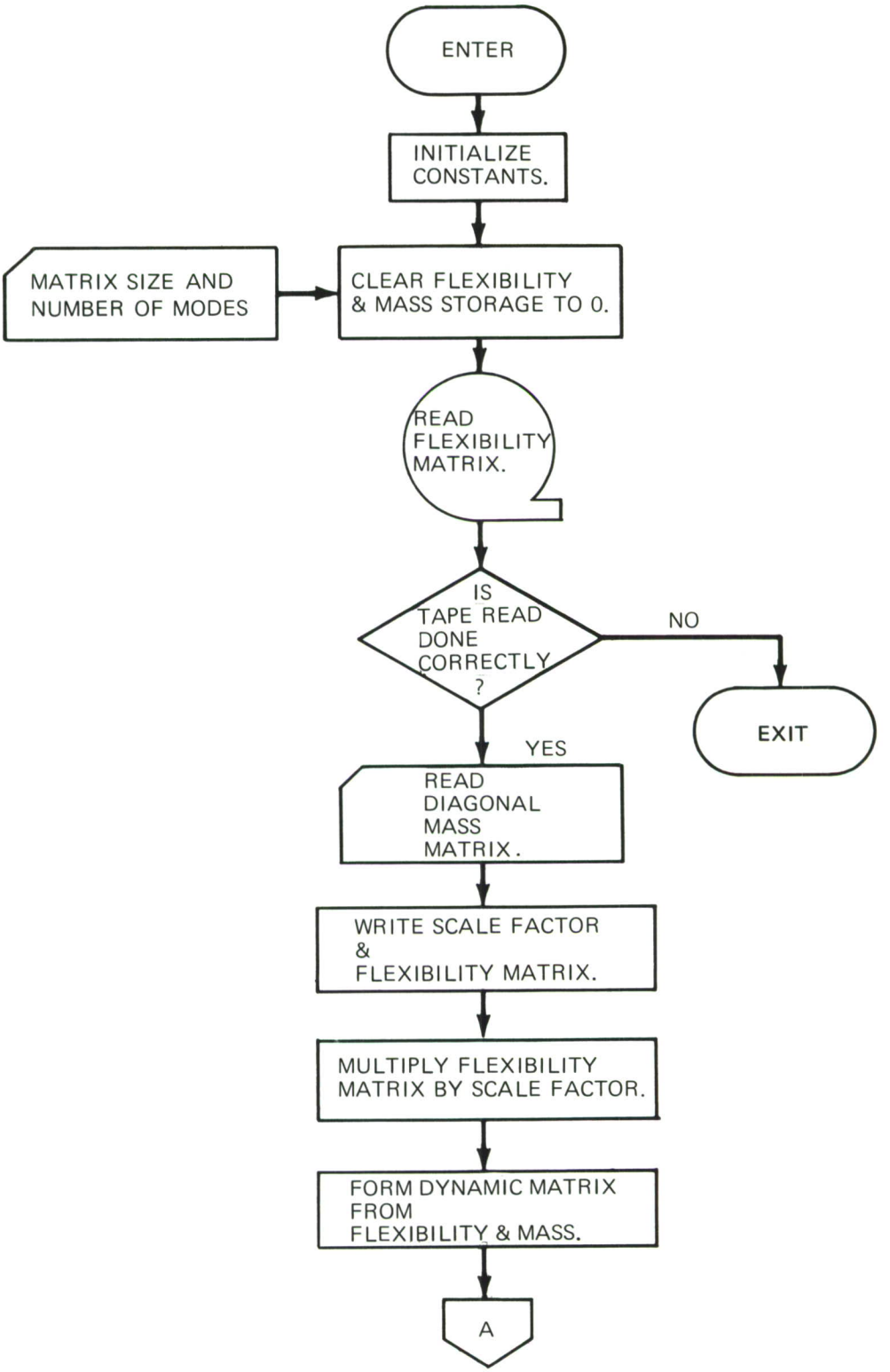


Figure 24. FREMOD Program Organization

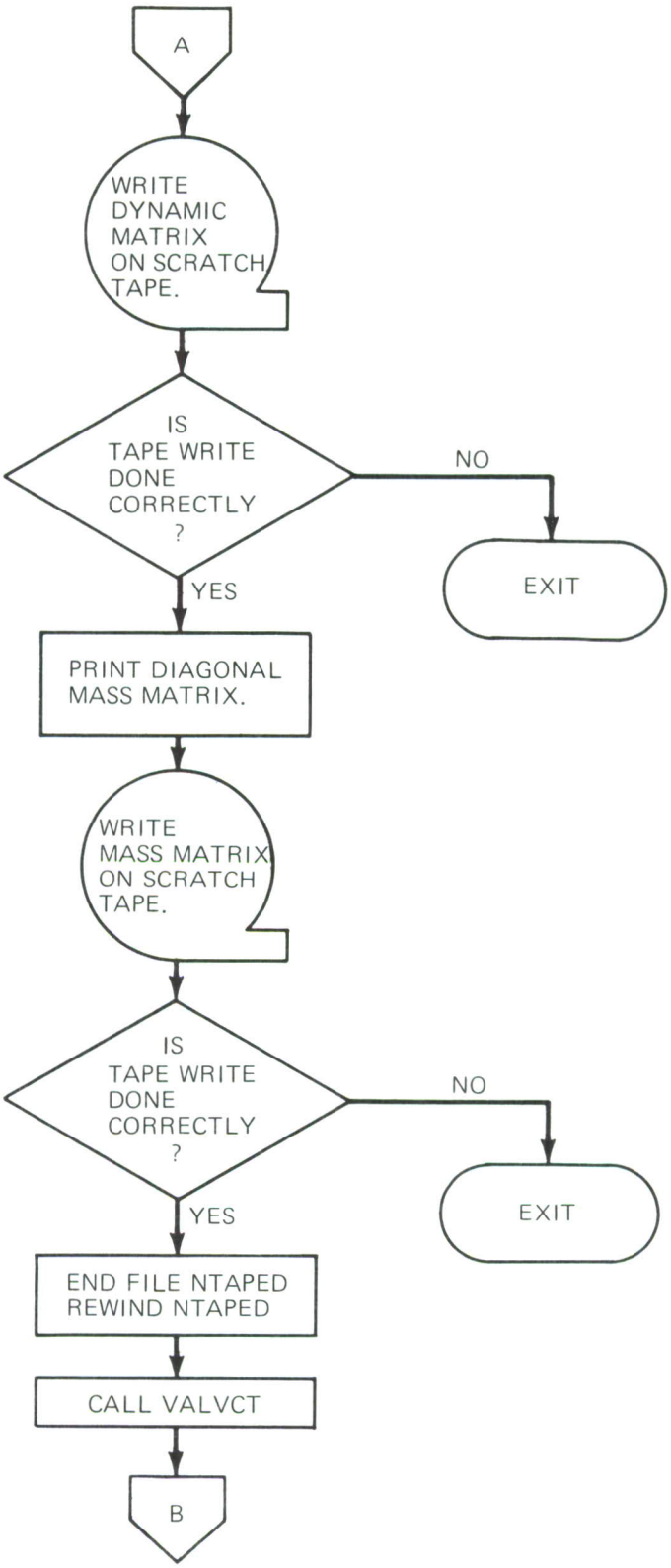


Figure 24—Continued

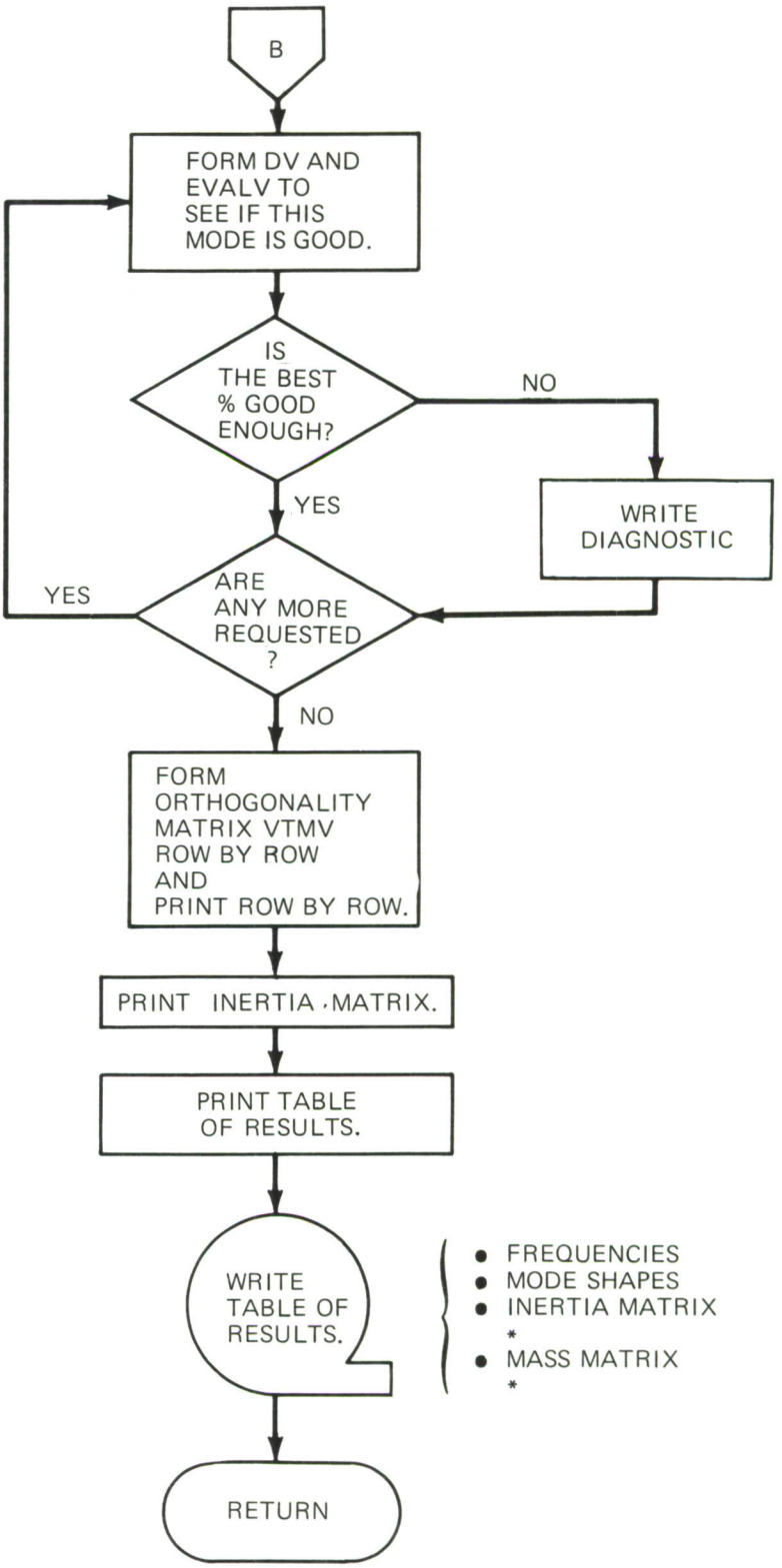


Figure 24—Concluded

b. Programming Organization

The FREMOD module consists of a control routine and seven subroutines. Routine FREMOD controls generation of the eigenvalue-eigenvectors. The functions of the seven subroutines follow.

(1) Subroutine Descriptions

Subroutine VALVCT calls six subroutines that together comprise the QR algorithm. The subroutine VALVCT input/output parameters are DYNMAT, N, MODES, EVAL, and VECMAT:

Parameter	Function
DYNMAT	Dynamic matrix (not returned)
N	Size of current dynamic matrix
MODES	Number of modes desired
EVAL	Array containing eigenvalues
VECMAT	Array containing eigenvectors

The function of VALVCT is to generate eigenvalues and eigenvectors. Given the matrix equation

$$\left[[D] - r[I]\right] \{\phi\} = 0 \qquad (1)$$

where: $[D]$ = dynamic matrix

r = eigenvalue

$[I]$ = identity matrix

$\{\phi\}$ = eigenvector

subroutine VALVCT and its associated subroutines find the set of characteristic roots r (eigenvalues) and the vectors $\{\phi\}$. Natural frequencies are calculated in the main program from the eigenvalues by the equation

$$\omega = \frac{1}{2\pi \sqrt{r}} \qquad (2)$$

The eigenvalue solution is accomplished using the technique commonly known as the QR algorithm.

The QR algorithm is based on the concept of a similarity transformation. To illustrate the similarity transformation and show its usefulness, let us premultiply equation (1) by the arbitrary nonsingular matrix [A] yielding:

$$[A]\left[[D] - r[I]\right]\{\phi\} = 0 \tag{3}$$

Now assume a vector $\{X\}$ such that $[A]^{-1}\{X\} = \{\phi\}$ and substitute the value into equation (3) yielding:

$$\begin{aligned}\left[[A][D] - r[A][I]\right][A]^{-1}\{X\} &= 0\\ \left[[A][D][A]^{-1} - r[I]\right]\{X\} &= 0\end{aligned} \tag{4}$$

Examination of equation (4) shows that the new matrix $[A][D][A]^{-1}$ has the same eigenvalues as equation (1). However, the transformation [A] is required to get back to the original vector space containing vectors $\{\phi\}$, i.e. $[A]\{\phi\} = \{X\}$. By judicious choice of a series of [A] matrices, the form of the original matrix may be changed to suit one's needs. The transformation represented by equation (4) is known as the similarity transformation. The process is as follows.

(a) Step 1

Subroutine HESSEN transforms the dynamic matrix [D] (N-by-N) to "upper-Hessenberg" form [H] ($h_{ij} = 0,\ i > j + 1$).

By using similarity transformations, the eigenvalues are guaranteed to be left unchanged. One premultiplication of [D] is required in eliminating the terms below the subdiagonal in the j^{th} column, and postmultiplication is required to complete the similarity transformation. However, before premultiplication and postmultiplication at the j^{th} step, a row and column interchange is performed to ensure that the term $d_{j+1,j}$ is larger than any term below it in column j. The equation is

$$[H] = \left[S_{n-2}\right]\left[U_{n-2}\right]\cdots\left[S_1\right]\left[U_1\right]\left[D\right]\left[U_1\right]^{-1}\left[S_1\right]^{-1}\cdots\left[U_{n-2}\right]^{-1}\left[S_{n-2}\right]^{-1} \tag{5}$$

where $[S_j]$ eliminates the terms below the subdiagonal in column j, and $[U_j]$ performs the row interchange to maintain numerical stability.

(b) Step 2

Subroutine QRITER performs the following calculations:

A matrix $[P]$ is found such that $[P][H]$ is upper triangular and $[P][H][P]^{-1}$ is again of upper-Hessenberg form. The sequence of matrices $[H_1], [H_2], \ldots [H_k]$ is now formed as shown below:

$$\left[H_1\right] = \left[P_1\right]\left[H_o\right]\left[P_1\right]^{-1}$$

$$\left[H_2\right] = \left[P_2\right]\left[H_1\right]\left[P_2\right]^{-1}$$

$$\vdots$$

$$\left[H_k\right] = \left[P_k\right]\left[H_{k-1}\right]\left[P_k\right]^{-1}$$

It has been demonstrated (reference 4) that this sequence converges to an upper triangular matrix if the roots r are real.

Convergence is approximately inversely proportional to the ratio $(r_i/r_{i-1})^k$ for the i^{th} root where $|r_1| > |r_2| > |r_3| \ldots > |r_n|$. This being the case, the acceleration technique consists of a shift of origin by an amount that is a close approximation of the root r_i . It is easy to find a close approximation since $h_{ii} \rightarrow r_i$. (Actually, the roots of the 2-by-2 matrix whose diagonal terms are $h_{i-1,i-1}$ and h_{ii} are used.) The shifting is performed by subtracting the approximation (say $\bar{r}_i$) off the diagonal of $[H]$ at the k^{th} step. The appropriate transformations are then applied and then $\bar{r}_i$ is added back to the diagonal. It is easy to show that the roots were unchanged, but the roots of the matrix $[H]$ during the actual step were $r_i - \bar{r}_i$. Therefore, the rate of convergence during that step was proportional to $\left(r_{i-1} - \bar{r}_i\right)^k / \left(r_i - \bar{r}_i\right)^k$, which will be a large number.

(c) Step 3

We now have a triangular matrix in which the diagonal elements are the required eigenvalues. Subroutine SORTRT orders the roots according to absolute value and stores them in a new array.

(d) Step 4

We have left only the problem of computing the vectors using the triangular matrix $[H_k]$. We first compute the vectors $\{\phi_T\}$ corresponding to the triangular matrix $[H_k]$. To obtain the vector corresponding to a particular eigenvalue (say the i^{th} diagonal term of $[H_k]$, i.e. r_i) , we use the following equation and observe that all elements of $\{\phi_T\}$ below the i^{th} must be zero.

$$\left[[H_k] - r_i[I]\right] \{\phi_T\} = 0 \tag{6}$$

We may arbitrarily choose the value 1.0 for the i^{th} element of $\{\phi_T\}$ and then proceed back up the vector obtaining each successive term by solving the equation represented by the corresponding row in the matrix equation. This is done by subroutine VECTOR.

Subroutine TRANS1 performs the following calculations in step 5.

(e) Step 5

The transformations that were required to perform step 2 are now retrieved from tape and accumulated into the matrix $[P]^{-1}$. That is

$$\left[P\right]^{-1} = \left[P_1\right]^{-1}\left[P_2\right]^{-1} \cdots \left[P_k\right]^{-1} \tag{7}$$

A vector $\{\phi_H\}$ corresponding to the Hessenberg matrix $[H]$ may now be computed as follows:

$$\{\phi_H\} = [P]^{-1} \{\phi_T\} \tag{8}$$

Subroutine TRANS2 performs the calculations of step 6.

(f) Step 6

The quantities necessary to reconstruct the transformations $[S_1]$, $[S_2]$, . . . $[S_{n-2}]$ of step 1 were temporarily stored in the lower part of matrix $[\bar{D}]$ for conservation of storage. (The matrix $[S]$ here is not to be confused with the stress deflection matrix used earlier.) Only n terms of storage were necessary to contain the information required to reproduce the transformations $[U_1]^{-1}$, $[U_2]^{-1}$, . . . $[U_{n-2}]$. The following product is now computed:

$$\{\phi\} = \left[U_1\right]^{-1}\left[S_1\right]^{-1} \cdots \left[U_{n-2}\right]^{-1}\left[S_{n-2}\right]^{-1} \{\phi_H\} \tag{9}$$

To compute $[U_i]$, $[U_1]^{-1}$, interchange rows ℓ and m and form $[U_i]$ as follows:

$$U_{jj} = 1,\ j \neq \ell,\ m$$

$$U_{\ell m} = U_{m\ell} = 1$$

All other terms are zero.

Also note that

$$\left[U_i\right] = \left[U_i\right]^{-1}$$

To compute $[S_i]$, $[S_i]^{-1}$, use the notation $[\bar{D}] = [d_{ij}]$ for the dynamic matrix $[D]$ and $[S] = [S_{ij}]$. The terms below the diagonal in column $(k + 1)$ of the general matrix $[S_k]$ are $-d_{k+2,k}/d_{k+1,k}$, $-d_{k+1,k}/d_{k+1,k}$, . . . $-d_{nk}/d_{k+1,k}$. In addition,

$$S_{ij} = 1,\ i = j$$

All others terms are zero.

Equation (10) is the formulation for $[S_1]$ for $n = 5$. It can be readily noted that $[S_1]^{-1}$ may be obtained from $[S_1]$ by changing the sign of the elements below the diagonal.

$$[S_1] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & -\dfrac{d_{31}}{d_{21}} & 1 & 0 & 0 \\ 0 & -\dfrac{d_{41}}{d_{21}} & 0 & 1 & 0 \\ 0 & -\dfrac{d_{51}}{d_{21}} & 0 & 0 & 1 \end{bmatrix} \tag{10}$$

To compute $[P_i]$,

$$\left[P_i\right] = \left[\bar{P}_{n-1}\right] \cdots \left[\bar{P}_2\right]\left[\bar{P}_1\right] \tag{11}$$

where $[\bar{P}_k]$ eliminates the subdiagonal terms in column k . At the time that the rotational transformations $[P_i]$ are required, the original dynamic matrix [D] has been reduced to Hessenberg form [H] :

$$[H] = [h_{ij}] \tag{12}$$

Specifically $[\bar{P}_k]$ is formed as follows:

$$L_k = \left(h_{kk}^2 + h_{k+1,k}^2\right)^{1/2}$$

$$\cos \theta_k = h_{kk} L_k$$

$$\sin \theta_k = h_{k+1,k} L_k$$

If

$$[\bar{P}_k] = [p_{ij}]$$

Then

$$\bar{P}_{ii} = 1, \quad i \neq k, k+1$$

$$\bar{P}_{kk} = \cos \theta \ , \ \bar{P}_{k,k+1} = \sin \theta$$

$$\bar{P}_{k+1,k} = -\sin \theta \ , \ \bar{P}_{k+1,k+1} = \cos \theta$$

$$\bar{P}_{ij} = 0$$

for all other i and j,

The matrix described above is known as Givens' rotational matrix. Note that no additional time is required to compute the inverse since

$$[\bar{P}_k]^{-1} = [\bar{P}_k]^T$$

(g) General-Purpose Subroutines

Subroutine	Function
INRPRD	To form the inner product of two vectors (See appendix II.)
READTP WRTETP	Binary tape input/output subroutines for TL01 compatibility (See appendix II.)

(2) Flow Diagrams

Flow diagrams for FREMOD subroutines are shown in figures 25 through 32.

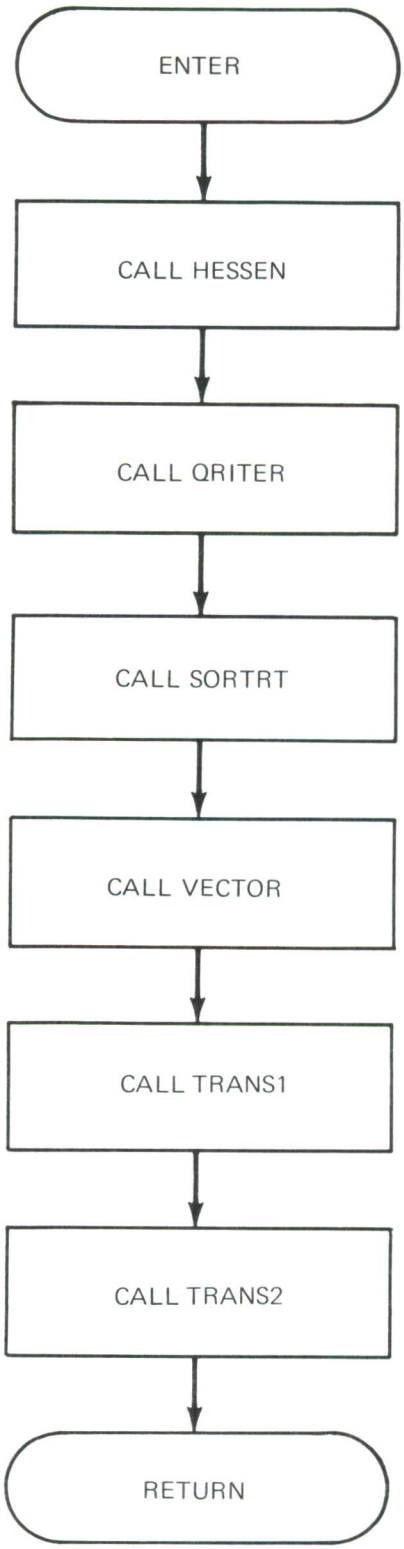


Figure 25. VALVCT Flow Chart

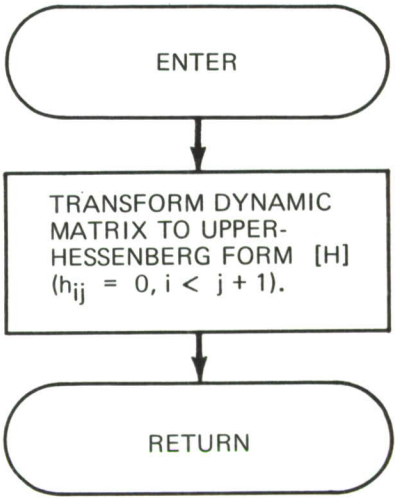


Figure 26. HESSEN Flow Chart

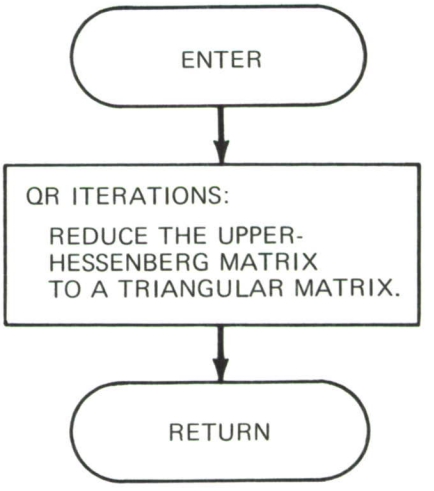


Figure 27. QRITER Flow Chart

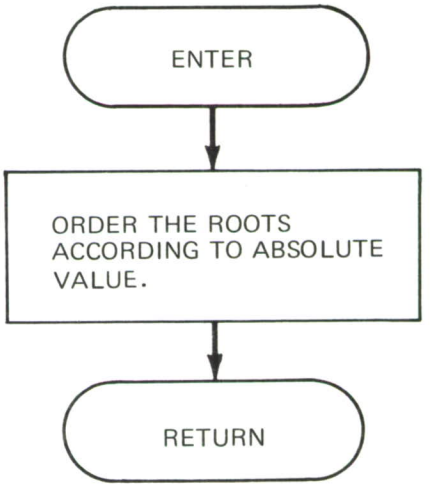


Figure 28. SORTRT Flow Chart

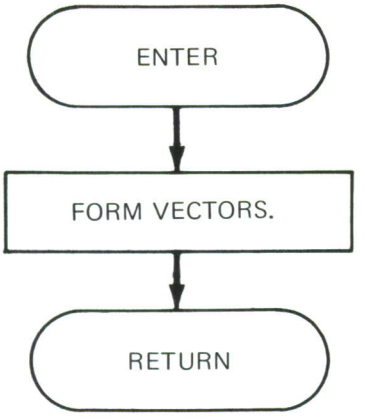


Figure 29. VECTOR Flow Chart

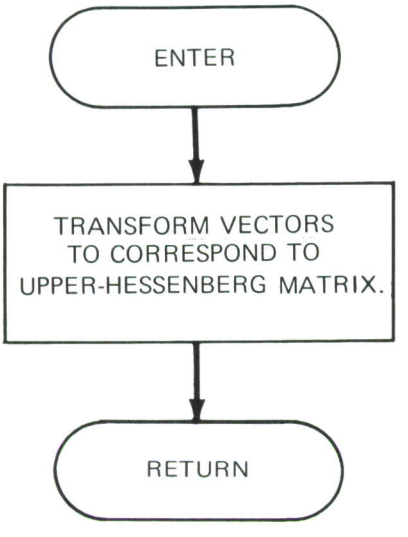


Figure 30. TRANS1 Flow Chart

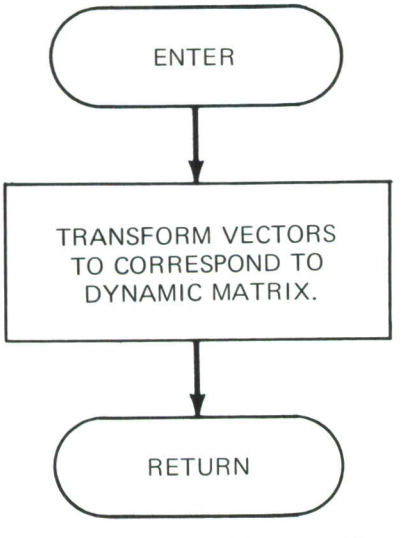


Figure 31. TRANS2 Flow Chart

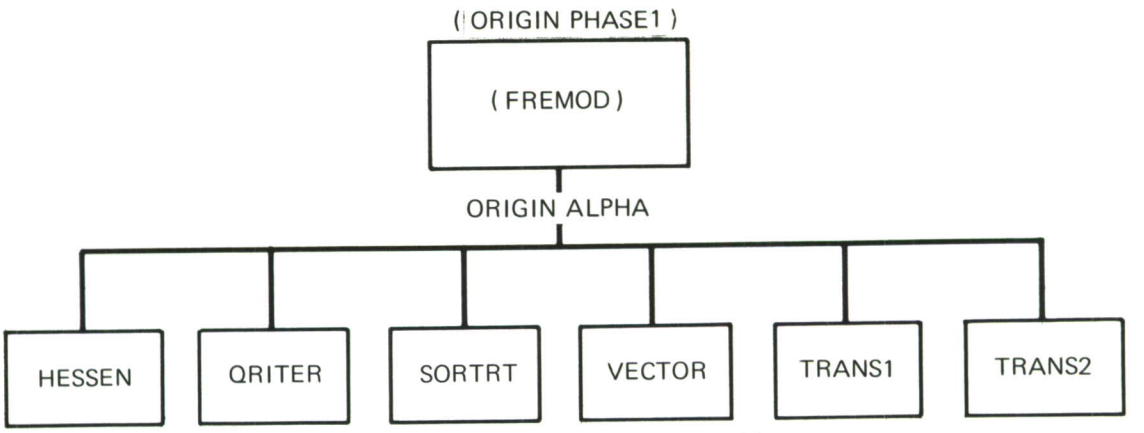


Figure 32. FREMOD Overlay Map

c. FREMOD Subroutine List

Subroutine	Function
FREMOD	This is the main program. It handles input, output, and generation of the dynamic matrix from the flexibility and mass matrices. Subroutine VALVCT is called for the QR solution
VALVCT	Calls successive subroutines, which together form an eigenvalue-eigenvector solution package using the QR algorithm
HESSEN	Transforms the dynamic matrix to upper-Hessenberg form
QRITER	Reduces the upper-Hessenberg matrix to a triangular matrix, the eigenvalues being the diagonal terms (QR iteration scheme)
SORTRT	Orders the eigenvalues according to absolute value and stores them in a new array. This array is needed in forming the eigenvectors.
VECTOR	Computes the eigenvectors
TRANS1	Transforms the eigenvectors (found using the triangular matrix) to correspond to the upper-Hessenberg matrix
TRANS2	Transforms the eigenvectors corresponding to the upper-Hessenberg matrix to vectors corresponding to the dynamic matrix

IV

PHASE II—RANDOM LOAD AND RESPONSE PROGRAMS

Phase II is an integrated set of computer programs for determining sonic loads and random structural responses. The execution of this phase is controlled by the PHASE2 routine. The RANLOD module and RANSO modules are called from the control program. The program listings for phase II are included in appendix IV.

1. RANDOM LOADING MODULE (RANLOD)

a. General Description

RANLOD generates force cross-power spectral density (cross-PSD) matrices describing the applied forces. The mathematical model is based on properties of decayed progressive sound waves. The RANLOD module consists of five FORTRAN-coded subroutines and associated system and general input/output (I/O) subroutines.

Other loading modules can be used in place of RANLOD when the analysis requires a different form of force loading. These modules must be compatible with the basic RANVIB system.

In addition to basic problem information, RANLOD requires data input on cards describing the panel geometry, wave data, and option control parameters. A detailed discussion of the input and card format is contained in the Engineering User's Guide, reference 3.

Figure 33 illustrates the logical placement of RANLOD in the RANVIB system. The overlay structure is discussed in section IV1.b.

The RANLOD module is called once during the execution of phase II to generate the required matrices. In addition to the frequencies, solution options, and control parameters stored in the labeled common blocks by the phase II control program, RANLOD reads card inputs. It then proceeds to generate the force cross-PSD matrices for the required solution option. They are stored on tape and printed as they are generated. The matrices are stored on tape in a compatible format for the various solution options the user desires. The printing is controlled by the user as described in the card input. Figure 34 illustrates RANLOD input/output data flow.

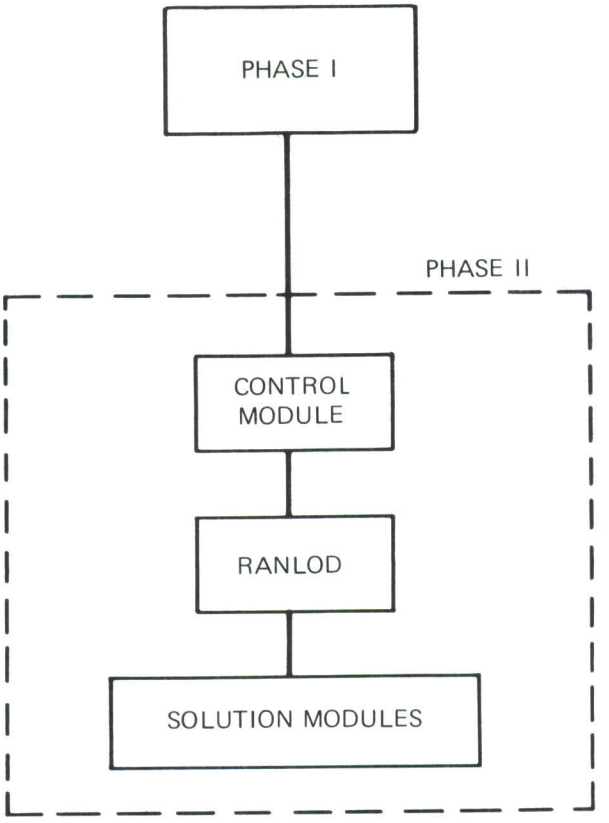


Figure 33. RANLOD Placement

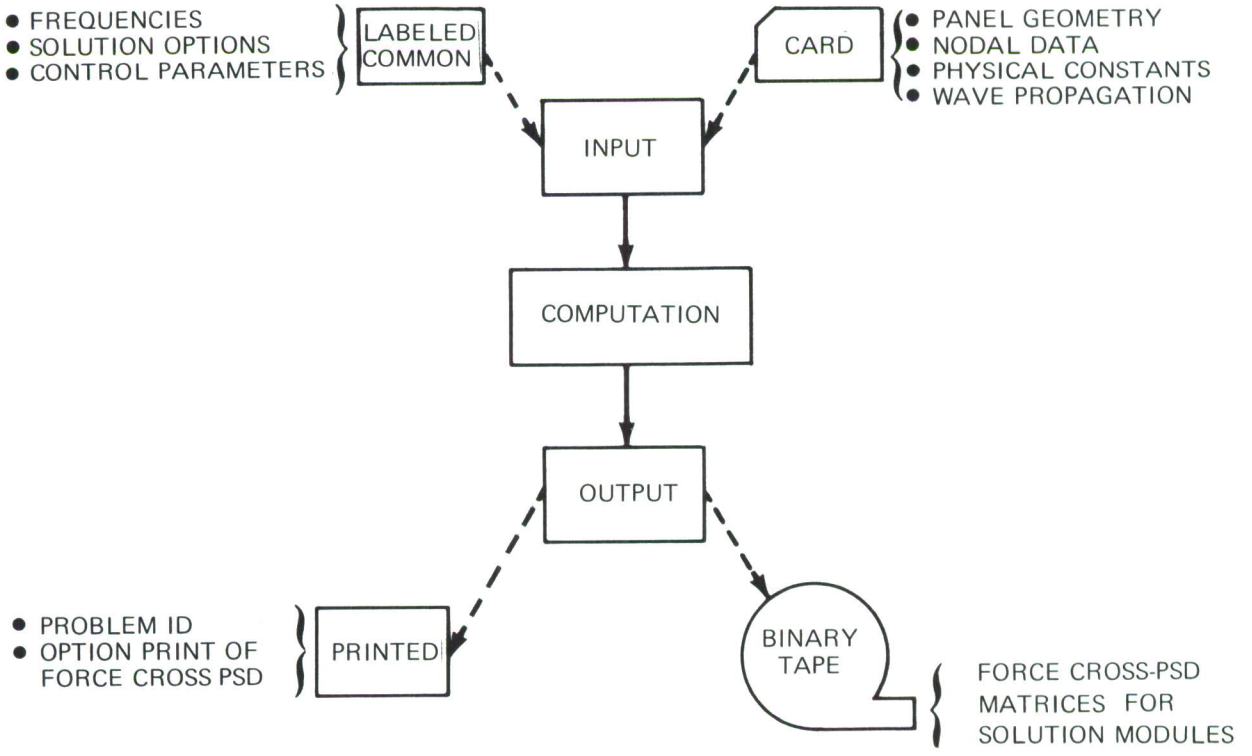


Figure 34. RANLOD Input/Output Diagram

The five subroutines that form the RANLOD module are as follows:

Subroutine	Function
RANLOD	Control subroutine that reads input and problem initializations and calls all subroutines needed for generation
ARIA	Calculates the areas associated with the retained nodes
CONST	Calculates problem constants needed in generation
NOISOR	Computes the force cross-PSD matrices
OUTPUT	Controls printing and formation of binary output tapes

The logic flow of the RANLOD module is shown in figure 35.

(1) Input/Output Functions

All card inputs are via logical tape 5, and all printed outputs are via logical tape 6. Logical tapes 2, 17, and 14 are used for all binary outputs of the force cross-PSD matrices. These tapes are written from subroutine OUTPUT using subroutine WRTETP. A detailed map of these tapes is given in the description of subroutine OUTPUT.

(2) Restrictions

The RANLOD restrictions are discussed in detail in the Engineering User's Guide, reference 3. These restrictions are basic problem size and mathematical limitations in analyses. If it is desired to replace RANLOD with another loading module, the following items must be adhered to.

(a) The labeled common blocks BLK1, BLK2, and BLK3 must be retained and the size maintained compatible with phase II definitions. The force cross-PSD matrix size must be maintained to less than or equal to the phase II limits.

(b) The binary tape output of the force cross-PSD matrices must be maintained in the compatible format for the solution modules.

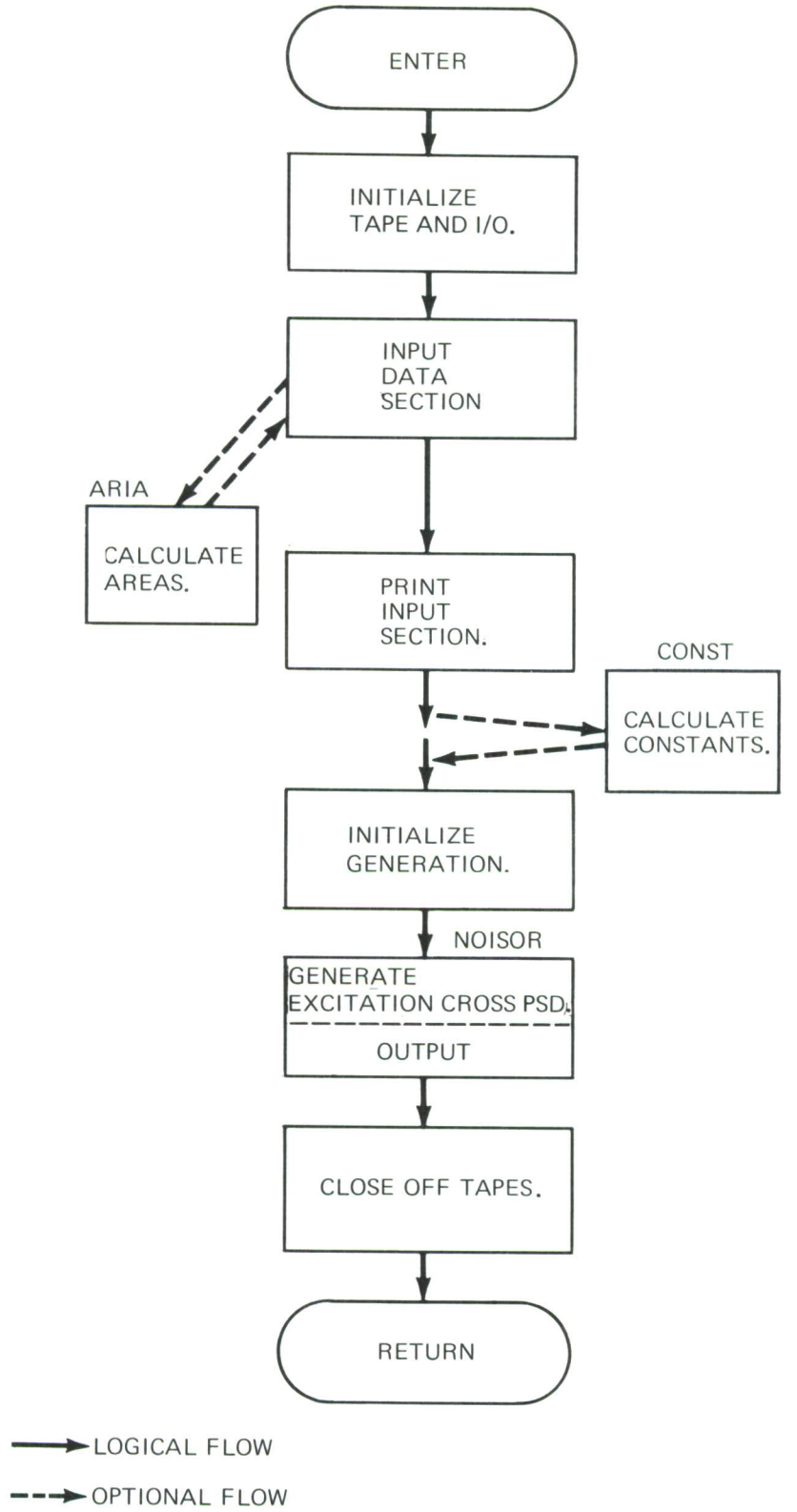


Figure 35. RANLOD Subroutine Flow

b. Programming Organization

(1) Subroutine Descriptions

This section discusses the purpose, methodology, restrictions, and inputs/outputs of the five FORTRAN subroutines.

(a) Subroutine RANLOD

Subroutine RANLOD is the controlling routine for the loading module. It is called by the phase II control program and returns to this program when generation is complete.

Method: Subroutine RANLOD accepts control from the phase II main program and controls the generation of the force cross-PSD matrices. It uses the information in labeled common blocks BLK1, BLK2, and BLK3 to generate the required solution option. Labeled common blocks BLK4, BLK5, and BLK6 are established for communication between RANLOD subroutines.

Figure 36 is a flow chart of the RANLOD subroutine.

Input: Input is via two modes. The first is labeled common from phase II control. The second mode of input is cards. The content and card format are discussed in detail in reference 3.

Error diagnostics: None

Subroutines required: ARIA, CONST, NOISOR

Argument list: None

Length: 37001_8

(b) Subroutine ARIA

Subroutine ARIA computes the areas associated with the retained nodes in the structural idealization.

Method: Using the set of line-to-origin distances, the area A for node k is calculated using

$$A_k = 1/4\ (x_{i+1} - x_{i-1})\ (y_{j+1} - y_{j-1})$$

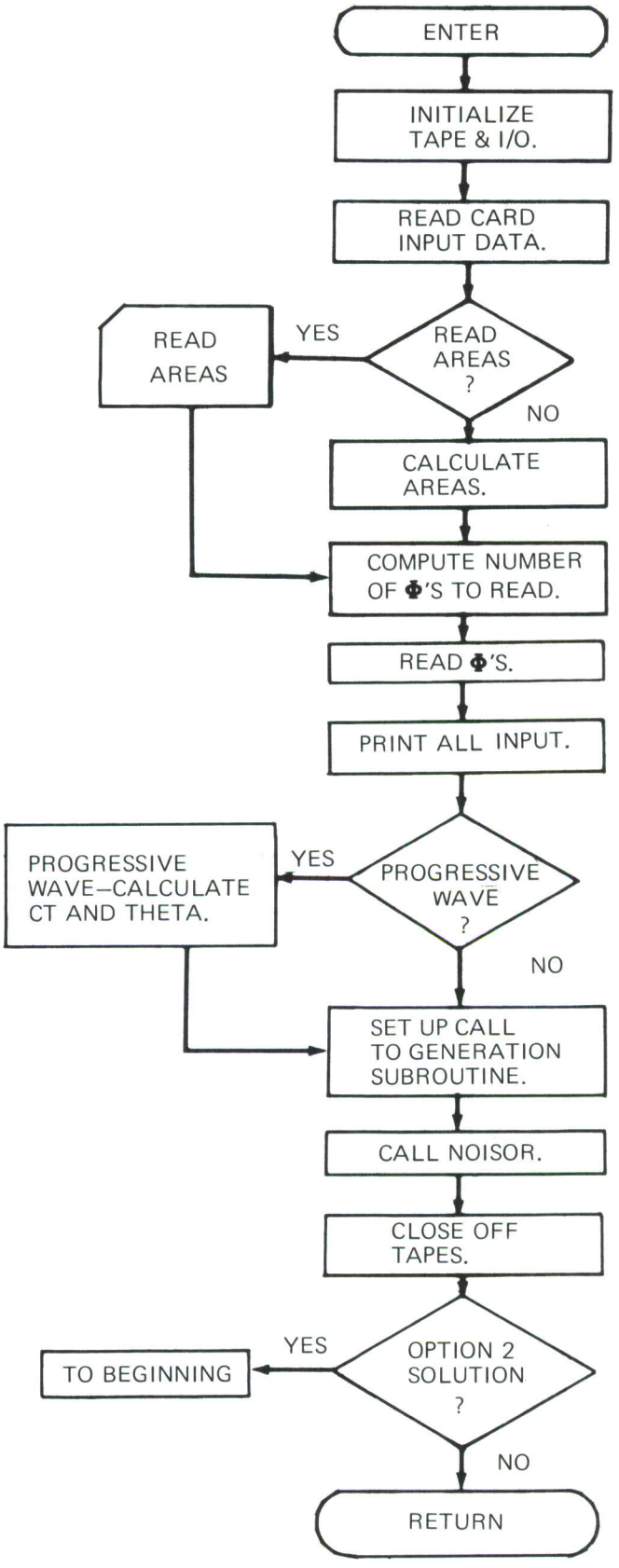


Figure 36. RANLOD Flow Chart

where: k = 1, number of retained nodes

i = 2, number of nodes in x direction

j = 2, number of nodes in y direction

Figure 37 illustrates subroutine ARIA flow.

Input: Input is via labeled common.

Error diagnostics: None

Subroutines required: None

Argument list: None

Length: 101_8

(c) Subroutine CONST

Subroutine CONST calculates the angle θ and trace velocity c_t . Angle θ defines the direction that the trace of the pressure wave fronts propagate over the panel surface.

Method: Subroutine CONST is only called for a progressive wave. The methodology and input control for this calculation is described in reference 3. Figure 38 illustrates subroutine CONST flow.

Input: There is no input except through the subroutine argument list.

Error diagnostics: None

Subroutines required: None

Argument list: CX input—The phase velocity along the panel in the x direction

CY input—The phase velocity along the panel in the y direction

CT output—Trace velocity

THETA output— θ , the angle between the direction of sound propagation and the x and y axes of the panel

Length: 141_8

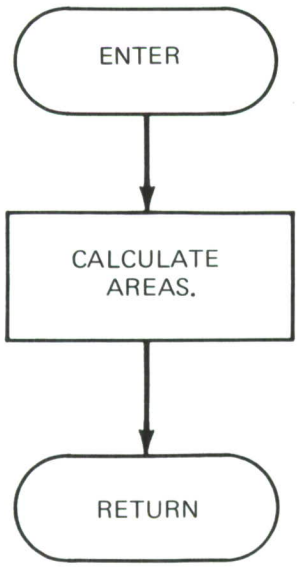


Figure 37. ARIA Flow Chart

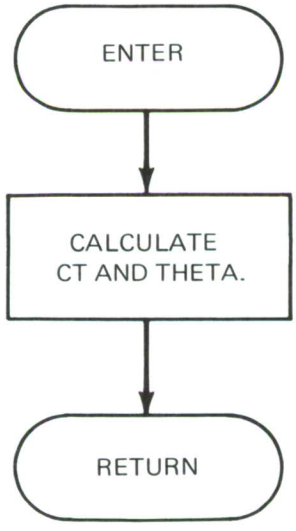


Figure 38. CONST Flow Chart

(d) Subroutine NOISOR

Subroutine NOISOR calculates the force cross-PSD matrices. It receives control of the generation from routine RANLOD and returns to this subroutine when generation is complete.

Method: The detailed mathematical model from which subroutine NOISOR computes force cross PSD is discussed in reference 1. Figure 39 describes the flow of subroutine NOISOR. The subroutine computes the matrices in a direct manner with the mathematical model. The one exception to this is the computation of the separation distances ξ_{ij} and η_{ij} from the set of line-to-origin distances x_i and y_i. The separation distances are calculated as needed in the calculation of force cross PSD. This is done to conserve storage. The equations used in the separation calculation are as follows:

$$\xi_{ij} = x_j - x_i$$

and

$$\eta_{ij} = y_j - y_i$$

The computation of the subscripts is done by use of FORTRAN IV function subroutine MOD. This is done to take advantage of the repeating properties of the nodal geometry. A detailed flow chart of the separation calculation in program notation is shown in figure 40.

Input: Input to subroutine NOISOR is via labeled common and the subroutine argument list.

Error diagnostics: None

Subroutines required: OUTPUT

Argument list: ILIM input—Limit on outer frequency loop
KLIM input—Limit on inner frequency loop
D input—Decay constant

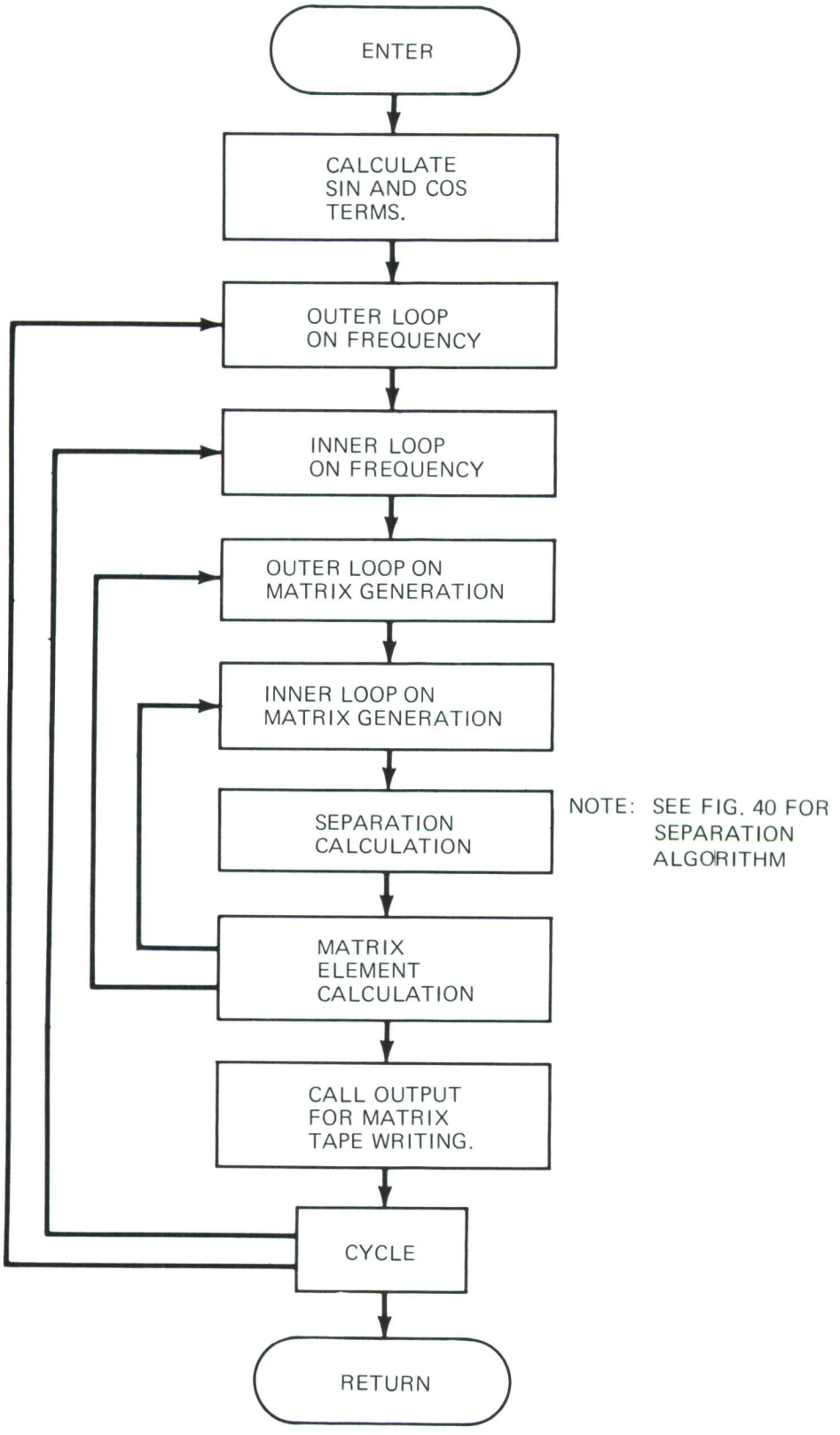


Figure 39. NOISOR Flow Chart

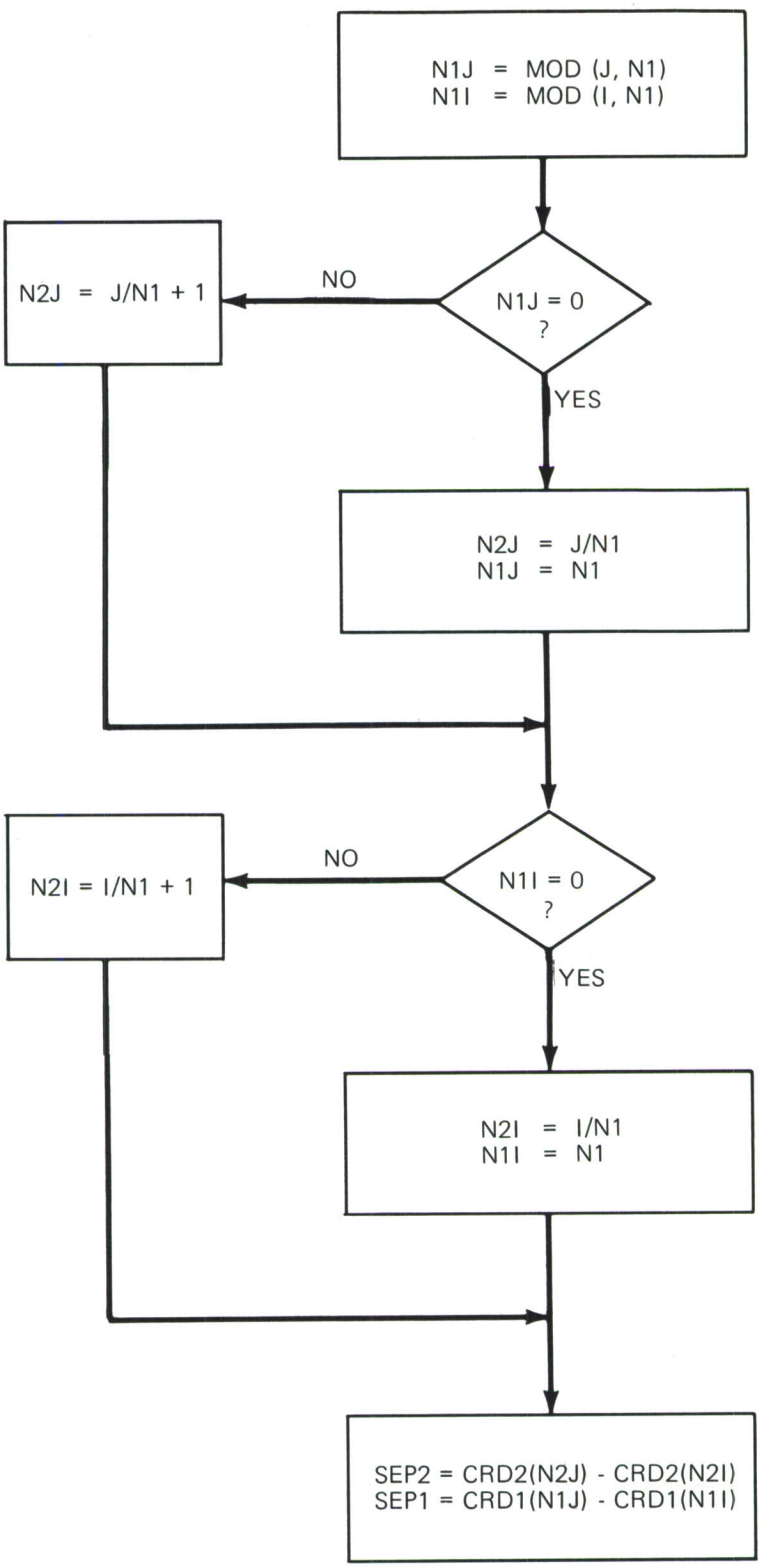


Figure 40. Separation-Algorithm Flow Chart

CT input—Trace velocity

THETA input— θ , propagation angle

CX input—Phase velocity in x direction

CY input—Phase velocity in y direction

Length: 601_8

(e) Subroutine OUTPUT

Subroutine OUTPUT is called from subroutine NOISOR each time the matri. output is required, i.e. each frequency computation. The return is to subroutine NOISOR upon completion of OUTPUT.

Method: The subroutine has two output functions. One is to print the force cross-PSD matrices for as many frequencies as requested by the user. The other is to write the force cross-PSD matrices on binary tape in a format compatible with the solution modules. Item 2 is accomplished by the use of subroutine WRTETP, which is discussed in appendix I. Figure 41 describes the general flow of subroutine OUTPUT.

Input: Input is via labeled common and the subroutine argument list.

Error diagnostics: The error return from subroutine WRTETP is tested, and the appropriate comment printed if an error has occurred.

Subroutines required: WRTETP

Argument list: NPHI input—Count of the number of frequencies for which calculation has been completed

OMG input—The value of the current frequency (used in printing for identification)

Length: 573_8

(2) Tape Use

The detailed maps of the binary tapes for the various solution options are shown in figure 42.

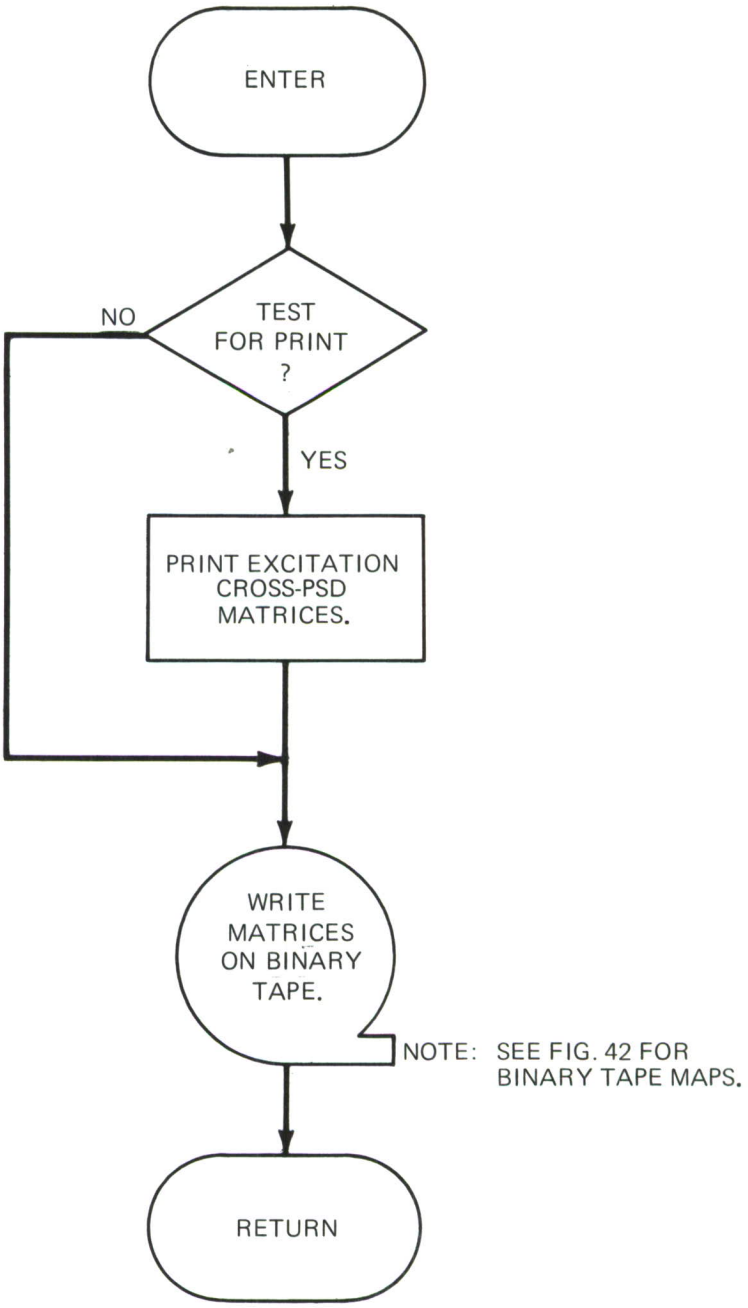


Figure 41. OUTPUT Flow Chart

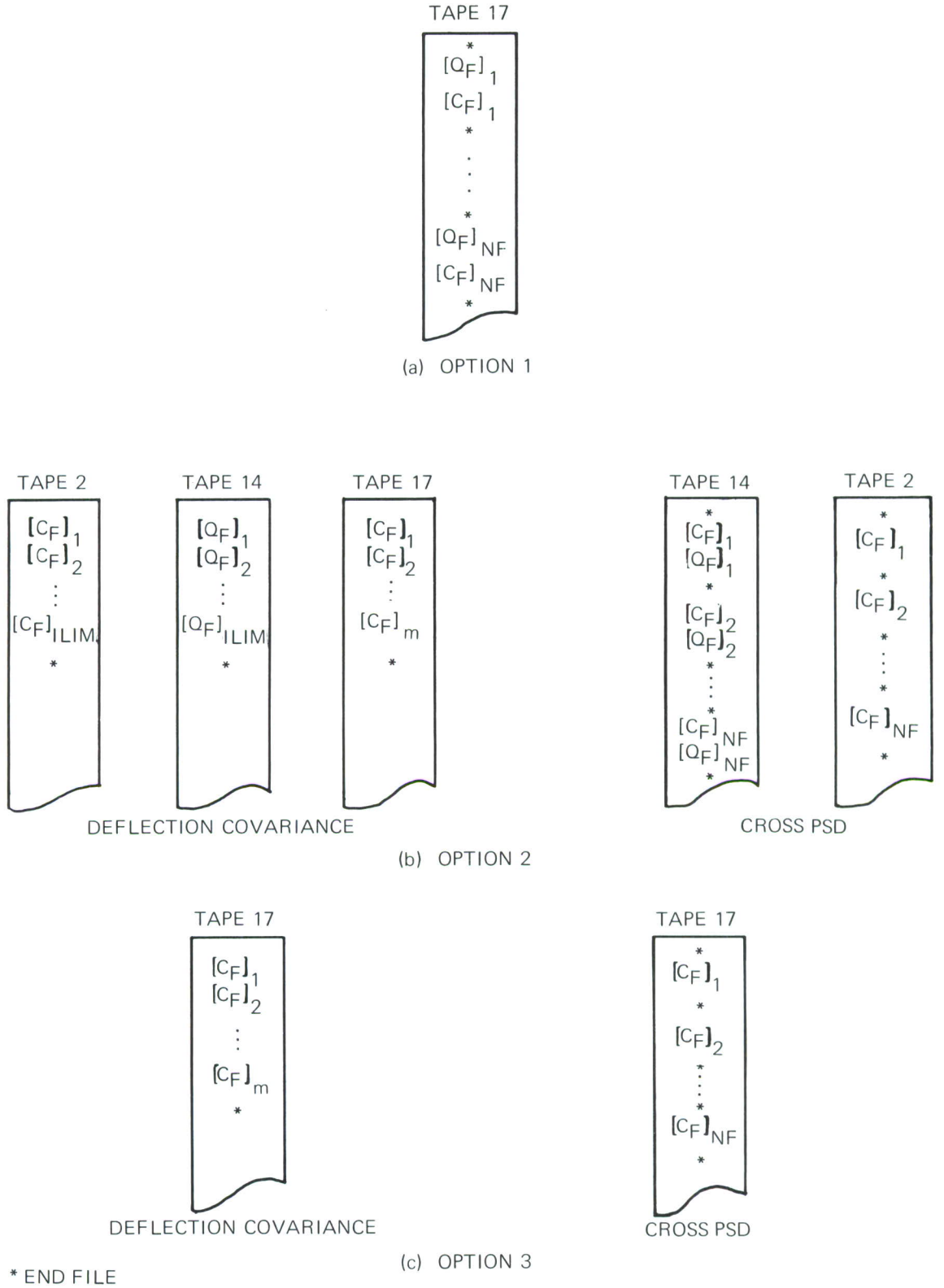


Figure 42. Binary Output Tape Maps from RANLOD

2. RANDOM-RESPONSE SOLUTION PROGRAM (RANSO)

a. General Description

(1) Purpose—Logical Organization

The purpose of the RANSO (random-response solutions) module is to calculate random deflections and stress response solutions via matrix methods for complex structures subjected to random excitations. All matrix manipulations and solutions are performed in real matrix form. The logical flow through the RANSO solution is shown in figure 43. There are three solution options:

(1) Option 1—General viscous damping

(2) Option 2—Normal modes

(3) Option 3—Normal modes without cross terms

Options 2 and 3 are used when the excitation pressures vary slowly with frequency, i.e. broadband. When the excitation is not broadband, option 1 is used. For each option, there are four basic solutions involved:

(1) Deflection cross PSD

(2) Deflection covariance and second spectral moments

(3) Stress cross PSD

(4) Stress covariance and second spectral moments within elements

(2) Input/Output Functions

The data from the phase I output tape will be used as part of the input of phase II. Additional input from cards is required as discussed in the Engineering User's Guide, reference 3. All intermediate results will be stored on scratch tapes for temporary storage before proceeding to the next operational subroutine.

(3) Phase II RANSO Subroutines

There are two types of subroutines in the RANSO module. They are the standard FORTRAN IV subroutines and TL01 subroutines. The FORTRAN IV subroutines are described as follows:

(a) Method

(b) Input/output

(c) Errors

(d) Subroutines required

(e) Argument list

(f) Subroutine length—number of storage locations in octal required by the subroutine when compiled on the 7094 Mod II Version 13

(g) Flow diagrams

The subroutines that are written in TL01 matrix language (appendix I) are described as follows:

(a) A description of the subroutine

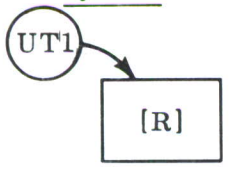
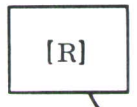
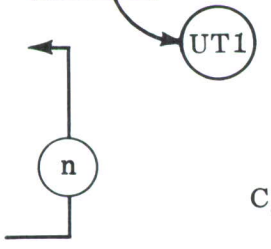
(b) Input tape storage and output tape storage

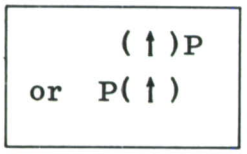
(c) Flow diagram

A discussion of READTP/WRTETP error messages is given in appendix I.

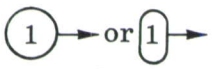
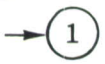
(4) FORTRAN and TL01 Flow-Diagram Conventions

(a) Symbols in TL01 Flow Charts

Symbol	Definition
	Read matrix [R] into core from tape unit UT1.
	Write matrix [R] onto tape unit UT1.
	Cycle n times over the enclosed operations.

Symbol	Definition
	Matrix operation
	Multiply the results from the preceding block by P.
n	Number of stress matrices
m	Number of mode shapes
NF	Number of frequencies

(b) Symbols for FORTRAN Diagrams Only

Symbol	Definition
*	File mark on tape
	Input from tape 1
	Output on tape 1
JD	Deflection covariance matrix
CPSD	Cross PSD

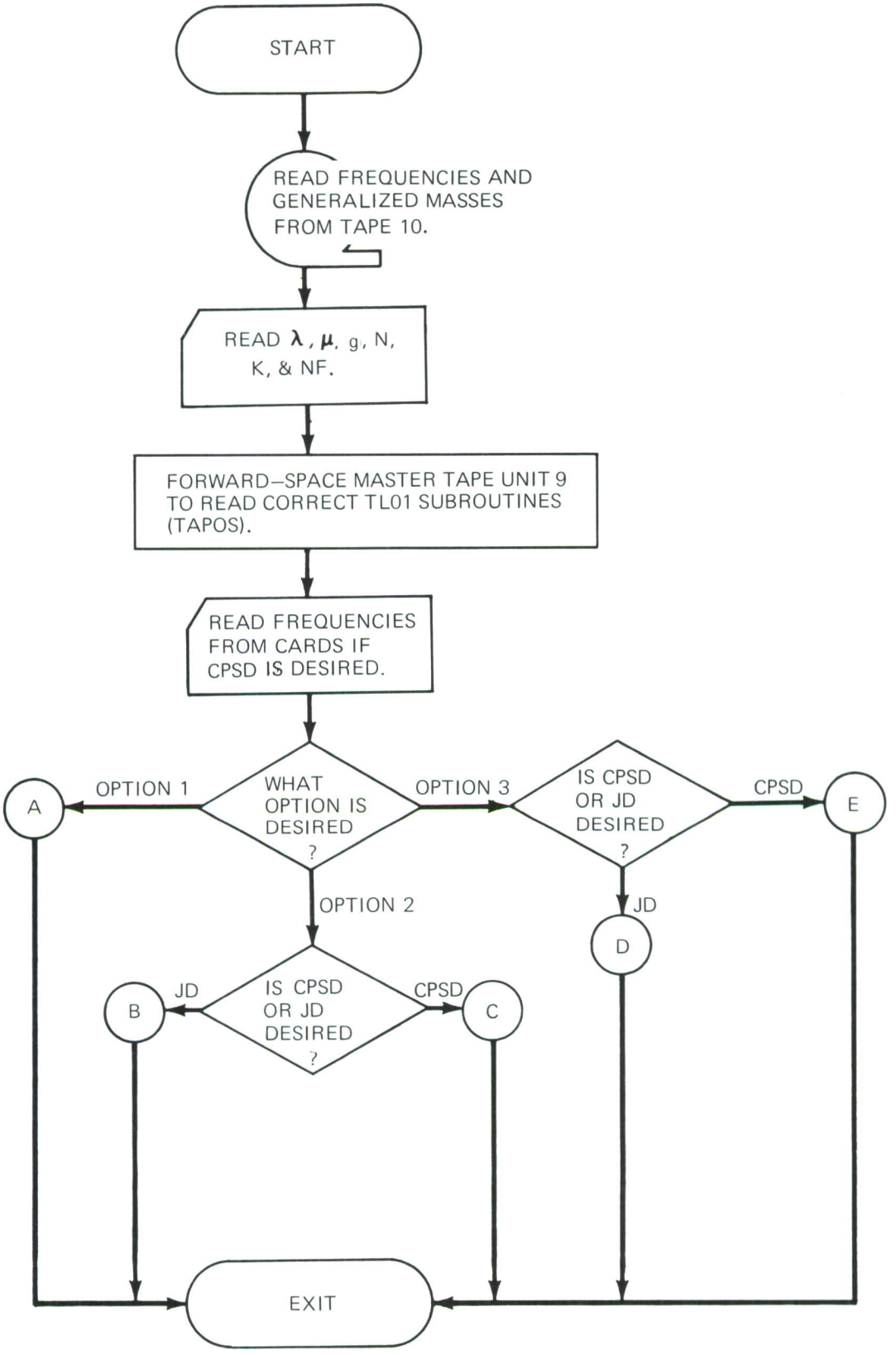


Figure 43. RANSO Flow Diagrams

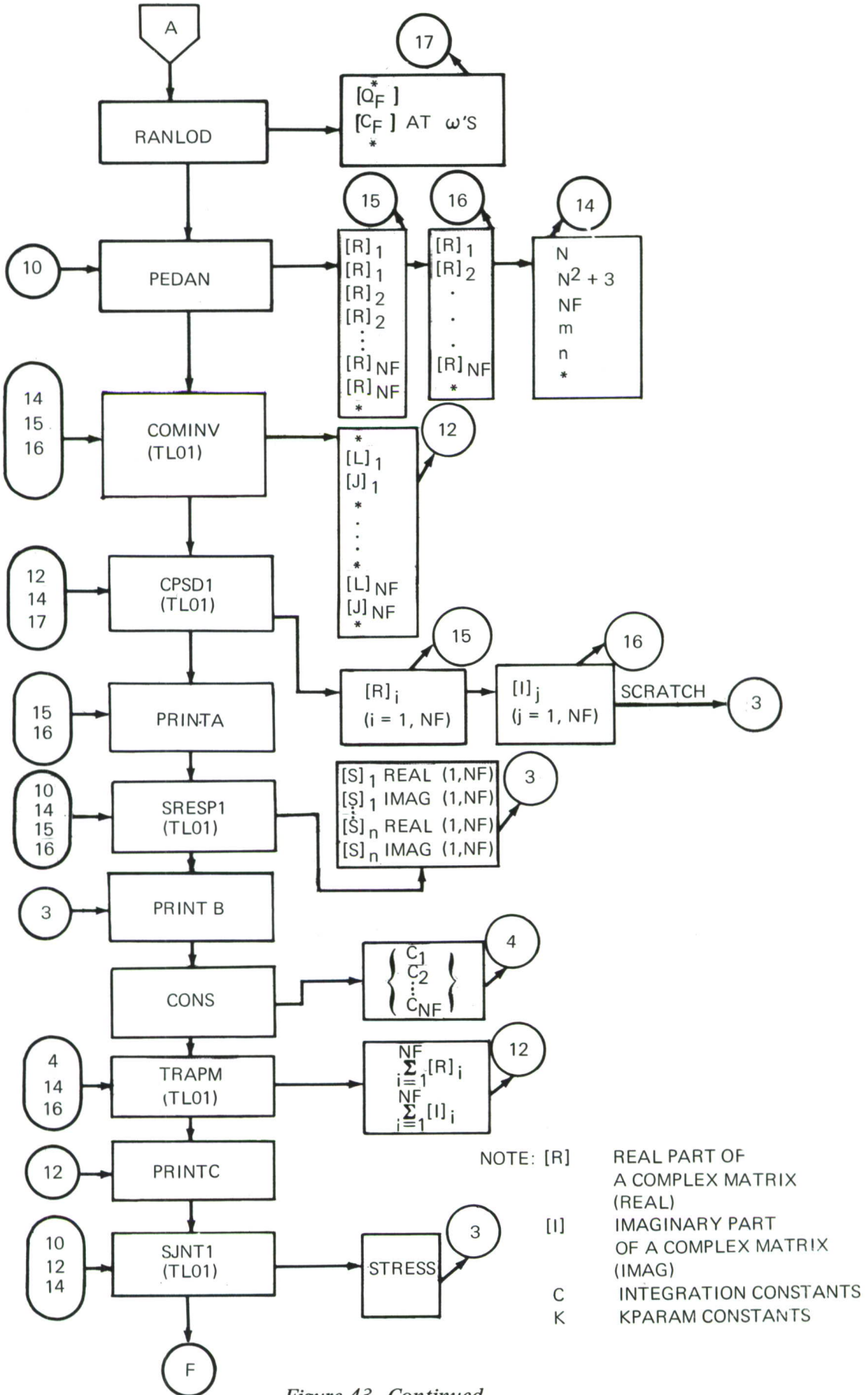


Figure 43—Continued

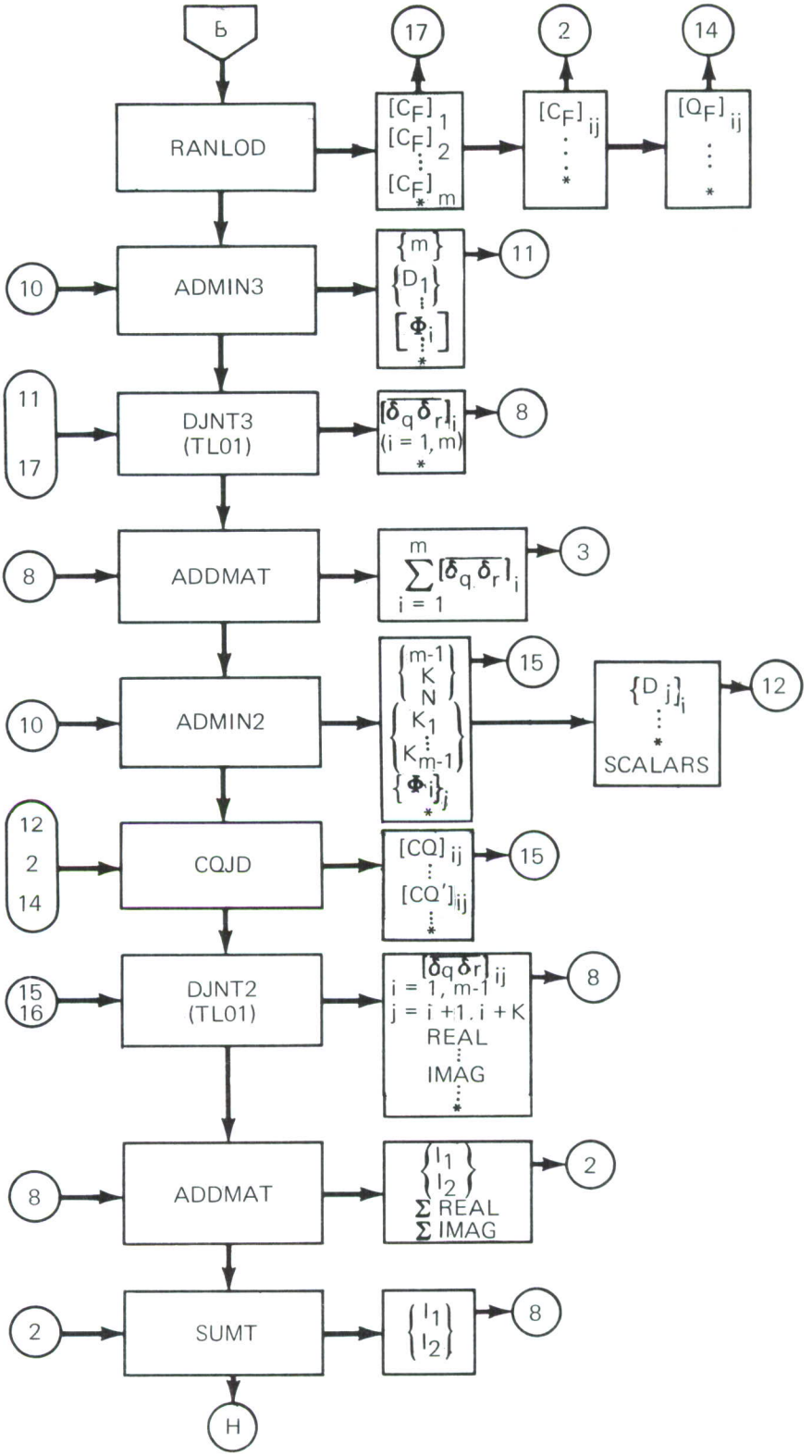


Figure 43—Continued

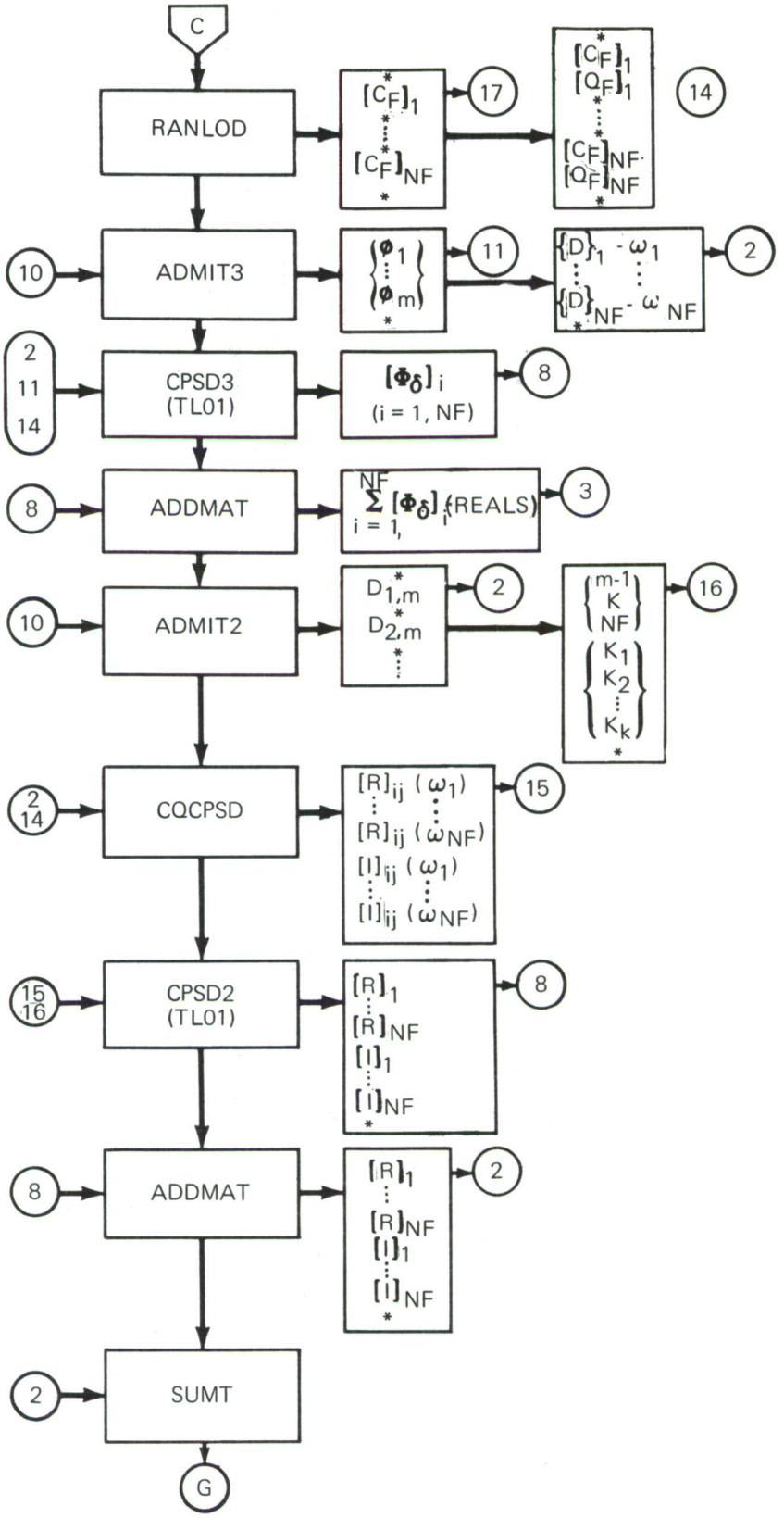


Figure 43—Continued

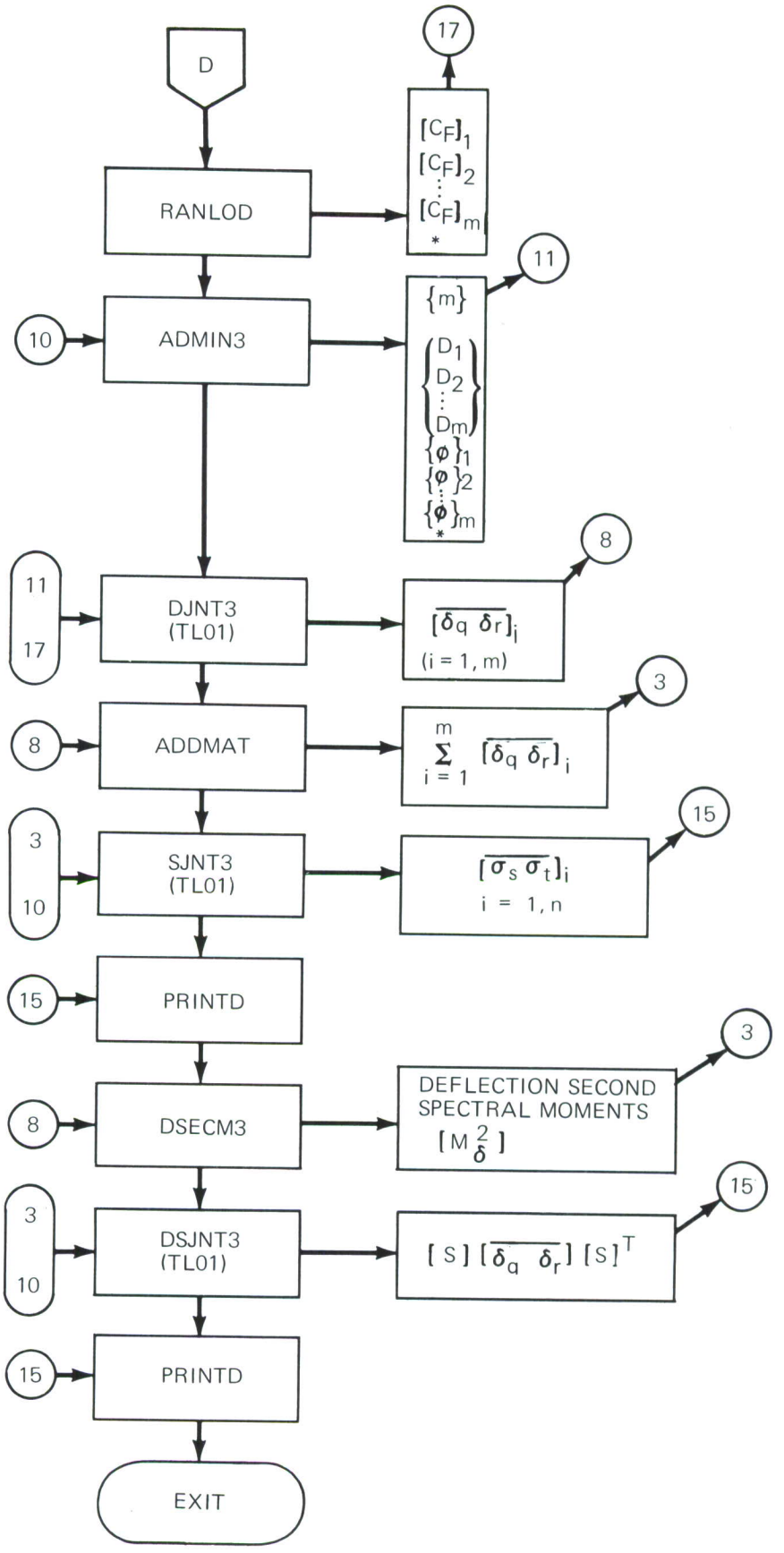


Fig. 43—Continued

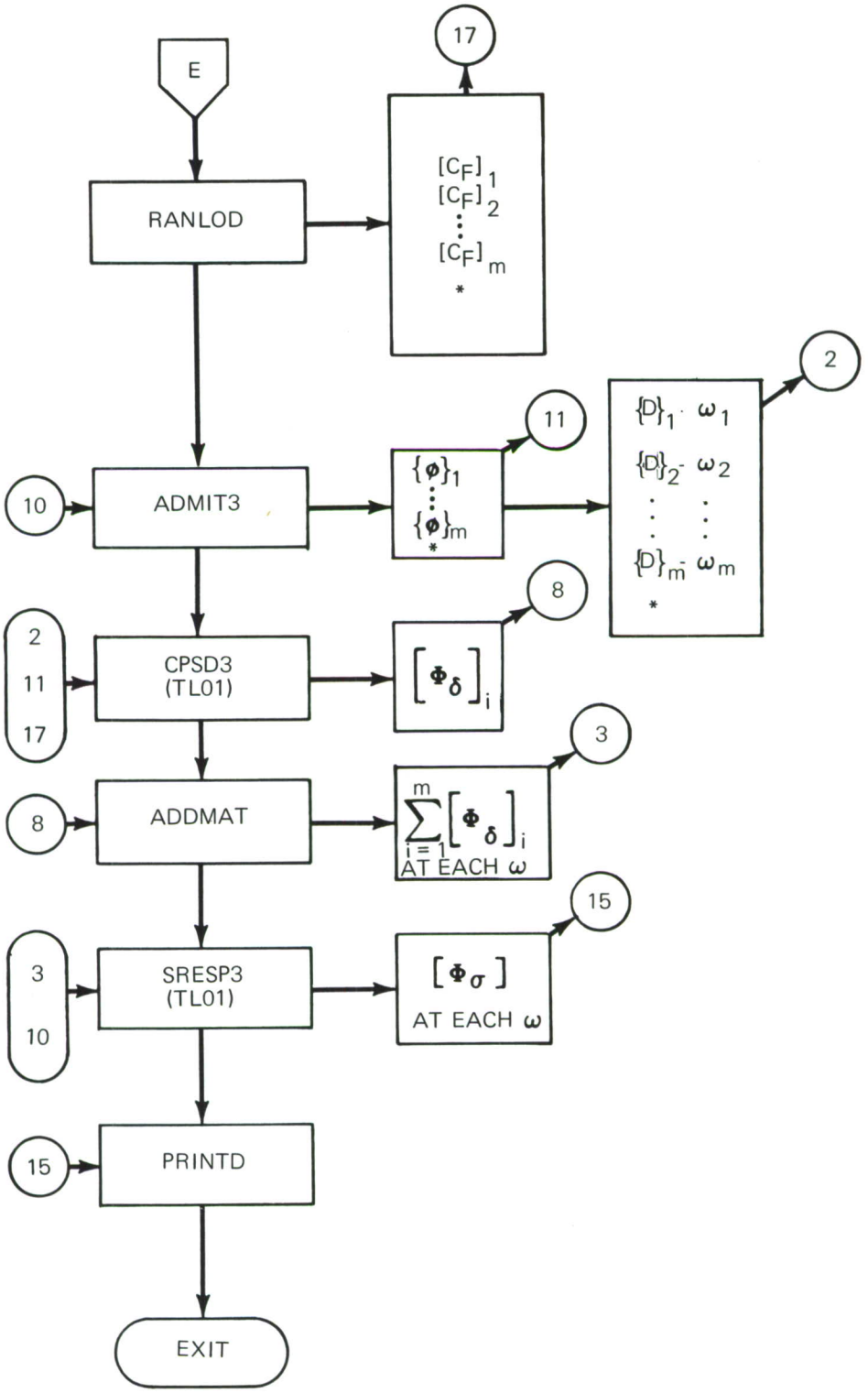


Figure 43-Continued

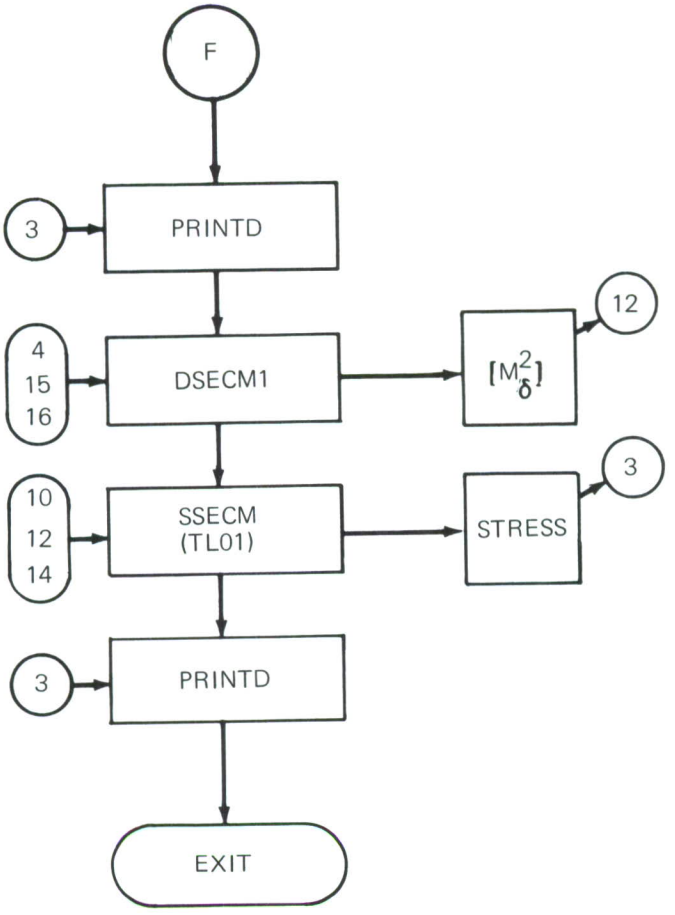


Figure 43—Continued

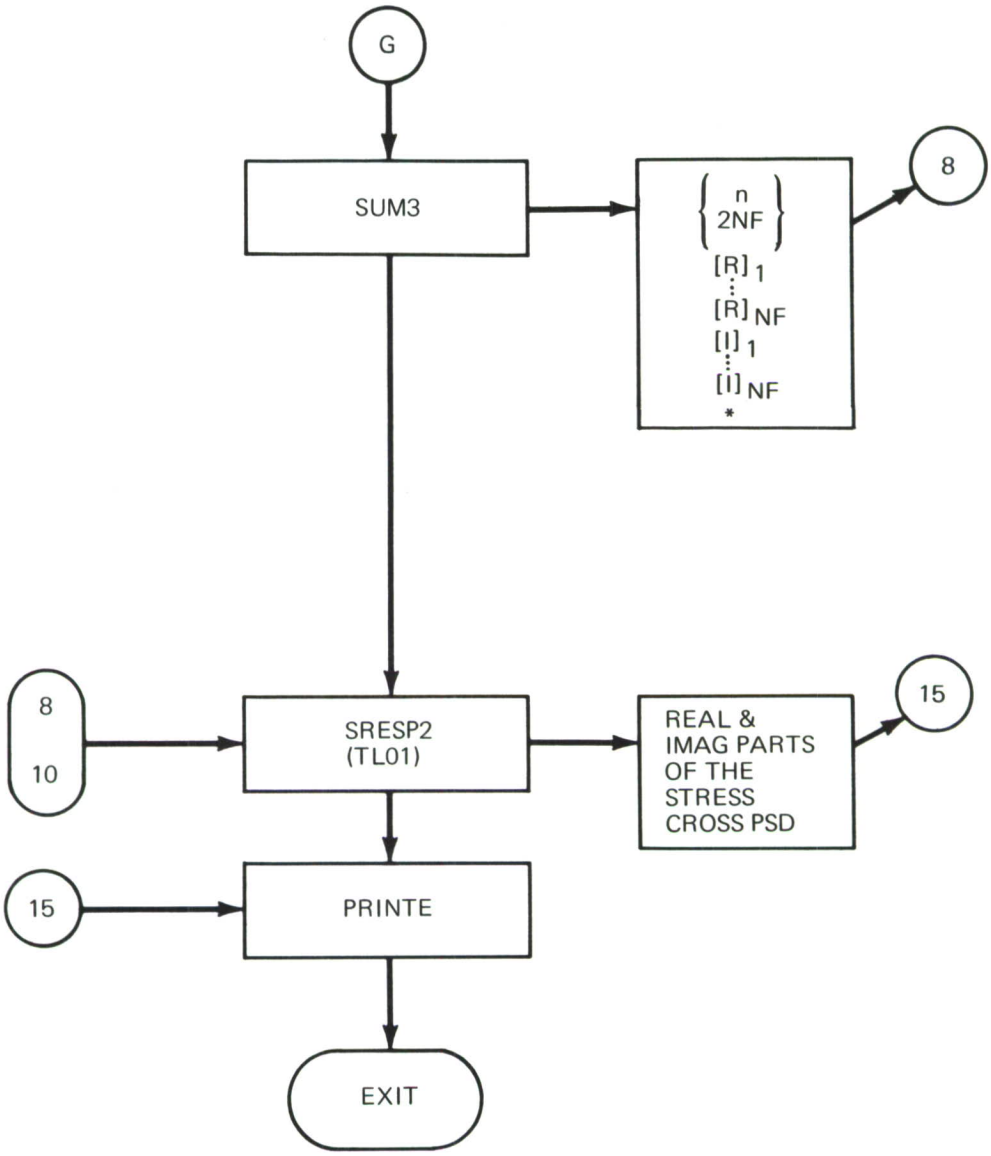


Figure 43—Continued

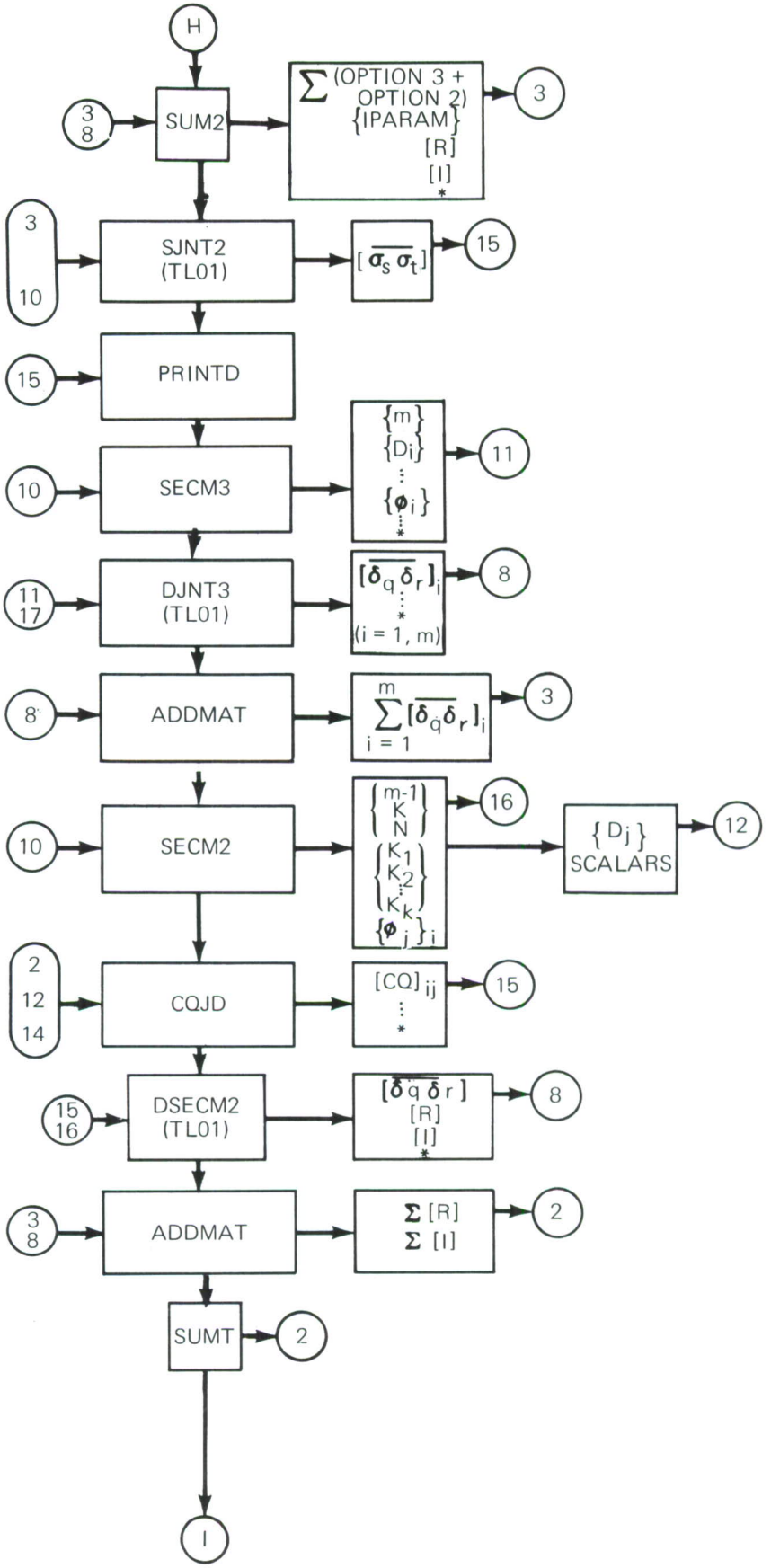


Figure 43—Continued

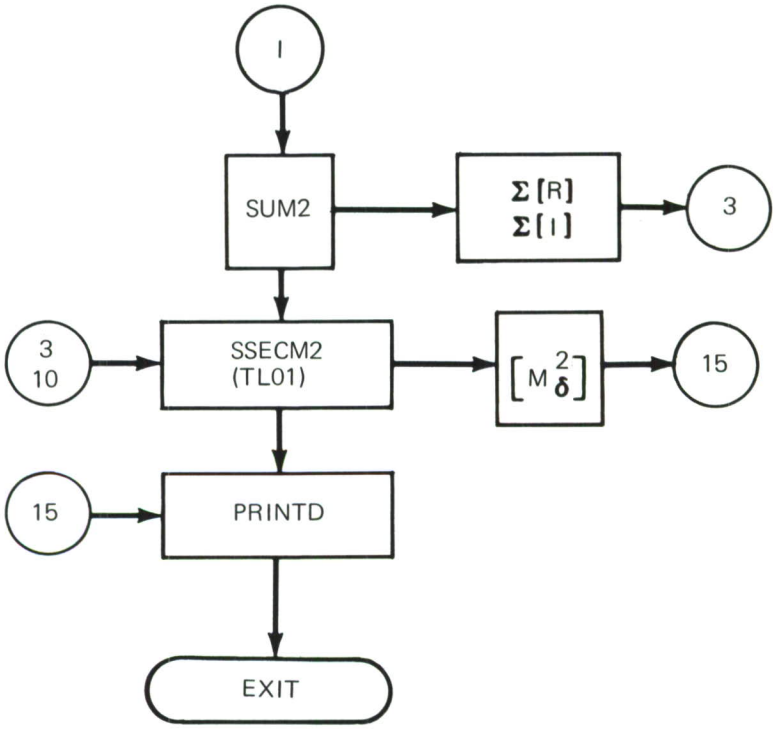


Figure 43—Concluded

(5) RANSO Subroutine Listing

Subroutine	Function
PEDAN	Impedance matrix
CONS	Calculates the constants used in the complex matrix integration using the trapezoidal rule
ADMIN2	Admittance integral scalars used in option 2 deflection covariance calculations
CQJD	Multiplies the co-PSD and quad-PSD matrices by the admittance scalar integrals used in the deflection covariance calculations and sums the products
SECM3	Calculates the scalars in the deflection-spectral-moment equations for option 3

Subroutine	Function
SECM2	Calculates the second-spectral-moment scalars for option 2 deflection covariance
DSECM3	Calculates the deflection second spectral moments for option 3
CQCPSD	Multiplies the co-PSD and quad-PSD matrices by the admittance scalars and sums the products for option 2
ADMIT2	Admittance scalars used in option 2 cross-PSD calculations
ADMIN3	Admittance integral scalars used in option 3 deflection covariance calculations
ADMIT3	Admittance scalars used in option 3 cross-PSD calculations
ADDMAT	Performs the matrix summation for option 2 and option 3 deflection covariance and cross-PSD calculations
TAPOS	Positions the master tape in proper positions to start reading TL01 data for all options
SUM2	Performs the matrix summation for option 2 deflection covariance and cross-PSD calculations
SUM3	Performs the matrix summation for option 3 deflection covariance and cross-PSD calculations
SUMT	Sums the option 2 and option 3 matrices
COMINV	Inverts a complex matrix
CPSD1	Calculates the deflection response cross PSD for option 1
SRESP1	Calculates stress response cross PSD for option 1
TRAPM	Performs the complex matrix integration over all frequencies to form the deflection covariance matrices for option 1
SJNT1	Calculates joint stress cross-PSD matrix for option 1
DSECM1	Calculates deflection second-spectral-moment matrix for option 1
SSECM	Calculates stress second spectral moments for option 1
DJNT3	Calculates deflection covariance matrix for option 3

Subroutine	Function
DJNT2	Calculates deflection covariance matrix for option 2
SJNT2	Calculates stress covariance matrix for option 2
DSECM2	Calculates deflection second-spectral-moment matrix for option 2
CPSD3	Calculates deflection cross-PSD matrix for option 3
CPSD2	Calculates deflection cross PSD for option 2
SJNT3	Calculates stress covariance matrix for option 3
DSJNT3	Calculates stress second-spectral-moment matrix for option 3
SRESP3	Calculates stress cross PSD for option 3
PRINTA	Prints the deflection cross-PSD matrices (option 1)
PRINTB	Prints the stress cross-PSD matrices for plates and beams (option 1)
PRINTC	Prints the deflection covariance matrices (option 1)
PRINTD	Prints the stress cross-PSD matrices in option 3 and prints the stress covariance matrix in all options
PRINTE	Prints the stress cross-PSD matrices (option 2)

b. Option 1 Solution Program—General Viscous Damping

(1) Organization

Option 1 is time consuming, because there is a complex matrix inversion involved for each frequency. When the excitation is not broadband, the mean-square deflections and stresses are calculated by numerical integration.

The damping matrix [C] is card input. The stiffness matrix [K] and the mass matrix [M] come from the phase I output tape. The desired frequencies for which the responses are formed are card inputs.

The solution steps and their associated subroutine names are described below. The limitations are as follows:

$$N \leq 60$$

$$m \leq 25$$

$$NF \leq 60$$

Step 1: The excitation-force matrices co-power $[C_F(\omega_i)]$ and quad-power $[Q_F(\omega_i)]$ spectral densities for each desired frequency are generated using subroutine RANLOD.

Step 2: The impedance matrix is calculated in subroutine PEDAN and is divided into a real part $-\omega^2[M] + [K]$ and an imaginary part $\omega[C]$. These matrices are used as inputs to the admittance-matrix formation.

Step 3: The admittance matrix $[H(i\omega)]$ is calculated for all frequencies using the impedance matrix as the input. A complex matrix-inversion process is used in the calculation for the admittance matrix at each specified frequency.

The TL01 subroutine COMINV is used to find the complex matrix inverse. See section IV 2.b.(2)(3) for the subroutine flow chart and a description of the method. Thus,

$$[H(i\omega)] = \left[-\omega^2 [M] + i\omega [C] + [K]\right]^{-1}$$

The admittance matrix is divided into a real part $[J(\omega)]$ and an imaginary part $[L(\omega)]$ and is stored on a scratch tape for the cross-PSD solution. Hence,

$$[H(i\omega)] = [J(\omega)] - i\,[L(\omega)]$$

Step 4: The deflection cross-PSD matrices $[\Phi_\delta(\omega)]$ are calculated for all frequencies by TL01 subroutine CPSD1. See section IV 2.b.(2)(j). The real part is formed by use of the relationship:

$$[C_\delta(\omega)] = [J]\left([C_F(\omega)][J] + [Q_F(\omega)][L]\right) + [L]\left([C_F(\omega)][L] - [Q_F(\omega)][J]\right)$$

The imaginary part is formed by use of the relationship:

$$[Q_\delta(\omega)] = [L]\left([C_F(\omega)][J] + [Q_F(\omega)][L]\right) - [J]\left([C_F(\omega)][L] - [Q_F(\omega)][J]\right)$$

The cross PSD calculated at each frequency is stored on tape for later use in the deflection second-spectral-moment calculation.

Step 5: If the stress cross-PSD solution is desired, TL01 subroutine SRESP1 is used. Thus,

$$[\Phi_{\sigma}(\omega)] = [S]\ [\Phi_{\delta}(\omega)]\ [S]^{T}$$

The stress matrices [S] come from the phase I output tape and the cross PSD $[\Phi_{\delta}(\omega)]$ are formed in step 4 above. See section IV 2. b. (2) (f).

Step 6: If the deflection covariances are desired, the cross PSD's are numerically integrated using the trapezoidal rule. To obtain a desired accuracy, an adequate number of cross PSD's should be defined over the frequency range. The constants used in the trapezoidal method are calculated in subroutine CONS, section IV 2. b. (2)(c), and stored on tape. These scalar constants are multiplied by the N-by-N cross-PSD matrices and summed over the frequency range to calculate the deflection covariance in TL01 subroutine TRAPM. See section IV 2.b.(2)(g).

Step 7: If the stress covariance matrices are desired, TL01 subroutine SJNT1 is used. See section IV 2. b. (2)(h). Thus,

$$[\overline{\sigma_s \sigma_t}] = [S]\ [\overline{\delta_q \delta_r}]\ [S]^{T}$$

Step 8: The deflection second spectral moments are obtained by multiplying the cross PSD formed in step 4 by the square of the frequency and by constants from CONS and then summing over all frequencies. This is accomplished in subroutine DSECM1. See section IV 2. b. (2)(d).

Step 9: The stress second spectral moments are calculated in TL01 subroutine SSECM, section IV 2. b. (2)(i), by using the N-by-N matrix calculation in step 8. This matrix is premultiplied by the stress matrix [S] and postmultiplied by $[S]^{T}$.

(2) Subroutine Descriptions

(a) Subroutine RANLOD (Refer to section IV 1.)

(b) Subroutine PEDAN (figure 44)

Method: The impedance real and imaginary matrices are calculated for NF frequencies.

- Real part: $-\omega^2$ [M] + [K]
- Imaginary part: ω[C]

Input: Mass matrix [M] and the stiffness matrix [K] come from the phase I output tape. The damping matrix [C] is card input.

Output: The impedance matrices (real part and imaginary part) are stored on a scratch tape.

Error: Standard READTP/WRTETP error messages

Subroutines required: READTP/WRTETP

Argument list: None

Length: 40526_8

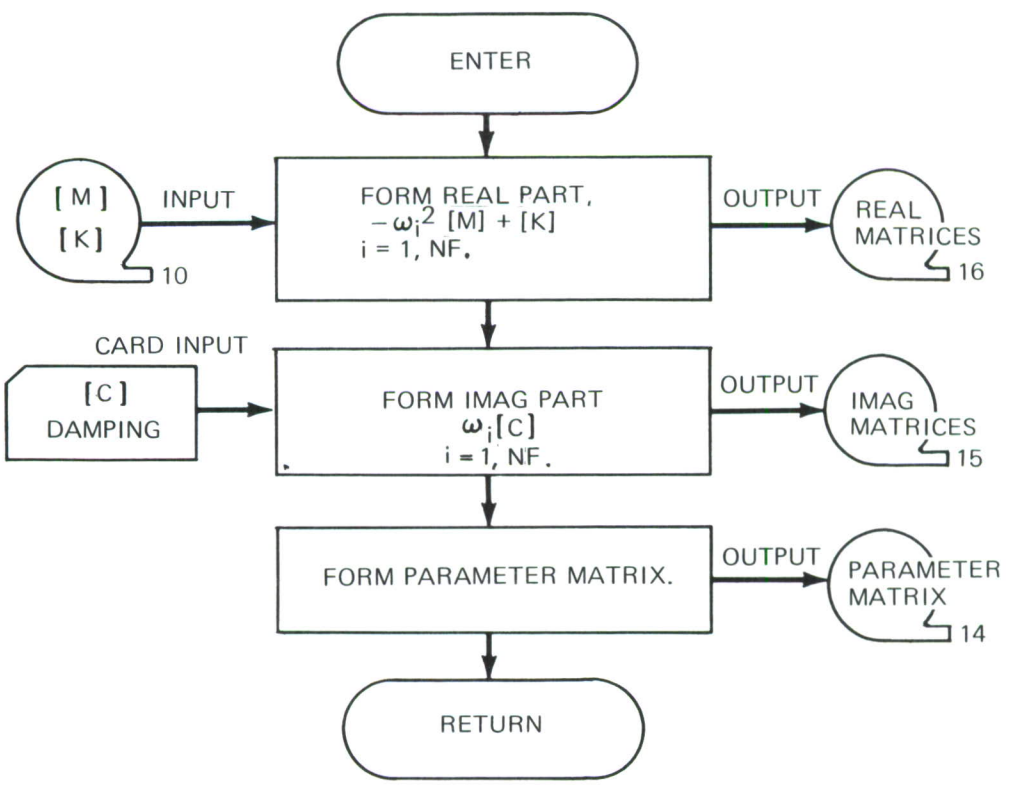


Figure 44. PEDAN Flow Chart

(c) Subroutine CONS (figure 45)

Method: This subroutine calculates the constants C used in the trapezoidal TL01 subroutine TRAPM at NF frequencies (ω_i) .

Form $H_i = \omega_{i+1} - \omega_i$, where $i = 1, NF-1$

Hence, $C_i = \frac{(H_{i-1} + H_i)}{2}$, where $i = 2, NF-1$

and $C_1 = \frac{H_1}{2}$, $C_{NF} = \frac{H_{(NF-1)}}{2}$

Input: Frequencies ω_i are stored in labeled common.

Output: Constants $C_i (i = 1, NF)$ are written on a scratch tape.

Error: Standard WRTETP error messages

Subroutines required: WRTETP

Argument list: None

Length: 526_8

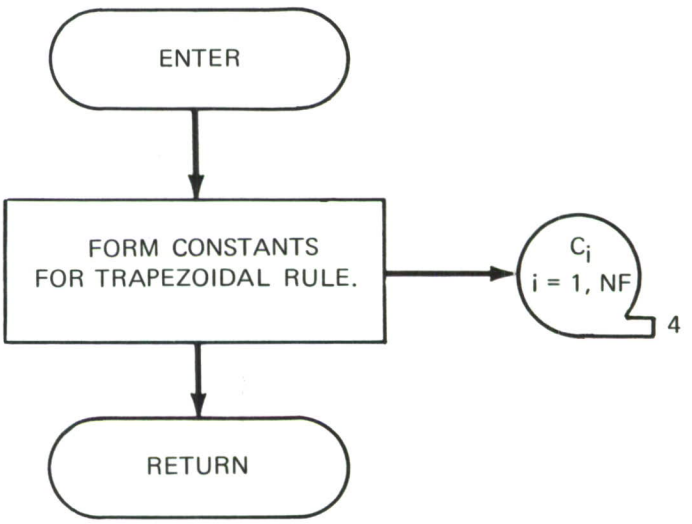


Figure 45. CONS Flow Chart

(d) Subroutine DSECM1 (figure 46)

Method: The deflection second spectral moments are calculated by multiplying the cross PSD found in subroutine CPSD1 by $\omega_i^2 C_i$. Thus,

$$\left[M_\delta^2\right] = \sum_{i=1}^{m} \left[\Phi_\delta(i\omega_i)\right] \omega_i^2 C_i$$

Input: Constants C_i and cross-PSD matrices $[\Phi_\delta(i\omega_i)]$ come from a scratch tape.

Output: Deflection second spectral moments are stored on tape 12.

Error: Standard READTP/WRTETP error messages

Subroutines required: READTP/WRTETP

Argument list: None

Length: 40514_8

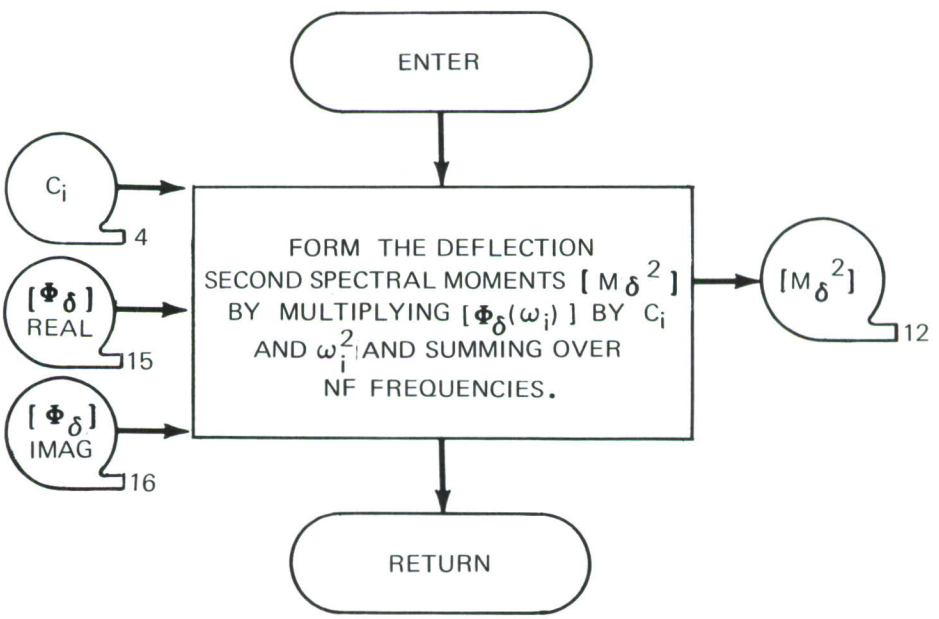


Figure 46. DSECM1 Flow Chart

(e) TL01 Subroutine COMINV

Solve for

$$[C] = ([A] - i\,[B])^{-1} .$$

The real part can be expressed as

$$([A] + [B]\,[A]^{-1}\,[B])^{-1}$$

and the imaginary part as

$$-\,[A]^{-1}\,[B]\left([A] + [B]\,[A]^{-1}\,[B]\right)^{-1} .$$

where: A = real part of a complex matrix

B = imaginary part of a complex matrix

The flow chart for the complex matrix-inversion subroutine is shown in figure 47.

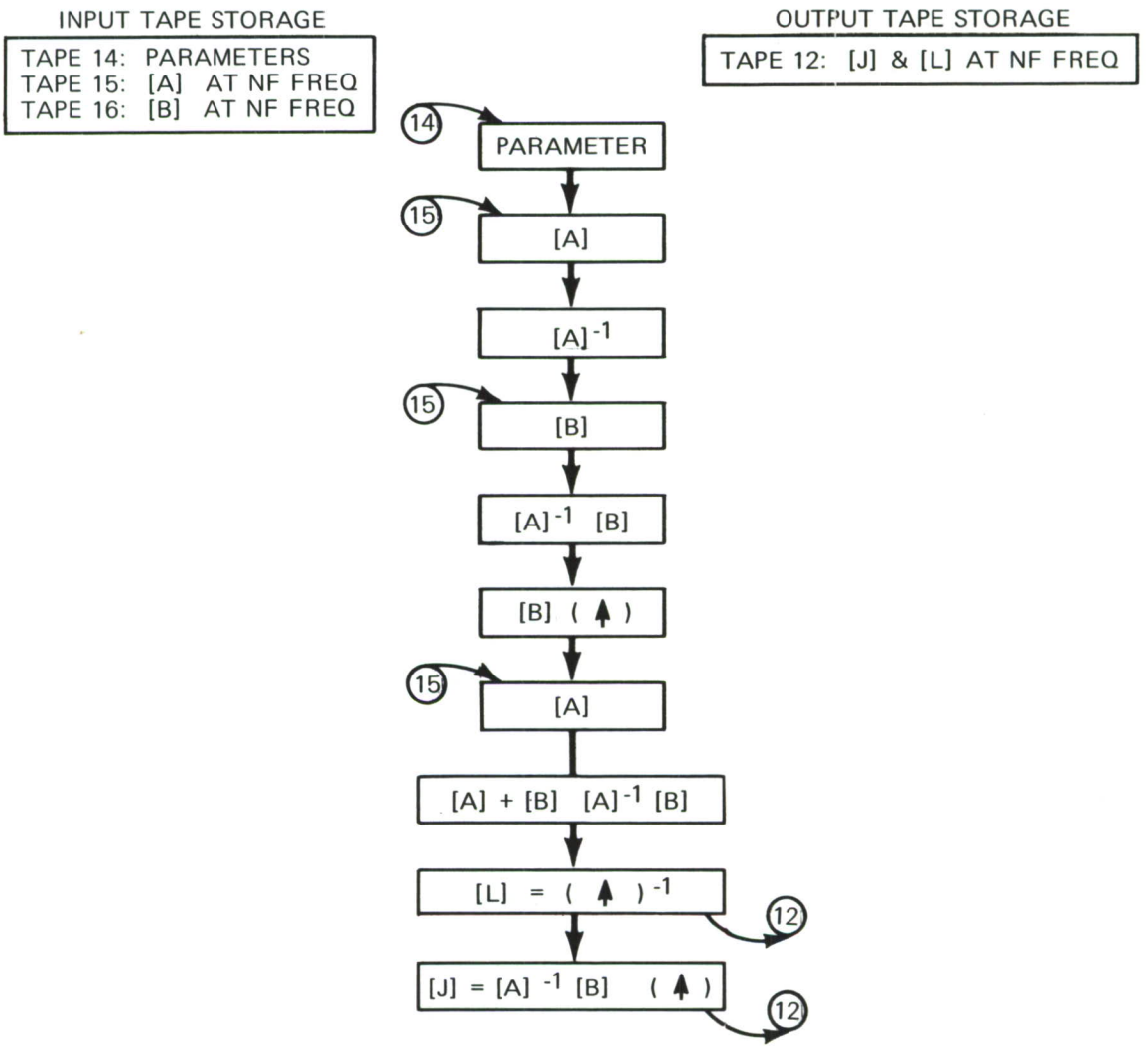


Figure 47. COMINV Flow Chart

(f) TL01 Subroutine SRESP1

The stress response cross-PSD matrices $[\Phi_\sigma(i\omega)]$ are calculated as shown in figure 48.

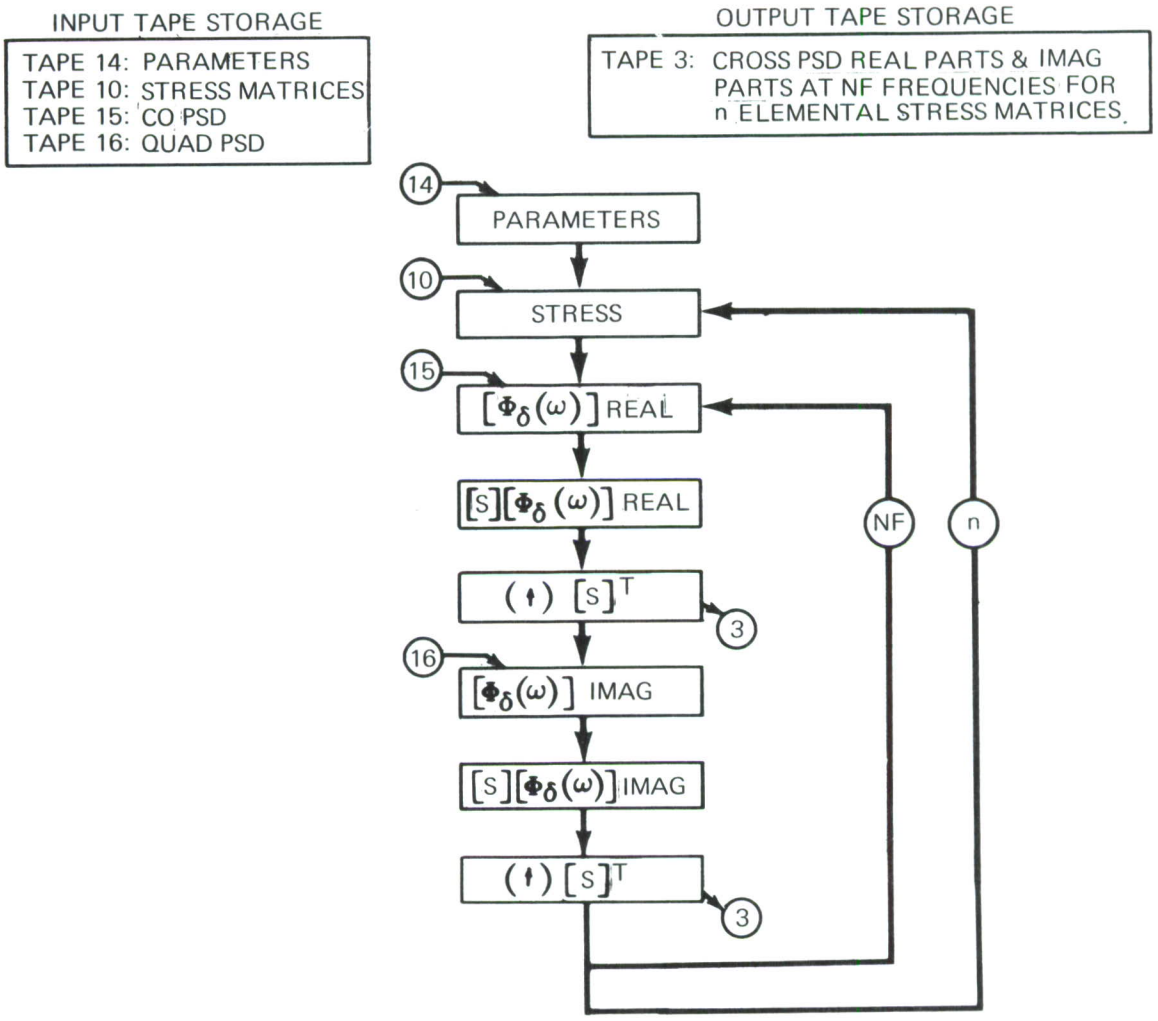


Figure 48. SRESP1 Flow Chart

(g) TL01 Subroutine TRAPM

The deflection covariance matrix $[\overline{\delta_q \delta_r}]$ is found by integrating the cross-PSD $[\Phi_\delta(i\omega)]$ matrices over NF frequencies by the trapezoidal method. Refer to figure 49.

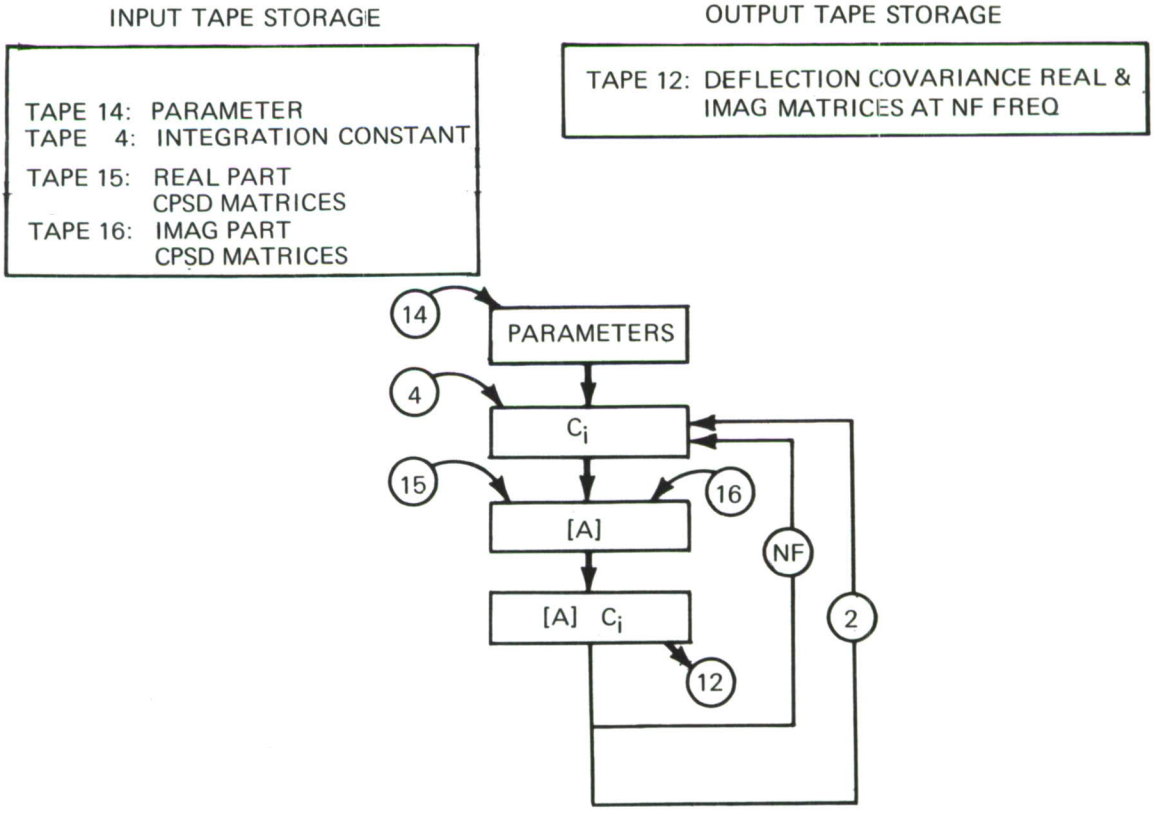


Figure 49. TRAPM Flow Chart

(h) TL01 Subroutine SJNT1

The stress covariance matrix $[\overline{\sigma_s \sigma_t}]$ at NF frequencies is calculated as shown in figure 50.

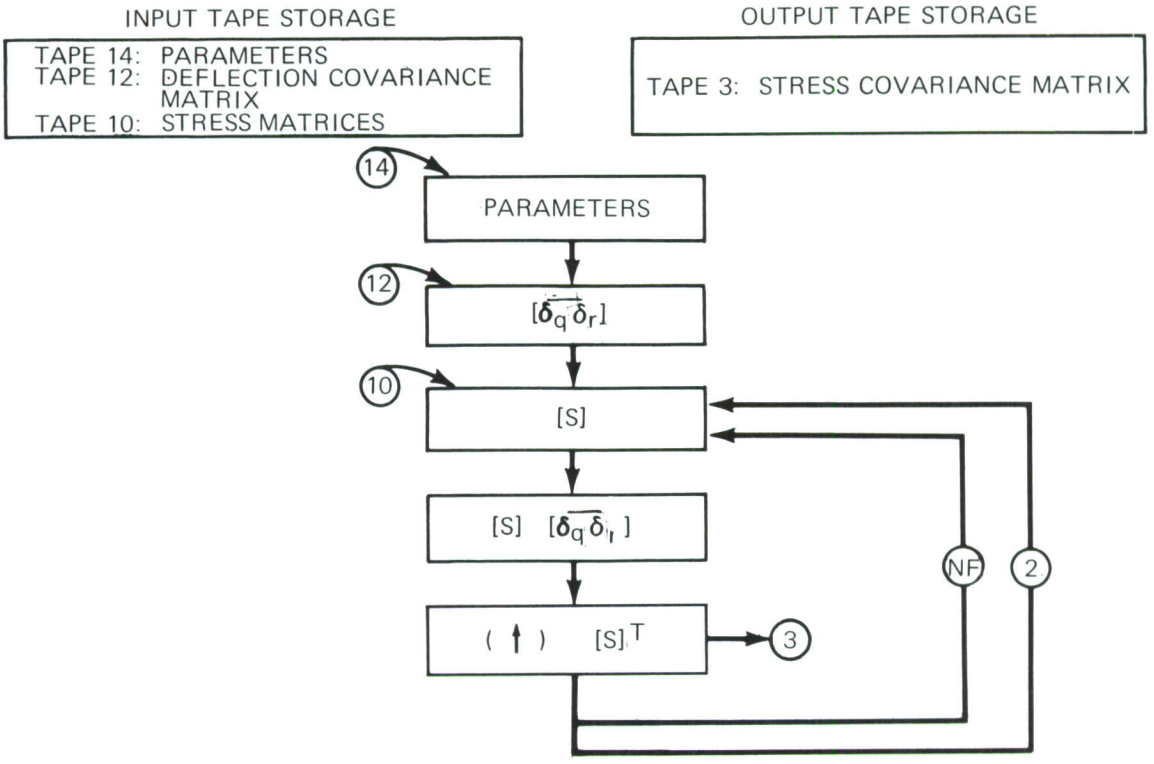


Figure 50. SJNT1 Flow Chart

(i) TL01 Subroutine SSECM

The stress second-spectral-moment matrix is calculated as shown in figure 51.

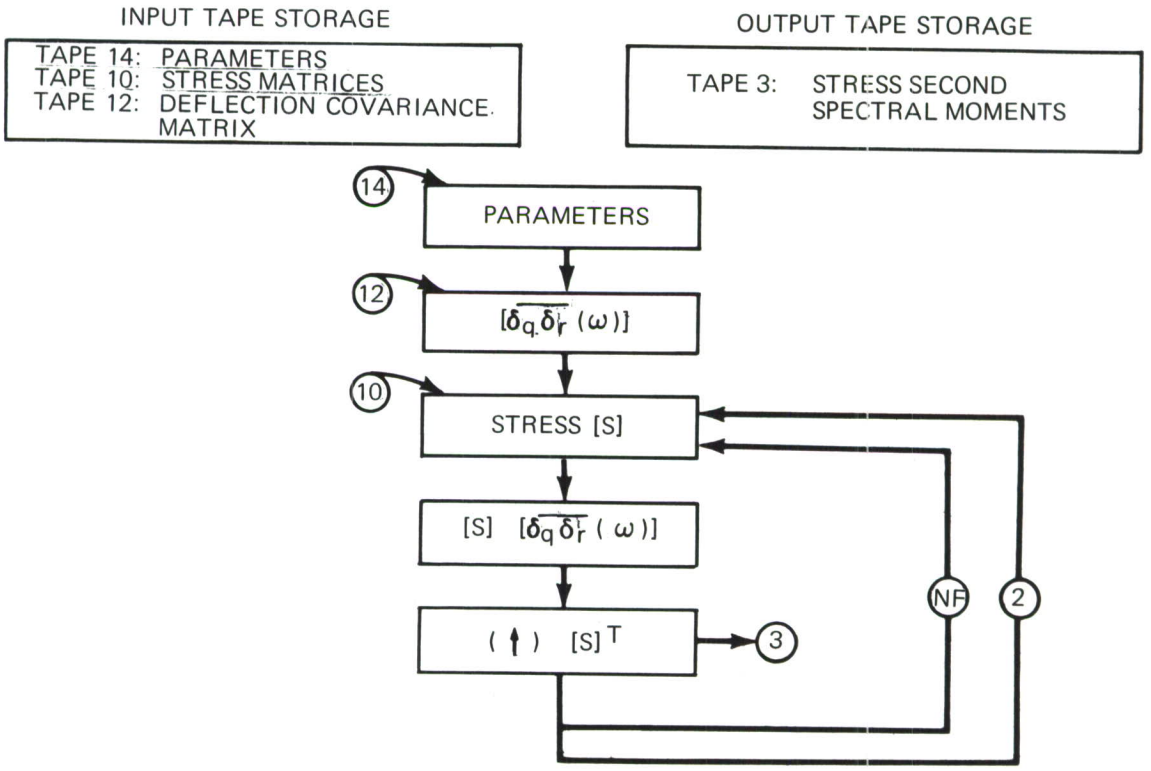


Figure 51. SSECM Flow Chart

(j) TL01 Subroutine CPSD1

The deflection response cross-PSD matrices are calculated for option 1. Refer to figure 52.

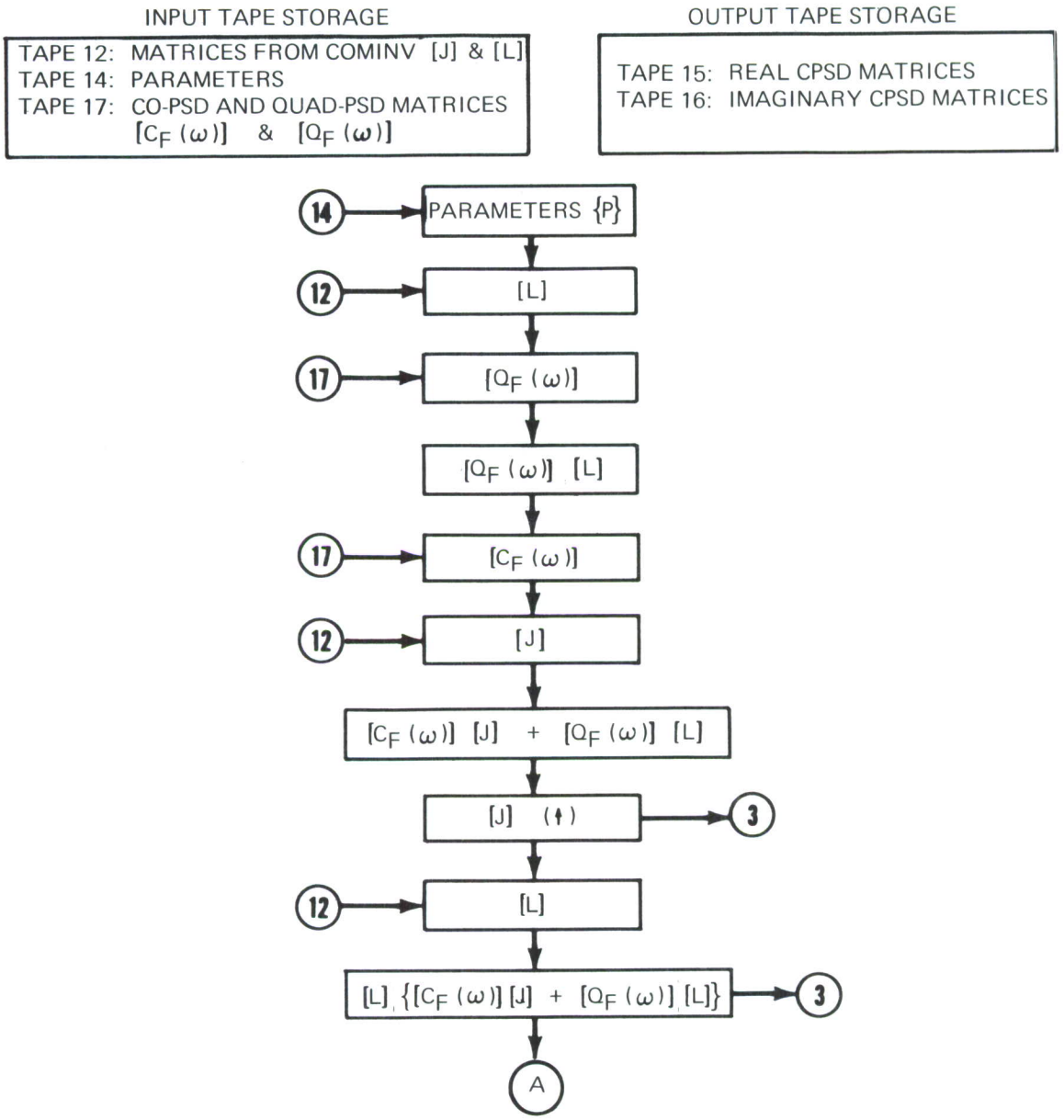


Figure 52. CPSD1 Flow Chart

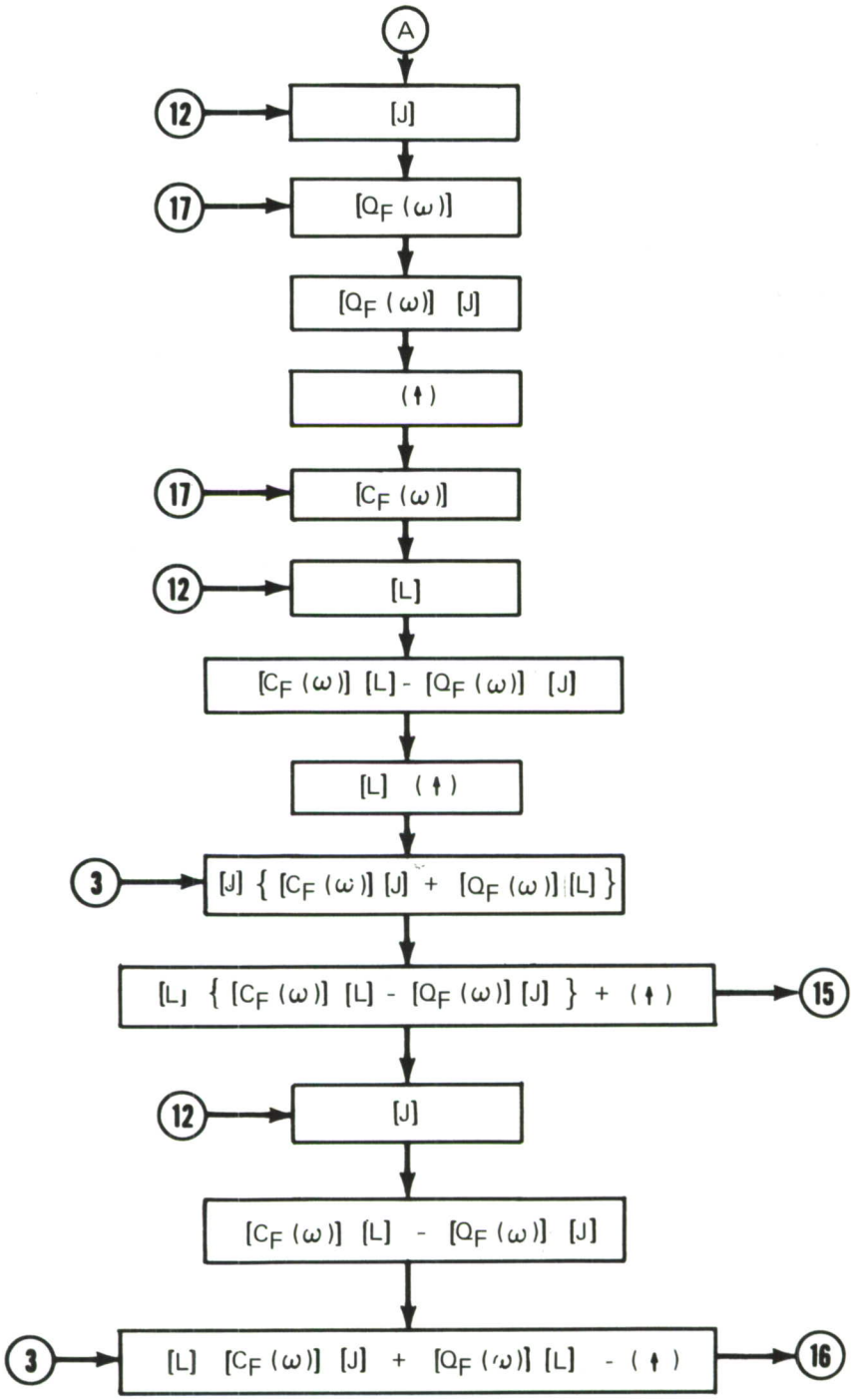


Figure 52. —Concluded

c. Option 2 Solution Program—Normal Modes

(1) Organization

The solution of option 2 is divided into two segments: (1) covariance, and (2) cross PSD. For simplicity , each segment is divided into a part that calculates contributions from like modes (same as option 3 and identified as option 3) and a part that calculates contributions from unlike modes. The summation of the like and unlike modal contributions results in the option 2 solution.

The effects of some of the cross modal terms are insignificant. A parameter K can be specified that limits the number of cross modal terms retained in the calculation of the solution. Cross PSD's are generated in two sets. The first set is for the natural frequencies and the second set is for the cross-modal terms. Frequencies for the cross modal term are in the order

$$\omega_{ij} = \frac{\omega_i + \omega_j}{2};$$

where: $i = 1, 2 \ldots, m - 1$

$j = i + 1, \ldots, i + K \leq m$

K = an input parameter defining the number of cross terms in the calculation

m = number of natural frequencies

The limitations are as follows:

$$N \leq 90$$
$$m \leq 25$$
$$NF \leq 90$$
$$1 \leq K \leq m - 1$$

(a) Segment 1—Deflection Covariance

Step 1: The excitation co-PSD (real part) and quad-PSD (imaginary part) matrices are generated in subroutine RANLOD. The excitations are calculated for like and unlike modes.

Step 2: The admittance integral scalars for option 3 are calculated in subroutine ADMIN3. See section IV 2. d. (2)(b).

Step 3: The deflection covariance matrices for like modes are generated in TL01 subroutine DJNT3, section IV 2. d. (2)(f), for each normal mode. In subroutine ADDMAT, section IV 2. d. (2)(c), the deflection covariance matrices are summed over m normal modes.

Step 4: The admittance integral scalars are calculated in subroutine ADMIN2, section IV 2. c. (2)(c). If a parameter K is specified, there are K cross-product terms of the admittance integral scalars formed for each mode.

Step 5: In subroutine CQJD, section IV 2. c. (2)(e), the excitations calculated in step 1 are combined with the admittance integral scalars calculated in step 4, and the resultant matrices are stored on a scratch tape that is used in deflection covariance subroutine DJNT2. See section IV 2. c. (2)(m).

Step 6: The deflection covariance for K cross terms for each mode are calculated in TL01 subroutine DJNT2. Real and imaginary parts of the deflection covariance matrices are generated for each cross mode and stored on scratch tape. Subroutine ADDMAT sums all the cross-modal deflection covariance matrices for the real and imaginary parts.

Step 7: The real part of the matrix calculated in step 6 is added to its transpose to form the real part of the deflection covariance matrix. The imaginary part of the matrix calculated in step 6 is added to its negative transpose to form the imaginary part of the deflection covariance matrix. This is done in subroutine SUMT. See section IV 2. c. (2)(f).

Step 8: The deflection covariance matrices of like and unlike modes are calculated in subroutine SUM2, section IV 2. c. (2)(g). This subroutine adds contributions from like modes obtained in option 3 to contributions from the unlike modes obtained in option 2.

Step 9: The stress covariance matrices are generated in TL01 subroutine SJNT2, section IV 2. c. (2)(n). The deflection covariance matrices (real and imaginary parts) are premultiplied by [S] and postmultiplied by $[S]^T$.

Step 10: The second-spectral-moment scalars of option 3 are calculated in subroutine SECM3. See section IV 2. c. (2)(h).

Step 11: The second-spectral-moment matrices for like modes (same as in option 3) are calculated in TL01 subroutine DJNT3, section IV 2. d. (2)(f), for each mode and stored on tape. Subroutine ADDMAT takes the m deflection second-spectral-moment matrices and sums for the normal modes.

Step 12: The scalars from the integrals used in the second spectral moments are calculated in subroutine SECM2, section IV 2. c. (2)(i), for like and unlike modes.

Step 13: The excitation matrices are combined with the scalars (same as in step 5 with different scalar values) in subroutine CQJD, and the resultant matrices are stored on scratch tape.

Step 14: The deflection second spectral moments are generated in TL01 subroutine DSECM2, section IV 2. c. (2)(o), for each cross mode. Subroutine ADDMAT is used to sum the deflection second-spectral-moment matrices over m unlike modes.

Step 15: The real matrix calculated in step 14 is added to its transpose to form the real part of the deflection second-spectral-moment matrix. The imaginary part of the matrix calculated in step 14 is added to its negative transpose to form the real part of the deflection second-spectral-moment matrix. This is done in subroutine SUMT.

Step 16: The summation of the deflection second-spectral-moment matrix results from contributions from both like and unlike modes for the real and imaginary parts as done in subroutine SUM2.

Step 17: The calculation of the stress second-spectral-moment matrix for real and imaginary parts is done in TL01 subroutine SSECM2. See section IV 2. c. (2)(p).

(b) Segment 2—Cross-PSD Solution

Step 1: Excitation co-PSD and quad-PSD matrices are generated in subroutine RANLOD for each frequency. The option 3 excitation co-PSD matrices are also calculated for each frequency. The NF selected frequencies are card input.

Step 2: The admittance scalars are calculated in subroutine ADMIT3 for each frequency. See section IV 2. d. (2)(e).

Step 3: The like-mode contributions to the deflection cross-PSD matrix solutions are generated in TL01 subroutine CPSD3. See section IV 2. d. (2)(i).

Step 4: The like-mode contributions to the deflection cross-PSD matrix solutions calculated in step 3 are summed in subroutine ADDMAT over m normal modes for each NF frequency.

Step 5: The admittance scalars are calculated in subroutine ADMIT2, section IV 2. c. (2)(j), for each frequency. If a parameter K is specified, there are K cross-product scalars formed at each mode for each frequency.

Step 6: In subroutine CQCPSD, section IV 2. c. (2)(k), the excitations calculated in step 1 are combined with the admittance scalars calculated in step 5 for each frequency, and the resultant matrices are stored on scratch tape.

Step 7: A component of each of the deflection co- and quad-PSD matrices is calculated at each mode for NF frequencies in TL01 subroutine CPSD2. See section IV 2. c. (2)(q). The ADDMAT subroutine is used to sum the cross-modal matrices for each mode at NF frequencies.

Step 8: The component co-PSD matrix calculated in step 7 is added to its transpose to form the deflection co-PSD matrix. The component quad-PSD matrix calculated in step 7 is added to its negative transpose to form the deflection quad-PSD matrix. There are NF co- and quad-PSD solutions for cross-modal contributions. This is done in subroutine SUMT.

Step 9: Subroutine SUM3 sums matrices obtained in steps 4 and 8 to obtain the deflection cross-PSD matrices.

Step 10: The stress cross PSD is generated in TL01 subroutine SRESP2. See section IV 2. c. (2)(v).

(2) Subroutine Descriptions

(a) Subroutine RANLOD (Refer to section IV 1.)

(b) Subroutines ADMIN3, ADDMAT, DJNT3, ADMIT3, and CPSD3 (See the subroutine descriptions for option 3, section IV 2. d. (2).)

(c) Subroutine ADMIN2

This subroutine (figure 53) calculates and stores on tape the admittance integral scalars used in the calculation of deflection covariance.

Method: The admittance integral scalars

$$\int_0^\infty D_i E_i \, d\omega, \quad \int_0^\infty D_j E_i \, d\omega$$

and

$$\int_0^\infty (D_i D_j + E_i E_j) \, d\omega$$

are combined in the following manner:

$$DE_{ij} = \int_0^\infty D_i E_j \, d\omega - \int_0^\infty D_j E_i \, d\omega$$

$$ED_{ij} = \int_0^\infty D_j E_i \, d\omega - \int_0^\infty D_i E_j \, d\omega$$

$$DDEE_{ij} = \int_0^\infty (D_i D_j + E_i E_j) \, d\omega$$

where: $i = 1, m - 1$

$j = i + 1, \; i + K \leq m$

See the Engineering User's Guide, reference 3, for definitions of the integrals.

Input: Mode shapes from phase I output tape

Output: Admittance integral scalars are stored on tape.

Error: READTP/WRTETP error messages

Subroutines required: READTP/WRTETP, SCALE

Length: 15301_8

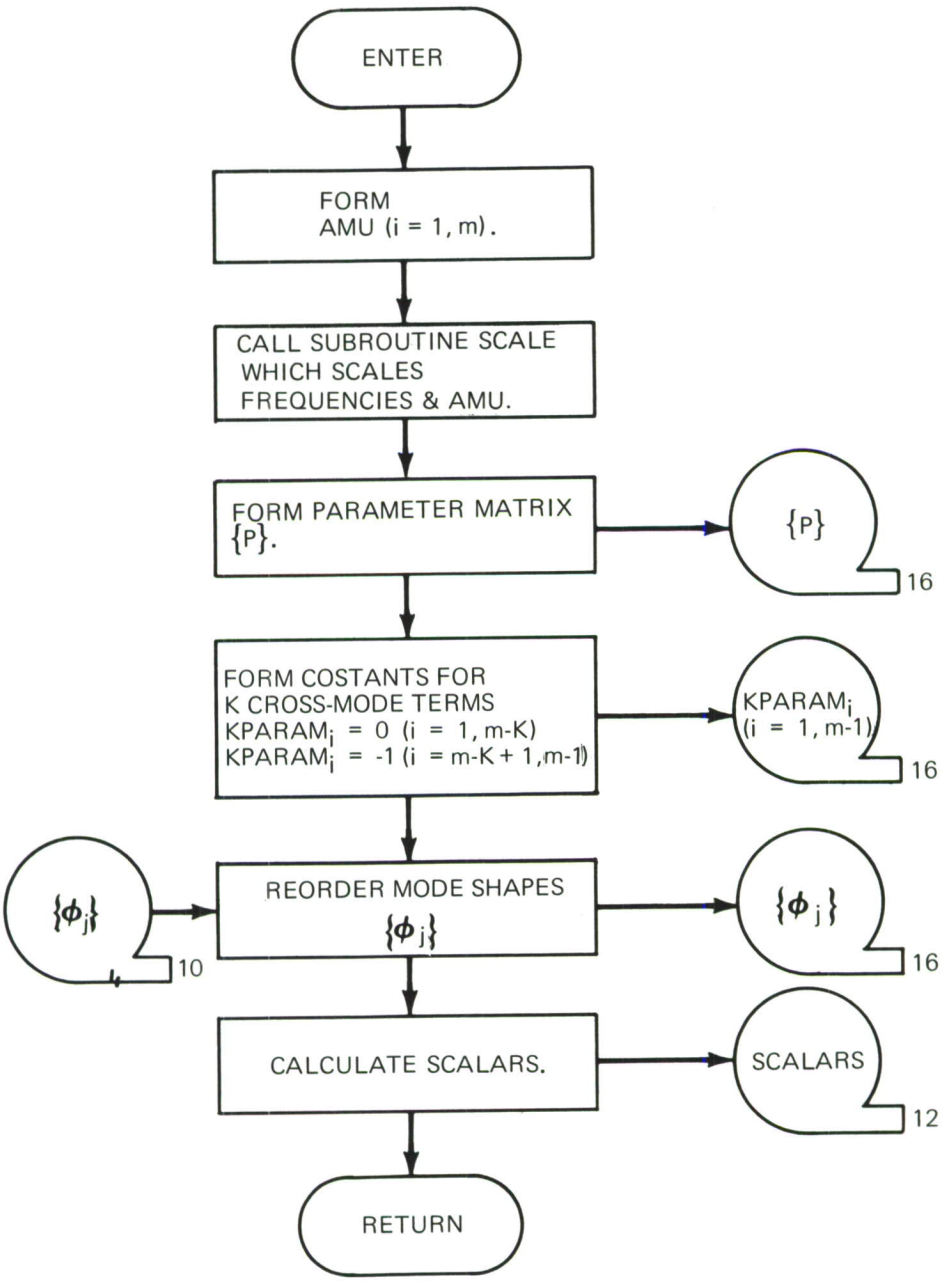


Figure 53. ADMIN2 Flow Chart

(d) Subroutine SCALE

This subroutine (figure 54) examines the magnitude of the first natural frequency and then uses an appropriate factor to scale frequencies.

Method: This subroutine will only scale frequencies ω from 0 to 100,000 Hz. If $0 \leq \omega \leq 100$

$100 < \omega \leq 1,000$

$1,000 < \omega \leq 10,000$

$10,000 < \omega \leq 100,000$

the scale factors are

10

100

1,000

10,000 respectively.

Input: Frequencies and m

Output: Scale factor and the scaled frequencies

Argument list: FREQ = frequencies

M = number of mode shapes

SCAL = scale factor

Length: 105_8

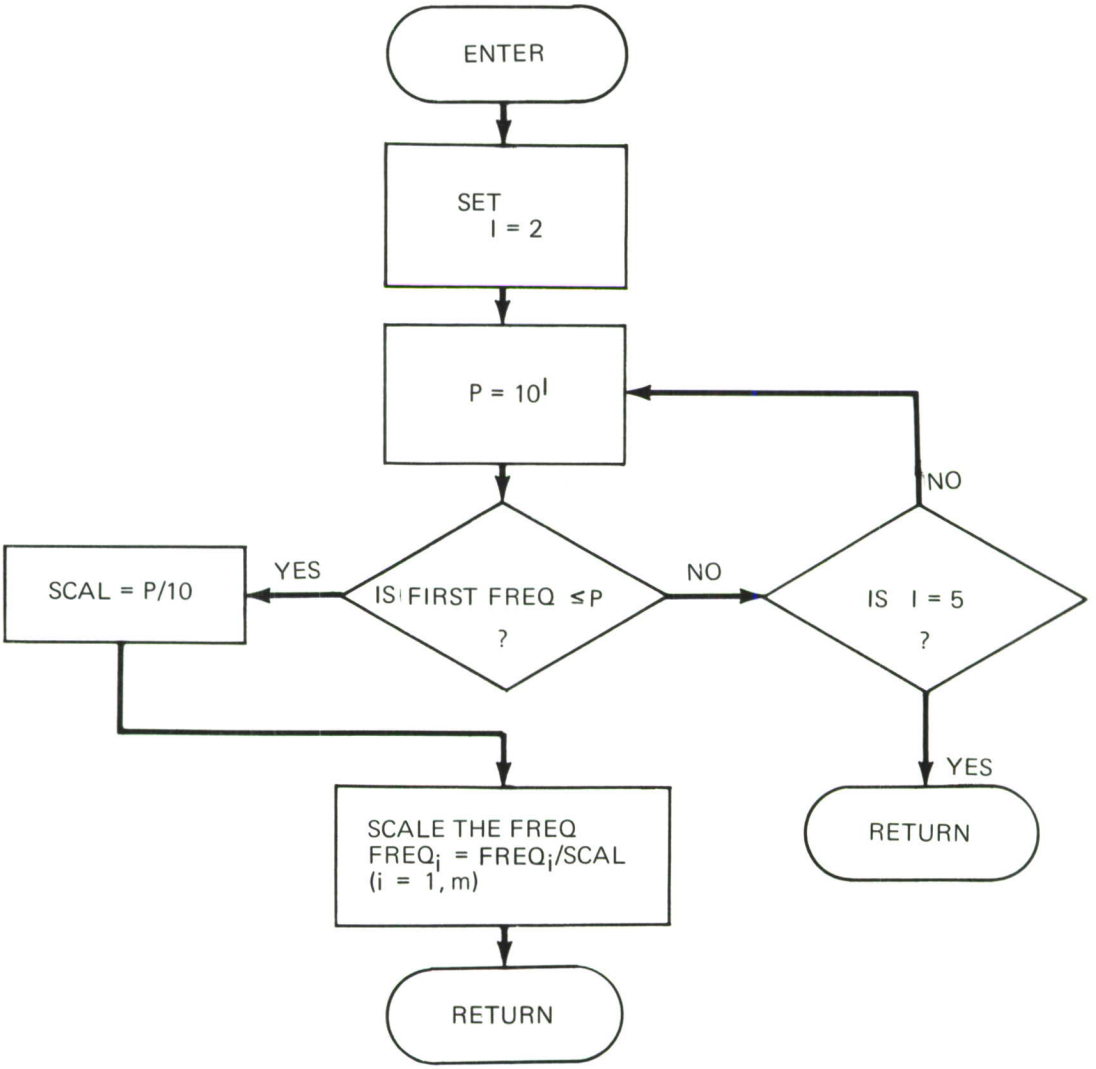


Figure 54. SCALE Flow Chart

(e) Subroutine CQJD

This subroutine (figure 55) combines the excitation co-power $[C_F(\omega)]$ and quad-power $[Q_F(\omega)]$ spectral density matrices and the admittance integral scalars when calculating the deflection covariance.

Method: A component of the deflection co-PSD is

$$[CQ]_{ij} = \alpha_{ij} [C_F]_{ij} + \gamma_{ij} [Q_F]_{ij}$$

A component of the deflection quad-PSD is

$$[CQ']_{ij} = - \gamma_{ij} [C_F]_{ij} + \alpha_{ij} [Q_F]_{ij}$$

where: $\alpha_{ij} = \int_0^\infty (D_i D_j + E_i E_j)\, d\omega$

$$\gamma_{ij} = \int_0^\infty D_i E_j \, d\omega - \int_0^\infty D_j E_i \, d\omega$$

$i = 1, m - 1$

$j = i + 1, i + K$

Input: Admittance integral scalars and the excitation co- and quad-PSD matrices from tape

Output: The $[CQ]_{ij}$ and $[CQ']_{ij}$ matrices are outputs on tape

Error: READTP/WRTETP error messages

Subroutines required: READTP/WRTETP

Length: 40300_8

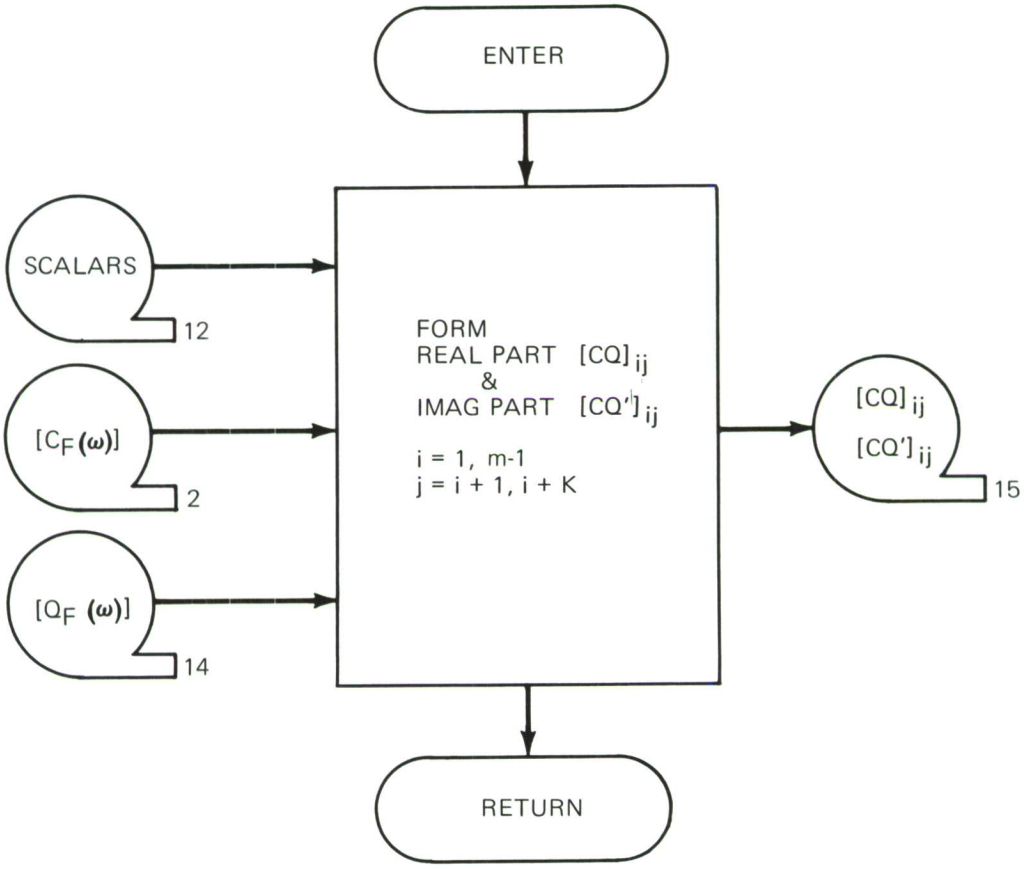


Figure 55. CQJD Flow Chart

(f) Subroutine SUMT

This subroutine (figure 56) sums a matrix with its transpose matrix.

Method: A matrix [A] is read from tape. The transpose of this matrix is added to itself, and the resultant matrix is the real part of the deflection covariance. To form the corresponding imaginary part, the transpose of this matrix is subtracted from itself. For the deflection cross-PSD calculations, this operation is repeated NF times, one for each frequency. For the real part,

$$[A] + [A]^T$$

and for the imaginary part,

$$[A] - [A]^T$$

Input: Tape input of matrices

Output: Tape output of the final deflection covariance or cross-PSD matrices

Error: READTP/WRTETP error messages

Subroutines required: READTP/WRTETP

Length: 40247_8

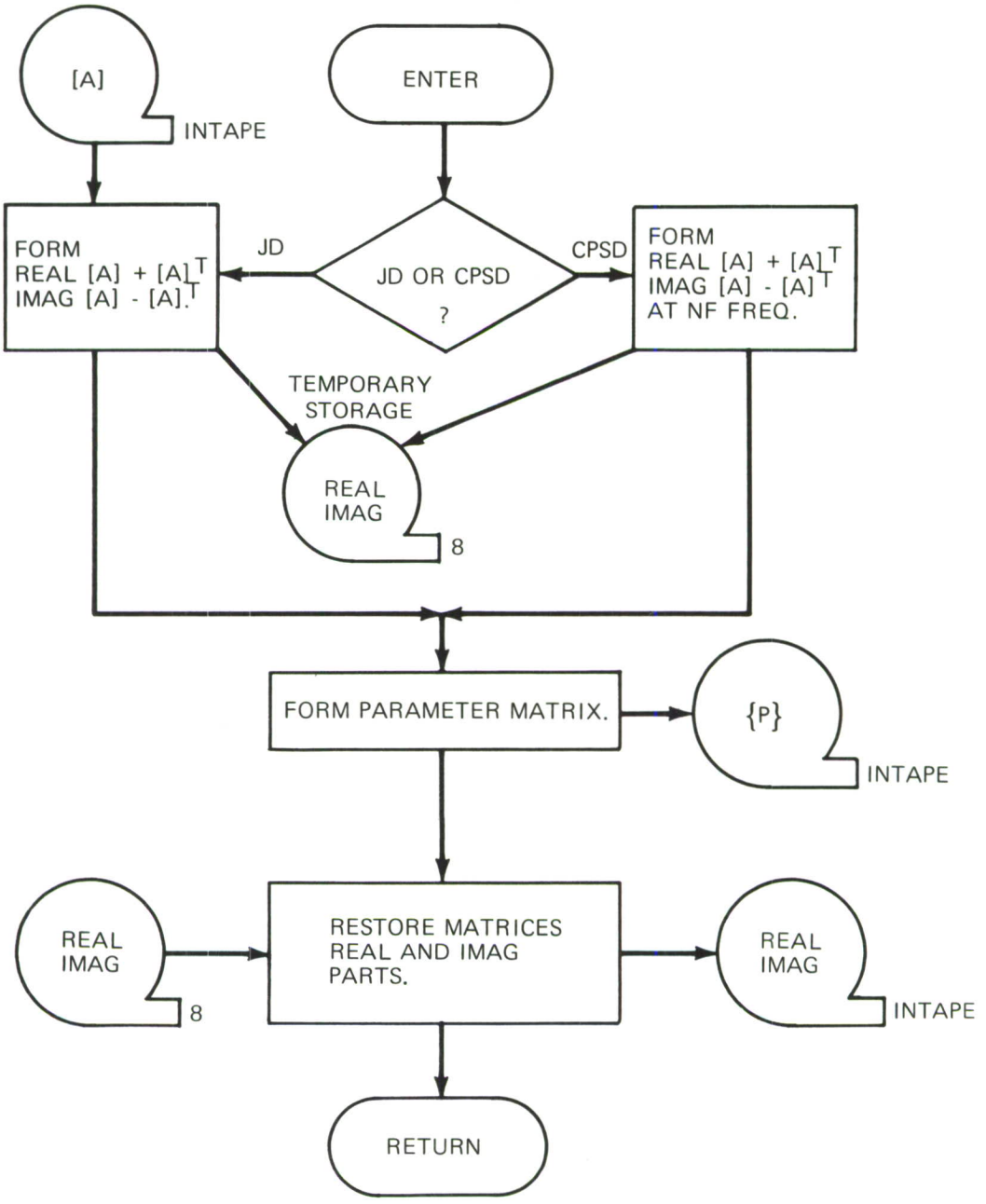


Figure 56. SUMT Flow Chart

(g) Subroutine SUM2

Method: This subroutine (figure 57) sums contributions to the real part of the deflection covariance matrix from like and unlike modes. The result is stored on an output tape.

The imaginary part of the deflection covariance matrix is transferred from its input tape to the output tape. Contributions come from only the unlike modes.

Input: Deflection covariance matrices for like modes stored on ITP1

Output: The deflection covariance matrices including contributions from like and unlike modes. This is stored on tape ITP1.

Error: READTP/WRTETP error messages

Subroutines required: READTP/WRTETP

Argument list: ITP1—Option 3 like-mode contributions to the deflection covariance matrices are stored on tape ITP1.

ITP2—Option 2 unlike-mode contributions to the deflection covariance matrices (real and imaginary parts) are stored on tape ITP2.

NO—If NO = 1, then calculate option 3
If NO = 2, then calculate option 2

NCN—If NCN = 1, then sum the deflection covariance matrices
If NCN = 2, then sum the deflection second spectral moment matrices

Length: 40370_8

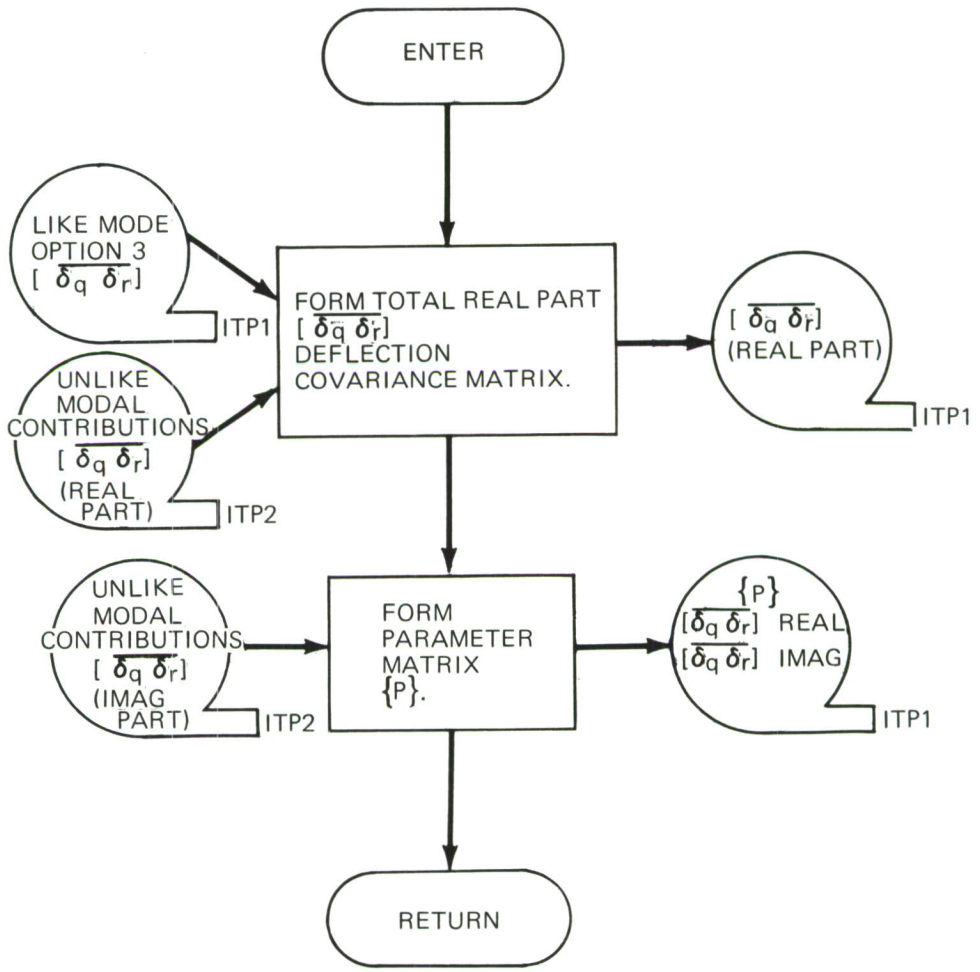


Figure 57. SUM2 Flow Chart

(h) Subroutine SECM3

This subroutine (figure 58) calculates the admittance integral scalars used in the deflection second spectral moments. The subroutine also reads mode shapes from the phase I output tape and restores them on another tape.

Method: $\left[M_{\delta}^{2}\right] = \pi/\left(2\mu_i M_i^2\right)$

where: $\mu_i = \mu + \lambda\omega_i^2 + g\,\omega_i$

$i = 1, m$

Input: Mode shapes $\{\phi_i\}$ come from phase I output tape.

Output: A parameter matrix, admittance integral scalars, and modes stored on tape.

Error: READTP/WRTETP error messages

Subroutines required: READTP/WRTETP

Length: 5423_8

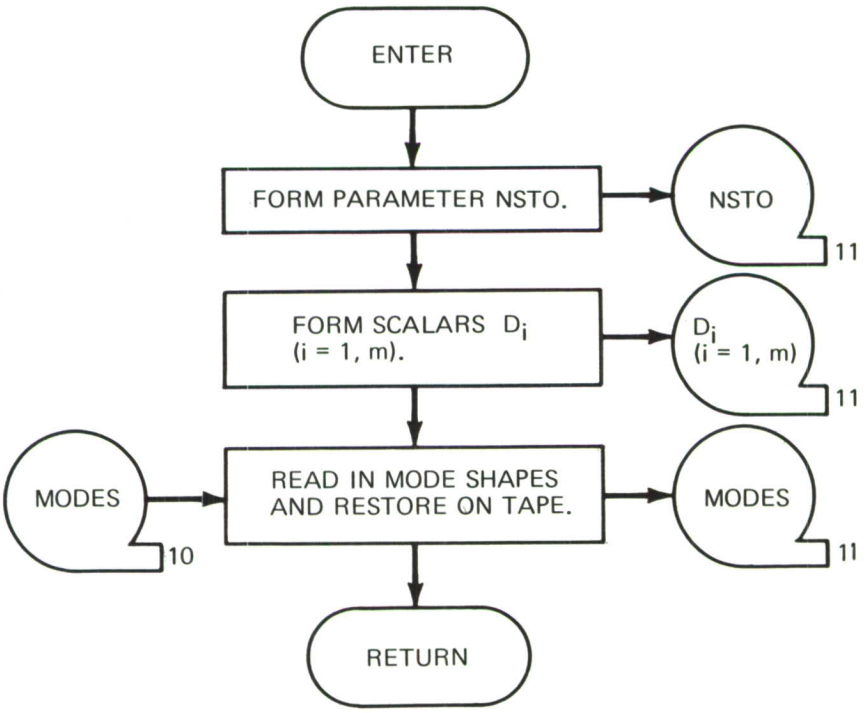


Figure 58. SECM3 Flow Chart

(i) Subroutine SECM2

The second-spectral-moment scalars for option 2 deflection covariance are calculated in subroutine SECM2 in the same manner as in subroutine ADMIN2, section IV 2. c. (2)(c).

(j) Subroutine ADMIT2

This subroutine (figure 59) calculates the admittance scalars used in the formation of cross PSD. The cross-modal constants and mode shapes are stored on tape.

Method: The admittance scalars are combined in the following manner:

$$DDEE_{ij} = D_i D_j + E_i E_j$$

$$DE_{ij} = D_i E_j - D_j E_i$$

$$ED_{ij} = D_j E_i - D_i E_j$$

where: $i = 1, m - 1$

$j = i + 1, i + K$

See the Engineering User's Guide, reference 3, for a definition of D_i and E_i .

Input: Modes shapes from phase I output tape

Output: Admittance scalars on tape

Error: Standard READTP/WRTETP error messages

Subroutines required: READTP/WRTETP

Length: 14566_8

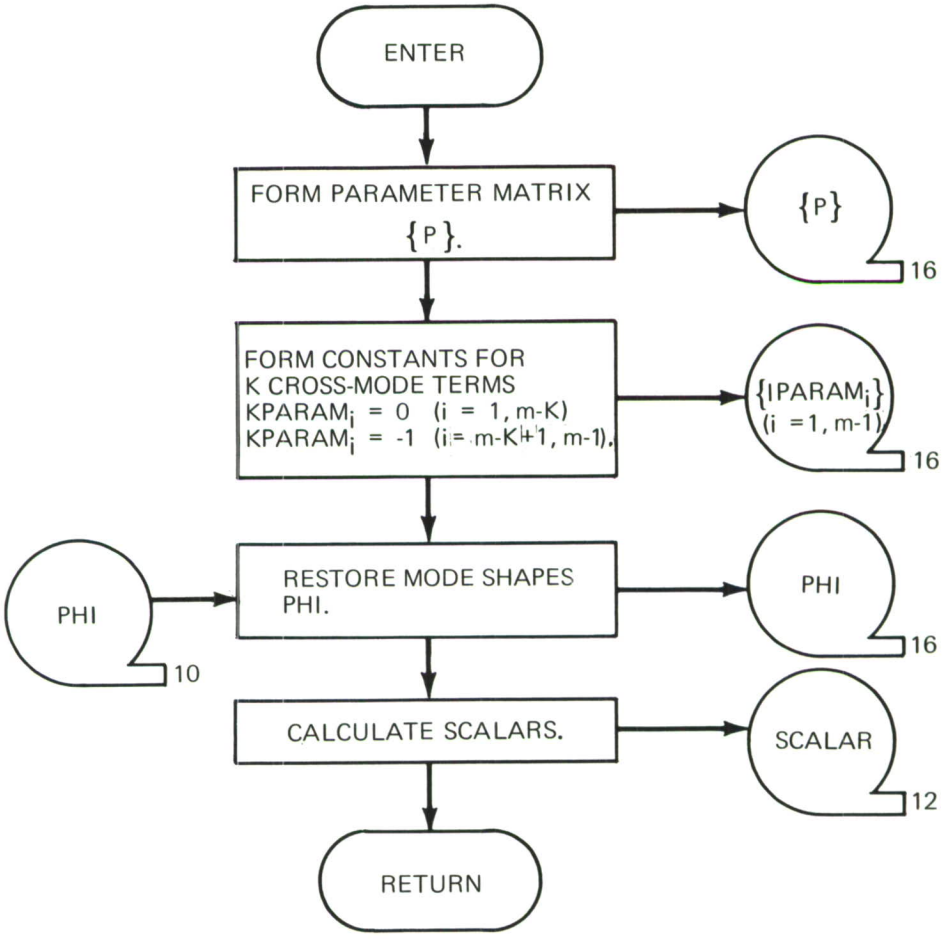


Figure 59. ADMIT2 Flow Chart

(k) Subroutine CQCPSD

This subroutine (figure 60) combines the excitation matrices and the admittance scalars for the real part and imaginary part in generating the cross-PSD matrices.

Method: For the real part,

$$[CQ]_{ij} = \alpha_{ij} [C_F]_{ij} + \gamma_{ij} [Q_F]_{ij}$$

For the imaginary part,

$$[CQ']_{ij} = -\gamma_{ij} [C_F]_{ij} + \alpha_{ij} [Q_F]_{ij}$$

where: $\alpha_{ij} = D_i D_j + E_i E_j$

$\gamma_{ij} = D_i E_j - D_j E_i$

$i = 1, m - 1$

$j = i + 1, i + K$

Input: Admittance scalars and the excitation co- and quad-PSD matrices from tapes

Output: The resultant matrices are stored on tape.

Error: Standard READTP/WRTETP error messages

Subroutines required: READTP/WRTETP

Length: 15155_8

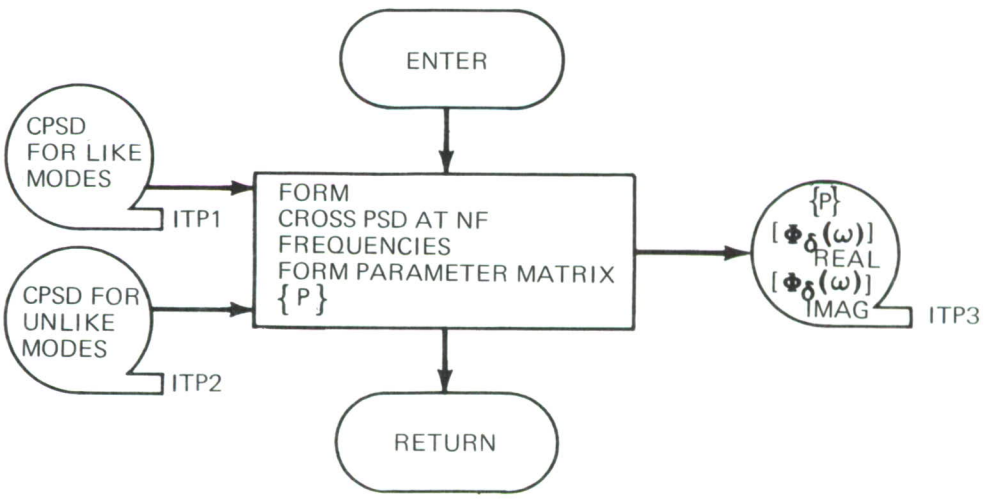


Figure 60. CQCPSD Flow Chart

(l) Subroutine SUM3

Method: This subroutine (figure 61) sums to obtain the deflection co-PSD matrices from like and unlike modal contributions.

The deflection quad-PSD matrices are transferred from an input tape to an output tape.

Input: Cross-PSD matrices for like and unlike modes are stored on ITP1 and ITP2

Output: Deflection cross-PSD matrices for NF frequencies are stored on ITP3

Error: READTP/WRTETP error messages

Subroutines required: READTP/WRTETP

Argument list: ITP1—Option 3 deflection cross-PSD matrices (like modal contributions) are stored on this tape.
ITP2—Cross-PSD matrices (contributions from unlike modes only) are stored on this tape.
ITP3—Output tape of summation of all deflection cross-PSD matrices for NF frequencies.

Length: 40343_8

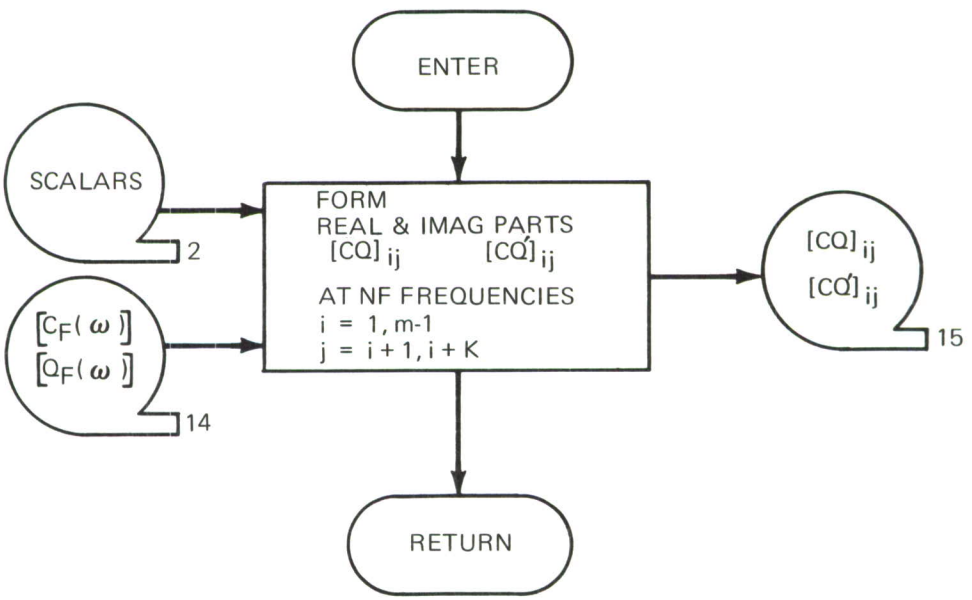


Figure 61. SUM3 Flow Chart

(m) TL01 Subroutine DJNT2

This subroutine (figure 62) calculates the deflection covariance matrices.

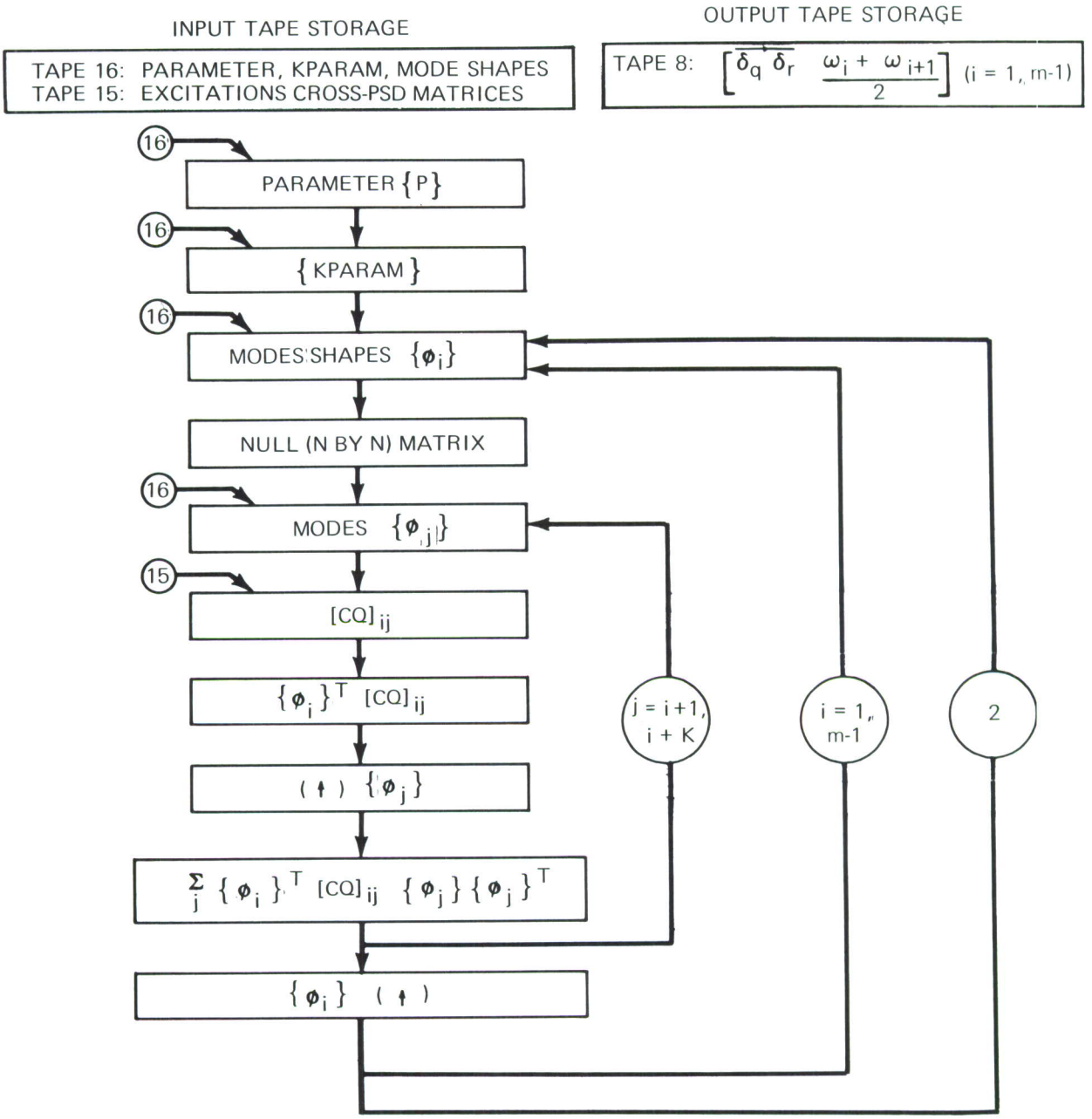


Figure 62. DJNT2 Flow Chart

(n) TL01 Subroutine SJNT2

This subroutine (figure 63) calculates the joint stress $[\overline{\sigma_s \sigma_t}]$.

INPUT TAPE STORAGE

TAPE 3: PARAMETERS $\{P\}$, DEFLECTION COVARIANCE MATRICES $[\overline{\delta_q \delta_r}]$
TAPE 10: STRESS MATRICES

OUTPUT TAPE STORAGE

TAPE 15: STRESS COVARIANCE MATRICES $[\overline{\sigma_s \sigma_t}]$

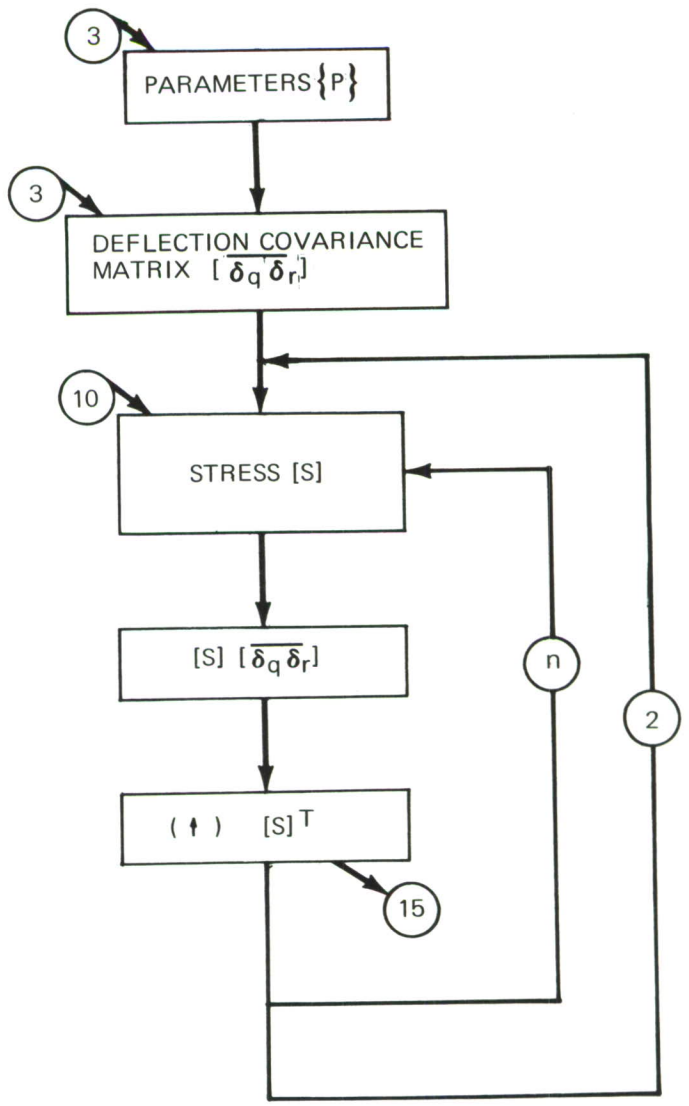


Figure 63. SJNT2 Flow Chart

(o) TL01 Subroutine DSECM2

This subroutine (figure 64) calculates the deflection second spectral moments.

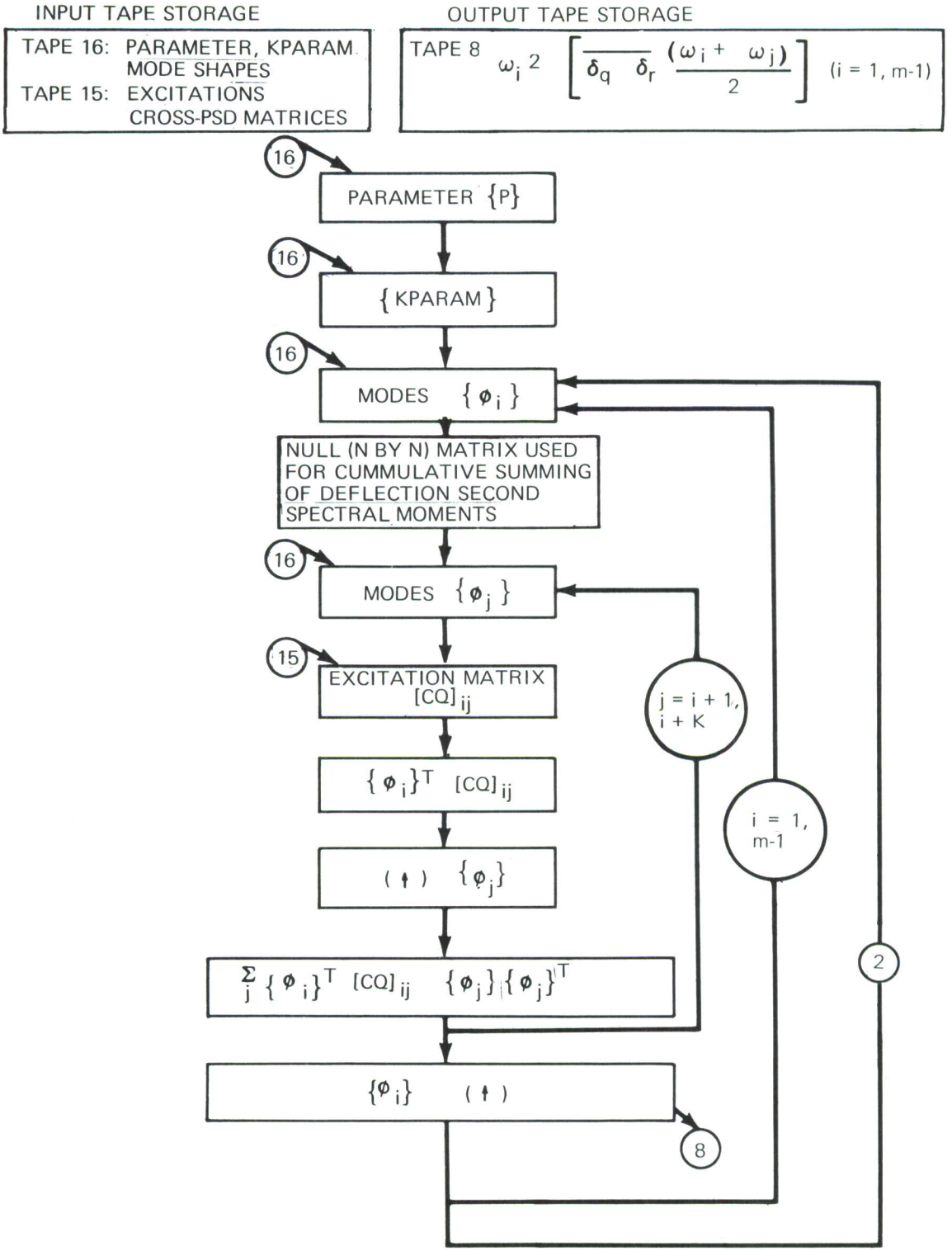


Figure 64. DSECM2 Flow Chart

(p) TL01 Subroutine SSECM2

This subroutine (figure 65) calculates the joint stress second spectral moments.

INPUT TAPE STORAGE	
TAPE 3:	PARAMETER, DEFLECTION SECOND SPECTRAL MOMENT MATRIX
TAPE 10:	STRESS MATRICES

OUTPUT TAPE STORAGE	
TAPE 15:	STRESS SECOND-SPECTRAL-MOMENT MATRIX

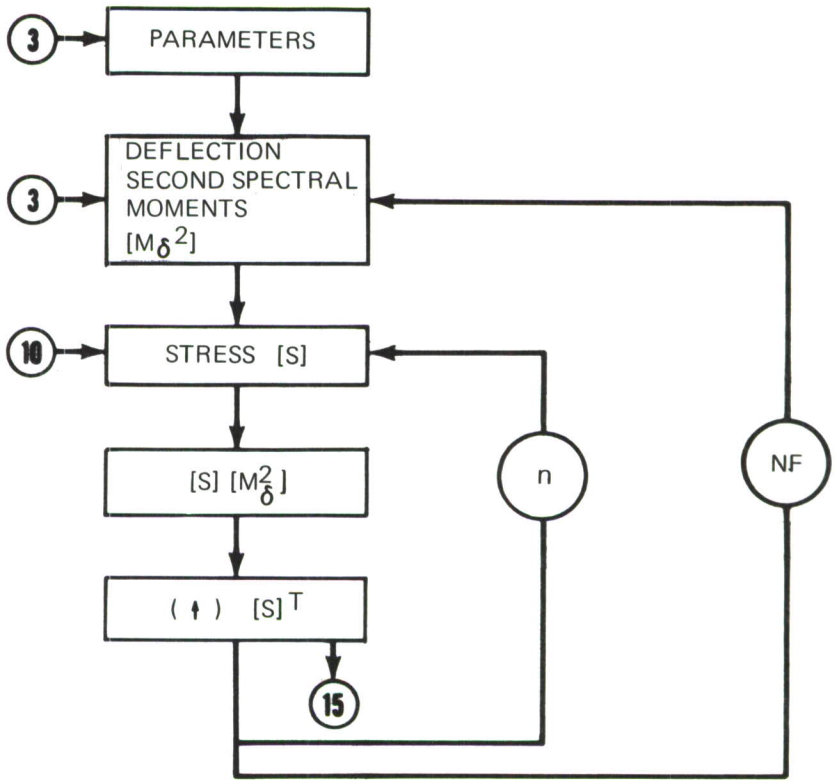


Figure 65. SSECM2 Flow Chart

(q) TL01 Subroutine CPSD2

This subroutine (figure 66) calculates the cross modal contributions to the deflection cross-PSD matrices at NF frequencies

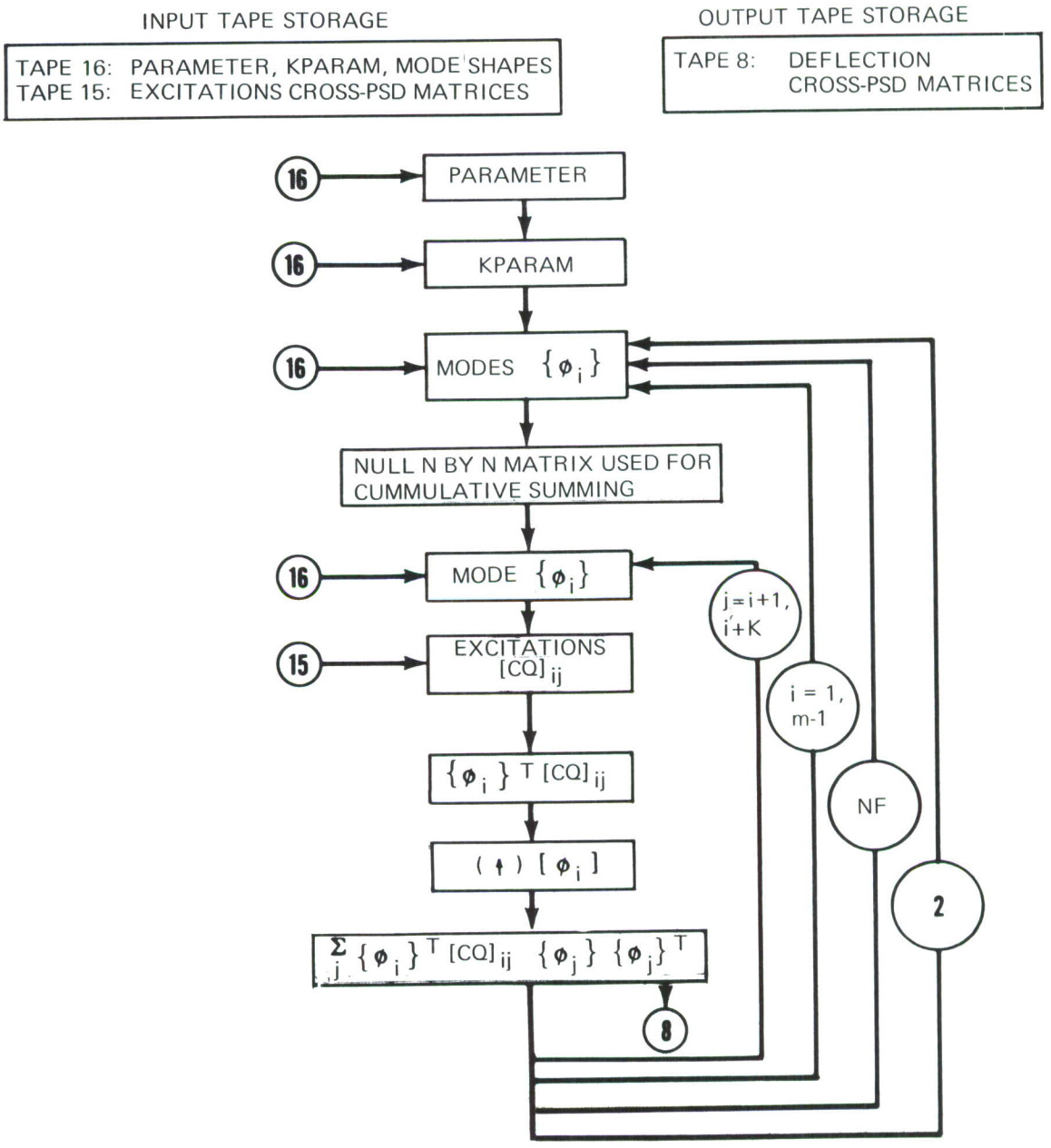


Figure 66. CPSD2 Flow Chart

(r) TL01 Subroutine SRESP2

This subroutine (figure 67) calculates the stress cross-PSD matrices for NF frequencies.

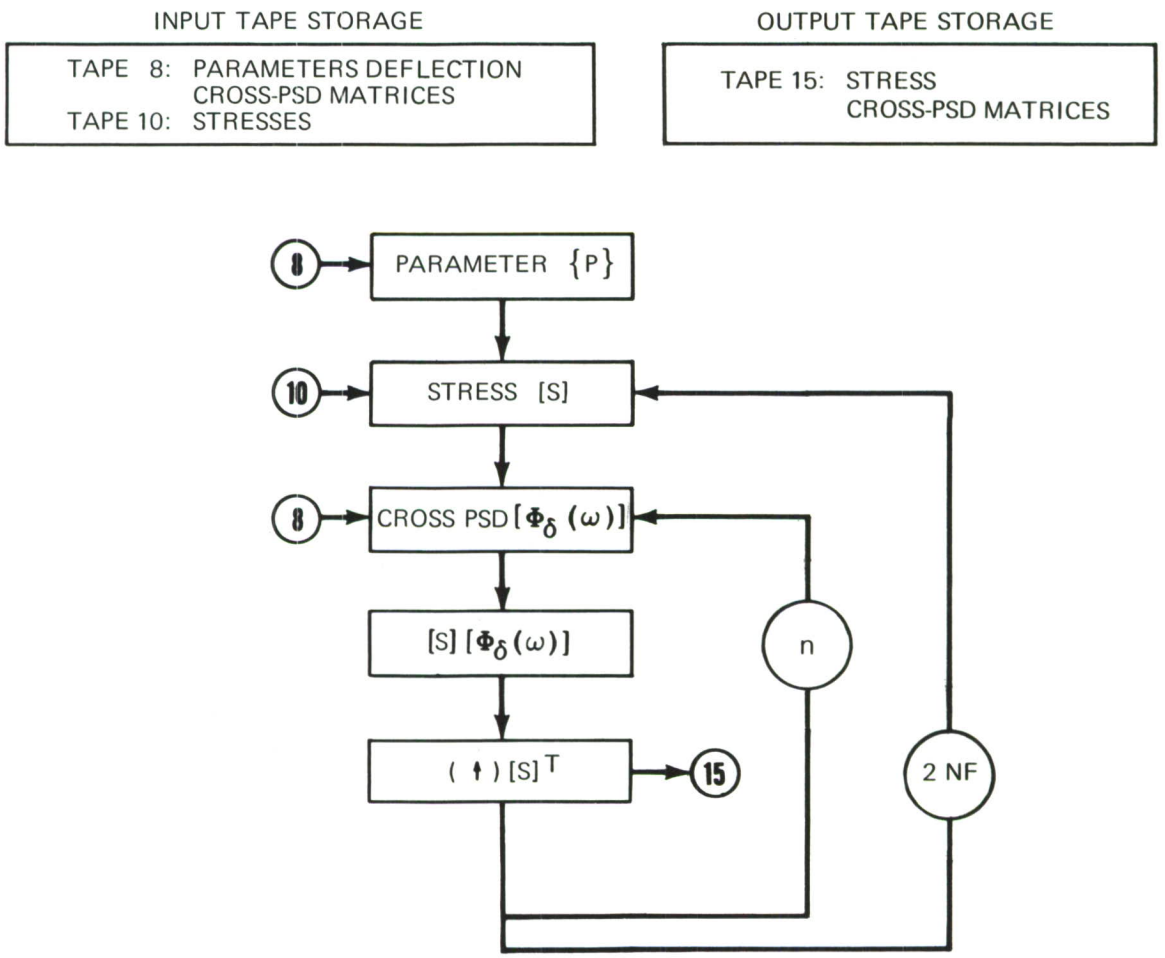


Figure 67. SRESP2 Flow Chart

d. Option 3 Solution Program—Normal Modes without Cross Terms

(1) Organization

Option 3 is in two segments: (1) deflection covariance and (2) deflection cross PSD. In this option, all modal cross-product terms are omitted. The solutions for the imaginary parts are null. The limitations are as follows:

$$N \leq 90$$
$$m \leq 25$$
$$NF \leq 90$$

(a) Segment 1—Deflection Covariance

Step 1: The excitation co-PSD matrices are calculated in subroutine RANLOD. See section IV 1.

Step 2: The admittance integral scalars are calculated in subroutine ADMIN3. See section IV 2. d. (2)(b).

Step 3: The deflection covariance matrix is calculated in TL01 subroutine DJNT3. See section IV 2. d. (2)(f). There are m (number of normal modes) deflection covariance matrices stored on a scratch tape. Subroutine ADDMAT, section IV 2. d. (2)(c), sums the deflection covariance matrices over m normal modes.

Step 4: The stress covariance matrices are calculated in TL01 subroutine SJNT3, section IV 2. d. (2)(g), using the stress matrices from the phase I output tape and the deflection covariance calculated in step 3. The deflection covariance is premultiplied by the stress matrices and postmultiplied by the transpose of these matrices.

Step 5: The deflection second-spectral-moment matrices are calculated in subroutine DSECM3. See section IV 2. d. (2)(d). The deflection covariance matrices calculated in step 3 are multiplied by ω_i^2 and summed over m normal modes.

Step 6: The stress second spectral moments are calculated in TL01 subroutine DSJNT3. See section IV 2. d. (2)(h). The deflection second-spectral-moment matrices calculated in step 5 are premultiplied by $[S]$ and postmultiplied by $[S]^T$.

(b) Segment 2—Cross-PSD Solution

Step 1: The excitation cross PSD's are calculated in subroutine RANLOD, section IV 1, for each frequency. The NF frequencies are card inputs.

Step 2: The admittance scalars are calculated in subroutine ADMIT3, section IV 2. d. (2)(e), for each frequency.

Step 3: The deflection cross-PSD solutions are calculated in TL01 subroutine CPSD3. See section IV 2. d. (2)(i).

Step 4: The deflection cross-PSD matrices calculated in step 3 are summed in subroutine ADDMAT, section IV 2. d. (2)(c), over m normal modes for each of the NF frequencies.

Step 5: The stress cross-PSD matrices are calculated in TL01 subroutine SRESP3. (See section IV 2. d. (2)(j).) The deflection cross-PSD matrices calculated in step 4 are premultiplied by the stress matrices and postmultiplied by the transpose of these matrices for each of the NF frequencies.

(2) Subroutine Descriptions

(a) Subroutine RANLOD (section IV 1.)

(b) Subroutine ADMIN3 (figure 68)

The admittance integral scalars D_i are calculated and stored on tape. The mode shapes are read from the phase I output tape and re-stored on tape.

Method:

$$D_i = \pi / \left(2 \mu_i \omega_i^2 M_i^2 \right)$$

where

$$\mu_i = \mu + \lambda \omega_i^2 + g \omega_i \quad (i = 1, m)$$

Input: All inputs come from labeled common.

Output: The scalars and mode shapes are stored on tape.

Error: READTP/WRTETP error messages

Subroutines required: READTP/WRTETP

Argument list: None

Length: 5540_8

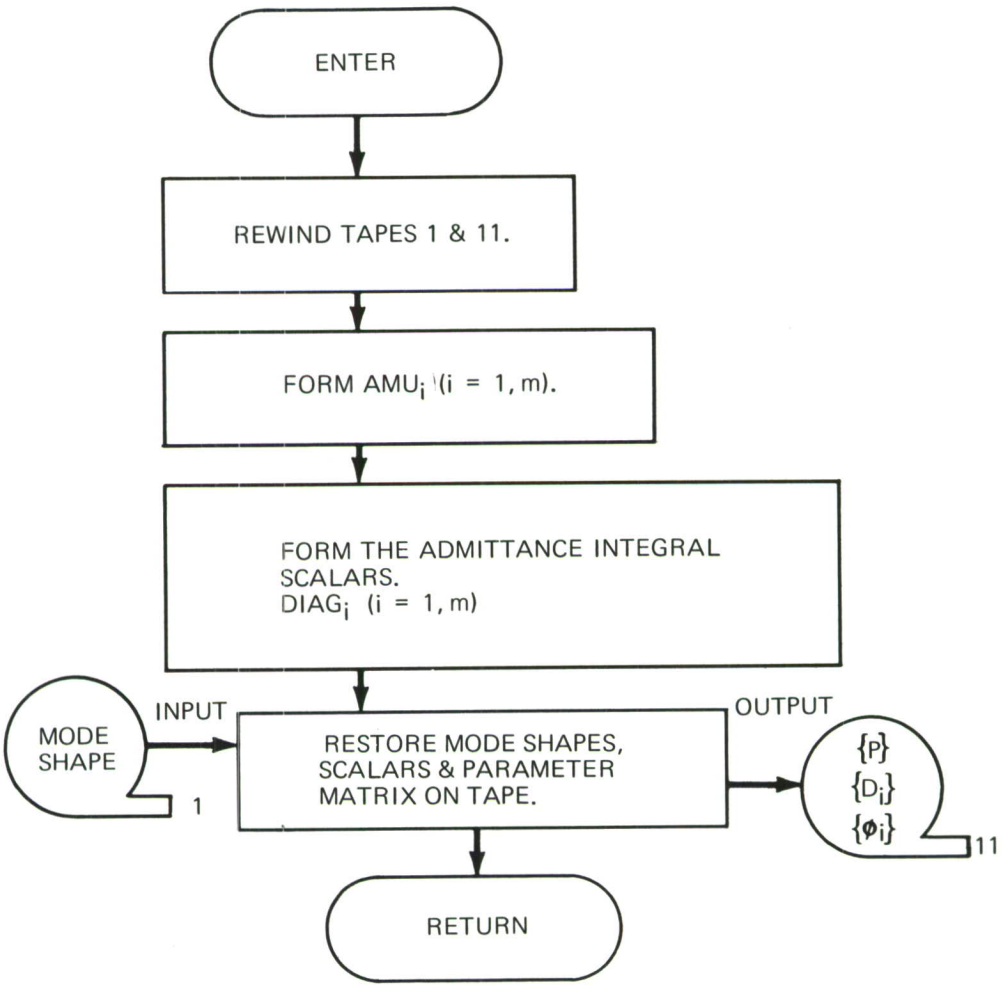


Figure 68. ADMIN3 Flow Chart

(c) Subroutine ADDMAT

This subroutine (figure 69) adds deflection covariance or cross-PSD matrices that are stored on tape.

Method: The matrices that are to be summed come from a tape (INTAPE). Each matrix is read into the core, summed, and stored on tape (OUTAPE).

Input: Matrices are input from INTAPE.

Output: The resultant matrix is stored on OUTAPE.

Error: Standard READTP/WRTETP error messages

Subroutines required: READTP/WRTETP

Argument list: INTAPE—Number of the tape that contains the matrices to be summed
OUTAPE—Number of the tape that contains the summation of the matrices from INTAPE
NO—For the cross-PSD calculation, this variable equals NF . For deflection covariance calculations, this variable equals 1.

Length: 40330_8

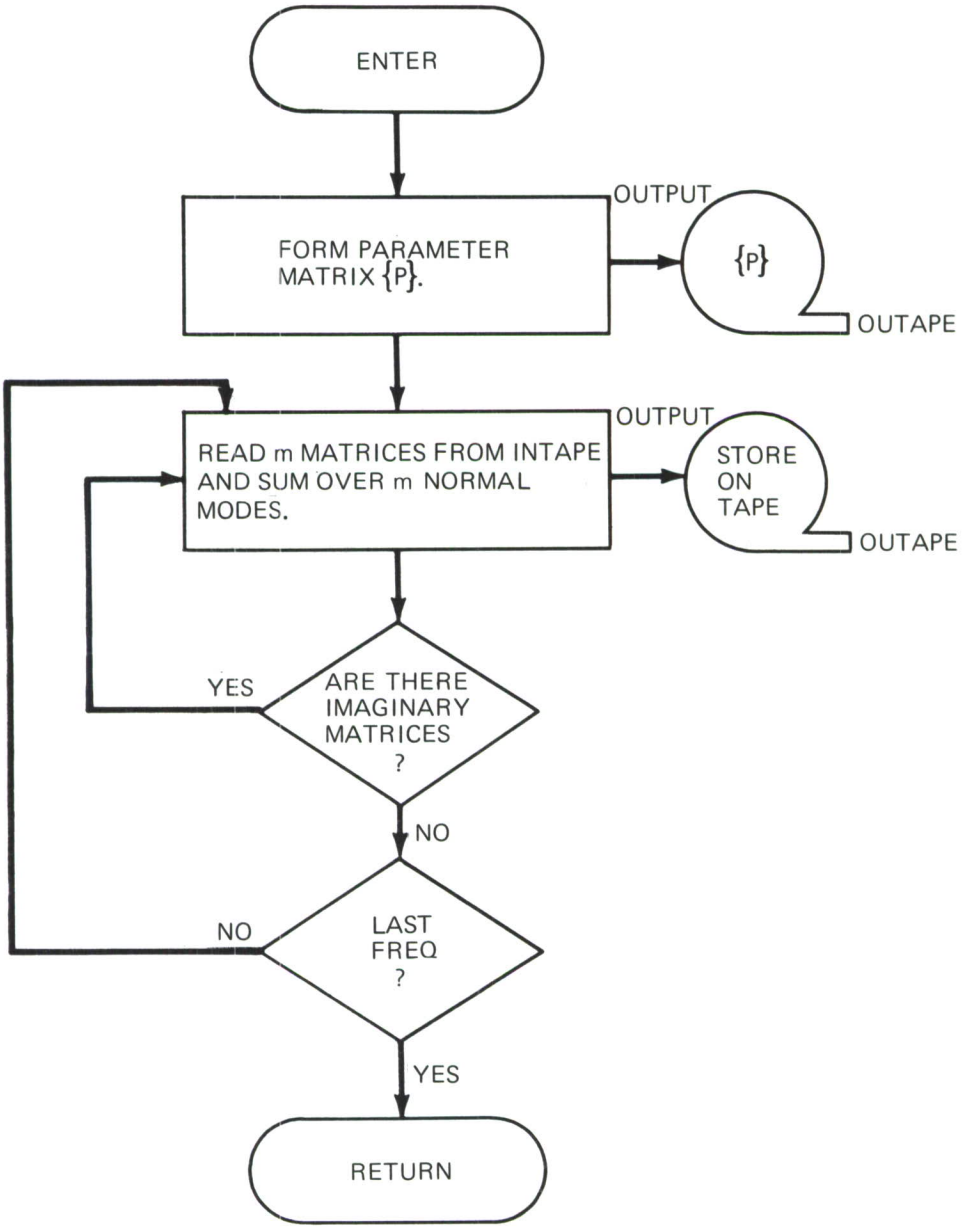


Figure 69. ADDMAT Flow Chart

(d) Subroutine DSECM3 (figure 70)

Method: The deflection covariance matrices that are calculated in subroutine DJNT3 are multiplied by the square of the frequency for each mode and summed to form the deflection second spectral moments.

$$\left[M_{\delta}{}^{2}\right] = \sum_{i=1}^{m} \left[\overline{\delta_q \, \delta_r}\right]_i \omega_i{}^{2}$$

Input: Deflection covariance matrices are from tape.

Output: Deflection second-spectral-moment matrix and a parameter matrix

Error: Standard READTP/WRTETP error messages

Subroutine required: READTP/WRTETP

Argument list: None

Length: 40175_8

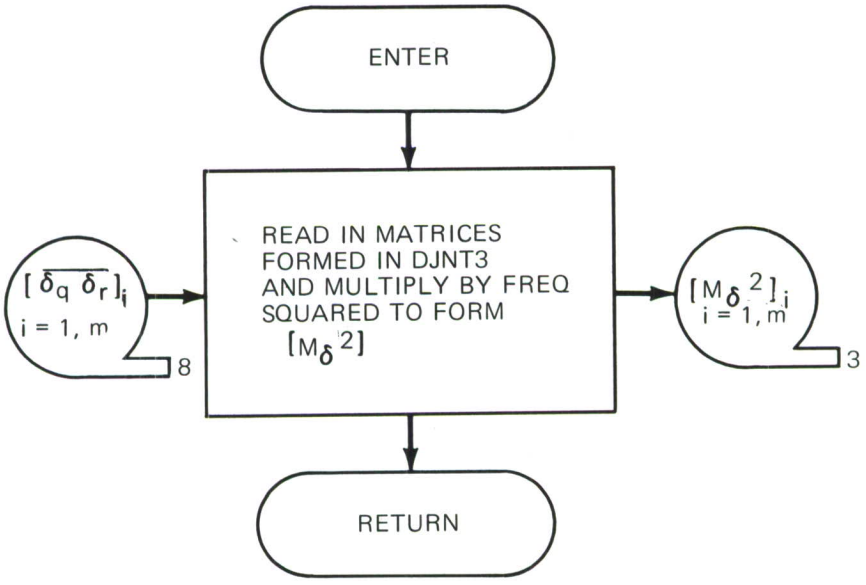


Figure 70. DSECM3 Flow Chart

(e) Subroutine ADMIT3 (figure 71)

Method: This subroutine reads the mode shapes from the phase I output tape and re-stores them on another tape. The admittance scalars are calculated and stored on tape.

$$D_i = \frac{\omega_i^2 - \omega^2}{M_i \left[\left(\omega_i^2 - \omega^2 \right)^2 + \omega^2 (\mu_i)^2 \right]}$$

$$E_i = \frac{\omega \mu_i}{M_i \left[\left(\omega_i^2 - \omega^2 \right)^2 + \omega^2 (\mu_i)^2 \right]}$$

where

$$\mu_i = \mu + \lambda \omega_i^2 + \frac{g \omega_i^2}{\omega}$$

The admittance scalars are then squared and summed to form $D_i^2 + E_i^2$ (i = 1, m) .

Input: From phase I output tape

Output: Mode shapes and the admittance scalars for each mode are stored on tape.

Error: Standard READTP/WRTETP error messages

Subroutines required: READTP/WRTETP

Argument list: None

Length: 6013_8

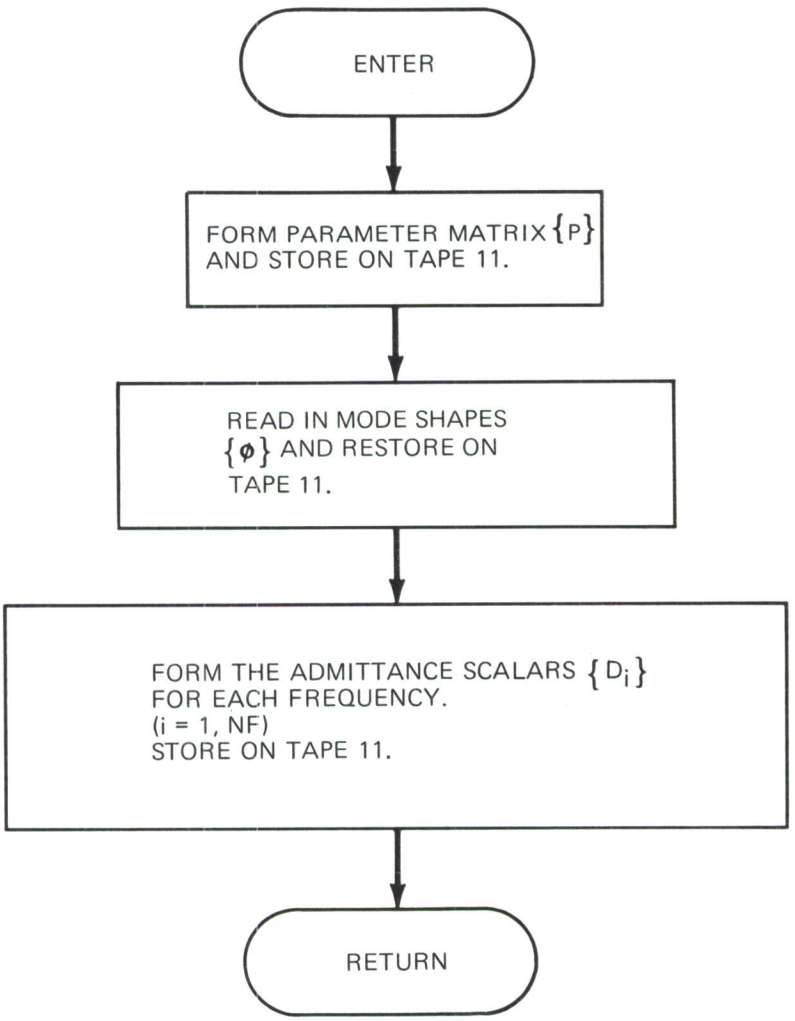


Figure 71. ADMIT3 Flow Chart

(f) TL01 Subroutine DJNT3 (figure 72)

The deflection covariance matrices are calculated.

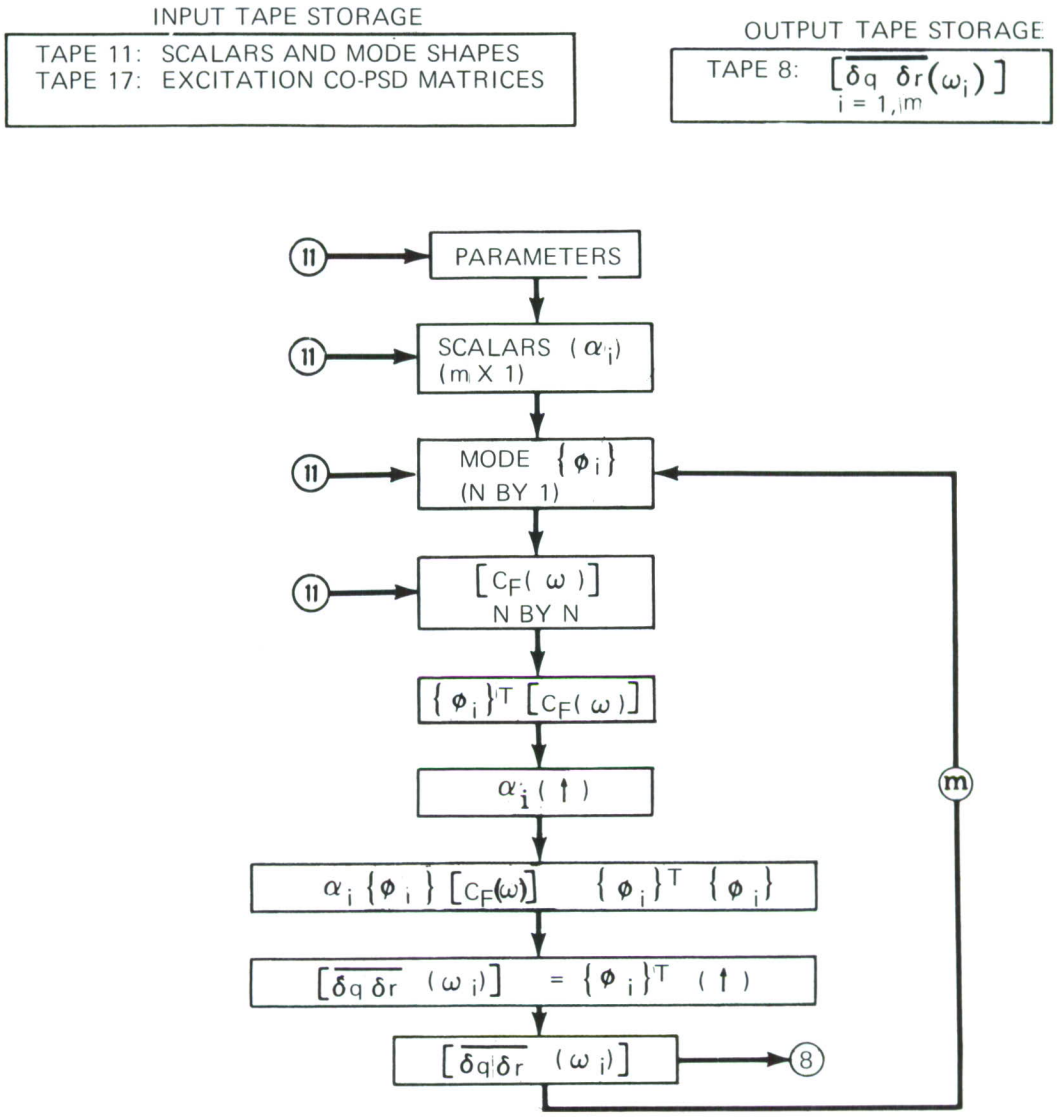


Figure 72. DJNT3 Flow Chart

(g) TL01 Subroutine SJNT3 (figure 73)

This subroutine calculates the stress covariance matrix.

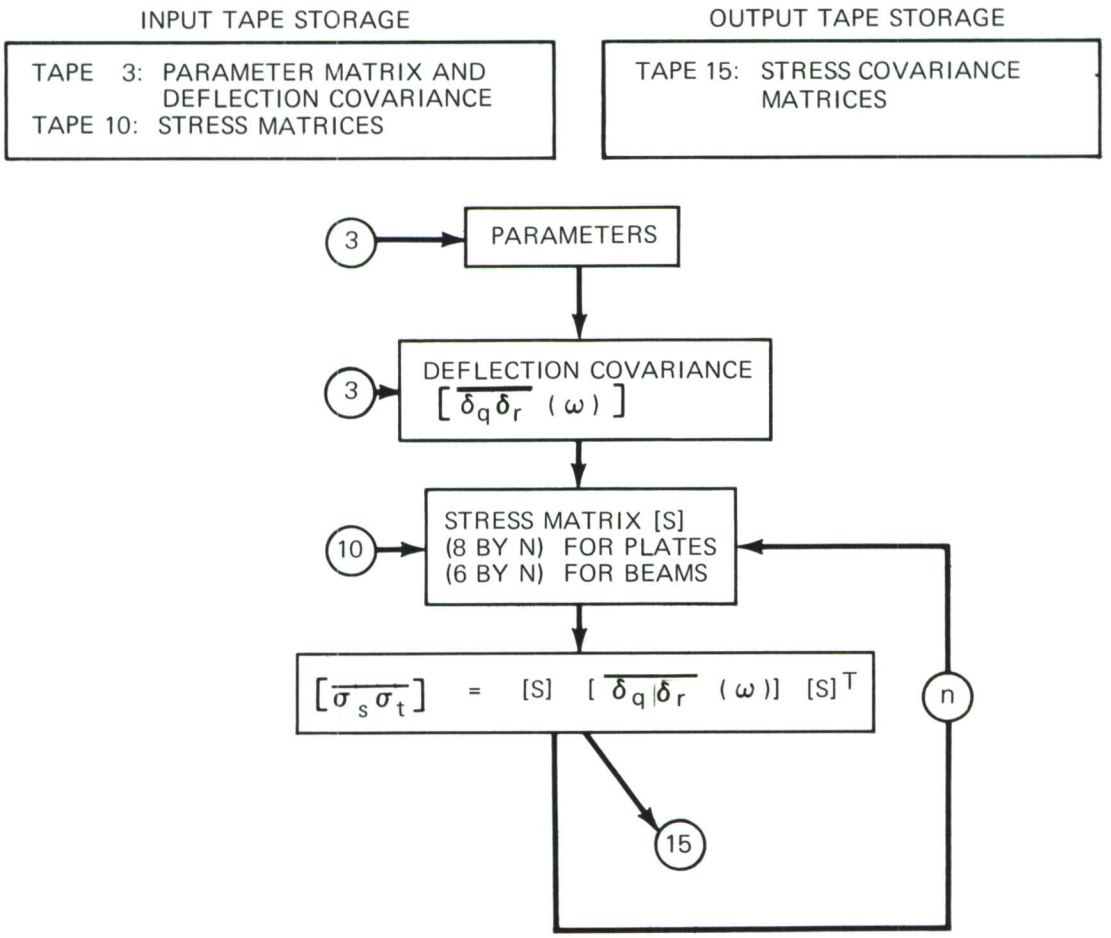


Figure 73. SJNT3 Flow Chart

(h) TL01 Subroutine DSJNT3 (figure 74)

This subroutine calculates the stress second-spectral-moment matrix.

INPUT TAPE STORAGE

TAPE 3:	PARAMETER MATRIX AND DEFLECTION SECOND-SPECTRAL-MOMENT MATRICES
TAPE 10:	STRESS MATRICES

OUTPUT TAPE STORAGE

TAPE 15:	STRESS SECOND SPECTRAL MOMENT MATRICES

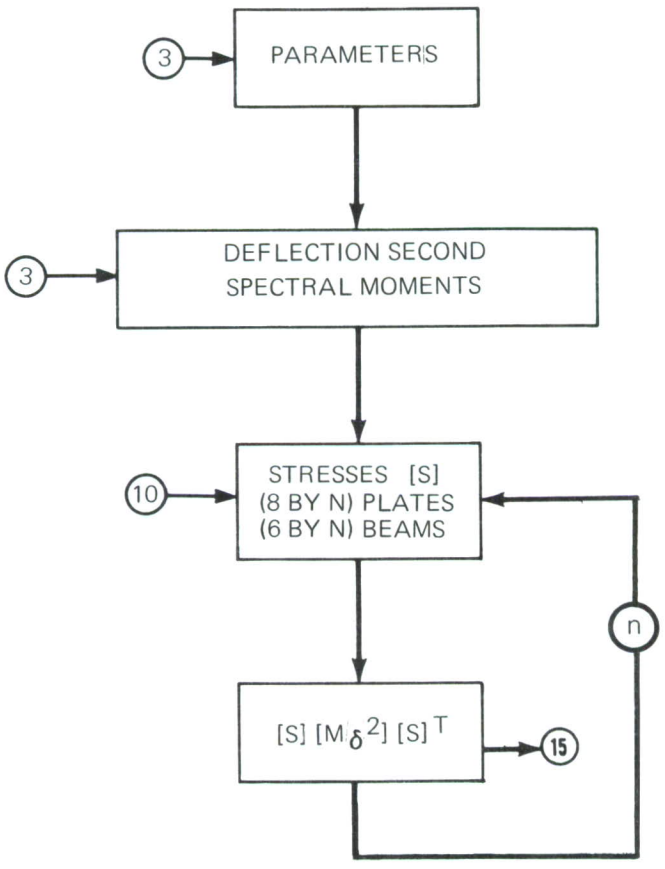


Figure 74. DSJNT3 Flow Chart

(i) TL01 Subroutine CPSD3 (figure 75)

This subroutine calculates the deflection cross-PSD matrices at NF frequencies.

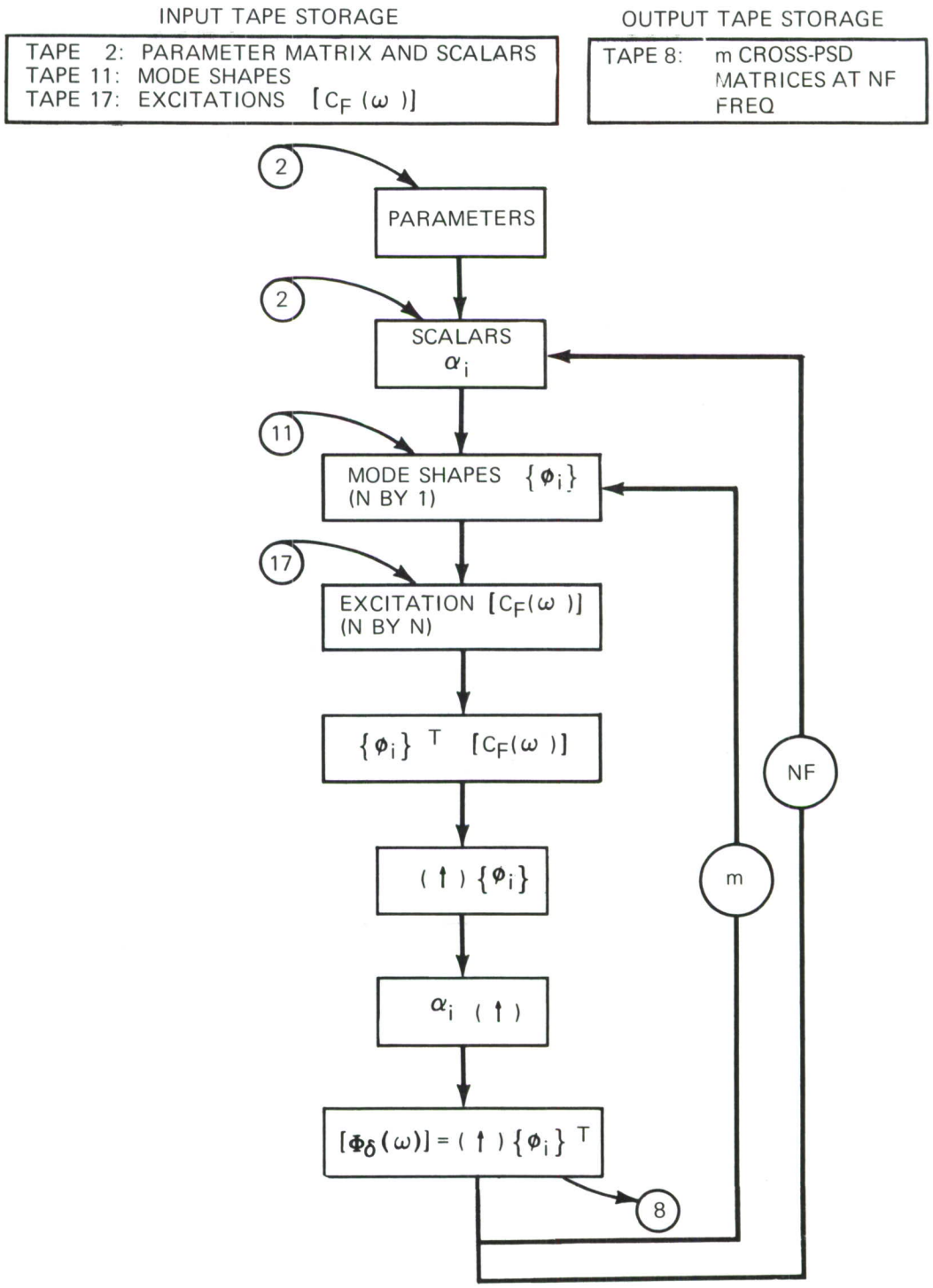


Figure 75. CPSD3 Flow Chart

(j) TL01 Subroutine SRESP3 (figure 76)

The stress cross-PSD matrices are calculated at NF frequencies.

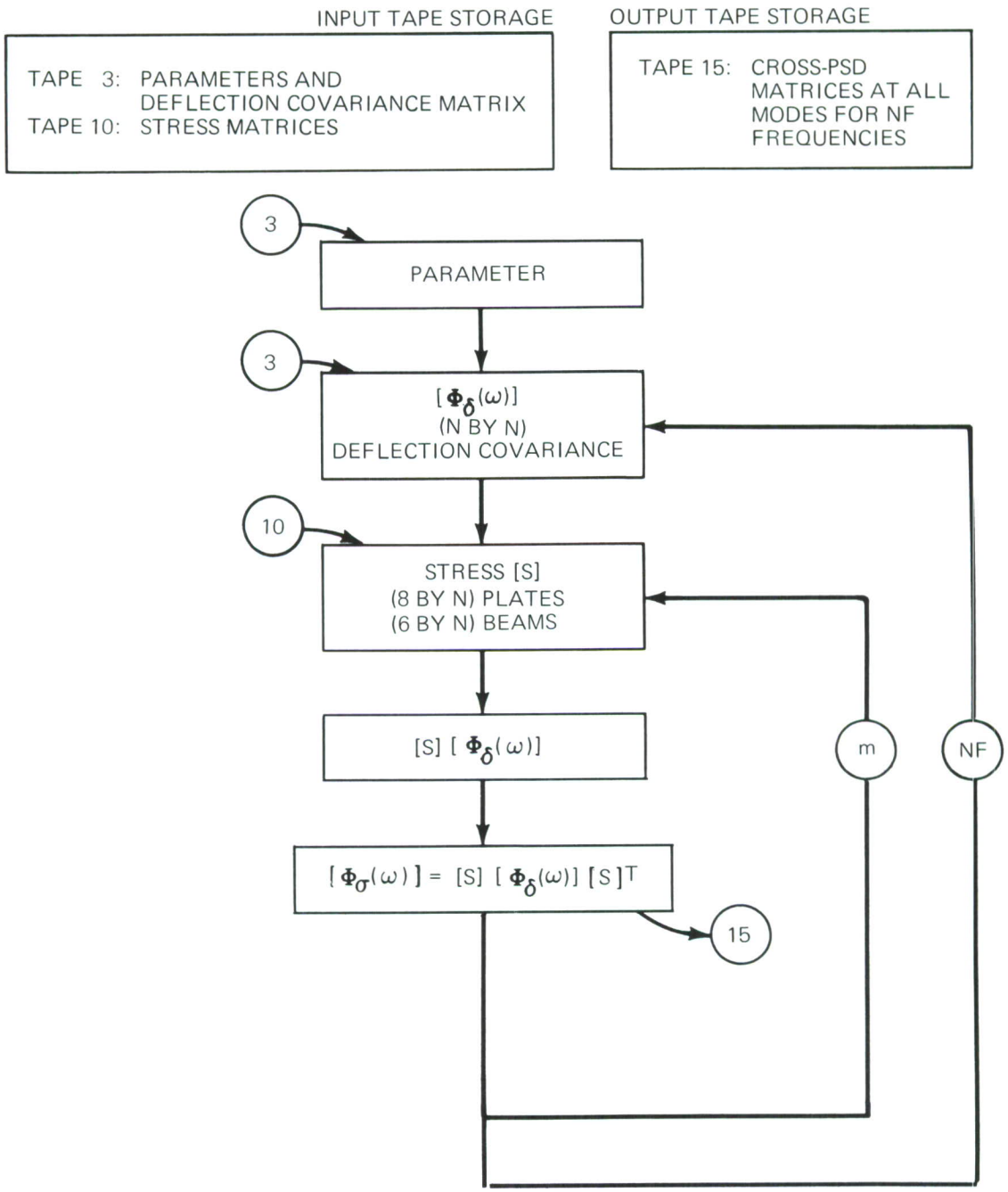


Figure 76. SRESP3 Flow Chart

APPENDIX I

TL01 DESCRIPTION AND LISTING

Matrix interpretive scheme TL01 is written in MAP assembly language. The program performs algebraic and manipulative operations on matrices.

The TL01 program instructions are executed from data inputs consisting of one data card for each TL01 instruction. Standard matrix storage is row-wise sequential as opposed to normal FORTRAN order.

The required subroutines are as follows:

Subroutine	Function
KRD	Reads all card inputs
FSR	Controls forward tape record spacing
FSF	Controls forward tape file spacing
BSF	Controls backward tape file spacing
INV4DS	Calculates the inverse of a matrix
DATASB	Establishes data storage

The flow charts for the TL01 program are in figure 77.

This appendix contains the following listings:

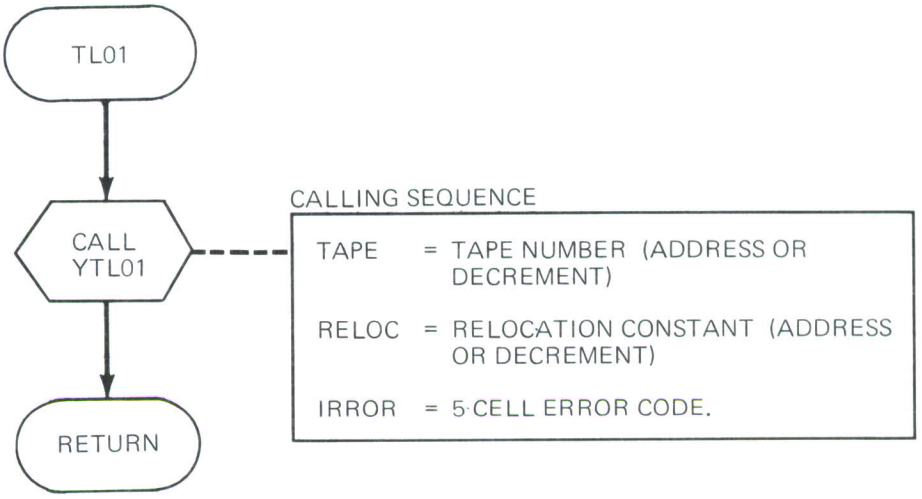


Figure 77. TL01 Flow Chart

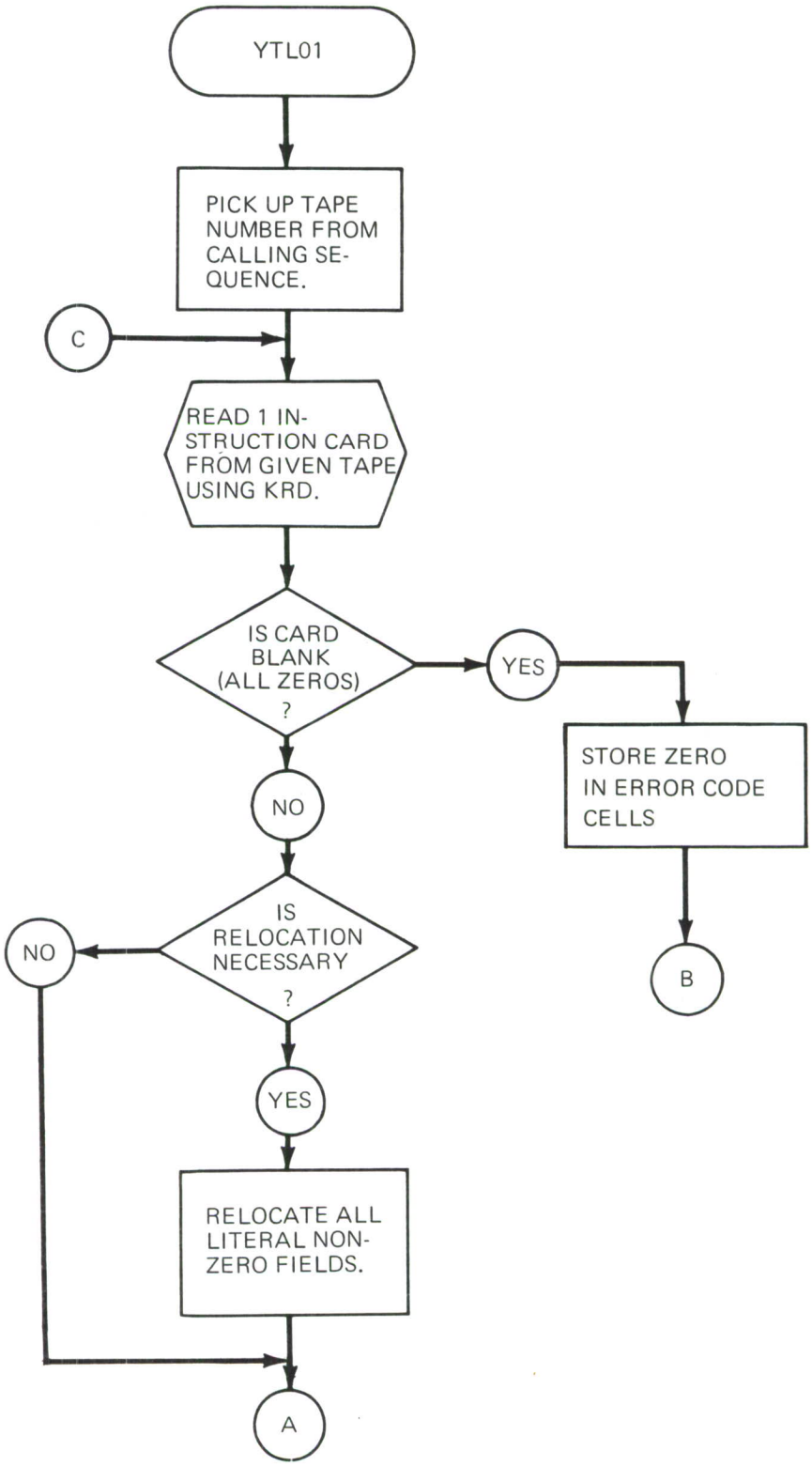


Figure 77—Continued

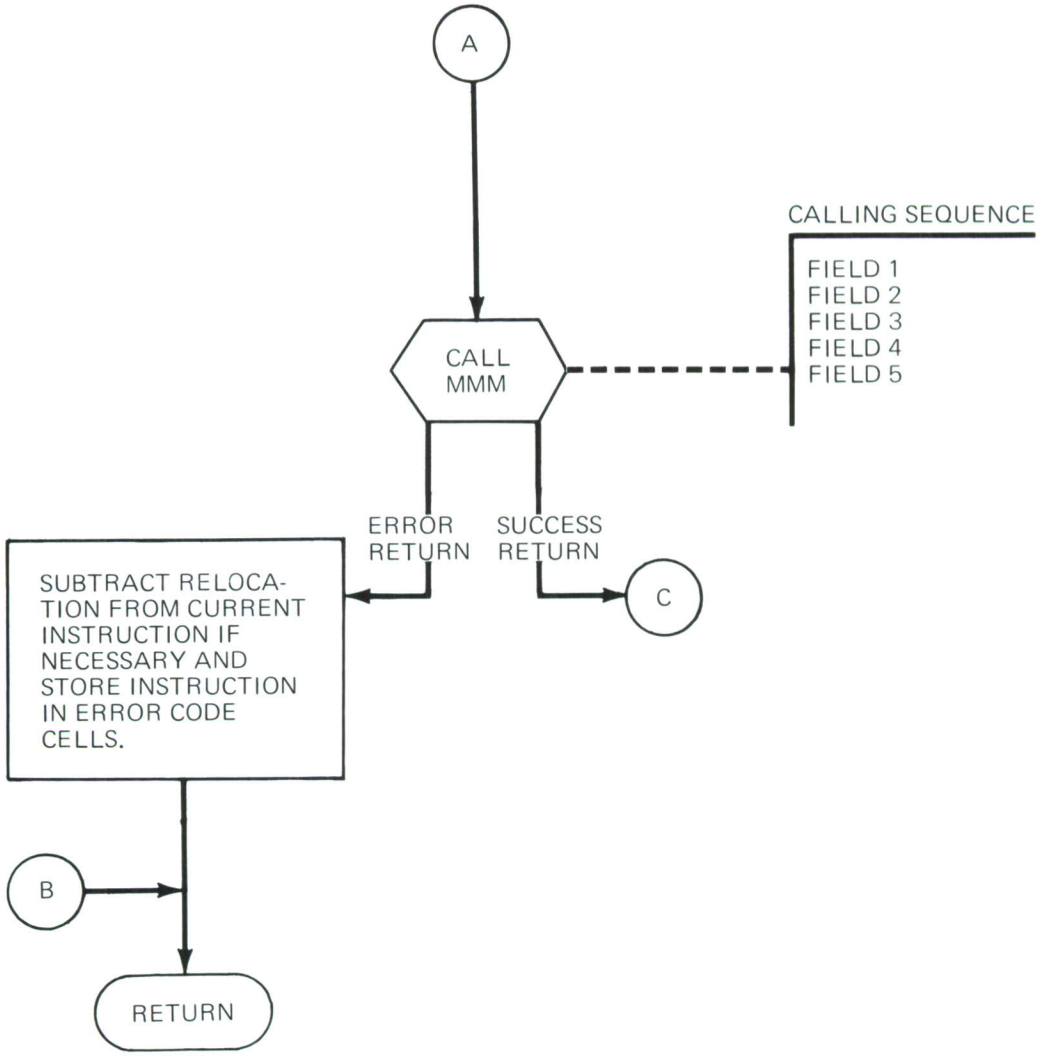


Figure 77—Continued

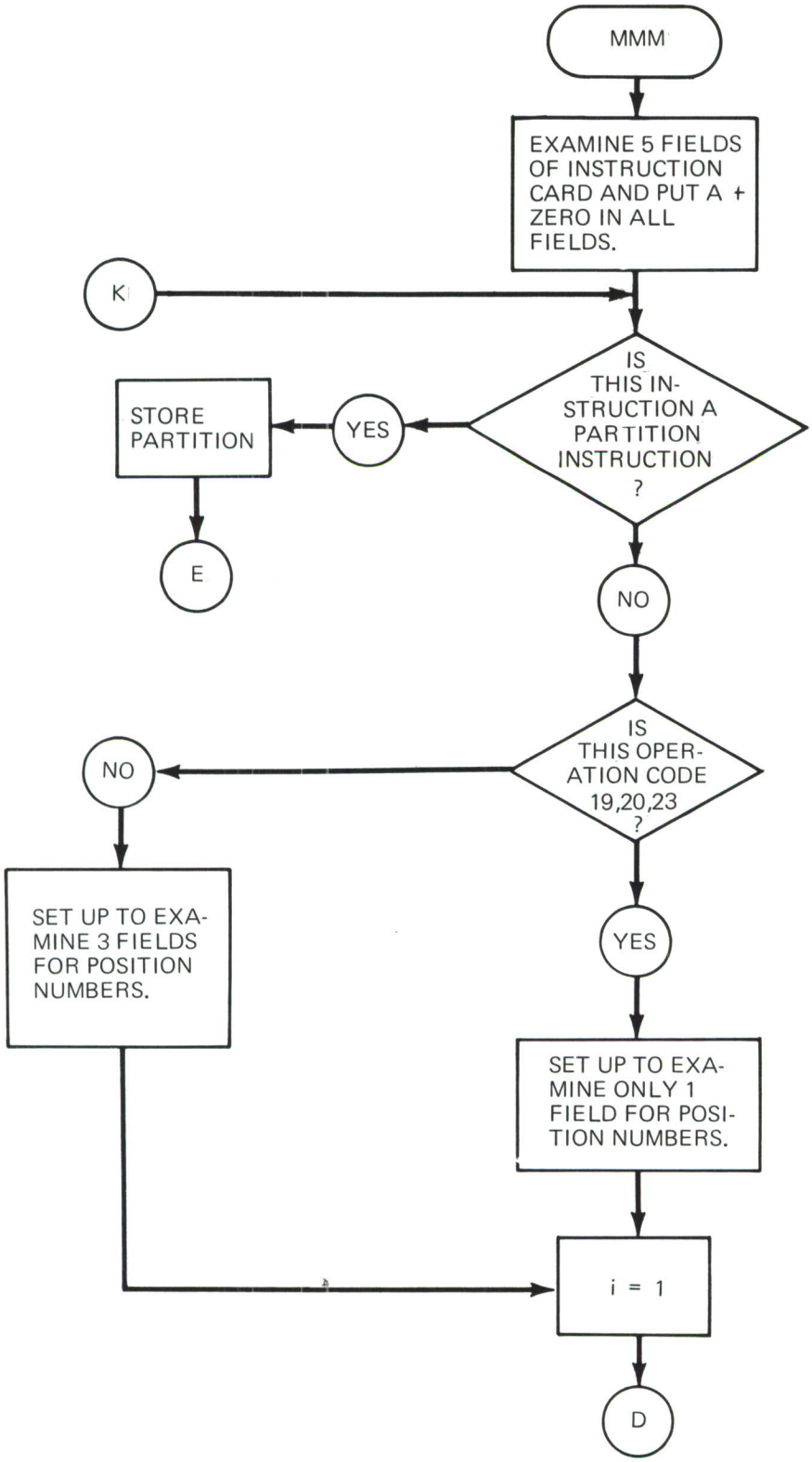


Figure 77—Continued

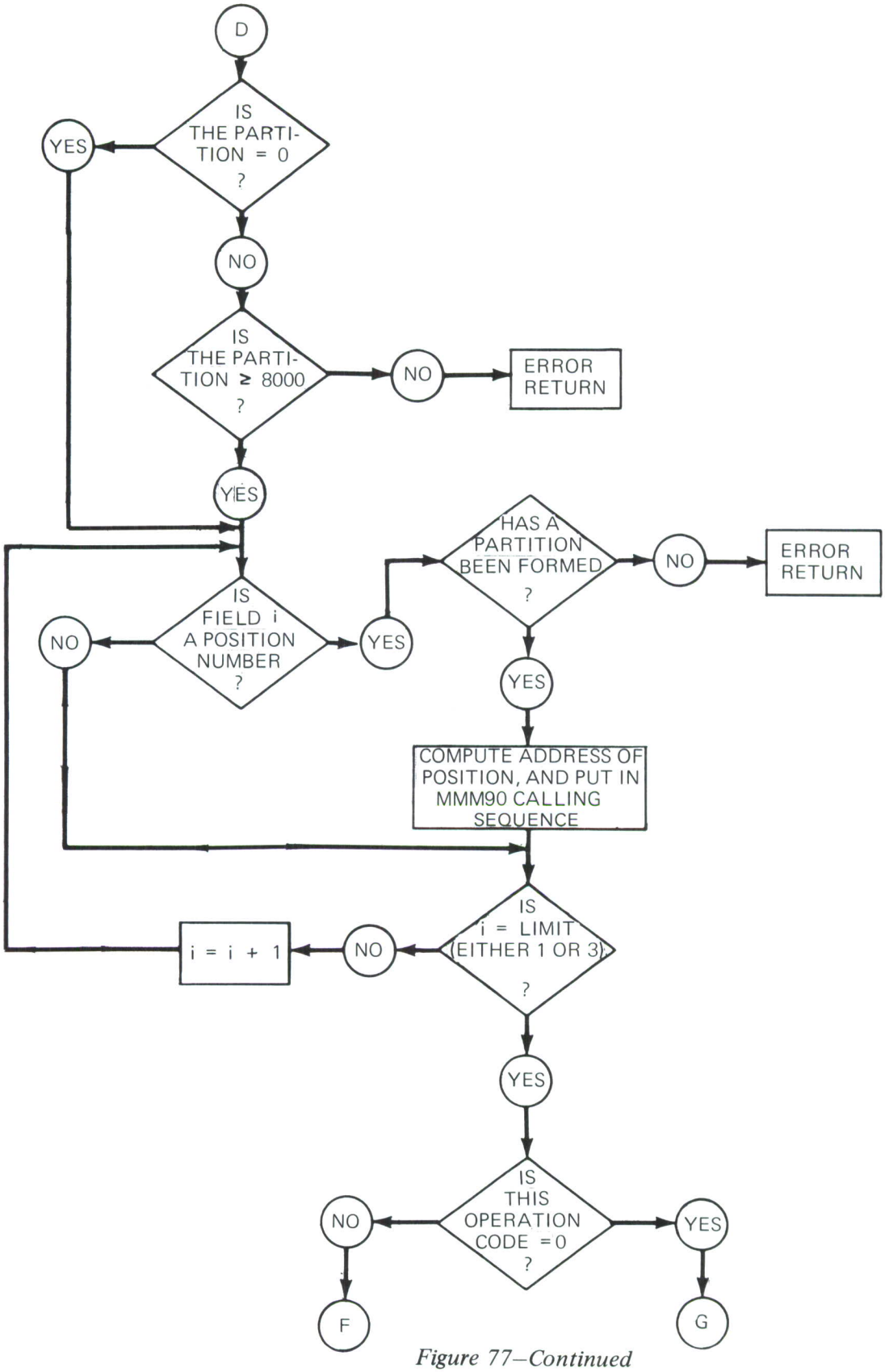


Figure 77—Continued

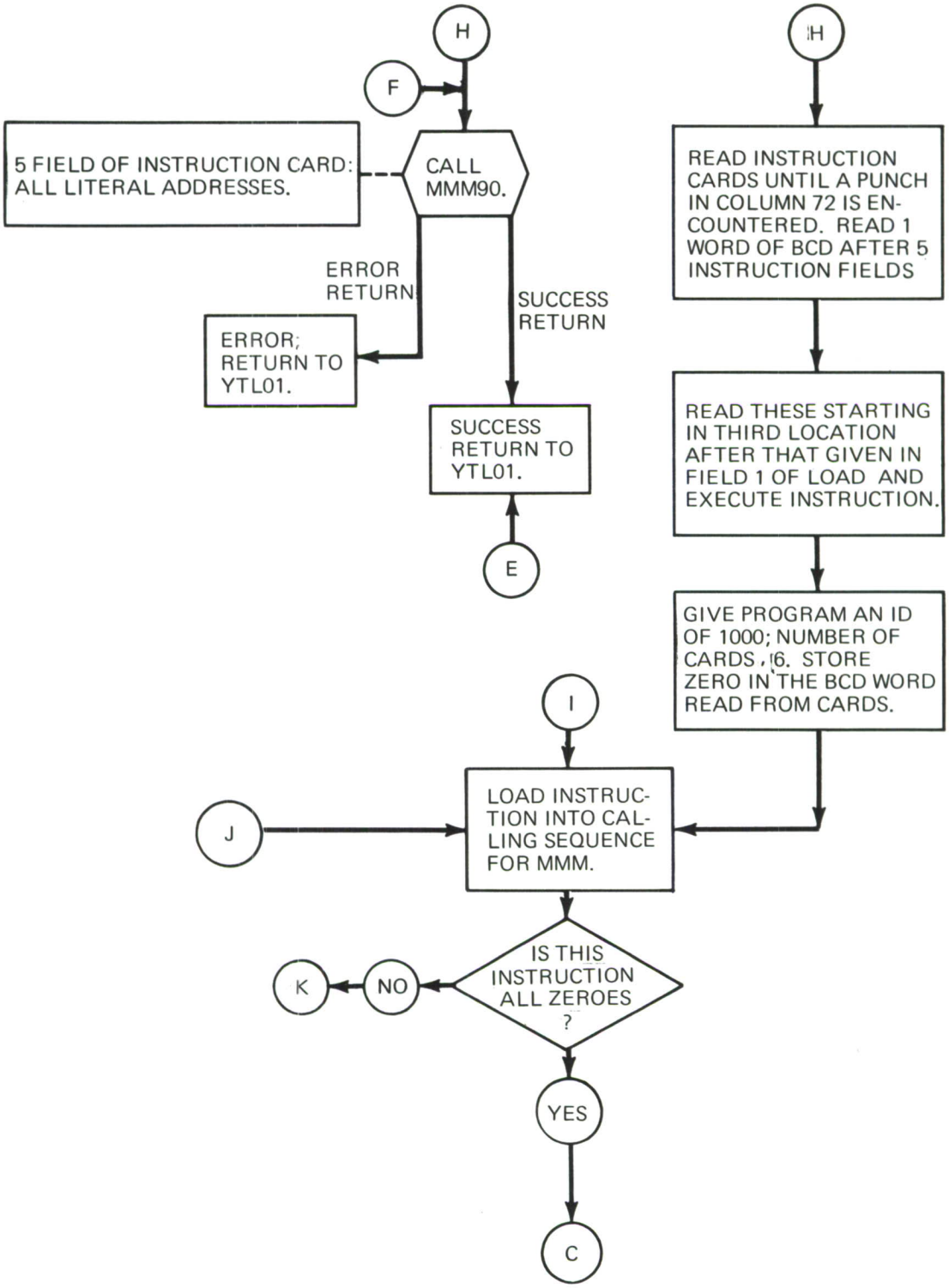


Figure 77—Continued

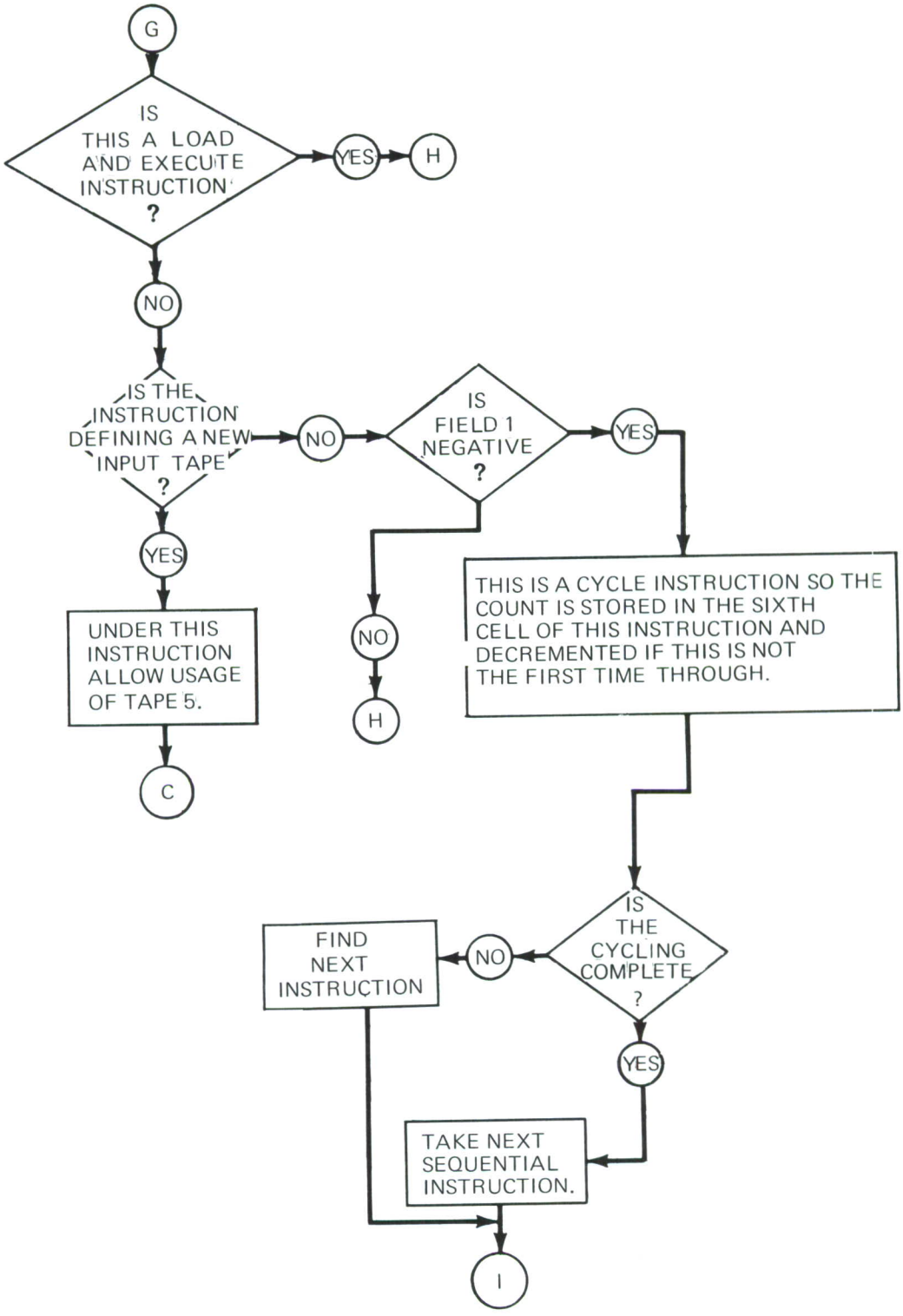


Figure 77—Continued

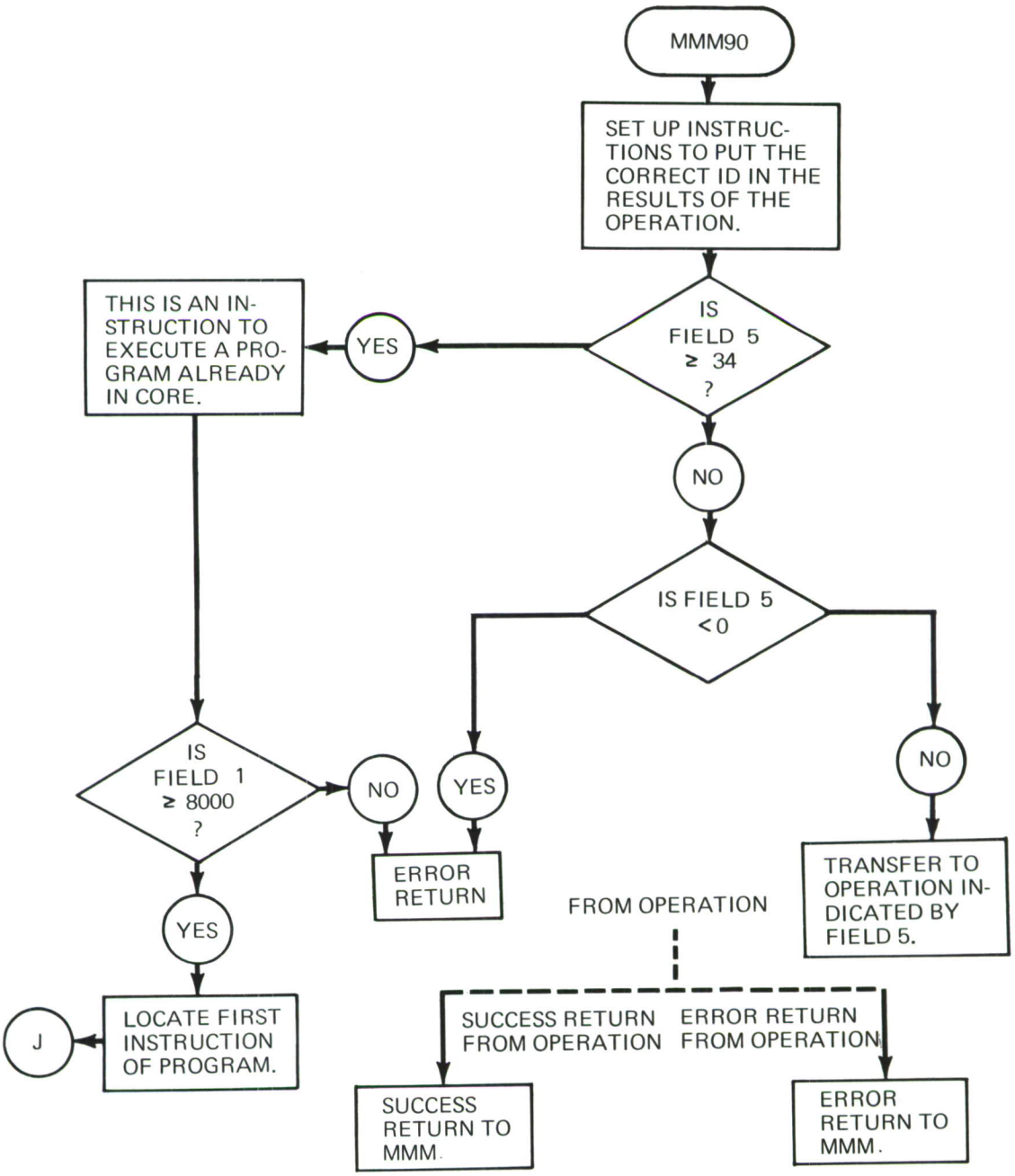


Figure 77—Concluded

TLO1 LISTING

```
$IBMAP HELP     DECK
*               SUBROUTINE TO ACT AS TLO1MN
       ENTRY    TLO1
 TLO1  SXA      SAVE4,4
       CLA      3,4
       STA      SUBR+3
       CLA      ADRSA
       SUB      RELOC1
       ADD*     4,4
       STO      RELOC
       CLA      5,4
       STA      SUBR+5
 SUBR  CALL     YTLO1(0,RELOC,0)
 SAVE4 AXT      0,4
       TRA      6,4
 ADRSA PZE      DATA
RELOC1 PZE      8000
RELOC  PZE      0
       END
```

```
$IBMAP YTLOSB  3200,DECK,M94/2                                          YTL00000
*YTL01         ****  YTL01 7094 IBSYS ASSEMBLY,  AUGUST, 1967  ****    YTL00010
*                                                                       -00020
*                                                                       -00030
*                                                                      YTL00040
*                                                                      YTL00050
*                                                                      YTL00060
*                                                                      YTL00070
*                                                                      YTL00080
*              SUBROUTINES MMM AND YTL01                               YTL00090
*                                                                      YTL00100
       ENTRY   YTL01                                                  YTL00110
*              CALL YTL01(NTAPE,NSHIFT,IRROR)                          YTL00120
*              DIMENSION IRROR(5)                                      YTL00130
*              CALLING SEQUENCE IS STRICTLY                            YTL00140
*              TO FORTRAN CONVENTIONS                                  YTL00150
*                                                                      YTL00160
       ENTRY   YTL01                                                  YTL00170
*              DUMMY ENTRY TO ELIMINATE POSSIBLE ERRORS                YTL00180
*                                                                      YTL00190
       ENTRY   MMM                                                    YTL00200
*              CALL MMM(XLOCA,XLOCB,XLOCC,NAMEC,NCODE)                 YTL00210
*                                                                      YTL00220
*                                                                      YTL00230
*              THESE ENTRY POINTS ARE FOR DUMP PURPOSES ONLY           YTL00240
*                                                                      YTL00250
*                                                                      YTL00260
       ENTRY   ERRCDE         ERROR CODE                              YTL00270
       ENTRY   TAPE           TAPE INFORMATION                        YTL00280
*                                                                      YTL00290
*                                                                      YTL00300
 YTL01 TTR     YTL01                                                   YTL00310
*                                                                      YTL00320
 YTL01 LMTM                                                           YTL00330
       EFTM                                                           YTL00340
       SXD     XR1,1                                                  YTL00350
       SXD     XR2,2                                                  YTL00360
       SXD     XR4,4                                                  YTL00370
       CAL     8                                                      YTL00380
       SLW     TMPFPT         SAVE CELL 8 FOR RESTORE                 YTL00390
       CAL     FPTTRA         SET UP FOR FL. PT. TRAP                 YTL00400
       SLW     8                                                      YTL00410
       STZ     CELL0          CLEAR CELL FOR STORING CELL 0           YTL00420
       STZ     ERRCDE         ZERO ERROR CODE                         YTL00430
       CAL     5,4                                                    YTL00440
       ACL     =5                                                     YTL00450
       STA     INS                                                    YTL00460
       CLA*    3,4            TAPE UNIT                               YTL00470
       SLW     TAP                                                    YTL00480
       TZE*    *+2            TAP=0                                   YTL00490
       CAS     =19                                                    YTL00500
       TRA     TOBIG          TAP GREATER THAN 22                     YTL00510
       TRA     *+1                                                    YTL00520
       SUB     =6                                                     YTL00530
       TNZ     YENT                                                   YTL00540
       CLA     =14            DO NOT ALLOW A READ OF TAPE 6           YTL00550
       STO     ERRCDE                                                 YTL00560
```

```
        TRA     YTER                                                    YTL00570
 TOBIG  CLA     =13             ILLEGAL TAPE NO                         YTL00580
        TRA     *-3                                                     YTL00590
*                                                                       YTL00600
  YENT  CALL    KRD(LOC,=0,IRROR,=5,=10,=0,TAP)                         YTL00610
        CLA     IRROR                                                   YTL00620
        TZE     *+4                                                     YTL00630
        ADD     =100            KRD READ CODE                           YTL00640
        STO     ERRCDE                                                  YTL00650
        TRA     YTER                                                    YTL00660
        AXT     4,1                                                     YTL00670
        CLA     LOC                                                     YTL00680
        ADD     LOC+5,1                                                 YTL00690
        TIX     *-1,1,1                                                 YTL00700
        TZE     YTOUT           FINISHED                                YTL00710
*                                                                       YTL00720
        LXD     XR4,4                                                   YTL00730
        CLA*    4,4             RELOCATION                              YTL00740
        TZE     YTEX                                                    YTL00750
        STO     YSHIFT                                                  YTL00760
*                                                                       YTL00770
*                                                                       YTL00780
  YTEX  CALL    MMM(LOC,LOC+1,LOC+2,LOC+3,LOC+4)YTER                    YTL00790
        TRA     YENT            NEXT CARD                               YTL00800
*                                                                       YTL00810
 YTOUT  AXT     5,2                                                     YTL00820
        STZ*    INS             CLEAR ERROR AREA                        YTL00830
        TIX     *-1,2,1                                                 YTL00840
 YEXIT  LXD     XR1,1                                                   YTL00850
        LXD     XR2,2                                                   YTL00860
        LXD     XR4,4                                                   YTL00870
        CAL     TMPFPT          RESTORE CELL 8                          YTL00880
        SLW     8                                                       YTL00890
        TRA     6,4             SUCCESS RETURN                          YTL00900
*                                                                       YTL00910
 YTER   NZT     YSHIFT                                                  YTL00920
        TTR     YTER3+1         TRANSFER ON NO RELOCATION               YTL00930
        AXT     3,1                                                     YTL00940
 YTER2  CLA     LOC+3,1                                                 YTL00950
        TZE     YTER3           TRANSFER ON BLANK FIELD                 YTL00960
        SSP                                                             YTL00970
        SUB     SMLTAP                                                  YTL00980
        TPL     YTER3           TRANSFER ON TAPE NUMBER                 YTL00990
        CLA     LOC+3,1                                                 YTL01000
        SSP                                                             YTL01010
        SUB     LOCORE                                                  YTL01020
        TMI     YTER3           TRANSFER ON POSITION NUMBER             YTL01030
        CAL     LOC+3,1                                                 YTL01040
        SUB     YSHIFT                                                  YTL01050
        STA     LOC+3,1         ORIGINAL ADDRESS                        YTL01060
 YTER3  TIX     YTER2,1,1                                               YTL01070
        CALL    .FWRD.(.UN06.,ERRPNT)           PRINT ERROR MESSAGE     YTL01080
        CLA     ERRCDE                                                  YTL01090
        TSX     .FCNV.,4                                                YTL01100
        CLA     INSCNT                                                  YTL01110
        TSX     .FCNV.,4                                                YTL01120
        CALL    .FFIL.                                                  YTL01130
        CAL     KEY                                                     YTL01140
        TZE     NOTPRG                                                  YTL01150
        SUB     YSHIFT                                                  YTL01160
```

```
        STO     YMAA                                                    YTL01170
        CLA     INST            GET PHYSICAL INSTRUCTION                YTL01180
        SUB     KEY             IF ERROR DURING CORE PROG.              YTL01190
        XCA                                                             YTL01200
        ZAC                                                             YTL01210
        DVP     =6                                                      YTL01220
        STQ     PHYSCT          PHYSICAL COUNT                          YTL01230
        CALL    .FWRD.(.UN06.,PROGER)           FOR CORE PROGRAM ONLY   YTL01240
        CLA     PHYSCT                                                  YTL01250
        TSX     .FCNV.,4                                                YTL01260
        CALL    .FFIL.                                                  YTL01270
NOTPRG  CLA     LOC                                                     YTL01280
        ADM     LOC+1                                                   YTL01290
        ADM     LOC+2                                                   YTL01300
        ADM     LOC+3                                                   YTL01310
        ADM     LOC+4                                                   YTL01320
        TNZ     LDERR                                                   YTL01330
        CLA     MILYN           PUT 1,000,000. IN FIELD 5 WHEN ERROR    YTL01340
        STO     LOC+4           WAS MADE WHEN THE CURRENT INSTR. IS ZERO YTL01350
 LDERR  AXT     5,1                                                     YTL01360
        AXT     5,2                                                     YTL01370
        CAL     LOC+5,1                                                 YTL01380
        SLW*    INS             LOAD CARD IN ERROR                      YTL01390
        TIX     *+1,1,1                                                 YTL01400
        TIX     *-3,2,1                                                 YTL01410
        NZT     KEY                                                     YTL01420
        TRA     YEXIT                                                   YTL01430
        STZ     ERRCDE                                                  YTL01440
        STZ     KEY                                                     YTL01450
PNTPRG  CALL    MMM(YMAA,=1,=0,=0,=20)PNTER                             YTL01460
        TRA     YEXIT                                                   YTL01470
PNTER   CALL    .FWRD.(.UN06.,CORERR)                                   YTL01480
        CLA     ERRCDE                                                  YTL01490
        TSX     .FCNV.,4                                                YTL01500
        CALL    .FFIL.                                                  YTL01510
        TRA     YEXIT                                                   YTL01520
*                                                                       YTL01530
        REM MMM   GENERAL MATRIX ROUTINE                                YTL01540
        REM CALLING SEQUENCE                                            YTL01550
*               CALL MMM(LA,LB,LC,NC,CODE)ERRRET                        YTL01560
        REM                                                             YTL01570
        REM CONTROL PACKAGE                                             YTL01580
        REM                                                             YTL01590
*                                                                       YTL01600
   MMM  LMTM                                                            YTL01610
        EFTM                                                            YTL01620
        NZT     FIRST                                                   YTL01630
        TRA     REST                                                    YTL01640
        CLA     ADRSDA                                                  YTL01650
        STA     LOCORE                                                  YTL01660
        ADD*    ADRSDA                                                  YTL01670
        SUB     =1                                                      YTL01680
        STA     HICORE                                                  YTL01690
        CLA*    ADRSDA                                                  YTL01700
        STA     MXDATA                                                  YTL01710
        CLA     HICORE                                                  YTL01720
        SUB     =63                                                     YTL01730
        STA     HIPROG                                                  YTL01740
        CLA     ADRSDA                                                  YTL01750
        ADD     =1                                                      YTL01760
```

```
        STA     *+1                                                 YTL.01770
        CLA     **          LOWEST ABSOLUTE CORE ADD. REFERENCED    YTL.01780
        STA     PROGLO                                              YTL.01790
        STZ     FIRST                                               YTL.01800
REST    SXD     IR1,1                                               YTL.01810
        SXD     IR2,2                                               YTL.01820
        SXD     IR3,3                                               YTL.01830
        SXD     IR4,4                                               YTL.01840
        SXD     IR5,5                                               YTL.01850
        SXD     IR6,6                                               YTL.01860
        SXD     IR7,7                                               YTL.01870
        STZ     MCROCT      COUNT FOR STORING RETURNS               YTL.01880
        STZ     KEY                                                 YTL.01890
        CLA*    3,4         AA                                      YTL.01900
        TNZ     *+2                                                 YTL.01910
        SSP                                                         YTL.01920
        STO     LOC                                                 YTL.01930
        STO     YMAA                                                YTL.01940
        CLA*    4,4                                                 YTL.01950
        TNZ     *+2                                                 YTL.01960
        SSP                                                         YTL.01970
        STO     LOC+1                                               YTL.01980
        STO     YMBB                                                YTL.01990
        CLA*    5,4         CC                                      YTL.02000
        TNZ     *+2                                                 YTL.02010
        SSP                                                         YTL.02020
        STO     LOC+2                                               YTL.02030
        STO     YMCC                                                YTL.02040
        CLA*    6,4         NAME OF C                               YTL.02050
        TNZ     *+2                                                 YTL.02060
        SSP                                                         YTL.02070
        STO     LOC+3                                               YTL.02080
        STO     YMNC                                                YTL.02090
        CLA*    7,4                                                 YTL.02100
        TNZ     *+2                                                 YTL.02110
        SSP                                                         YTL.02120
        STO     LOC+4                                               YTL.02130
        STO     YMOP                                                YTL.02140
*                                                                   YTL.02150
YMIN    AXT     3,1                                                 YTL.02160
YTA     NZT     LOC+3,1                                             YTL.02170
        TTR     YTB         ZERO FIELD                              YTL.02180
        CLA     LOC+3,1                                             YTL.02190
        SSP                                                         YTL.02200
        SUB     SMLTAP                                              YTL.02210
        TPL     YTB                                                 YTL.02220
        CLA     LOC+3,1                                             YTL.02230
        SSP                                                         YTL.02240
        SUB     PROGLO                                              YTL.02250
        TMI     YTB                                                 YTL.02260
        CAL     LOC+3,1                                             YTL.02270
        ADD     YSHIFT      RELOCATE CORE ADDRESSES                 YTL.02280
        STA     LOC+3,1                                             YTL.02290
        STA     YMAA+3,1                                            YTL.02300
YTB     TIX     YTA,1,1                                             YTL.02310
        CLA     YMOP                                                YTL.02320
        TNZ     YMA         NOT OPERATION ZERO                      YTL.02330
        CLA     YMAA                                                YTL.02340
        TNZ     YMA                                                 YTL.02350
        CLA     YMBB                                                YTL.02360
```

```
        TMI     YMA                                                     YTL02370
        SUB     SMLTAP                                                  YTL02380
        TPL     YMA                                                     YTL02390
        CLA     YMBB                                                    YTL02400
        TZE     YMEXEC                                                  YTL02410
        STO     XMODE           FIELD 2 IS A PARTITION                  YTL02420
        STZ     PARTON          ZERO PARTON WHEN A PARTITION IS FORMED  YTL02430
        TSX     CHKRNG,4        CHECK TO SEE IF NO. IS                  YTL02440
        TRA     *+2             BETWEEN 8000 AND 32563                  YTL02450
        TRA     WHERE           OK ON PARTITION ADDRESS                 YTL02460
        CLA     =21                                                     YTL02470
        TRA     YMERR           TOO LOW OK TOO HI PARTITION             YTL02480
*                                                                       YTL02490
YMA     AXT     1,1                                                     YTL02500
        SXA     TEMP-1,1        3 FIELDS                                YTL02510
        CLA     YMOP                                                    YTL02520
        SUB     =19                                                     YTL02530
        TZE     YYY             CODE 19                                 YTL02540
        CLA     YMOP                                                    YTL02550
        SUB     =20                                                     YTL02560
        TZE     YYY             CODE 20                                 YTL02570
        CLA     YMOP                                                    YTL02580
        SUB     =23                                                     YTL02590
        TZE     YYY             CODE 23                                 YTL02600
        AXT     3,1                                                     YTL02610
        SXA     TEMP-1,1        1 FIELD FOR READ                        YTL02620
*                                                                       YTL02630
YYY     LXA     TEMP-1,1                                                YTL02640
        AXT     0,2                                                     YTL02650
        CLA     XMODE                                                   YTL02660
        STO     TEMP            PARTITION                               YTL02670
YMB     CLA     YMAA,2          PARAMETER                               YTL02680
        TMI     YMC                                                     YTL02690
        TZE     YMC                                                     YTL02700
        SUB     TOPPOS                                                  YTL02710
        TPL     YMC             NOT A POS. NO.                          YTL02720
        CLA     PARTON                                                  YTL02730
        TZE     *+3                                                     YTL02740
        CLA     =22             POSITION REFERENCED WITHOUT PARTITION   YTL02750
        TRA     YMERR                                                   YTL02760
        CLA     YMAA,2                                                  YTL02770
        SUB     =1                                                      YTL02780
        TZE     YMD             POSITION NO. IS ONE                     YTL02790
        PAX     0,4                                                     YTL02800
YME     CAL     TEMP                                                    YTL02810
        ADD     =1                                                      YTL02820
        SLW     TEMP-1          ADDRESS OF M                            YTL02830
        ADD     =1                                                      YTL02840
        SLW     TEMP-2          ADDRESS OF N                            YTL02850
        LDQ*    TEMP-1                                                  YTL02860
        MPY*    TEMP-2                                                  YTL02870
        TNZ     WAYTOB                                                  YTL02880
        STQ     TEMP-1                                                  YTL02890
        XCA                                                             YTL02900
        TZE     GOODDM                                                  YTL02910
        TMI     *+5                                                     YTL02920
        CAS     MXDATA                                                  YTL02930
        TRA     *+3                                                     YTL02940
        TRA     GOODDM                                                  YTL02950
        TRA     GOODDM                                                  YTL02960
```

```
WAYTOB STZ     JEMP-1                                                  YTL02970
       LDQ     YMAA,2                                                  YTL02980
       MPY     THOUSN                                                  YTL02990
       STQ     TEMP                                                    YTL03000
       CLA     YMAA,2                                                  YTL03010
       SXA     TEMP-1,4                                                YTL03020
       SUB     TEMP-1                                                  YTL03030
       ADD     TEMP                                                    YTL03040
       ADD     =2000000                                                YTL03050
       TRA     YMERR                                                   YTL03060
GOODDM CLA     TEMP-1                                                  YTL03070
       ACL     =3                                                      YTL03080
       ACL     TEMP                                                    YTL03090
       SLW     TEMP            NEXT MATRIX                             YTL03100
       TIX     YME,4,1                                                 YTL03110
 YMD   CLA     TEMP                                                    YTL03120
       STO     YMAA,2          INSERT ADDRESS                          YTL03130
 YMC   TXI     *+1,2,-1                                                YTL03140
       CLA     XMODE                                                   YTL03150
       TIX     YMB-1,1,1                                               YTL03160
*                                                                      YTL03170
*              SEPARATION OF ZERO OP. CODES                            YTL03180
*                                                                      YTL03190
 YSEP  CLA     YMOP                                                    YTL03200
       TNZ     YMEXEC          NOT OP. CODE 0                          YTL03210
 SEPA  CLA     YMBB                                                    YTL03220
       SSP                                                             YTL03230
       ADD     YMCC                                                    YTL03240
       ADM     YMNC                                                    YTL03250
       TNZ     SEPC                                                    YTL03260
       CLA     YMAA                                                    YTL03270
       SUB     SMLTAP                                                  YTL03280
       TPL     SEPB            TRANSFER OF CONTROL                     YTL03290
       TTR     MLOAD           LOAD PROGRAM                            YTL03300
 SEPB  CLA     YMAA                                                    YTL03310
       TSX     TP,4                                                    YTL03320
       STO     TAP                                                     YTL03330
       XCA                                                             YTL03340
       TZE     *+3                                                     YTL03350
       CLA     =13                                                     YTL03360
       TRA     YMERR                                                   YTL03370
       XCA                                                             YTL03380
       CAS     =5              ALLOW 5                                 YTL03390
       TRA     *+2                                                     YTL03400
       TRA     WHERE                                                   YTL03410
       STO     TAPE                                                    YTL03420
       TSX     TPCK,4                                                  YTL03430
       TRA     WHERE                                                   YTL03440
 SEPC  CLA     YMBB                                                    YTL03450
       SSP                                                             YTL03460
       ADD     YMCC                                                    YTL03470
       TNZ     YMEXEC                                                  YTL03480
       CLA     YMAA                                                    YTL03490
       TMI     CYCLE+2                                                 YTL03500
       CAS     LOCORE                                                  YTL03510
       TRA     MLOAD                                                   YTL03520
       TRA     MLOAD                                                   YTL03530
       TRA     CYCLE                                                   YTL03540
*                                                                      YTL03550
 MLOAD CLA     YMAA                                                    YTL03560
```

```
         CAS     LOCORE          PROGRAM MUST LOAD BETWEEN          YTL03570
         TRA     *+4             ADRSDA AND 32500,INCLUSIVE         YTL03580
         TRA     *+3                                                YTL03590
         CLA     =23                                                YTL03600
         TRA     YMERR                                              YTL03610
         CAS     HIPROG                                             YTL03620
         TRA     *-3                                                YTL03630
         TRA     *+1                                                YTL03640
         ADD     =3                                                 YTL03650
         STA     SEPD+3                                             YTL03660
         CLA     TAP                                                YTL03670
         STA     SEPE            CURRENT TAPE                       YTL03680
SEPD     CALL    KRD(**,=1,IRROR,=5,=10,=1,SEPE)                    YTL03690
         STO     TEMP            CARD COUNT                         YTL03700
         CLA     IRROR           KRD ERROR CODE                     YTL03710
         TZE     CONT1                                              YTL03720
         XCA                                                        YTL03730
         MPY     THOUSN                                             YTL03740
         XCA                                                        YTL03750
         ADD     TEMP                                               YTL03760
         ADD     MILYN                                              YTL03770
         TRA     YMERR                                              YTL03780
 CONT1   CLA     YMAA            STUFF ADDRESS OF DIMENSION         YTL03790
         ADD     =1                                                 YTL03800
         STA     NOINST          ROW DIMENSION                      YTL03810
         ADD     =1                                                 YTL03820
         STA     ALWYS6                                             YTL03830
         CLA     TEMP                                               YTL03840
NOINST   STO     **                                                 YTL03850
         CLA     =6                                                 YTL03860
ALWYS6   STO     **                                                 YTL03870
         CLA     YMNC                                               YTL03880
         TZE     *+3                                                YTL03890
         STO*    YMAA                                               YTL03900
         TRA     SEPF                                               YTL03910
         CLA     THOUSN                                             YTL03920
         STO*    YMAA                                               YTL03930
         CLA     SEPD+3                                             YTL03940
         STO     INST            NEXT INSTRUCTION                   YTL03950
         CLA     INSCNT                                             YTL03960
         ADD     =1                                                 YTL03970
         STO     INSCNT                                             YTL03980
         CLA     KEY                                                YTL03990
         STA     SAVKEY                                             YTL04000
         TNZ     *+4                                                YTL04010
         CLA     YMAA            FROM CARD PROGRAM                  YTL04020
         STO     KEY                                                YTL04030
         TRA     SEPF                                               YTL04040
         CLA     YMAA            FROM CORE PROGRAM                  YTL04050
         STO     KEY                                                YTL04060
         TRA     SUBLNK          STORE RETURN LOCATION              YTL04070
*                                                                   YTL04080
  SEPF   LDQ*    NOINST                                             YTL04090
         MPY     =6                                                 YTL04100
         XCA                                                        YTL04110
         PAX     0,1                                                YTL04120
         ADD     SEPD+3                                             YTL04130
         STA     GETFLD                                             YTL04140
         AXT     5,2                                                YTL04150
GETFLD   CLA     **,1                                               YTL04160
```

```
         TNZ      *+2                                                    YTL04170
         STZ*     GETFLD                                                 YTL04180
         TXI      *+1,1,-1                                               YTL04190
         TIX      GETFLD,2,1                                             YTL04200
         STZ*     GETFLD                                                 YTL04210
         TIX      GETFLD-1,1,1                                           YTL04220
         CLA      YMNC                                                   YTL04230
         TNZ      WHERE                                                  YTL04240
         STZ      INSCNT                                                 YTL04250
*                                                                        YTL04260
         TTR      COREX                                                  YTL04270
*                                                                        YTL04280
CYCLE    SUB      THOUSN                                                 YTL04290
         STO      YMAA                                                   YTL04300
         CLA      KEY                                                    YTL04310
         TNZ      *+3                                                    YTL04320
         CLA      =24                                                    YTL04330
         TRA      YMERR          CANNOT CYCLE FROM TAPE                  YTL04340
         CLA      YMNC           COUNT                                   YTL04350
         TZE      *+4            ZERO COUNT                              YTL04360
         TPL      *+3                                                    YTL04370
NEGCT    CLA      =25                                                    YTL04380
         TRA      YMERR          NEGATIVE COUNT                          YTL04390
         STO      TEMP                                                   YTL04400
         SUB      PROGLO                                                 YTL04410
         TMI      CYAT                                                   YTL04420
         CLA      YMNC                                                   YTL04430
         ADD      YSHIFT                                                 YTL04440
         STA      YMNC                                                   YTL04450
         CLA*     YMNC                                                   YTL04460
         TZE      *+2            ZERO COUNT                              YTL04470
         TMI      NEGCT                                                  YTL04480
         STO      TEMP           COUNT IS IN TEMP                        YTL04490
         SUB      THOUSN                                                 YTL04500
         TMI      CYA                                                    YTL04510
         CLA      =26                                                    YTL04520
         TRA      YMERR          COUNT MUST BE LTE TO 1000               YTL04530
CYAT     CLA      YMNC                                                   YTL04540
         SUB      THOUSN                                                 YTL04550
         TMI      *+2                                                    YTL04560
         TRA      CYAT-2                                                 YTL04570
CYA      CLA      INST                                                   YTL04580
         SUB      =1             ADDRESS OF SUB-COUNT                    YTL04590
         STO      TEMP-1                                                 YTL04600
         CAL*     TEMP-1                                                 YTL04610
         ANA      =0077777000000                                         YTL04620
         TNZ      CYB            PRIMED                                  YTL04630
         CLA      TEMP                                                   YTL04640
         STO*     TEMP-1         PRIME                                   YTL04650
         ALS      18             PUT TOTAL COUNT IN DECREMENT            YTL04660
         ORS*     TEMP-1                                                 YTL04670
CYB      CLA*     TEMP-1                                                 YTL04680
         ANA      =0000000077777                                         YTL04690
         TZE      COUT                                                   YTL04700
         SUB      =1                                                     YTL04710
         TZE      COUT                                                   YTL04720
         STA*     TEMP-1         NEW SUB-COUNT                           YTL04730
         CLA      YMAA                                                   YTL04740
         SUB      =1                                                     YTL04750
         XCA                                                             YTL04760
```

```
        MPY     =6                                                      YTL04770
        XCA                                                             YTL04780
        ADD     INST                                                    YTL04790
        STO     INST            NEXT CARD                               YTL04800
        STO     SAVINS          INST. LOC. FOR RETURN ARRAY             YTL04810
        TRA     COUT+1                                                  YTL04820
*                                                                       YTL04830
 COUT   STZ*    TEMP-1                                                  YTL04840
        CLA     INSCNT                                                  YTL04850
        ADD     =1                                                      YTL04860
        STO     INSCNT                                                  YTL04870
        TTR     COREX                                                   YTL04880
*                                                                       YTL04890
 COREX  AXT     5,1                                                     YTL04900
        CLA*    INST            LOAD CARD                               YTL04910
        STO     YMAA+5,1                                                YTL04920
        STO     LOC+5,1         FOR ERROR TRACE                         YTL04930
        CAL     INST                                                    YTL04940
        ADD     =1                                                      YTL04950
        SLW     INST                                                    YTL04960
        TIX     COREX+1,1,1                                             YTL04970
        CAL     INST                                                    YTL04980
        ADD     =1                                                      YTL04990
        SLW     INST            NEXT CARD                               YTL05000
        SLW     SAVINS                                                  YTL05010
        AXT     5,1                                                     YTL05020
        PXD     0,0                                                     YTL05030
        ADM     YMAA+5,1                                                YTL05040
        TIX     *-1,1,1                                                 YTL05050
        TNZ     YMIN                                                    YTL05060
        CLA     MCROCT                                                  YTL05070
        TNZ     *+3                                                     YTL05080
        STZ     KEY                                                     YTL05090
        TRA     WHERE+2         GO OUT OF CORE PROGRAM                  YTL05100
        PAC     0,1             SET UP RETURN LOCATION                  YTL05110
        CAL     RETURN-1,1                                              YTL05120
        STA     INST            NEXT INSTRUCTION                        YTL05130
        ARS     18                                                      YTL05140
        STA     XMODE           PARTITION                               YTL05150
        CAL     PRGLOC-1,1      PROGRAM LOCATION                        YTL05160
        STA     KEY                                                     YTL05170
        ARS     18                                                      YTL05180
        STA     INSCNT                                                  YTL05190
        CAL     MCROCT                                                  YTL05200
        SUB     =1                                                      YTL05210
        SLW     MCROCT                                                  YTL05220
        TRA     COREX                                                   YTL05230
*                                                                       YTL05240
 WHERE  CLA     INSCNT          INCREMENT COUNT OF                      YTL05250
        ADD     =1              INSTRUCTIONS SUCCESSFULLY               YTL05260
        STO     INSCNT          COMPLETED                               YTL05270
        CLA     KEY                                                     YTL05280
        TZE     YMEXIT                                                  YTL05290
        TTR     COREX                                                   YTL05300
*                                                                       YTL05310
YMEXEC  TSX     MMM90,4                                                 YTL05320
 YMAA   PZE                                                             YTL05330
 YMBB   PZE                                                             YTL05340
 YMCC   PZE                                                             YTL05350
 YMNC   PZE                                                             YTL05360
```

```
 YMOP   PZE                                                          YTL05370
        TTR     YMERR                                                YTL05380
        TTR     WHERE                                                YTL05390
*                                                                    YTL05400
YMEXIT  LXD     IR4,4                                                YTL05410
        LXD     IR2,2                                                YTL05420
        LXD     IR3,3                                                YTL05430
        LXD     IR1,1                                                YTL05440
        LXD     IR5,5                                                YTL05450
        LXD     IR6,6                                                YTL05460
        LXD     IR7,7                                                YTL05470
        TRA     9,4             SUCCESS                              YTL05480
*                                                                    YTL05490
 YMERR  NZT     ERRCDE          DO NOT STORE CODE IF PREVIOUSLY DONE YTL05500
        STO     ERRCDE                                               YTL05510
        LXD     IR4,4                                                YTL05520
        TXI     YMEXIT+1,4,1    ERROR RETURN TO 6,4                  YTL05530
*                                                                    YTL05540
*                                                                    YTL05550
 MMM90  SXD     SPOT4,4                                              YTL05560
        CLA 1,4                 STORE PARAMETERS                     YTL05570
        ADD     =1                                                   YTL05580
        STA MMMA                                                     YTL05590
        ADD     =1                                                   YTL05600
        STA MMMA+2                                                   YTL05610
        ADD     =1                                                   YTL05620
        STO LA11                                                     YTL05630
        CLA 2,4                                                      YTL05640
        ADD     =1                                                   YTL05650
        STA MMMB                                                     YTL05660
        ADD     =1                                                   YTL05670
        STA MMMB+2                                                   YTL05680
        ADD     =1                                                   YTL05690
        STO LB11                                                     YTL05700
        CLA 3,4                                                      YTL05710
        STA     LOCC                                                 YTL05720
        ADD     =1                                                   YTL05730
        STA     LOCMC                                                YTL05740
        ADD     =1                                                   YTL05750
        STA     LOCNC                                                YTL05760
        ADD     =1                                                   YTL05770
        STO LC11                                                     YTL05780
        STZ     ANULL                                                YTL05790
        STZ     BNULL                                                YTL05800
        CLA     WDMNL                                                YTL05810
        CAS*    LA11                                                 YTL05820
        TRA     *+2                                                  YTL05830
        STO     ANULL           A IS NULL                            YTL05840
        CAS*    LB11                                                 YTL05850
        TRA     *+2                                                  YTL05860
        STO     BNULL           B IS NULL                            YTL05870
 MMMA   CLA 0                   LOC MA                               YTL05880
        STO MA                                                       YTL05890
 MMNA   CLA     0               LOC NA                               YTL05900
        STO NA                                                       YTL05910
 MMMB   CLA 0                   LOC MB                               YTL05920
        STO MB                                                       YTL05930
        CLA 0                   LOC NB                               YTL05940
        STO NB                                                       YTL05950
 MMMT   CLA 5,4                                                      YTL05960
```

```
        PAX  0,1                    CODE                                YTL05970
        TMI      *+3                                                    YTL05980
        SUB      HIPROG             FIELD 5 MUST BE POSTIVE AND         YTL05990
        TMI      *+3                LESS THAN 32500                     YTL06000
        CLA      =20                                                    YTL06010
        TRA      MMME                                                   YTL06020
        TXH      MACRO,1,33                                             YTL06030
        TRA      LYST,1                                                 YTL06040
        TRA      MMM33                                                  YTL06050
        TRA      MMM32                                                  YTL06060
        TRA      MMM31                                                  YTL06070
        TRA      MMM30                                                  YTL06080
        TRA      MMM29                                                  YTL06090
        TRA      MMM28                                                  YTL06100
        TRA      MMM27                                                  YTL06110
        TRA      MMM26                                                  YTL06120
        TRA      MMM25                                                  YTL06130
        TRA      MMM24                                                  YTL06140
        TRA      MMM23A                                                 YTL06150
        TRA  MMM22                                                      YTL06160
        TRA  MMM21                                                      YTL06170
        TRA  MMM20                                                      YTL06180
        TRA  MMM19                                                      YTL06190
        TRA  MMM18                                                      YTL06200
        TRA  MMM17                                                      YTL06210
        TRA  MMM16                                                      YTL06220
        TRA  MMM15                                                      YTL06230
        TRA  MMM14                                                      YTL06240
        TRA  MMM13                                                      YTL06250
        TRA  MMM12                                                      YTL06260
        TRA  MMM11                                                      YTL06270
        TRA  MMM10                                                      YTL06280
        TRA  MMM9                                                       YTL06290
        TRA  MMM8                                                       YTL06300
        TRA  MMM7                                                       YTL06310
        TRA  MMM6                                                       YTL06320
        TRA  MMM5                                                       YTL06330
        TRA  MMM4                                                       YTL06340
        TRA  MMM3                                                       YTL06350
        TRA  MMM2                                                       YTL06360
        TRA  MMM1                                                       YTL06370
 LYST   TRA      MMM0                                                   YTL06380
*                                                                       YTL06390
*                                                                       YTL06400
MACRO   TXH      *+3,1,TSTMAC                                           YTL06410
        CLA      =20                FIELD 5 MUST BE GTE 8000            YTL06420
        TRA      MMME                                                   YTL06430
        PXA      0,1                                                    YTL06440
        ADD      YSHIFT                                                 YTL06450
        STO      YMOP                                                   YTL06460
        ADD      =3                                                     YTL06470
        STO      INST                                                   YTL06480
        STA      SEPD+3                                                 YTL06490
        SUB      =1                                                     YTL06500
        STO      TEMP-1                                                 YTL06510
        STA      *+1                                                    YTL06520
        CLA      **                 N                                   YTL06530
        SUB      =6                                                     YTL06540
        TZE      *+3                                                    YTL06550
        CLA      =6                 MUST HAVE N=6                       YTL06560
```

```
        TRA     MMME                                                YTL06570
        CAL     TEMP-1                                              YTL06580
        SUB     =1                                                  YTL06590
        STA     NOINST                                              YTL06600
        CLA     KEY                                                 YTL06610
        STA     SAVKEY                                              YTL06620
        TNZ     *+4             CORE PROGRAM                        YTL06630
        CLA     YMOP                                                YTL06640
        STO     KEY                                                 YTL06650
        TRA     SEPF                                                YTL06660
        CLA     YMOP                                                YTL06670
        STO     KEY                                                 YTL06680
        CLA     INSCNT                                              YTL06690
        ADD     =1                                                  YTL06700
        STO     INSCNT                                              YTL06710
SUBLNK  CLA     MCROCT                                              YTL06720
        CAS     =5              CHECK FOR FULL RETURN BUFFER        YTL06730
        TRA     *+3                                                 YTL06740
        TRA     *+2                                                 YTL06750
        TRA     NOTFUL                                              YTL06760
        LXA     =4,1                                                YTL06770
        CLA     PRGLOC+5,1                                          YTL06780
        STO     PRGLOC+4,1                                          YTL06790
        CLA     RETURN+5,1      MOVE RETURN+1 THRU RETURN+4         YTL06800
        STO     RETURN+4,1      INTO RETURN THRU RETURN+3           YTL06810
        TIX     *-4,1,1                                             YTL06820
        LAC     MCROCT,1                                            YTL06830
        TRA     *+5                                                 YTL06840
NOTFUL  CAL     MCROCT                                              YTL06850
        ADD     =1                                                  YTL06860
        SLW     MCROCT                                              YTL06870
        PAC     0,1                                                 YTL06880
        CAL     SAVINS                                              YTL06890
        SLW     RETURN-1,1                                          YTL06900
        CAL     XMODE                                               YTL06910
        ALS     18                                                  YTL06920
        STD     RETURN-1,1                                          YTL06930
        CLA     SAVKEY                                              YTL06940
        SLW     PRGLOC-1,1                                          YTL06950
        CLA     INSCNT                                              YTL06960
        ALS     18                                                  YTL06970
        STD     PRGLOC-1,1                                          YTL06980
        TRA     SEPF                                                YTL06990
*                                                                   YTL07000
*                                                                   YTL07010
 MMME   STO     ERRCDE          SAVE ERROR CODE                     YTL07020
        LXD     SPOT4,4                                             YTL07030
        TRA     6,4                                                 YTL07040
 MMMR   TSX     TAST,4                                              YTL07050
        LXD     SPOT4,4                                             YTL07060
        CLA     4,4                                                 YTL07070
  LOCC  STO     0               LOC C                               YTL07080
        CLA     MC                                                  YTL07090
 LOCMC  STO     0               LOC MC                              YTL07100
        CLA     NC                                                  YTL07110
 LOCNC  STO     0               LOC NC                              YTL07120
        TRA     7,4                                                 YTL07130
*                                                                   YTL07140
  TAST  ZET     SIGN                                                YTL07150
        TTR     TEST2                                               YTL07160
```

```
       CLA     =0077777000000                                          YTL07170
       STO     SIGN                                                    YTL07180
       CLA     HICORE                                                  YTL07190
       ADD     =1                                                      YTL07200
       STA     *+2                                                     YTL07210
       CLA     COREND                                                  YTL07220
       STO     **                                                      YTL07230
       TTR     1,4                                                     YTL07240
TEST2  CLA*    TEST2-2                                                 YTL07250
       SUB     COREND                                                  YTL07260
       TZE     TEST2-1                                                 YTL07270
       CLA     =19             CORE LIMIT EXCEEDED                     YTL07280
       TRA     MMME                                                    YTL07290
       REM                                                             YTL07300
       REM PACKAGE 1     MMM0,1,2                                      YTL07310
       REM MATRIX TRANSFER, MATRIX ADD, MATRIX SUBTRACT                YTL07320
       REM                                                             YTL07330
MMM0   CAL     2,4                                                     YTL07340
       TNZ POSIT                                                       YTL07350
       CLA 3,4                                                         YTL07360
       SUB     SMLTAP                                                  YTL07370
       TPL RITE                                                        YTL07380
       CLA 1,4                                                         YTL07390
       SUB     SMLTAP                                                  YTL07400
       TPL REDE                                                        YTL07410
       TRA MOVE                                                        YTL07420
*                                                                      YTL07430
 RITE  NZT     MA                                                      YTL07440
       TRA     *+5                                                     YTL07450
       NZT     NA                                                      YTL07460
       TRA     *+3                                                     YTL07470
       TSX     CKDM1,4                                                 YTL07480
       LXD     SPOT4,4                                                 YTL07490
       CLA*    1,4                                                     YTL07500
       TNZ     *+2                                                     YTL07510
       CLA     =0                                                      YTL07520
       STO     SAVE            NAME                                    YTL07530
       CLA     MA                                                      YTL07540
       STO     SAVE+1          M                                       YTL07550
       CLA     NA                                                      YTL07560
       STO     SAVE+2          N                                       YTL07570
       STZ     TESTC                                                   YTL07580
*      SPMTC1=0 IF MATRIX IS NOT SPARSE,OTHERWISE,SPARSE               YTL07590
       STZ     SPMTC1                                                  YTL07600
       STZ     SAVE+4                                                  YTL07610
       LDQ     MA                                                      YTL07620
       MPY     NA                                                      YTL07630
       XCA                                                             YTL07640
       PAX     0,2             M * N                                   YTL07650
       STO     MTN             SAVE M*N                                YTL07660
       TNZ     NZRO                                                    YTL07670
       CAL     SAVE                                                    YTL07680
       ACL     SAVE+1                                                  YTL07690
       ACL     SAVE+2                                                  YTL07700
       SLW     SAVE+3                                                  YTL07710
       TRA     WID                                                     YTL07720
 NZRO  ADD     =3                                                      YTL07730
       PAX     0,1             M * N + 3                               YTL07740
       ADD     1,4             A(1,1) + M * N                          YTL07750
       STA     GTCSM                                                   YTL07760
```

```
        STA     WRMAT                                                   YTL07770
        SXA     MNP3,1                                                  YTL07780
CKNUL1  CLA*    WRMAT           TEST FOR WORD M=NULL                    YTL07790
        SUB     WDMNL                                                   YTL07800
        TZE     NULLM           WORD M=NULL PRESENT                     YTL07810
CKNUL2  ZET*    WRMAT           TEST FOR ALL ZEROS                      YTL07820
        TRA     GETCKS          NON-ZERO ELEMENT                        YTL07830
        TIX     *-2,2,1                                                 YTL07840
 NULLM  CAL     WDMNL           FORM CHECK SUM FOR NULL MATRIX          YTL07850
        ACL     SAVE                                                    YTL07860
        ACL     SAVE+1                                                  YTL07870
        ACL     SAVE+2                                                  YTL07880
        SLW     TESTC                                                   YTL07890
        TRA     STCKS                                                   YTL07900
*               THE FOLLOWING SET OF INSTRUCTIONS                       YTL07910
*               DETERMINE IF A MATRIX IS SPARSE. IF                     YTL07920
*               THE MATRIX IS SPARSE, THE CHECKSUM IS                   YTL07930
*               CALCULATED NEGLECTING NEGATIVE ZEROS.                   YTL07940
*               THE CONTROL WORDS ARE FORMED AND PLACED                 YTL07950
*               IN THE MATRIX BEFORE THE CHECKSUM IS                    YTL07960
*               FORMED. AFTER THE MATRIX IS WRITTEN                     YTL07970
*               ON TAPE, THE CONTROL WORDS WILL BE                      YTL07980
*               REMOVED FROM THE MATRIX.                                YTL07990
GETCKS  LXA     MTN,2                                                   YTL08000
 CKSP1  CLA*    WRMAT           TEST FOR SPARSE MATRIX                  YTL08010
        TZE     *+3                                                     YTL08020
        ANA     =0377400000000                                          YTL08030
        TZE     LDCKS           MATRIX NOT SPARSE                       YTL08040
        TIX     *-4,2,1                                                 YTL08050
        LXA     MTN,2                                                   YTL08060
        SXA     SPMTC1,2                                                YTL08070
*                                                                       YTL08080
*               THE FOLLOWING FORMS THE CONTROL WORDS                   YTL08090
        AXT     0,1             FIX CONTROL WORD                        YTL08100
SPTST1  CLA*    WRMAT                                                   YTL08110
        TNZ     SPTST2-1                                                YTL08120
        TXI     *+1,1,1         ADD 1 TO ZERO COUNT                     YTL08130
        TIX     SPTST1,2,1                                              YTL08140
        PXA     0,1                                                     YTL08150
        STO*    WRMAT           STORE CONTROL WO-D                      YTL08160
        TRA     SPCKS           GO FORM CHECKSUM                        YTL08170
        TXH     SPTST2+2,1,0                                            YTL08180
SPTST2  TIX     SPTST1,2,1                                              YTL08190
        TRA     SPCKS           GO FORM CHECKSUM                        YTL08200
        TXI     *+1,2,1                                                 YTL08210
        PXA     0,1                                                     YTL08220
        STO*    WRMAT           STORE CONTROL WORD                      YTL08230
        TIX     SPTST1-1,2,2                                            YTL08240
*                                                                       YTL08250
 SPCKS  PXA     0,0             FORM SPARSE MATRIX CK SUM               YTL08260
        LXA     MNP3,1                                                  YTL08270
 CKSP2  ZET*    GTCSM                                                   YTL08280
 CKSP3  ACL*    GTCSM                                                   YTL08290
        TIX     *-2,1,1                                                 YTL08300
        ACL     SPARSE                                                  YTL08310
        LDQ     SPARSE                                                  YTL08320
        STQ     SAVE+4                                                  YTL08330
        TRA     STCKS                                                   YTL08340
*                                                                       YTL08350
 LDCKS  PXA     0,0                                                     YTL08360
```

```
 GTCSM ACL     **,1                                                  YTL08370
       TIX     *-1,1,1                                               YTL08380
 STCKS SLW     SAVE+3          CHECKSUM IN FOURTH WORD               YTL08390
       LXA     MTN,2                                                 YTL08400
   WID CLA     3,4                                                   YTL08410
       SSP                                                           YTL08420
       TSX     TP,4            GET TAPE NUMBER                       YTL08430
       STO     TAPE                                                  YTL08440
       TSX     TPCK,4                                                YTL08450
       CALL    .FVIO.(TAPE,TAPEIB)                                   YTL08460
       CALL    .FWRB.(TAPEIB)                                        YTL08470
       CALL    .FBLO.(SAVE,=16)                                      YTL08480
       CALL    .FWLR.                                                YTL08490
       CALL    .FWRB.(TAPEIB)                                        YTL08500
       NZT     MTN                                                   YTL08510
       TRA     MINR                                                  YTL08520
       ZET     SPMTC1                                                YTL08530
       TRA     WRSPRS          SPARSE MATRIX                         YTL08540
       ZET     TESTC                                                 YTL08550
       TRA     WRNULL          NULL MATRIX                           YTL08560
 WRMAT CLA     **,2            WRITE MATRIX                          YTL08570
       TSX     .FBLT.,4                                              YTL08580
       TIX     *-2,2,1                                               YTL08590
   MINR CLA    =16                                                   YTL08600
       SUB     MTN                                                   YTL08610
  WRMR TMI     ENDWR                                                 YTL08620
       TZE     ENDWR                                                 YTL08630
       PAX     0,2             M*N LESS THAN 16. WRITE MORE          YTL08640
       ZAC                                                           YTL08650
       TSX     .FBLT.,4                                              YTL08660
       TIX     *-2,2,1                                               YTL08670
       TRA     ENDWR                                                 YTL08680
*              WRITE SPARSE MATRIX                                   YTL08690
 WRSPRS AXT    0,1                                                   YTL08700
       CLA*    WRMAT                                                 YTL08710
       TZE     SPSZP1          ZERO ELEMENT, DO NOT WRITE            YTL08720
       TSX     .FBLT.,4                                              YTL08730
       TXI     *+1,1,1                                               YTL08740
       LDQ*    WRMAT                                                 YTL08750
       PXA     0,0                                                   YTL08760
       LLS     9                                                     YTL08770
       TNZ     *+2                                                   YTL08780
  SPSZ STZ*    WRMAT           STORE ZERO OVER CONTROL WORD          YTL08790
 SPSZP1 TIX    WRSPRS+1,2,1                                          YTL08800
       SXA     WDWC,1                                                YTL08810
       CLA     =16                                                   YTL08820
       SUB     WDWC                                                  YTL08830
       TRA     WRMR                                                  YTL08840
 WRNULL CLA    WDMNL           M=NULL                                YTL08850
       TSX     .FBLT.,4                                              YTL08860
       AXT     15,2                                                  YTL08870
       ZAC                                                           YTL08880
       TSX     .FBLT.,4                                              YTL08890
       TIX     *-2,2,1                                               YTL08900
 ENDWR CALL    .FWLR.                                                YTL08910
       TRA     DONE                                                  YTL08920
   REDE CLA    1,4                                                   YTL08930
       TSX     TP,4            GET TAPE NUMBER                       YTL08940
       STO     TAPE                                                  YTL08950
       TSX     TPCK,4                                                YTL08960
```

```
         CALL    .FVIO.(TAPE,TAPEIB)                                    YTL08970
         CALL    .FRDB.(TAPEIB)                                         YTL08980
         CALL    .FBLI.(SAVE,=16)                                       YTL08990
         CALL    .FRLR.                                                 YTL09000
         LXD     SPOT4,4                                                YTL09010
         CLA     SAVE            NAME                                   YTL09020
         STO*    3,4                                                    YTL09030
         NZT     4,4             IS CARD NAME =0                        YTL09040
         TRA     *+5             YES                                    YTL09050
         SUB     4,4             NO                                     YTL09060
         TZE     *+3                                                    YTL09070
         CLA     =17             NAME ON TAPE DOES NOT CHECK            YTL09080
         TRA     MMME                                                   YTL09090
         CLA     SAVE+1          YES                                    YTL09100
         STO*    LOCMC                                                  YTL09110
         STO     MC                                                     YTL09120
         LDQ     SAVE+2                                                 YTL09130
         STQ     NC                                                     YTL09140
         STQ*    LOCNC                                                  YTL09150
         MPY     SAVE+1                                                 YTL09160
         TNZ     DIMOK-2                                                YTL09170
         XCA                                                            YTL09180
         PAX     0,1             M * N                                  YTL09190
         STO     MTN                                                    YTL09200
         TZE     DIMOK                                                  YTL09210
         TMI     *+2                                                    YTL09220
         CAS     MXDATA                                                 YTL09230
         TRA     *+3                                                    YTL09240
         TRA     DIMOK                                                  YTL09250
         TRA     DIMOK                                                  YTL09260
         CLA     =28                                                    YTL09270
         TRA     MMME                                                   YTL09280
DIMOK    ADD     =3                                                     YTL09290
         PAX     0,2             M * N + 3                              YTL09300
         ADD     3,4             A(1,1) + M * N                         YTL09310
         STA     RDMAT                                                  YTL09320
         STA     GTCSM1                                                 YTL09330
         STA     FIXNM                                                  YTL09340
         TSX     CHKCOR,4                                               YTL09350
RZM      CALL    .FRDB.(TAPEIB)                                         YTL09360
         CLA     SAVE+4                                                 YTL09370
         SUB     SPARSE                                                 YTL09380
         TNZ     CKNUL3                                                 YTL09390
         STZ     RSCKSM          SPARSE MATRIX CHECKSUM                 YTL09400
*                                                                       YTL09410
*        THE FOLLOWING SET OF INSTRUCTIONS READ A                       YTL09420
*        SPARSE MATRIX AND FORMS ITS CHECKSUM                           YTL09430
*                                                                       YTL09440
SPRD     TSX     .FBLT.,4                                               YTL09450
         STO     READTP          SAVE WORD FROM TAPE                    YTL09460
         CAL     READTP                                                 YTL09470
         ACL     RSCKSM          ADD TO CHECKSUM                        YTL09480
         SLW     RSCKSM                                                 YTL09490
         CAL     READTP                                                 YTL09500
         ANA     =0377400000000                                         YTL09510
         TZE     KCW             CONTROL WORD                           YTL09520
         CLA     READTP                                                 YTL09530
         STO*    RDMAT                                                  YTL09540
         TIX     SPRD,1,1                                               YTL09550
RLR1     CALL    .FRLR.                                                 YTL09560
```

```
        CAL     RSCKSM          FORM CHECKSUM                    YTL09570
        ACL     SPARSE                                           YTL09580
        ACL     SAVE                                             YTL09590
        ACL     SAVE+1                                           YTL09600
        ACL     SAVE+2                                           YTL09610
        TRA     RDCKS                                            YTL09620
 KCW    LXA     READTP,2        STORE ZEROS                      YTL09630
        STZ*    RDMAT                                            YTL09640
        TXI     *+1,1,-1                                         YTL09650
        TIX     *-2,2,1                                          YTL09660
        TXL     RLR1,1,0                                         YTL09670
        TRA     SPRD                                             YTL09680
CKNUL3  TSX     .FBLT.,4                                         YTL09690
        NZT     MTN                                              YTL09700
        TRA     RDMAT+2                                          YTL09710
        STO     TESTC                                            YTL09720
        SUB     WDMNL           M=NULL                           YTL09730
        TNZ     RDMAT-3                                          YTL09740
        CALL    .FRLR.                                           YTL09750
        TRA     NULLMR          MATRIX IS NULL                   YTL09760
*                                                                YTL09770
        CLA     TESTC           GET A(1,1) BACK                  YTL09780
        TRA     *+2                                              YTL09790
        TSX     .FBLT.,4                                         YTL09800
 RDMAT  STO     **,1            READ IN MATRIX                   YTL09810
        TIX     *-2,1,1                                          YTL09820
        CALL    .FRLR.                                           YTL09830
        CAL     SAVE+3          IS CHECKSUM = 0                  YTL09840
        TZE     DONE            YES, DONE                        YTL09850
        ERA     =1.0            IS CHECKSUM = 1.0                YTL09860
        TZE     DONE            YES, DONE                        YTL09870
        PXA     0,0             NO, COMPARE CHECKSUMS            YTL09880
GTCSM1  ACL     **,2                                             YTL09890
        TIX     *-1,2,1                                          YTL09900
 RDCKS  ERA     SAVE+3                                           YTL09910
        TZE     DONE                                             YTL09920
        CLA     =18             CHECKSUM ERROR                   YTL09930
        TRA     MMME                                             YTL09940
NULLMR  LXA     MTN,2           STORE ZEROS FOR NULL MATRIX      YTL09950
 FIXNM  STZ     **,2                                             YTL09960
        TIX     *-1,2,1                                          YTL09970
        CAL     WDMNL                                            YTL09980
        SLW*    LC11                                             YTL09990
        ACL     SAVE            FORM NULL CHECKSUM               YTL10000
        ACL     SAVE+1                                           YTL10010
        ACL     SAVE+2                                           YTL10020
        TRA     RDCKS                                            YTL10030
 POSIT  STZ     TAPE+2                                           YTL10040
        STP     TAPE+2          SAVE SIGN OF TAPE INSTRUCTION    YTL10050
        CLA     2,4                                              YTL10060
        SSP                                                      YTL10070
        TSX     TP,4            GET TAPE NO. IN DECREMENT        YTL10080
        STO     TAPE                                             YTL10090
        STQ     TAPE+1          FILE IN ADDRESS, MATRICES IN DEC. YTL10100
        TSX     TPCK,4                                           YTL10110
        CALL    .FVIO.(TAPE,TAPEIB)                              YTL10120
        CLA     TAPE+1                                           YTL10130
        TNZ     SPACE1          IS TAPE POSITIONING REQUESTED    YTL10140
        CLA     TAPE+2                                           YTL10150
        TPL     EOF                                              YTL10160
```

```
        CALL    .FRWT.(TAPEIB)                          REWIND          YTL10170
        TTR     DONE                                                    YTL10180
EOF     CALL    .FEFT.(TAPEIB)                          WRITE END OF FILE YTL10190
        TTR     DONE                                                    YTL10200
SPACE1  CLA     TAPE+2                                                  YTL10210
        TPL     FST             FORWARD SPACING                         YTL10220
        CAL     TAPE+1                                                  YTL10230
        ANA     =0000000077777  LEAVE ONLY FILE COUNT                   YTL10240
        SUB     =1                                                      YTL10250
        SLW     TAPE+1                                                  YTL10260
        CALL    BSF(TAPE+1,TAPE,ERR)                                    YTL10270
        CLA     ERR                                                     YTL10280
        TZE     DONE                                                    YTL10290
        ADD     THOUSN                                                  YTL10300
        TRA     MMME                                                    YTL10310
  FST   CAL     TAPE+1                                                  YTL10320
        ANA     =0000000077777                                          YTL10330
        TZE     RCDOLY          NO FILE SPACING                         YTL10340
        SLW     TAPE+2                                                  YTL10350
        CALL    FSF(TAPE+2,TAPE,ERR)                                    YTL10360
        CLA     ERR                                                     YTL10370
        TZE     *+3                                                     YTL10380
        ADD     =2000           FSF ERROR                               YTL10390
        TRA     MMME                                                    YTL10400
        CAL     TAPE+1                                                  YTL10410
        ANA     =0077777000000                                          YTL10420
        TZE     DONE            FILE SPACE ONLY                         YTL10430
        SLW     TAPE+1                                                  YTL10440
RCDOLY  CLA     TAPE+1                                                  YTL10450
        ARS     18                                                      YTL10460
        STO     TAPE+1                                                  YTL10470
        CALL    FSR(TAPE+1,TAPE,ERR)                                    YTL10480
        CLA     ERR                                                     YTL10490
        TZE     DONE                                                    YTL10500
        ADD     =3000           FSR ERROR                               YTL10510
        TRA     MMME                                                    YTL10520
*                                                                       YTL10530
  DONE  TSX     TAST,4                                                  YTL10540
        LXD     SPOT4,4                                                 YTL10550
        TRA     7,4                                                     YTL10560
*                                                                       YTL10570
*               INTERPRET TAPE NUMBER                                   YTL10580
*                                                                       YTL10590
   TP   SXD     TEMP,1          SAVE INDEX                              YTL10600
        AXT     0,1                                                     YTL10610
        SUB     SMLTAP                                                  YTL10620
        TMI     TP2             TRANSFER IF BST OR FST                  YTL10630
        TXI     *+1,1,1         COUNTER                                 YTL10640
        TNZ     *-3                                                     YTL10650
        LDQ     =0              CLEAR MQ IFONLY A TAPE NUMBER           YTL10660
        PXA     0,1             TAPE NUMBER IN AC(ADDRESS)              YTL10670
  TP1   LXD     TEMP,1          LOAD INDEX                              YTL10680
        TTR     1,4             RETURN                                  YTL10690
  TP2   STZ     TEMP-1                                                  YTL10700
        SXA     TEMP-1,1        SAVE TAPE NUMBER                        YTL10710
        ADD     MILYN                                                   YTL10720
        AXT     0,1                                                     YTL10730
        SUB     THOUSN                                                  YTL10740
        TMI     TP4             TRANSFER IF FST                         YTL10750
        TXI     *+1,1,1         COUNTER                                 YTL10760
```

```
        TNZ     *-3                                                     YTL10770
        PXD     0,0             CLEAR MQ AND AC                         YTL10780
        XCA                                                             YTL10790
        PXA     0,1              NO. OF FILES IN AC(ADDRESS)            YTL10800
        XCA                         NOW IN MQ                           YTL10810
 TP3    CLA     TEMP-1           TAPE NUMBER                            YTL10820
        TTR     TP1                                                     YTL10830
 TP4    STZ     TEMP-2                                                  YTL10840
        SXA     TEMP-2,1         SAVE NO. OF FILES                      YTL10850
        ADD     THOUSN                                                  YTL10860
        ALS     19               2*NO. OF RECDS IN DECREMENT            YTL10870
        STD     TEMP-2                                                  YTL10880
        LDQ     TEMP-2           FILES IN ADDR, RECDS IN DECRE.         YTL10890
        TTR     TP3                                                     YTL10900
*                                                                       YTL10910
*               CHECK TAPE NUMBER                                       YTL10920
*                                                                       YTL10930
 TPCK   LXA     TAPE,1                                                  YTL10940
        TXL     *+3,1,MXTAPS                                            YTL10950
        CLA     =13                                                     YTL10960
        TRA     MMME                                                    YTL10970
        TTR     TPCK0,1                                                 YTL10980
TPCK19  TTR     1,4                                                     YTL10990
TPCK18  TTR     1,4                                                     YTL11000
TPCK17  TTR     1,4                                                     YTL11010
TPCK16  TTR     1,4                                                     YTL11020
TPCK15  TTR     1,4                                                     YTL11030
TPCK14  TTR     1,4                                                     YTL11040
TPCK13  TTR     1,4                                                     YTL11050
TPCK12  TTR     1,4                                                     YTL11060
TPCK11  TTR     1,4                                                     YTL11070
TPCK10  TTR     1,4                                                     YTL11080
TPCK9   TTR     1,4                                                     YTL11090
 TPCK8  TTR     1,4                                                     YTL11100
 TPCK7  TTR     MMMTP           STACKED PUNCH OUTPUT                    YTL11110
 TPCK6  TTR     MMMTP           STACKED OUTPUT                          YTL11120
 TPCK5  TTR     MMMTP           STACKED INPUT ONLY                      YTL11130
 TPCK4  TTR     1,4                                                     YTL11140
 TPCK3  TTR     1,4                                                     YTL11150
 TPCK2  TTR     1,4                                                     YTL11160
 TPCK1  TTR     1,4                                                     YTL11170
 TPCK0  TRA     TPCK+2                                                  YTL11180
 MMMTP  CLA     =14              LEGAL TAPE, BUT NOT ACCESSIBLE         YTL11190
        TRA     MMME                                                    YTL11200
*                                                                       YTL11210
*               THIS ROUTINE CHECKS THE LEGALITY OF                     YTL11220
*               DIMENSIONS OF FIELD 1 OR 2, OR BOTH                     YTL11230
*               M * N MUST BE LESS THAN 24561 , NON-ZERO, AND POSITIVE  YTL11240
*                                                                       YTL11250
 CKDM1  LDQ     MA               FIELD 1 CHECK                          YTL11260
        MPY     NA                                                      YTL11270
        TNZ     BADDM1                                                  YTL11280
        XCA                                                             YTL11290
        TZE     BADDM1                                                  YTL11300
        TMI     BADDM1                                                  YTL11310
        CAS     MXDATA                                                  YTL11320
        TRA     BADDM1                                                  YTL11330
        TRA     1,4                                                     YTL11340
        TRA     1,4                                                     YTL11350
 CKDM2  LDQ     MB               FIELD 2 CHECK                          YTL11360
```

```
          MPY     NB                                                    YTL11370
          TNZ     BADDM2                                                YTL11380
          XCA                                                           YTL11390
          TZE     BADDM2                                                YTL11400
          TMI     BADDM2                                                YTL11410
          CAS     MXDATA                                                YTL11420
          TRA     BADDM2                                                YTL11430
          TRA     1,4                                                   YTL11440
          TRA     1,4                                                   YTL11450
CKDM12    SXA     CKDMRT,4        CHECK BOTH FIELDS                     YTL11460
          TSX     CKDM1,4                                               YTL11470
          TSX     CKDM2,4                                               YTL11480
CKDMRT    AXT     **,4                                                  YTL11490
          TRA     1,4                                                   YTL11500
BADDM1    CLA     =3                                                    YTL11510
          TRA     MMME                                                  YTL11520
BADDM2    CLA     =4                                                    YTL11530
          TRA     MMME                                                  YTL11540
*                                                                       YTL11550
*                 CHECK TO SEE IF A DIAGONAL MATRIX                     YTL11560
*                 IS A 1 X N OR A M X 1                                 YTL11570
*                                                                       YTL11580
 CKDG1    CLA     =1                                                    YTL11590
          CAS     MA                                                    YTL11600
          TRA     BADDM1                                                YTL11610
          TRA     1,4                                                   YTL11620
          CAS     NA                                                    YTL11630
          TRA     BADDM1                                                YTL11640
          TRA     1,4                                                   YTL11650
 NOTDG    CLA     =5                                                    YTL11660
          TRA     MMME                                                  YTL11670
 CKDG2    CLA     =1                                                    YTL11680
          CAS     MB                                                    YTL11690
          TRA     BADDM2                                                YTL11700
          TRA     1,4                                                   YTL11710
          CAS     NB                                                    YTL11720
          TRA     BADDM2                                                YTL11730
          TRA     1,4                                                   YTL11740
          TRA     NOTDG                                                 YTL11750
*                 CHECK TO SEE IF RESULT MATRIX WILL OVERFLOW CORE      YTL11760
CHKCOR    LDQ     MC                                                    YTL11770
          MPY     NC                                                    YTL11780
          XCA                                                           YTL11790
          ADD     LC11                                                  YTL11800
          CAS     HICORE                                                YTL11810
          TRA     OVER                                                  YTL11820
          TRA     1,4                                                   YTL11830
          TRA     1,4                                                   YTL11840
  OVER    CLA     =19                                                   YTL11850
          TRA     MMME                                                  YTL11860
*                 CHECK TO SEE IF AN ADDRESS IS BETWEEN                 YTL11870
*                 8000 AND 32563,INCLUSIVE                              YTL11880
CHKRNG    CAS     HICORE          THE RETURN IS MADE AS                 YTL11890
          TRA     1,4             WITH A CAS INSTRUCTION                YTL11900
          TRA     2,4             ABOVE - 1,4                           YTL11910
          CAS     LOCORE          INBETWEEN - 2,4                       YTL11920
          TRA     2,4             BELOW - 3,4                           YTL11930
          TRA     2,4                                                   YTL11940
          TRA     3,4                                                   YTL11950
*                                                                       YTL11960
```

```
*               FLOATING POINT TRAP ANALYSIS FOR ALL                    YTL11970
*               OF THE OPERATION CODES                                  YTL11980
*                                                                       YTL11990
FPSPIL  STO     TEMPAC                                                  YTL12000
        STQ     TEMPMQ                                                  YTL12010
        CLA     0                                                       YTL12020
        STD     CELL0                                                   YTL12030
        CLA     CELL0                                                   YTL12040
        CAS     =3B17                                                   YTL12050
        TRA     TSTDIV          NOT UNDERFLOW FROM MPY, ADD             YTL12060
        TRA     ACMQUF          UNDERFLOW IN BOTH AC AND MQ             YTL12070
        CLA     TEMPAC          UNDERFLOW IN MQ ONLY                    YTL12080
        LDQ     =0                                                      YTL12090
        STZ     CELL0           CLEAR CELL IF UNDERFLOW                 YTL12100
        TRA*    0                                                       YTL12110
ACMQUF  PXA     0,0             CLEAR AC                                YTL12120
        TRA     *-4                                                     YTL12130
TSTDIV  CAS     =9B17                                                   YTL12140
        TRA     *+4             IT IS A DIVIDE, TEST FURTHER            YTL12150
        TRA     ACMQUF          MQ UNDERFLOW ON DIVIDE                  YTL12160
        CLA     =8              OVERFLOW ON MPY OR ADD                  YTL12170
        TRA     MMME                                                    YTL12180
        CAS     =11B17                                                  YTL12190
        TRA     POSSTO          OVERFLOW ON DIVIDE OR IT                YTL12200
*                                COULD ALSO BE DOUBLE PRECISION          YTL12210
*                               STORAGE TRAP                            YTL12220
        TRA     ACMQUF          AC,MQ UNDERFLOW ON DIVIDE               YTL12230
        PXA     0,0             AC UNDERFLOW ONLY                       YTL12240
        LDQ     TEMPMQ                                                  YTL12250
        TRA     ACMQUF-2                                                YTL12260
POSSTO  CAS     =13B17                                                  YTL12270
        TRA     *+4                                                     YTL12280
        TRA     *+1                                                     YTL12290
        CLA     =9                                                      YTL12300
        TRA     MMME            DIVIDE OVERFLOW                         YTL12310
        CLA     =27                                                     YTL12320
        TRA     MMME            D.P. STORAGE TRAP.                      YTL12330
*               THE FOLLOWING ARE ADDITIONS TO CONSIDER                 YTL12340
*               ADDITION AND MULTIPLICATION OF NULL                     YTL12350
*               MATRICES. MARCH 1963                                    YTL12360
 NADD1  LDQ     FILL5           B IS NULL                               YTL12370
        STQ     LDMT                                                    YTL12380
        STA     STMT                                                    YTL12390
        SUB     LC11                                                    YTL12400
        ADD     LA11                                                    YTL12410
        STA     LDMT                                                    YTL12420
 LDMT                           LA11+MNA                                YTL12430
 STMT   STO     **,1            LC11+MNC                                YTL12440
        TIX     *-2,1,1                                                 YTL12450
        TRA     MMMR                                                    YTL12460
*                               A IS NULL FOR MMM3 OR MMM4              YTL12470
 LDMT1                          LB11+MNB                                YTL12480
 STMT1  STO     **,2            LC11+MNC                                YTL12490
        TIX     STMT1+3,1,0     NB                                      YTL12500
        TXI     STMT1+1,1,0     MN-1                                    YTL12510
        TIX     LDMT1,2,1                                               YTL12520
        TRA     MMMR                                                    YTL12530
*                                                                       YTL12540
 NMPY1  SXA     NLP1,2                                                  YTL12550
        LXA     NLP1,4                                                  YTL12560
```

```
NMPY  STA   NLP1          NULL MATRIX MULTIPLY                    YTL12570
NLP1  STZ   **,4                                                  YTL12580
      TIX   *-1,4,1                                               YTL12590
      CLA   WDMNL                                                 YTL12600
      STO*  LC11                                                  YTL12610
      TRA   MMMR                                                  YTL12620
MMMCF CLA   =1            NON CONFORMABLE MATRICES.               YTL12630
      TRA   MMME                                                  YTL12640
*                                                                 YTL12650
MOVE  CLA FILL0                                                   YTL12660
      STO STAL1                                                   YTL12670
      LDQ   MA                                                    YTL12680
      MPY   NA                                                    YTL12690
      TNZ   *+3                                                   YTL12700
      XCA                                                         YTL12710
      TZE   *+2               ALLOW A MOVE OF 0 X 0, M X 0, 0 X N YTL12720
      TSX   CKDM1,4           CHECK DIMENSIONS IN FIELD 1         YTL12730
      LDQ MA                                                      YTL12740
      STQ MC                                                      YTL12750
      CLA NA                                                      YTL12760
      STO NC                                                      YTL12770
      TRA ELXEL                                                   YTL12780
MMM1  LDQ   FILL5             MATRIX ADD                          YTL12790
      CLA   FILL1                                                 YTL12800
      TRA   *+3                                                   YTL12810
MMM2  LDQ   FILL6             MATRIX SUB                          YTL12820
      CLA   FILL2                                                 YTL12830
      STQ   LDMT                                                  YTL12840
      STO STAL1                                                   YTL12850
      TSX   CKDM12,4          CHECK DIMENSIONS IN FIELDS 1,2      YTL12860
      CLA MA                                                      YTL12870
      STO MC                                                      YTL12880
      SUB MB                                                      YTL12890
      TNZ   MMMCF             NON-CONFORMABLE                     YTL12900
      CLA NA                                                      YTL12910
      STO NC                                                      YTL12920
      SUB NB                                                      YTL12930
      TNZ   MMMCF             NON-CONFORMABLE                     YTL12940
ELXEL LDQ MA                  LOOP CONTROL                        YTL12950
      MPY NA                                                      YTL12960
      XCA                                                         YTL12970
      PAX 0,1                 MNA,B,C                             YTL12980
      ADD LA11                                                    YTL12990
      STA ITR81                                                   YTL13000
      SUB LA11                                                    YTL13010
      ADD LB11                                                    YTL13020
      STA STAL1                                                   YTL13030
      STA     LDMT                                                YTL13040
      SUB LB11                                                    YTL13050
      ADD LC11                                                    YTL13060
      STA     STMT                                                YTL13070
      STA STAL1+1                                                 YTL13080
      TSX   CHKCOR,4                                              YTL13090
      LXD   SPOT4,4                                               YTL13100
      NZT   5,4                                                   YTL13110
      TRA   MOVEUP                                                YTL13120
      ZET   BNULL                                                 YTL13130
      TRA   NADD1             B IS NULL                           YTL13140
      ZET   ANULL                                                 YTL13150
      TRA   LDMT              A IS NULL                           YTL13160
```

```
 ITR81 CLA 0,1                 LA11+MNA                                  YTL13170
 STAL1                         LB11+MNB                                  YTL13180
       STO 0,1                 LC11+MNC                                  YTL13190
       TIX ITR81,1,1                                                     YTL13200
       TRA MMMR                                                          YTL13210
MOVEUP CLA     3,4                                                      YTL13220
       SUB     1,4                                                      YTL13230
       TMI     ITR81           MOVE DOWN                                YTL13240
       CLA     ITR81           MOVE UP                                  YTL13250
       STA     MSPOT1                                                   YTL13260
       CLA     STAL1+1                                                  YTL13270
       STA     MSPOT2                                                   YTL13280
       SXD     MSPOT3,1                                                 YTL13290
       AXT     1,1                                                      YTL13300
MSPOT1 CLA     0,1             LA11+MNA                                 YTL13310
MSPOT2 STO     0,1             LC11+MNC                                 YTL13320
       TXI     *+1,1,1                                                  YTL13330
MSPOT3 TXL     MSPOT1,1,**     MNA                                      YTL13340
       TRA     MMMR                                                     YTL13350
       REM PACKAGE 2     MMM3,4                                         YTL13360
       REM ADD AND SUBTRACT TRANSPOSE                                   YTL13370
       REM                                                              YTL13380
 MMM3  LDQ     FILL5           ADD TRANSPOSE                            YTL13390
       CLA     FILL3                                                    YTL13400
       TRA     *+3                                                      YTL13410
 MMM4  LDQ     FILL6           SUB TRANSPOSE                            YTL13420
       CLA     FILL4                                                    YTL13430
       STQ     LDMT1                                                    YTL13440
       STO STAL2                                                        YTL13450
       TSX     CKDM12,4        CHECK DIMENSIONS OF FIELDS 1,2           YTL13460
       CLA MA                                                           YTL13470
       STO MC                                                           YTL13480
       SUB NB                                                           YTL13490
       TNZ     MMMCF           NON-CONFORMABLE                          YTL13500
       CLA NA                                                           YTL13510
       STO NC                                                           YTL13520
       SUB MB                                                           YTL13530
       TNZ     MMMCF           NON-CONFORMABLE                          YTL13540
       TSX     CHKCOR,4                                                 YTL13550
       LDQ MA                  LOOP CONTROL                             YTL13560
       MPY NA                                                           YTL13570
       XCA                                                              YTL13580
       PAX     0,1             MNA,MNC                                  YTL13590
       PAX     0,2             MNB                                      YTL13600
       SUB     =1                                                       YTL13610
       ALS 18                                                           YTL13620
       STD     STAL2+3                                                  YTL13630
       STD     STMT1+2                                                  YTL13640
       ARS 18                                                           YTL13650
       ADD     =1                                                       YTL13660
       ADD LA11                                                         YTL13670
       STA ITR82                                                        YTL13680
       SUB LA11                                                         YTL13690
       ADD LB11                                                         YTL13700
       STA STAL2                                                        YTL13710
       STA     LDMT1                                                    YTL13720
       SUB LB11                                                         YTL13730
       ADD LC11                                                         YTL13740
       STA     STMT1                                                    YTL13750
       ZET     BNULL                                                    YTL13760
```

```
        TRA     NADD1           B IS NULL                       YTL13770
        STA STAL2+1                                             YTL13780
        CLA NB                                                  YTL13790
        ALS 18                                                  YTL13800
        STD     STMT1+1                                         YTL13810
        ZET     ANULL                                           YTL13820
        TRA     LDMT1                                           YTL13830
        STD     STAL2+2                                         YTL13840
ITR82   CLA 0,1                 LA11+MNA                        YTL13850
STAL2                           LB11+NMB                        YTL13860
        STO 0,1                 LC11+MNC                        YTL13870
        TIX     STAL2+4,2,0     NB                              YTL13880
        TXI     STAL2+2,2,0     MN-1                            YTL13890
        TIX ITR82,1,1                                           YTL13900
        TRA MMMR                                                YTL13910
        REM                                                     YTL13920
        REM PACKAGE 3     MMM5                                  YTL13930
        REM SCALAR MULTIPLY                                     YTL13940
        REM                                                     YTL13950
 MMM5   TSX     CKDM2,4         CHECK DIMENSIONS IN FIELD 2     YTL13960
        LDQ     MB                                              YTL13970
        STQ MC                                                  YTL13980
        CLA NB                                                  YTL13990
        STO NC                                                  YTL14000
        TSX     CHKCOR,4                                        YTL14010
        LDQ     MB                                              YTL14020
        MPY NB                                                  YTL14030
        XCA                                                     YTL14040
        PAX 0,1                 MNB                             YTL14050
        ADD LB11                                                YTL14060
        STA ITR83+1                                             YTL14070
        SUB LB11                                                YTL14080
        ADD LC11                                                YTL14090
        STA ITR83+2                                             YTL14100
        XCA                                                     YTL14110
        CLA     YMAA                                            YTL14120
        STA ITR83                                               YTL14130
        PXA     0,1                                             YTL14140
        PAX     0,4                                             YTL14150
        XCA                                                     YTL14160
        ZET     BNULL                                           YTL14170
        TRA     NMPY            B IS NULL                       YTL14180
ITR83   LDQ 0                   LOC A                           YTL14190
        FMP 0,1                 LOC BMN+1                       YTL14200
        STO 0,1                 LOC CMN+1                       YTL14210
        TIX ITR83,1,1                                           YTL14220
        TRA MMMR                                                YTL14230
*                                                               YTL14240
*               PACKAGE 4,MMM6,MMM7,MMM8                        YTL14250
*               THESE ARE THE THREE MULTIPLY ROUTINES           YTL14260
*               WHICH ARE WRITTEN FOR THE 7094                  YTL14270
*               THEY USE COMMON MULTIPLY INSTRUCTIONS           YTL14280
*               AND A DFAD. INCLUDES A NULL MULTIPLY            YTL14290
*               ADDED 5/63                                      YTL14300
*                                                               YTL14310
 MMM6   CLA     NA              MULTIPLY                        YTL14320
        SUB     MB                                              YTL14330
        TNZ     MMMCF           CONFORMABILITY CHECK            YTL14340
        TSX     CKDM12,4        CHECK DIMENSIONS                YTL14350
        LXA     MA,3            TOTAL ROWS OF A                 YTL14360
```

```
        LXA     NB,4            TOTAL COLUMNS OF B                      YTL14370
        SXA     NC,4            COLUMN DIMENSION OF PRODUCT             YTL14380
                                                                        YTL14390
        LDQ     MA                                                      YTL14400
        STQ     MC              ROW DIMENSION OF PRODUCT                YTL14410
        MPY     NB                                                      YTL14420
        XCA                                                             YTL14430
        PAX     0,5             TOTAL NO. OF ELEMENTS IN PRODUCT        YTL14440
        ADD     LC11                                                    YTL14450
        STA     ANSWER          STORAGE OF ANSWER                       YTL14460
        TSX     CHKCOR,4                                                YTL14470
        LXA     NB,4                                                    YTL14480
        CAL     ANULL           CHECK FOR A OR B NULL                   YTL14490
        ACL     BNULL                                                   YTL14500
        TNZ     ZROMPY                                                  YTL14510
        CLA     NA                                                      YTL14520
        STO     TOTA            NUMBER OF ELEMENTS IN ROW OF A          YTL14530
        STO     INCRMA          STORAGE INCREMENT FROM ROW TO ROW       YTL14540
        ADD     LA11                                                    YTL14550
        STA     ELEMA           ADDRESS FOR A                           YTL14560
        LDQ     MB                                                      YTL14570
        MPY     NB                                                      YTL14580
        XCA                                                             YTL14590
        STO     TOTB            TOTAL NUMBER OF ELEMENTS IN B           YTL14600
        ADD     LB11                                                    YTL14610
        STA     ELEMB           ADDRESS FOR B                           YTL14620
        LXA     =1,1                                                    YTL14630
        SXD     MPTST1,1        DECREMENT FOR A                         YTL14640
        SXA     INCRMB,1        STORAGE INCREMENT FROM COL. TO COL.     YTL14650
        LXA     NB,1                                                    YTL14660
        SXD     MPTST1+1,1      DECREMENT FOR B                         YTL14670
        TRA     MULTPY                                                  YTL14680
*                                                                       YTL14690
*       SET UP CONTROL NUMBERS FOR MMM7                                 YTL14700
*                                                                       YTL14710
MMM7    CLA     NA              POST MULTIPLY BY TRANSPOSE              YTL14720
        SUB     NB                                                      YTL14730
        TNZ     MMMCF           CONFORMABILITY CHECK                    YTL14740
        TSX     CKDM12,4        CHECK DIMENSIONS                        YTL14750
        LXA     MA,3            TOTAL ROWS IN A                         YTL14760
        LXA     MB,4            TOTAL ROWS IN B                         YTL14770
        SXA     NC,4            COLUMN DIMENSION OF PRODUCT             YTL14780
        LDQ     MA                                                      YTL14790
        STQ     MC              ROW DIMENSION OF PRODUCT                YTL14800
        MPY     MB                                                      YTL14810
        XCA                                                             YTL14820
        PAX     0,5             TOTAL NUMBER OF ELEMENTS IN PRODUCT.    YTL14830
        ADD     LC11                                                    YTL14840
        STA     ANSWER          STORAGE OF ANSWER                       YTL14850
        TSX     CHKCOR,4                                                YTL14860
        LXA     MB,4                                                    YTL14870
        CAL     ANULL           CHECK FOR A OR B NULL                   YTL14880
        ACL     BNULL                                                   YTL14890
        TNZ     ZROMPY                                                  YTL14900
        CLA     NA                                                      YTL14910
        STO     TOTA            NUMBER OF ELEMENTS IN ROW OF A          YTL14920
        STO     INCRMA          STORAGE INCREMENT FROM ROW TO ROW       YTL14930
        ADD     LA11                                                    YTL14940
        STA     ELEMA           ADDRESS FOR A                           YTL14950
        CLA     NB              NUMBER OF ELEMENTS IN ROW OF B          YTL14960
        STO     TOTB
```

```
        STO     INCRMB          STORAGE INCREMENT FROM ROW TO ROW       YTL14970
        ADD     LB11                                                    YTL14980
        STA     ELEMB                                                   YTL14990
        LXA     =1,1                                                    YTL15000
        SXD     MPTST1,1        DECREMENT FOR A                         YTL15010
        SXD     MPTST1+1,1      DECREMENT FOR B                         YTL15020
        TRA     MULTPY                                                  YTL15030
*                                                                       YTL15040
*               SET UP CONTROL NUMBERS FOR MMM8                         YTL15050
*                                                                       YTL15060
MMM8    CLA     MA              PRE MULTIPLY BY TRANSPOSE               YTL15070
        SUB     MB                                                      YTL15080
        TNZ     MMMCF           CONFORMABILITY CHECK                    YTL15090
        TSX     CKDM12,4        CHECK DIMENSIONS                        YTL15100
        LXA     NA,3            TOTAL COLUMNS OF A                      YTL15110
        SXA     MC,3            ROW DIMENSION OF PRODUCT                YTL15120
        LXA     NB,4            TOTAL COLUMNS OF B                      YTL15130
        SXA     NC,4            COLUMN DIMENSION OF PRODUCT             YTL15140
        LDQ     NA                                                      YTL15150
        MPY     NB                                                      YTL15160
        XCA                                                             YTL15170
        PAX     0,5             TOTAL NUMBER OF ELEMENTS IN PRODUCT     YTL15180
        ADD     LC11                                                    YTL15190
        STA     ANSWER          STORAGE OF ANSWER                       YTL15200
        TSX     CHKCOR,4                                                YTL15210
        LXA     NB,4                                                    YTL15220
        CAL     ANULL           CHECK FOR A OR B NULL                   YTL15230
        ACL     BNULL                                                   YTL15240
        TNZ     ZROMPY                                                  YTL15250
        LXA     NA,1                                                    YTL15260
        SXD     MPTST1,1        DECREMENT FOR A                         YTL15270
        LDQ     NA                                                      YTL15280
        MPY     MA                                                      YTL15290
        XCA                                                             YTL15300
        STO     TOTA            TOTAL NUMBER OF ELEMENTS IN A           YTL15310
        ADD     LA11                                                    YTL15320
        STA     ELEMA           ADDRESS FOR A                           YTL15330
        LXA     NB,1                                                    YTL15340
        SXD     MPTST1+1,1      DECREMENT FOR B                         YTL15350
        LDQ     NB                                                      YTL15360
        MPY     MB                                                      YTL15370
        XCA                                                             YTL15380
        STO     TOTB            TOTAL NUMBER OF ELEMENTS IN B           YTL15390
        ADD     LB11                                                    YTL15400
        STA     ELEMB           ADDRESS FOR B                           YTL15410
        CLA     =1                                                      YTL15420
        STO     INCRMA          INCREMENT FOR A                         YTL15430
        STO     INCRMB          INCREMENT FOR B                         YTL15440
        TRA     MULTPY                                                  YTL15450
ZROMPY  CLA     MAD                                                     YTL15460
        STZ     MAD                                                     YTL15470
        TNZ     MMMR                                                    YTL15480
        CAL     WDMNL                                                   YTL15490
        SLW*    LC11                                                    YTL15500
        TNX     MMMR,5,1                                                YTL15510
        STZ*    ANSWER                                                  YTL15520
        TIX     *-1,5,1                                                 YTL15530
        TRA     MMMR                                                    YTL15540
*                                                                       YTL15550
*               MULTIPLY LOOP USED BY MMM6,7,8                          YTL15560
```

```
*                                                                      YTL15570
MULTPY PXA      DOTPRD,0                                               YTL15580
       CLA      *-1                                                    YTL15590
       LBT                                                             YTL15600
       TRA      *+2                                                    YTL15610
       ADD      =1                                                     YTL15620
       STA      SUMPRD                                                 YTL15630
       CLA      MAD                                                    YTL15640
       TZE      NOADD                                                  YTL15650
       CAL      MADYES         PUT IN ADD FOR MPY AND ADD              YTL15660
       SLW      ANSWER-2                                               YTL15670
       STZ      MAD                                                    YTL15680
       TRA      *+3                                                    YTL15690
 NOADD CAL      MADNO          MULTIPLY ONLY,USE DUMMY                 YTL15700
       SLW      ANSWER-2                                               YTL15710
STRTLP LXA      TOTA,1                                                 YTL15720
       LXA      TOTB,2                                                 YTL15730
       LDQ      =0                                                     YTL15740
       PXA      0,0.                                                   YTL15750
       DST*     SUMPRD         ZERO DOT PRODUCT                        YTL15760
ZROTST NZT*     ELEMA                                                  YTL15770
       TRA      MPTST1                                                 YTL15780
       NZT*     ELEMB                                                  YTL15790
       TRA      MPTST1                                                 YTL15800
 ELEMA LDQ      **,1           ELEMENT OF A                            YTL15810
 ELEMB FMP      **,2           ELEMENT OF B                            YTL15820
SUMPRD DFAD     **             RUNNING PRODUCT                         YTL15830
       DST*     *-1                                                    YTL15840
MPTST1 TIX      *+1,1,**       DECREMENT FOR A                         YTL15850
       TIX      ZROTST,2,**    DECREMENT FOR B                         YTL15860
       HTR      MMMR           VARIABLE CELL FOR OP. CODES 30,31,32    YTL15870
       FRN                                                            YTL15880
ANSWER STO      **,5           STORE DOT PRODUCT                       YTL15890
       TIX      *+1,5,1                                                YTL15900
       TNX      NEWRCA,4,1     TRA IF A NEW ROW OR COLUMN OF A IS NEEDED YTL15910
       CLA      ELEMB          GET NEW COLUMN OR ROW OF B.             YTL15920
       ADD      INCRMB                                                 YTL15930
       STA      ELEMB                                                  YTL15940
       TRA      STRTLP         NEXT ELEMENT OF ROW OF PRODUCT          YTL15950
NEWRCA TNX      MMMR,3,1       COUNT OF ROWS OR COLUMNS OF A USED      YTL15960
       LXA      NC,4           RE-SET COUNT OF ROWS OR COLS OF B       YTL15970
       CLA      ELEMA          GET NEW ROW/COLUMN OF A                 YTL15980
       ADD      INCRMA                                                 YTL15990
       STA      ELEMA                                                  YTL16000
       CLA      LB11           START OVER WITH STORAGE ACCESS TO B      YTL16010
       ADD      TOTB                                                   YTL16020
       STA      ELEMB                                                  YTL16030
       TRA      STRTLP                                                 YTL16040
       REM                                                            YTL16050
       REM PACKAGE 5     MMM9                                         YTL16060
       REM MATRIX TRANSPOSE                                           YTL16070
       REM                                                            YTL16080
  MMM9 TSX      CKDM1,4                                                YTL16090
       CLA      MA                                                     YTL16100
       STO NC                                                         YTL16110
       ALS 18                                                         YTL16120
       STD ITR85+3                                                    YTL16130
       LDQ NA                                                         YTL16140
       STQ MC                                                         YTL16150
       TSX      CHKCOR,4                                              YTL16160
```

```
        LDQ     NA                                                  YTL16170
        MPY MA                                                      YTL16180
        XCA                                                         YTL16190
        PAX     0,1                                                 YTL16200
        PAX     0,4                                                 YTL16210
        ADD LA11                                                    YTL16220
        STA ITR85                                                   YTL16230
        SUB LA11                                                    YTL16240
        ADD LC11                                                    YTL16250
        ZET     ANULL                                               YTL16260
        TRA     NMPY            A IS NULL                           YTL16270
        STA ITR85+1                                                 YTL16280
        SUB LC11                                                    YTL16290
        ALS 18                                                      YTL16300
        CHS                                                         YTL16310
        ADD ITR85+3                                                 YTL16320
        COM                                                         YTL16330
        STD ITR85+4                                                 YTL16340
ITR85   CLA 0,1                 LA11+MNA                            YTL16350
        STO 0,4                 LC11+MNC                            YTL16360
        TNX MMMR,1,1                                                YTL16370
        TIX ITR85,4,0           NC                                  YTL16380
        TXI ITR85,4,0           MNC-NC-1                            YTL16390
        REM                                                         YTL16400
        REM PACKAGE 6       MMM10,11                                YTL16410
        REM ADD, SUBTRACT DIAGONAL MATRIX                           YTL16420
        REM                                                         YTL16430
        REM                                                         YTL16440
MMM10   CLA FIL10               ADD DIAGONAL                        YTL16450
        CAS FIL10                                                   YTL16460
MMM11   CLA FIL11               SUB DIAGONAL                        YTL16470
        STO STAL6                                                   YTL16480
        CLA     FILL5           INITIALIZE ITR86 WITH CLA 0,1       YTL16490
        STO     ITR86                                               YTL16500
        TSX     CKDM12,4        CHECK DIMENSIONS IN FIELDS 1,2      YTL16510
        TSX     CKDG2,4                                             YTL16520
        CLS     =1              LOOP CONTROL                        YTL16530
        ADD MB                                                      YTL16540
        ADD NB                                                      YTL16550
        SUB MA                                                      YTL16560
        TNZ     MMMCF           NON-CONFORMABLE                     YTL16570
        ADD MA                                                      YTL16580
        ADD LB11                                                    YTL16590
        STA STAL6                                                   YTL16600
        SUB LB11                                                    YTL16610
        ALS 18                                                      YTL16620
        STD     ITR86+3                                             YTL16630
        ARS 18                                                      YTL16640
        PAX 0,2                 NB                                  YTL16650
        SUB NA                                                      YTL16660
        TNZ     MMMCF           NON-CONFORMABLE                     YTL16670
        LDQ MA                                                      YTL16680
        STQ MC                                                      YTL16690
        STQ NC                                                      YTL16700
        TSX     CHKCOR,4                                            YTL16710
        LDQ     MA                                                  YTL16720
        MPY NA                                                      YTL16730
        XCA                                                         YTL16740
        PAX 0,1                 MNA                                 YTL16750
        ADD LA11                                                    YTL16760
```

```
        STA ITR86                                                       YTL16770
        SUB LA11                                                        YTL16780
        ADD LC11                                                        YTL16790
        STA     ITR86+5                                                 YTL16800
        AXT     1,4                                                     YTL16810
        ZET     BNULL                                                   YTL16820
        TRA     NADD1           B IS NULL                               YTL16830
        CLA     FILL7                                                   YTL16840
        ZET     ANULL                                                   YTL16850
        STO     ITR86           A IS NULL                               YTL16860
ITR86   CLA 0,1                 LA11+MNA                                YTL16870
        TIX     ITR86+5,4,1                                             YTL16880
STAL6                           LB11+NB                                 YTL16890
        TXI     ITR86+4,4,0     MA                                      YTL16900
        TNX     ITR86+5,2,1                                             YTL16910
        STO 0,1                 LC11+MNC                                YTL16920
        TIX ITR86,1,1                                                   YTL16930
        TRA MMMR                                                        YTL16940
        REM                                                             YTL16950
        REM PACKAGE 7     MMM12,13,14,15,16                             YTL16960
        REM MULTIPLY MATRIX OR TRANSPOSE PRE OR POST BY DIAGONAL        YTL16970
        REM MULTIPLY DIAGONAL BY DIAGONAL OR TWO MATRICES BY ELEMENT    YTL16980
        REM                                                             YTL16990
        REM PACKAGE 7.1   MMM12                                         YTL17000
        REM POST MULTIPLY BY DIAGONAL                                   YTL17010
MMM12   TSX     CKDM12,4        CHECK DIMENSIONS IN FIELDS 1,2          YTL17020
        TSX     CKDG2,4                                                 YTL17030
        CLA NA                  POST MPY BY DIAGONAL                    YTL17040
        STO NC                                                          YTL17050
        PAX 0,2                 NB                                      YTL17060
        ADD LB11                                                        YTL17070
        STA SCAR1+1                                                     YTL17080
        SUB LB11                                                        YTL17090
        ADD     =1                                                      YTL17100
        SUB MB                                                          YTL17110
        SUB NB                                                          YTL17120
        TNZ     MMMCF           NON-CONFORMABLE                         YTL17130
        CLA NA                                                          YTL17140
        SUB     =1                                                      YTL17150
        ALS 18                                                          YTL17160
        STD     SCAR1+5                                                 YTL17170
        LDQ MA                                                          YTL17180
        STQ MC                                                          YTL17190
        TSX     CHKCOR,4                                                YTL17200
        LDQ     MA                                                      YTL17210
        MPY NA                                                          YTL17220
        XCA                                                             YTL17230
        PAX 0,1                 MNA                                     YTL17240
        PAX     0,4                                                     YTL17250
        ADD LA11                                                        YTL17260
        STA SCAR1                                                       YTL17270
        SUB LA11                                                        YTL17280
        ADD LC11                                                        YTL17290
        ZET     ANULL                                                   YTL17300
        TRA     NMPY            A IS NULL                               YTL17310
        ZET     BNULL                                                   YTL17320
        TRA     NMPY            B IS NULL                               YTL17330
        STA     SCAR1+2                                                 YTL17340
SCAR1   LDQ 0,1                 LA11+MNA                                YTL17350
        FMP 0,2                 LB11+NB                                 YTL17360
```

```
        STO 0,1                 LC11+MNC                                YTL17370
        TNX MMMR,1,1                                                    YTL17380
        TIX SCAR1,2,1                                                   YTL17390
        TXI SCAR1,2,0           NB-1                                    YTL17400
        REM PACKAGE 7.2   MMM13                                         YTL17410
        REM PRE MULTIPLY BY DIAGONAL                                    YTL17420
MMM13   TSX     CKDM12,4        CHECK DIMENSIONS IN FIELDS 1,2          YTL17430
        TSX     CKDG1,4                                                 YTL17440
        CLA MB                  PRE MPY BY DIAGONAL                     YTL17450
        STO MC                                                          YTL17460
        PAX 0,1                 NA                                      YTL17470
        ADD LA11                                                        YTL17480
        STA SCAR2                                                       YTL17490
        SUB LA11                                                        YTL17500
        ADD     =1                                                      YTL17510
        SUB MA                                                          YTL17520
        SUB NA                                                          YTL17530
        TNZ     MMMCF           NON CONFORMABLE                         YTL17540
        CLA NB                                                          YTL17550
        STO NC                                                          YTL17560
        TSX     CHKCOR,4                                                YTL17570
        CLA     NB                                                      YTL17580
        PAX 0,4                 NB                                      YTL17590
        SUB     =1                                                      YTL17600
        ALS 18                                                          YTL17610
        STD     SCAR2+5                                                 YTL17620
        LDQ MB                                                          YTL17630
        MPY NB                                                          YTL17640
        XCA                                                             YTL17650
        PAX 0,2                 MNB                                     YTL17660
        ADD LB11                                                        YTL17670
        STA SCAR2+1                                                     YTL17680
        SUB LB11                                                        YTL17690
        ADD LC11                                                        YTL17700
        ZET     ANULL                                                   YTL17710
        TRA     NMPY1           A IS NULL                               YTL17720
        ZET     BNULL                                                   YTL17730
        TRA     NMPY1           B IS NULL                               YTL17740
        STA     SCAR2+2                                                 YTL17750
SCAR2   LDQ 0,1                 LA11+NA                                 YTL17760
        FMP 0,2                 LB11+MNB                                YTL17770
        STO 0,2                 LC11+MNC                                YTL17780
        TNX MMMR,2,1                                                    YTL17790
        TIX SCAR2,4,1                                                   YTL17800
        TXI     SCAR2+6,4,0     NB-1                                    YTL17810
        TIX SCAR2,1,1                                                   YTL17820
        REM PACKAGE 7.3   MMM14                                         YTL17830
        REM POST MULTIPLY TRANSPOSE BY DIAGONAL                         YTL17840
MMM14   TSX     CKDM12,4        CHECK DIMENSIONS IN FIELDS 1,2          YTL17850
        TSX     CKDG2,4                                                 YTL17860
        CLA MA                  POST MPY TRANSPOSE BY DIAGONAL          YTL17870
        STO NC                                                          YTL17880
        ADD LB11                                                        YTL17890
        STA SCAR3+2                                                     YTL17900
        SUB LB11                                                        YTL17910
        ADD     =1                                                      YTL17920
        SUB MB                                                          YTL17930
        SUB NB                                                          YTL17940
        TNZ     MMMCF           NON-CONFORMABLE                         YTL17950
        CLA NA                                                          YTL17960
```

```
        STO MC                                                          YTL17970
        ALS 18                                                          YTL17980
        STD     SCAR3+5                                                 YTL17990
        ADD     =1B17                                                   YTL18000
        STD PACH3                                                       YTL18010
        TSX     CHKCOR,4                                                YTL18020
        LDQ MA                                                          YTL18030
        MPY NA                                                          YTL18040
        XCA                                                             YTL18050
        PAX     0,1                                                     YTL18060
        PAX     0,4                                                     YTL18070
        ADD LA11                                                        YTL18080
        STA SCAR3+1                                                     YTL18090
        SUB LA11                                                        YTL18100
        PAX 0,2                                                         YTL18110
        ADD LC11                                                        YTL18120
        ZET     ANULL                                                   YTL18130
        TRA     NMPY            A IS NULL                               YTL18140
        ZET     BNULL                                                   YTL18150
        TRA     NMPY            B IS NULL                               YTL18160
        STA     SCAR3+3                                                 YTL18170
        TXI *+1,2,2                                                     YTL18180
PACH3   TIX *+1,2,**    **=NA+1                                         YTL18190
        TXI *+1,2,-2                                                    YTL18200
        SXD     SCAR3+7,2                                               YTL18210
SCAR3   LXA NC,2            NB                                          YTL18220
        LDQ 0,1                 LA11+NMA                                YTL18230
        FMP 0,2                 LB11+NB                                 YTL18240
        STO 0,4                 LC11+MNC                                YTL18250
        TNX MMMR,4,1                                                    YTL18260
        TNX     SCAR3+6,1,0     NA                                      YTL18270
        TIX SCAR3+1,2,1                                                 YTL18280
        TXI SCAR3,1,0           NMA-NA-1                                YTL18290
        REM PACKAGE 7.4   MMM15                                         YTL18300
        REM PRE MULTIPLY TRANSPOSE BY DIAGONAL                          YTL18310
MMM15   TSX     CKDM12,4        CHECK DIMENSIONS IN FIELDS 1,2          YTL18320
        TSX     CKDG1,4                                                 YTL18330
        CLA NB                  PRE MPY TRANSPOSE BY DIAGONAL           YTL18340
        STO MC                                                          YTL18350
        PAX 0,1                 NB                                      YTL18360
        SXD     SCAR4+4,1                                               YTL18370
        ADD LA11                                                        YTL18380
        STA SCAR4                                                       YTL18390
        SUB LA11                                                        YTL18400
        ADD     =1                                                      YTL18410
        PAX 0,2             NB+1                                        YTL18420
        SXD PACH4,2                                                     YTL18430
        SUB MA                                                          YTL18440
        SUB NA                                                          YTL18450
        TNZ     MMMCF           NON-CONFORMABLE                         YTL18460
        LDQ MB                                                          YTL18470
        STQ NC                                                          YTL18480
        TSX     CHKCOR,4                                                YTL18490
        LDQ     MB                                                      YTL18500
        MPY NB                                                          YTL18510
        XCA                                                             YTL18520
        PAX 0,2             NMB                                         YTL18530
        TXI *+1,2,2                                                     YTL18540
PACH4   TIX *+1,2,**    **=NB+1                                         YTL18550
        TXI *+1,2,-2                                                    YTL18560
```

```
        SXD     SCAR4+5,2                                                YTL18570
        PAX     0,4             NMC                                      YTL18580
        PAX     0,2             NMB                                      YTL18590
        ADD LB11                                                         YTL18600
        STA SCAR4+1                                                      YTL18610
        SUB LB11                                                         YTL18620
        ADD LC11                                                         YTL18630
        ZET     ANULL                                                    YTL18640
        TRA     NMPY            A IS NULL                                YTL18650
        ZET     BNULL                                                    YTL18660
        TRA     NMPY            B IS NULL                                YTL18670
        STA     SCAR4+2                                                  YTL18680
SCAR4   LDQ 0,1                 LA11+NA                                  YTL18690
        FMP 0,2                 LB11+NMB                                 YTL18700
        STO 0,4                 LC11+MNC                                 YTL18710
        TNX MMMR,4,1                                                     YTL18720
        TIX SCAR4,2,0           NB                                       YTL18730
        TXI     SCAR4+6,2,0     NMB-NB-1                                 YTL18740
        TIX SCAR4,1,1                                                    YTL18750
        REM PACKAGE 7.5   MMM16                                          YTL18760
        REM MULTIPLY DIAGONAL BY DIAGONAL, OR TWO MATRICES BY ELEMENT    YTL18770
MMM16   TSX     CKDM12,4        CHECK DIMENSIONS IN FIELDS 1,2           YTL18780
        TSX     CKDG1,4                                                  YTL18790
        TSX     CKDG2,4                                                  YTL18800
        CLA MA                  MPY DIAGONAL BY DIAGONAL                 YTL18810
        SUB MB                                                           YTL18820
        TNZ XCHK1                                                        YTL18830
        CLA NA                                                           YTL18840
        SUB NB                                                           YTL18850
        TNZ     MMMCF           NON-CONFORMABLE                          YTL18860
        TRA XCHK2                                                        YTL18870
XCHK1   CLA MA                                                           YTL18880
        SUB NB                                                           YTL18890
        TNZ     MMMCF           NON-CONFORMABLE                          YTL18900
        CLA NA                                                           YTL18910
        SUB MB                                                           YTL18920
        TNZ     MMMCF           NON-CONFORMABLE                          YTL18930
XCHK2   CLA     =1                                                       YTL18940
        STO MC                                                           YTL18950
        LDQ MA                                                           YTL18960
        MPY NA                                                           YTL18970
        XCA                                                              YTL18980
        STO NC                                                           YTL18990
        TSX     CHKCOR,4                                                 YTL19000
        CLA     NC                                                       YTL19010
        PAX 0,1                 NA,B,C                                   YTL19020
        PAX     0,4                                                      YTL19030
        ADD LA11                                                         YTL19040
        STA ITR87                                                        YTL19050
        SUB LA11                                                         YTL19060
        ADD LB11                                                         YTL19070
        STA ITR87+1                                                      YTL19080
        SUB LB11                                                         YTL19090
        ADD LC11                                                         YTL19100
        ZET     ANULL                                                    YTL19110
        TRA     NMPY            A IS NULL                                YTL19120
        ZET     BNULL                                                    YTL19130
        TRA     NMPY            B IS NULL                                YTL19140
        STA     ITR87+2                                                  YTL19150
ITR87   LDQ 0,1                 LA11+NA                                  YTL19160
```

```
        FMP 0,1                 LB11+NB                                 YTL19170
        STO 0,1                 LC11+NC                                 YTL19180
        TIX ITR87,1,1                                                   YTL19190
        TRA MMMR                                                        YTL19200
        REM                                                             YTL19210
        REM PACKAGE 8     MMM17                                         YTL19220
        REM ADD CONSTANT TIMES UNIT MATRIX                              YTL19230
        REM                                                             YTL19240
MMM17   CLA MA                  ADD KI                                  YTL19250
        SUB NA                                                          YTL19260
        TZE     *+3                                                     YTL19270
        CLA     =2              MATRIX NOT SQUARE                       YTL19280
        TRA     MMME                                                    YTL19290
        ZET     ANULL                                                   YTL19300
        STZ*    LA11                                                    YTL19310
        CLA MA                  LOOP CONTROL                            YTL19320
        STO MC                                                          YTL19330
        STO NC                                                          YTL19340
        ALS 18                                                          YTL19350
        STD     ITR88+3                                                 YTL19360
        LDQ MA                                                          YTL19370
        MPY NA                                                          YTL19380
        XCA                                                             YTL19390
        PAX     0,1                                                     YTL19400
        ADD LA11                                                        YTL19410
        STA ITR88                                                       YTL19420
        SUB LA11                                                        YTL19430
        ADD LC11                                                        YTL19440
        STA     ITR88+4                                                 YTL19450
        CLA 2,4                                                         YTL19460
        STA ITR88+2                                                     YTL19470
        TSX     CKDM1,4         CHECK DIMENSIONS IN FIELD 1             YTL19480
        TSX     CHKCOR,4                                                YTL19490
        CLA     =1                                                      YTL19500
        PAX 0,2                 ONE                                     YTL19510
ITR88   CLA 0=1                 TA11+MNA                                YTL19520
        TIX     ITR88+4,2,1                                             YTL19530
        FAD 0                   LOC B                                   YTL19540
        TXI     ITR88+4,2,0     MA                                      YTL19550
        STO 0,1                 LC11+MNC                                YTL19560
        TIX ITR8 ,1,1                                                   YTL19570
        CLA     LA11                                                    YTL19580
        SUB     LC11                                                    YTL19590
        TZE     MMMR                                                    YTL19600
        CLA     WDMNL                                                   YTL19610
        ZET     ANULL                                                   YTL19620
        STO*    LA11                                                    YTL19630
        TRA MMMR                                                        YTL19640
*                                                                       YTL19650
*               PACKAGE 9,  MMM18  MATRIX INVERSION                     YTL19660
*               USES INV4,  7094 INVERSION WHICH UTILIZES               YTL19670
*               THE HARDWARE DOUBLE PRECISION                           YTL19680
*                                                                       YTL19690
*               FIELD 2 OPTION DESCRIPTION                              YTL19700
*                                                                       YTL19710
*               FIELD 2 = -1, PRINT DETERMINANT                         YTL19720
*               FIELD 2 = -2 , HAVE CONDITIONING INFO. PRINTED          YTL19730
*               FIELD 2 = -3 , DO BOTH                                  YTL19740
*                                                                       YTL19750
*                                                                       YTL19760
```

```
MMM18 TSX     CKDM1,4       CHECK DIMENSIONS                                YTL19770
      CLA     MA                                                            YTL19780
      SUB     NA                                                            YTL19790
      TZE     *+3                                                           YTL19800
      CLA     =2            NOT SQUARE                                      YTL19810
      TRA     MMME                                                          YTL19820
      CLA     ANULL                                                         YTL19830
      TZE     NOTNUL                                                        YTL19840
      CLA     =11                                                           YTL19850
      TRA     MMME                                                          YTL19860
NOTNUL CLA    NA                                                            YTL19870
      STO     MC                                                            YTL19880
      STO     NC                                                            YTL19890
      LDQ     NA                                                            YTL19900
      MPY     NA                                                            YTL19910
      STQ     NSQ                                                           YTL19920
      RQL     1                                                             YTL19930
      STQ     TWONSQ        2NSQ                                            YTL19940
      CLA     TWONSQ                                                        YTL19950
      ADD     NA                                                            YTL19960
      ADD     NA                                                            YTL19970
      ADD     =6                                                            YTL19980
      ADD     LC11          LC11+2*N**2+2*N+6                               YTL19990
      CAS     HICORE                                                        YTL20000
      TRA     OVER                                                          YTL20010
      TRA     *+1                                                           YTL20020
      CLA     LC11                                                          YTL20030
      LBT                                                                   YTL20040
      TRA     *+2           MAKE SURE ADDRESS IS EVEN                       YTL20050
      ADD     =1                                                            YTL20060
      STA     GO+4                                                          YTL20070
      CLA     LA11                                                          YTL20080
      ADD     NSQ                                                           YTL20090
      STA     GET1                                                          YTL20100
      CLA     GO+4                                                          YTL20110
      ADD     TWONSQ                                                        YTL20120
      STA     STO1                                                          YTL20130
      CLA     LC11                                                          YTL20140
      CAS     LA11                                                          YTL20150
      TRA     TOPFST        IF INVERTED FORWARDS, MOVE FROM TOP DOWN        YTL20160
      TRA     TOPFST        IF INVERTED OVER ITSELF, MOVE FROM TOP DOWN     YTL20170
BOTFST LXA    NSQ,1         IF INVERTED BACKWARDS,MOVE  FROM BOTTOM FST     YTL20180
      LXA     TWONSQ,2                                                      YTL20190
  GET1 CLA    **,1          LA11 + NSQ                                      YTL20200
  STO1 STO    **,2          LC11(EVEN) + 2*N**2                             YTL20210
      TIX     *+1,2,1                                                       YTL20220
      STZ*    *-2                                                           YTL20230
      TIX     *+1,2,1                                                       YTL20240
      TIX     GET1,1,1                                                      YTL20250
      TRA     INVERT                                                        YTL20260
TOPFST LXA    NSQ,1         EXPAND THIS WAY FOR INVERSION OVER ITSELF       YTL20270
      SXD     LPEND,1                                                       YTL20280
      AXT     1,1                                                           YTL20290
      AXT     1,2                                                           YTL20300
  GET2 CLA*   GET1                                                          YTL20310
      STZ*    STO1                                                          YTL20320
      TXI     *+1,2,1                                                       YTL20330
      STO*    STO1                                                          YTL20340
      TXI     *+1,2,1                                                       YTL20350
      TXI     *+1,1,1                                                       YTL20360
```

```
 LPEND TXL     GET2,1,**       N**2 IN DECREMENT                                YTL20370
INVERT CLA     YMBB            CHECK FOR OPTIONAL DIAGONAL TERM PRINT           YTL20380
       TZE     GO                                                               YTL20390
       TMI     *+3                                                              YTL20400
       CLA     =35             FIELD MUST BE NEGATIVE                           YTL20410
       TRA     MMME                                                             YTL20420
       PAX     0,7                                                              YTL20430
       TXH     *-3,7,3         ONLY 1,2, OR 3                                   YTL20440
       TXL     GO+1,7,1        1 IS JUST DETERMINANT                            YTL20450
       CLA     NA                                                               YTL20460
       SSM                                                                      YTL20470
       STO     NA                                                               YTL20480
       TRA     *+2                                                              YTL20490
    GO LXA     =0,7            CLEAR XR7 IF FIELD 2=0                           YTL20500
       CALL    INV4DS(**,NA,IRR1,IRR2,SCALE,DET,NDETXP)                         YTL20510
       ZAC                                                                      YTL20520
       STP     NA                                                               YTL20530
       CLA     IRR1                                                             YTL20540
       TZE     GOOD                                                             YTL20550
INVE   CALL    .FWRD.(.UNO6.,INVFMT)                ERROR PRINT                 YTL20560
       CLA     IRR1                                                             YTL20570
       TSX     .FCNV.,4                                                         YTL20580
       CLA     IRR2                                                             YTL20590
       TSX     .FCNV.,4                                                         YTL20600
       CLA     SCALE                                                            YTL20610
       TSX     .FCNV.,4                                                         YTL20620
       CALL    .FFIL.                                                           YTL20630
  GOOD LXA     NSQ,1                                                            YTL20640
       LXA     TWONSQ,2                                                         YTL20650
       CLA     LC11                                                             YTL20660
       ADD     NSQ                                                              YTL20670
       STA     STO2                                                             YTL20680
 SQEZE CLA*    STO1                                                             YTL20690
       TIX     *+1,2,1                                                          YTL20700
       LDQ*    STO1                                                             YTL20710
       FRN                                                                      YTL20720
 STO2  STO     **,1                                                             YTL20730
       TIX     *+1,2,1                                                          YTL20740
       TIX     SQEZE,1,1                                                        YTL20750
       CLA     IRR1                                                             YTL20760
       TZE     *+3                                                              YTL20770
       CLA     =32                                                              YTL20780
       TRA     MMME                                                             YTL20790
       PXA     0,7             CHECK OPTION FOR DET PRINT                       YTL20800
       TZE     MMMR                                                             YTL20810
       TRA     *+4,7                                                            YTL20820
       TRA     PNTDET          FIELD 2 = -3                                     YTL20830
       TRA     MMMR            FIELD 2 = -2                                     YTL20840
PNTDET CALL    .FWRD.(.UNO6.,DETPNT)                                            YTL20850
       CLA     MA                                                               YTL20860
       TSX     .FCNV.,4                                                         YTL20870
       CLA     NA                                                               YTL20880
       TSX     .FCNV.,4                                                         YTL20890
       CLA*    YMAA                                                             YTL20900
       TSX     .FCNV.,4                                                         YTL20910
       CLA     DET                                                              YTL20920
       TSX     .FCNV.,4                                                         YTL20930
       CLA     NDETXP                                                           YTL20940
       TSX     .FCNV.,4                                                         YTL20950
       CALL    .FFIL.                                                           YTL20960
```

```
        TRA     MMMR                                                    YTL20970
*                                                                       YTL20980
*                                                                       YTL20990
*                                                                       YTL21000
MMM19   ZET     2,4                                                     YTL21010
        TRA     TRY3                                                    YTL21020
        NZT     3,4                                                     YTL21030
        TRA     RDBCD                                                   YTL21040
        CLA     =16             FIELD 2 OR FIELD 3 =0                   YTL21050
        TRA     MMME                                                    YTL21060
  TRY3  ZET     3,4                                                     YTL21070
        TRA     RDDECM                                                  YTL21080
        TRA     *-4                                                     YTL21090
RDBCD   CLA     4,4             FIELDS 2 AND 3 ARE ZERO                 YTL21100
        TZE     RDBCD1          READ 1 BCD CARD                         YTL21110
        STO     TAP1                                                    YTL21120
        CALL    .FVIO.(TAP1,TAPIB)                                      YTL21130
RDBCD1  CALL    .FRDD.(TAPIB,TLEIN)                                     YTL21140
        CLA     YMAA                                                    YTL21150
        STA     IN+3                                                    YTL21160
 IN     CALL    .FSLI.(**,=14)                                          YTL21170
        CALL    .FRTN.                                                  YTL21180
        TRA     DONE                                                    YTL21190
RDDECM  CLA     4,4                                                     YTL21200
        TNZ     *+2                                                     YTL21210
        CLA     TAP             CURRENT TAPE                            YTL21220
        STO     READ+7                                                  YTL21230
        SUB     =5                                                      YTL21240
        TZE     GOODTP          TAPE 5 IS PERMITTED                     YTL21250
        CLA     READ+7                                                  YTL21260
        STO     TAPE                                                    YTL21270
        TSX     TPCK,4          CHECK TAPE NO.                          YTL21280
        LXD     SPOT4,4                                                 YTL21290
GOODTP  CLA     3,4                                                     YTL21300
        STO     READ+5                                                  YTL21310
        CLA     2,4                                                     YTL21320
        STO     READ+4                                                  YTL21330
        CLA     1,4                                                     YTL21340
        STO     READ1+3                                                 YTL21350
READ1   CALL    KRD(**,=1,IRROR,KRDP4,KRDP5,=0,KRDP7)                   YTL21360
        STO     TEMP            CARD COUNT                              YTL21370
        CLA     IRROR                                                   YTL21380
        TZE     CK19NL-2                                                YTL21390
        XCA                                                             YTL21400
        MPY     THOUSN                                                  YTL21410
        XCA                                                             YTL21420
        ADD     TEMP                                                    YTL21430
        ADD     MILYN                                                   YTL21440
        TRA     MMME                                                    YTL21450
        NZT*    READ1+3                                                 YTL21460
        STZ*    READ1+3                                                 YTL21470
CK19NL  CLA*    LA11                                                    YTL21480
        SUB     WDMNL           ARE WE READING IN TO A NULL MATRIX      YTL21490
        TNZ     DONE            IF NOT, ALLDONE                         YTL21500
        LDQ*    MMMA                                                    YTL21510
        MPY*    MMNA                                                    YTL21520
        XCA                                                             YTL21530
        SUB     =1              M*N-1                                   YTL21540
        PAX     0,1                                                     YTL21550
        ADD     =1                                                      YTL21560
```

```
        ADD     LA11            LA11+M*N                                        YTL21570
        STA     *+1                                                             YTL21580
        CLA     **,1                                                            YTL21590
        TNZ     CLRWRD          IF NON-ZERO ELEMENT, CLEAR CODE WORD            YTL21600
        TIX     *-2,1,1                                                         YTL21610
        TRA     DONE                                                            YTL21620
CLRWRD  STZ*    LA11            CLEAR CODE WORD                                 YTL21630
        TRA     DONE                                                            YTL21640
        REM                                                                     YTL21650
*               MMM20 - OUTPUT ROUTINE FOR YTLO1                                YTL21660
*               CONVERTED TO FORTRAN METHOD IN JUNE 1962                        YTL21670
*               NEEDS (STH), (FIL), (TSH), (RTN)                                YTL21680
*               INSTRUCTION CARD IS,  FIELD 1 = LOCATION OF MATRIX              YTL21690
*                                     = 0 IF COMMENT IS TO BE                   YTL21700
*                                       PRINTED ON LINE - WORKS                 YTL21710
*                                       WITH CORE OR TAPE COMMENT               YTL21720
*                                 FIELD 2 = 0, FOR NEW PAGE                     YTL21730
*                                         = 1 FOR NO SPACING                    YTL21740
*                                 FIELD 3 = 0 TO PRINT MATRIX,CHKSUM            YTL21750
*                                         = 1 FOR CHECKSUM ONLY                 YTL21760
*                                 FIELD 4 = 0, IF MATRIX NO. WANTED             YTL21770
*                                         = K, IF 1 FULL TITLE CARD             YTL21780
*                                           TO BE READ FROM TAPE K              YTL21790
*                                         = 8000 OR GREATER THE                 YTL21800
*                                           COMMENT COMES FROM CORE             YTL21810
*                                           THIS IS CONNECTED WITH              YTL21820
*                                           NEW OPTION IN OP. CODE 19           YTL21830
*                                 FIELD 5 = OP. CODE 20                         YTL21840
*               MMM23A-ENTRY POINT TO ALLOW MMM23 TO UTILIZE THE                YTL21850
*                      MATRIX IDENTIFICATION SECTION OF MMM20                   YTL21860
*               FIRST PAGE HEADING AND NEW PAGE PRINT ARE                       YTL21870
*               USED BY MMM23 ALSO                                              YTL21880
MMM23A  CLA     =1              MAP ROUTINE WAS CALLED                          YTL21890
        STO     MAP                                                             YTL21900
        TRA     MMM20+1                                                         YTL21910
 MMM20  STZ     MAP                                                             YTL21920
        NZT     1,4             IS THIS AN ON LINE COMMENT                      YTL21930
        TRA     NSPACE          YES, DO NOT CHECK DIMENSIONS                    YTL21940
        CLA     MA                                                              YTL21950
        ADD     NA                                                              YTL21960
        TZE     *+2                                                             YTL21970
        TSX     CKDM1,4         CHECK DIMENSIONS IN FIELD 1                     YTL21980
        CLA     YMBB                                                            YTL21990
        STO     SPACE           IS FIELD 2 BLANK                                YTL22000
        TNZ     NSPACE                                                          YTL22010
        CALL    .FWRD.(.UNO6.,NPAGE)                                            YTL22020
        CALL    .FFIL.                                                          YTL22030
NSPACE  CLA     YMNC                                                            YTL22040
        STO     TTLE                                                            YTL22050
        TZE     PNTNO           IS FIELD 4 BLANK                                YTL22060
        CAS     =22             IS IT A TAPE NO.                                YTL22070
        TRA     *+3             NO                                              YTL22080
        TRA     TPTLE           YES                                             YTL22090
        TRA     TPTLE           YES                                             YTL22100
        AXT     14,1            CORE TITLE, COPY INTO TITLE                     YTL22110
        ADD     =14                                                             YTL22120
        ADD     YSHIFT                                                          YTL22130
        STA     *+1                                                             YTL22140
        CLA     **,1                                                            YTL22150
        STO     TITLE+14,1                                                      YTL22160
```

```
        TIX     *-2,1,1                                                 YTL22170
        TRA     PNTTLE                                                  YTL22180
TPTLE   CALL    .FVIO.(YMNC,TAP1IB)                                     YTL22190
        CALL    .FRDD.(TAP1IB,TLEIN)                                    YTL22200
        CALL    .FSLI.(TITLE,=14)                                       YTL22210
        CALL    .FRTN.                                                  YTL22220
PNTTLE  CLA     YMAA            CHECK FOR ON LINE PRINT                 YTL22230
        TNZ     WRTCMT                                                  YTL22240
        CALL    .FPRN.(TLEOUT)                                          YTL22250
        CALL    .FSLO.(TITLE,=14)                                       YTL22260
        CALL    .FFIL.                                                  YTL22270
        TRA     DONE                                                    YTL22280
WRTCMT  CALL    .FWRD.(.UN06.,TLEOUT)                                   YTL22290
        CALL    .FSLO.(TITLE,=14)                                       YTL22300
        CALL    .FFIL.                                                  YTL22310
        CLA     =14             LINE 1 OF ROW 1 STARTS ON LINE 14       YTL22320
        STO     LINE                                                    YTL22330
        TRA     PNTNO+2                                                 YTL22340
 PNTNO  CLA     =12             LINE 1 OF ROW 1 STARTS ON LINE 12       YTL22350
        STO     LINE                                                    YTL22360
        NZT     YMAA                                                    YTL22370
        TRA     NOTITL                                                  YTL22380
        CLA     MA                                                      YTL22390
        ADD     NA                                                      YTL22400
        TNZ     PRNTFL          GO AHEAD AND PRINT IF M OR N IS NON-ZERO YTL22410
        ZET     TTLE            DIMENSIONS ARE ZERO, WAS A TITLE PRINTED YTL22420
        TRA     DONE            YES, 0 DIMENSIONS HERE CAUSE AN EXIT    YTL22430
NOTITL  CLA     =7                                                      YTL22440
        TRA     MMME            NO,THEY ARE TRYING TO PRINT A 0 BY 0 MATRIXYTL22450
PRNTFL  CALL    .FWRD.(.UN06.,MATNO)                                    YTL22460
        CLA*    YMAA                                                    YTL22470
        T&X     .FCNV.,4                                                YTL22480
        CALL    .FFIL.                                                  YTL22490
  CKSM  LDQ     MA                                                      YTL22500
        MPY     NA                                                      YTL22510
        XCA                                                             YTL22520
        PAX     0,2             LOAD INDEX FOR FORMAT CHECK             YTL22530
        ADD     =3                                                      YTL22540
        PAX     0,1                                                     YTL22550
        ADD     YMAA                                                    YTL22560
        STA     *+2                                                     YTL22570
        PXA     0,0                                                     YTL22580
        ACL     **,1            COMPUTE CHECK SUM OF MATRIX             YTL22590
        TIX     *-1,1,1                                                 YTL22600
        SLW     CHKSUM                                                  YTL22610
        CALL    .FWRD.(.UN06.,ORDSUM)                                   YTL22620
        CLA     MA                                                      YTL22630
        TSX     .FCNV.,4                                                YTL22640
        CLA     NA                                                      YTL22650
        TSX     .FCNV.,4                                                YTL22660
        CLA     CHKSUM                                                  YTL22670
        TSX     .FCNV.,4                                                YTL22680
        CALL    .FFIL.                                                  YTL22690
        CLA     ANULL                                                   YTL22700
        TZE     ANN20                                                   YTL22710
        CALL    .FWRD.(.UN06.,NULMAT)                                   YTL22720
        CALL    .FFIL.                                                  YTL22730
        NZT     MAP                                                     YTL22740
        TRA     OUT20           NOT MAP                                 YTL22750
ANN20   ZET     MAP             WAS MAPPING CALLED                      YTL22760
```

```
        TRA     MMM23           YES                                     YTL22770
        ZET     YMCC            IS ONLY CHECK SUM WANTED                YTL22780
        TRA     CSMOLY          YES                                     YTL22790
        PXA     0,2             NO                                      YTL22800
        ADD     LA11                                                    YTL22810
        STA     FMTCK                                                   YTL22820
        AXT     0,1                                                     YTL22830
FMTCK   CLA     **,2              EXAMINE MATRIX TO FIND 1              YTL22840
        TNZ     *+2                                                     YTL22850
        TXI     *+3,1,1                                                 YTL22860
        ANA     =0377400000000  FLOATING POINT ELEMENT                  YTL22870
        TNZ     FLPPT             IF FL. PT , USE E FORMAT              YTL22880
        TIX     FMTCK,2,1                                               YTL22890
        LDQ     MA                                                      YTL22900
        MPY     NA                                                      YTL22910
        XCA                                                             YTL22920
        STZ     CHKSUM                                                  YTL22930
        SXA     CHKSUM,1                                                YTL22940
        SUB     CHKSUM                                                  YTL22950
        TZE     FLPPT                                                   YTL22960
        STZ     WHCHFT          0 IF INTEGER FORMAT                     YTL22970
        TRA     FMTDEF                                                  YTL22980
FLPPT   CLA     =1                                                      YTL22990
        STO     WHCHFT          =1 IF FLOATING POINT FORMAT             YTL23000
FMTDEF  CLA     =1                                                      YTL23010
        STO     I               COUNT OF ROWS                           YTL23020
        CLA     LA11                                                    YTL23030
        STA     PRNT                                                    YTL23040
        STZ     CHKSUM          ZERO FOR NEW PAGE TRANSFER              YTL23050
LOOP    CLA     NA                                                      YTL23060
        PAX     0,2             COLUMN COUNT                            YTL23070
        AXT     7,1             START COUNT OF ELEMENTS ON ONE LINE     YTL23080
        ADD     PRNT            INCREMENT PRINT ADDRESS BY 1 ROW        YTL23090
        STA     PRNT                                                    YTL23100
PRNT1   CLA     WHCHFT                                                  YTL23110
        TZE     FIXPRT                                                  YTL23120
        CALL    .FWRD.(.UNO6.,FLPPNT)                                   YTL23130
        TRA     PRNT-2                                                  YTL23140
FIXPRT  CALL    .FWRD.(.UNO6.,INTPNT)                                   YTL23150
        CLA     I               ROW  INDEX                              YTL23160
        TSX     .FCNV.,4                                                YTL23170
PRNT    CLA     **,2                                                    YTL23180
        TSX     .FCNV.,4                                                YTL23190
        TIX     COLUMN,2,1      IS THIS ROW FINISHED                    YTL23200
        CALL    .FFIL.                                                  YTL23210
        CLA     I               YES, ARE ALL ROW FINISHED               YTL23220
        CAS     MA                                                      YTL23230
        TRA     OUT20                                                   YTL23240
        TRA     OUT20                                                   YTL23250
        ADD     =1              MORE ROWS LEFT                          YTL23260
        STO     I                                                       YTL23270
        CLS     LINE            FIND OUT IF THERE ARE AT LEAST          YTL23280
        SUB     =1              3 LINES AVAILABLE FOR THIS NEW ROW      YTL23290
        ADD     =54                                                     YTL23300
        LDQ     =3                                                      YTL23310
        TLQ     KEPPNT                                                  YTL23320
        XCA                     3 OR LESS LINES AVAILABLE = K           YTL23330
        MPY     =7                                                      YTL23340
        XCA                                                             YTL23350
        CAS     NA              IS K SUFFICIENT TO HOLD 1 ROW           YTL23360
```

```
         TRA     KEPPNT                                                  YTL23370
         TRA     KEPPNT                                                  YTL23380
         CLA     =1                                                      YTL23390
         STO     CHKSUM                                                  YTL23400
         TRA     NEWPGE          NOT ENOUGH ,START NEW PAGE              YTL23410
COLUMN   TIX     PRNT,1,1        STILL PRINTING ROW I, SAME LINE         YTL23420
         AXT     7,1             START COUNT OF ELEMENTS IN A LINE       YTL23430
         CLA     SPACE           ARE WE SUPPRESSING NEW PAGES            YTL23440
         TNZ     PRNT            IGNORE LINE COUNT IF SPACE NOT ZERO     YTL23450
         CLA     LINE            START NEW LINE                          YTL23460
         ADD     =1                                                      YTL23470
         STO     LINE                                                    YTL23480
         STZ     CHKSUM          ZERO CONTROL FOR NEW PAGE TRANSFER      YTL23490
         CLA     LINE                                                    YTL23500
         CAS     =54             ARE WE ABOUT TO PRINT LINE 55           YTL23510
         TRA     CLOSIT                                                  YTL23520
         TRA     PRNT                                                    YTL23530
         TRA     PRNT                                                    YTL23540
CLOSIT   CALL    .FFIL.                                                  YTL23550
NEWPGE   CLA     TTLE            WHICH TITLE TO PRINT                    YTL23560
         TNZ     CDTLE                                                   YTL23570
         CALL    .FWRD.(.UN06.,NWPGE1)                                   YTL23580
         CLA*    YMAA                                                    YTL23590
         TSX     .FCNV.,4                                                YTL23600
         CALL    .FFIL.                                                  YTL23610
         TRA     BACK                                                    YTL23620
CDTLE    CALL    .FWRD.(.UN06.,NWPGE2)                                   YTL23630
         CALL    .FSLO.(TITLE,=14)                                       YTL23640
         CALL    .FFIL.                                                  YTL23650
  BACK   CLA     =8              RESET LINE COUNT                        YTL23660
         STO     LINE                                                    YTL23670
         PXA     0,0                                                     YTL23680
         CAS     CHKSUM                                                  YTL23690
         TRA     STPGE           RETURN TO MMM23                         YTL23700
         TRA     PRNT1           PRINT ROW INDEX AT START OF PAGE        YTL23710
         TRA     LOOP            NEW ROW AT START OF PAGE                YTL23720
KEPPNT   CLA     SPACE           DO NOT INCREMENT COUNT IF SPACING IGNORED YTL23730
         TNZ     LOOP                                                    YTL23740
         CLA     LINE            START NEW ROW ON CURRENT PAGE           YTL23750
         ADD     =2                                                      YTL23760
         STO     LINE                                                    YTL23770
         TRA     LOOP                                                    YTL23780
CSMOLY   CALL    .FWRD.(.UN06.,CSMPT)                                    YTL23790
         CALL    .FFIL.                                                  YTL23800
         TRA     DONE                                                    YTL23810
OUT20    CALL    .FWRD.(.UN06.,FINAL)                                    YTL23820
         CALL    .FFIL.                                                  YTL23830
         TRA     DONE                                                    YTL23840
*                PACKAGE 12 MMM21                                        YTL23850
*                DIAGONAL INVERSE                                        YTL23860
 MMM21   TSX     CKDM1,4                                                 YTL23870
         TSX     CKDG1,4                                                 YTL23880
         CLA     ANULL                                                   YTL23890
         TZE     *+3                                                     YTL23900
         CLA     =11             NULL MATRIX WILL NOT INVERT             YTL23910
         TRA     MMME                                                    YTL23920
         LDQ     MA                                                      YTL23930
         STQ     MC                                                      YTL23940
         MPY     NA                                                      YTL23950
         XCA                                                             YTL23960
```

```
        PAX     0,1                                                     YTL23970
        ADD     LA11                                                    YTL23980
        STA     CORE+1                                                  YTL23990
        SUB     LA11                                                    YTL24000
        ADD     LC11                                                    YTL24010
        STA     CORE+2                                                  YTL24020
        CLA     NA                                                      YTL24030
        STO     NC                                                      YTL24040
        TSX     CHKCOR,4                                                YTL24050
  CORE  CLA     =1.0                                                    YTL24060
        FDP     **,1                                                    YTL24070
        STQ     **,1                                                    YTL24080
        TIX     CORE,1,1                                                YTL24090
        DCT                                                             YTL24100
        TRA     *+2                                                     YTL24110
        TRA     MMMR                                                    YTL24120
        CLA     =10                                                     YTL24130
        TRA     MMME                                                    YTL24140
*               PACKAGE 13 MMM22                                        YTL24150
*               FORM P1 MATRIX                                          YTL24160
 MMM22  TSX     CKDM12,4        CHECK DIMENSIONS IN FIELDS 1,2          YTL24170
        ZET     ANULL                                                   YTL24180
        STZ*    LA11            A IS NULL                               YTL24190
        ZET     BNULL                                                   YTL24200
        STZ*    LB11            B IS NULL                               YTL24210
        CLA     =25                                                     YTL24220
        STO     NC                                                      YTL24230
        LDQ     MA                                                      YTL24240
        MPY     NA                                                      YTL24250
        XCA                                                             YTL24260
        STO     MC                                                      YTL24270
        TSX     CHKCOR,4                                                YTL24280
        CLA     MC                                                      YTL24290
        PAX     0,4                                                     YTL24300
        ADD     LA11                                                    YTL24310
        STA     INNR+6                                                  YTL24320
        SUB     LA11                                                    YTL24330
        ADD     LB11                                                    YTL24340
        STA     INNR+2                                                  YTL24350
        CLA     LC11                                                    YTL24360
        STA     INNR                                                    YTL24370
        LDQ     NB                                                      YTL24380
        MPY     MB                                                      YTL24390
        XCA                                                             YTL24400
        SUB     MC                                                      YTL24410
        TZE     OUTR                                                    YTL24420
        TRA     MMMCF                                                   YTL24430
  OUTR  AXT     25,2                                                    YTL24440
        CAL     INNR                                                    YTL24450
        ACL     =25                                                     YTL24460
        SLW     INNR                                                    YTL24470
        CLA     =1.0                                                    YTL24480
  MIDL  AXT     5,1                                                     YTL24490
        STO     CELL1                                                   YTL24500
  INNR  STO     **,2                                                    YTL24510
        XCA                                                             YTL24520
        FMP     **,4                                                    YTL24530
        TXI     *+1,2,-1                                                YTL24540
        TIX     INNR,1,1                                                YTL24550
        LDQ     CELL1                                                   YTL24560
```

```
        FMP     **,4                                                    YTL24570
        TIX     MIDL,2,0                                                YTL24580
        TIX     OUTR,4,1                                                YTL24590
        CLA     WDMNL                                                   YTL24600
        ZET     ANULL                                                   YTL24610
        STO*    LA11                                                    YTL24620
        ZET     BNULL                                                   YTL24630
        STO*    LB11                                                    YTL24640
        TRA     MMMR                                                    YTL24650
*               MMM23 - MATRIX MAPPING OR CHECKSUM ROUTINE              YTL24660
*               ADDED TO YTL01 JUNE 1962                                YTL24670
*               NEEDS (STH),(FIL), AND MMM20                            YTL24680
*               INSTRUCTION CARD IS,  FIELD 1 = LOCATION OF MATRIX      YTL24690
*                                     FIELD 2 = 0 FOR NEW PAGE          YTL24700
*                                             = 1 FOR NO SPACING        YTL24710
*                                     FIELD 3 = ADDRESS OF NUMBER FOR   YTL24720
*                                               ZERO LEVEL CHECKING-    YTL24730
*                                               A(I,J) MAPPED AS ZERO IF YTL24740
*                                               LESS THAN OR EQUAL TO   YTL24750
*                                               THIS NUMBER.            YTL24760
*                                     FIELD 4 = 0 IF MATRIX NO. WANTED  YTL24770
*                                             = K IF 1 FULL TITLE       YTL24780
*                                               CARD TO BE READ         YTL24790
*                                               FROM TAPE K             YTL24800
*                                     FIELD 5 = OP. CODE 23             YTL24810
*               PICK UP A NUMBER OF VARIABLES FROM MMM20 WHICH ARE      YTL24820
*               MARKED BY ASTRISK IN COL 73                             YTL24830
*                                                                       YTL24840
MMM23   ZET     ANULL                                                   YTL24850
        TRA     THRU            A IS NULL                               YTL24860
        NZT     YMCC                                                    YTL24870
        TRA     STOZE                                                   YTL24880
        CLA*    YMCC                                                    YTL24890
        SSP                                                             YTL24900
        STO     LEVEL           STORE NUMBER FOR ZERO LEVEL TESTING     YTL24910
        TRA     *+2                                                     YTL24920
STOZE   STZ     LEVEL                                                   YTL24930
        CALL    .FWRD.(.UN06.,LVLPT)                                    YTL24940
        CLA     LEVEL                                                   YTL24950
        TSX     .FCNV.,4                                                YTL24960
        CALL    .FFIL.                                                  YTL24970
        CLA     LINE            INCREASE LINE COUNT FOR EXTRA TITLE     YTL24980
        ADD     =2                                                      YTL24990
        STO     LINE                                                    YTL25000
        STZ     SVCLPT          ZERO COUNT FOR COLUMN I.D. COUNT        YTL25010
        STZ     NCLPT           ZERO FLAG FOR NEW PAGE COLUMN I.D.      YTL25020
        CLA     =110            GO AHEAD WITH MAPPING                   YTL25030
        LDQ     NA                                                      YTL25040
        TLQ     *+3                                                     YTL25050
        CLA     =34             TOO BIG TO MAP                          YTL25060
        TRA     MMME                                                    YTL25070
        CLA     =9              NO, PROCEED WITH MAPPING                YTL25080
        TLQ     LESS10          LESS THAN 10 COLUMNS                    YTL25090
        XCA                     MORE THAN 10 COLUMNS                    YTL25100
        SUB     =10             FIND NUMBER OF COLUMN LABELS - LABEL    YTL25110
        AXT     1,1             TO GO AT EACH INCREMENT OF 10 COLUMNS   YTL25120
        SUB     =10                                                     YTL25130
        TMI     *+2             IF N - K*10 = 0, IT WILL BE + 0         YTL25140
        TXI     *-2,1,1         COUNT OF LABELS LEFT IN XR1             YTL25150
        PXA     0,1                                                     YTL25160
        PAX     0,2
```

```
        STO     SVCLPT          SAVE COUNT OF LABELS                                    YTL25170
CLPT    CALL    .FWRD.(.UN06.,COLPT)                                                    YTL25180
        PXA     0,0                                                                     YTL25190
   PP1  ADD     =10                                                                     YTL25200
        PAX     0,5                                                                     YTL25210
        TSX     .FCNV.,4        PRINT LABEL ON EACH 10 COLUMNS                          YTL25220
        PXA     0,5                                                                     YTL25230
        TIX     PP1,1,1                                                                 YTL25240
        CALL    .FFIL.                                                                  YTL25250
        CALL    .FWRD.(.UN06.,COLPT1)           PRINT TO IDENTIFY COLS.                 YTL25260
        CLA     =0200000000000                                                          YTL25270
        TSX     .FCNV.,4                                                                YTL25280
        TIX     *-2,2,1                                                                 YTL25290
        CALL    .FFIL.                                                                  YTL25300
        CLA     NCLPT           IS THIS THE FIRST PAGE                                  YTL25310
        TNZ     NSPCE-2         NO, THIS WAS PRINT ON NEW PAGE                          YTL25320
LESS10  CLA     =10                                                                     YTL25330
        STO     TESTI           COUNT FOR IDENTIFYING EVERY 10 TH ROW                   YTL25340
        CLA     =1                                                                      YTL25350
        STO     I               ROW COUNT                                               YTL25360
        CLA     LA11            ADDRESS OF MATRIX -                                     YTL25370
        STA     TESTIJ                                                                  YTL25380
        CLA     =-1                                                                     YTL25390
        STO     CHKSUM          SET UP RETURN FROM MMM20 NEW PAGE PRINT                 YTL25400
MRROWS  CLA     NA                                                                      YTL25410
        PAX     0,1             NUMBER OF COLUMNS                                       YTL25420
        ADD     TESTIJ          INCREMENT ADDRESS OF CURRENT ROW                        YTL25430
        STA     TESTIJ                                                                  YTL25440
        CLA     =0606060606060                                                          YTL25450
        AXT     19,2                                                                    YTL25460
        STO     ROW+19,2        FILL 1 ROW OF MAP WITH BLANKS                           YTL25470
        TIX     *-1,2,1                                                                 YTL25480
        PXA     ROW,0                                                                   YTL25490
        CLA     *-1                                                                     YTL25500
        STA     RWCELL          SET UP ADDRESS OF 1 ST CELL OF ROW                      YTL25510
        AXT     6,2                                                                     YTL25520
        CAL     ZERO+6,2                                                                YTL25530
RWCELL  ANS     **              ZERO CURRENT COLUMN POSITION                            YTL25540
TESTIJ  CLA     **,1                                                                    YTL25550
        SSP                                                                             YTL25560
        CAS     LEVEL           IS ELEMENT (I,J) GREATER THAN GIVEN LEVEL               YTL25570
        TRA     *+3             YES                                                     YTL25580
        TRA     NCHNG           EQUAL                                                   YTL25590
        TRA     NCHNG           LESS THAN                                               YTL25600
        CAL     DOLSGN+6,2      GREATER THAN LEVEL                                      YTL25610
        ORS*    RWCELL          PUT $ IN CURRENT POSITION                               YTL25620
        TRA     *+3                                                                     YTL25630
 NCHNG  CAL     POINT+6,2       PUT DECIMAL POINT IN IF NO. IS BELOW LEVEL              YTL25640
        ORS*    RWCELL                                                                  YTL25650
        TIX     KPGONG,1,1                                                              YTL25660
        CLA     SPACE                                                                   YTL25670
        TNZ     NSPCE                                                                   YTL25680
        CLA     LINE                                                                    YTL25690
        CAS     =54                                                                     YTL25700
        TRA     NEWPGE          TO MMM20                                                YTL25710
        TRA     *+1                                                                     YTL25720
        ADD     =1                                                                      YTL25730
        STO     LINE                                                                    YTL25740
        TRA     NSPCE                                                                   YTL25750
KPGONG  TIX     RWCELL-1,2,1    HAVE WE USED A FULL CELL OF ROW                         YTL25760
```

```
        CLA     RWCELL                                                  YTL25770
        ADD     =1                                                      YTL25780
        STA     RWCELL          WANT TO USE NEXT CELL OF ROW            YTL25790
        TRA     RWCELL-2                                                YTL25800
 STPGE  NZT     SVCLPT          ARE THERE MORE THAN 10 COLUMNS          YTL25810
        TRA     NOID            NO                                      YTL25820
        CLA     =1              YES, PUT A 1 IN FLAG                    YTL25830
        STO     NCLPT                                                   YTL25840
        LXA     SVCLPT,1                                                YTL25850
        LXA     SVCLPT,2                                                YTL25860
        TRA     CLPT            PRINT NEW PAGE COLUMN I.D.              YTL25870
  NOID  CLA     =7              COLUMN I.D. IS NOT NECESSARY            YTL25880
        STO     LINE                                                    YTL25890
        TRA     NSPCE                                                   YTL25900
        CLA     =9              RETURN FROM PRINTING COLUMN I.D.        YTL25910
        STO     LINE                                                    YTL25920
 NSPCE  CLA     I                                                       YTL25930
        SUB     TESTI                                                   YTL25940
        TZE     RWIDNT          DO WE PRINT ROW IDENTIFICATION          YTL25950
        CALL    .FWRD.(.UN06.,RWPRNT)                                   YTL25960
        CALL    .FSLO.(ROW,=19)                                         YTL25970
        CALL    .FFIL.                                                  YTL25980
        TRA     TSTLMT                                                  YTL25990
RWIDNT  CALL    .FWRD.(.UN06.,RWIDPT)           PRINT ROW WITH IDENT.   YTL26000
        CLA     TESTI                                                   YTL26010
        TSX     .FCNV.,4                                                YTL26020
        CALL    .FSLO.(ROW,=19)                                         YTL26030
        CALL    .FFIL.                                                  YTL26040
        CLA     TESTI                                                   YTL26050
        ADD     =10                                                     YTL26060
        STO     TESTI                                                   YTL26070
TSTLMT  CLA     I                                                       YTL26080
        CAS     MA              ARE WE DONE                             YTL26090
        TRA     THRU            YES                                     YTL26100
        TRA     THRU                                                    YTL26110
        ADD     =1              NO                                      YTL26120
        STO     I                                                       YTL26130
        TRA     MRROWS                                                  YTL26140
THRU    CALL    .FWRD.(.UN06.,FINLPT)                                   YTL26150
        CALL    .FFIL.'                                                 YTL26160
   OUT  TRA     DONE                                                    YTL26170
*               MMM24 - UNLOAD TAPE UNIT                                YTL26180
*               FIELD 2 = TAPE NUMBER X 10**6                           YTL26190
*               NEEDS LIBRARY ROUTINE UNLOAD                            YTL26200
 MMM24  CLA     2,4                                                     YTL26210
        SSP                                                             YTL26220
        TSX     TP,4            GET TAPE NUMBER                         YTL26230
        STO     TAPE                                                    YTL26240
        TSX     TPCK,4          IS IT A VALID NUMBER                    YTL26250
        CALL    UNLOAD(TAPE)                                            YTL26260
        TRA     DONE                                                    YTL26270
*               PACKAGE 16 MMM25                                        YTL26280
*               SQUARE ROOT OF A DIAGONAL                               YTL26290
 MMM25  TSX     CKDM1,4         CHECK DIMENSIONS IN FIELD 1             YTL26300
        TSX     CKDG1,4                                                 YTL26310
        CLA     MA                                                      YTL26320
        STA     MC                                                      YTL26330
   SR1  LDQ     NA                                                      YTL26340
        STQ     NC                                                      YTL26350
        TSX     CHKCOR,4                                                YTL26360
```

```
        LDQ     NA                                                      YTL26370
        MPY     MA                                                      YTL26380
        XCA                                                             YTL26390
        PAX     0,4                                                     YTL26400
        STA     SR4                                                     YTL26410
        ACL     LA11                                                    YTL26420
        STA     SR5                                                     YTL26430
        PXA     0,4                                                     YTL26440
        ACL     LC11                                                    YTL26450
        ZET     ANULL                                                   YTL26460
        TRA     NMPY            A IS NULL                               YTL26470
        STA     SR6                                                     YTL26480
   SR4  AXT     **,1            ** IS MN                                YTL26490
   SR5  CLA     **,1            ** IS LA11+MN                           YTL26500
        STO     SQROOT                                                  YTL26510
        TZE     SR6                                                     YTL26520
        TPL     *+3                                                     YTL26530
        CLA     =12             NEGATIVE ELEMENT                        YTL26540
        TRA     MMME                                                    YTL26550
        CALL    SQRT(SQROOT)                                            YTL26560
   SR6  STO     **,1            ** IS LC11+MN                           YTL26570
        TIX     SR5,1,1                                                 YTL26580
        TTR     MMMR                                                    YTL26590
*               MMM26 - CALL IN NEW LINK OF CHAIN                       YTL26600
*                       FIELD 1 = TAPE NUMBER, MUST BE                  YTL26610
*                                 1,2, OR 3                             YTL26620
*                       FIELD 4 = LINK NUMBER                           YTL26630
MMM26   TRA     DONE                                                    YTL26640
*                                                                       YTL26650
*               OPERATION CODE 27 - FIXED POINT INCREMENT               YTL26660
*               OR DECREMENT OF A SINGLE CORE LOCATION                  YTL26670
                        OR IN A CORE PROGRAM IT CAN BE                  YTL26680
                        1000 + 101 + J, WHERE I IS THE NUMBER           YTL26690
                        OF CARDS FORWARD OR BACKWARDS FROM              YTL26700
                        THIS INSTRUCTION, AND J IS THE                  YTL26710
                        FIELD WITHIN THE INSTRUCTION.IF                 YTL26720
                        NEGATIVE, COUNT IS BACKWARDS.                   YTL26730
*               FIELD 1 = CORE LOCATION TO BE INCREM./DECREM.           YTL26740
*               FIELD 2 BLANK IF FIELD 4 IS THE INCREMENT               YTL26750
*               FIELD 2 NON-ZERO IF FIELD 4 IS THE INCREMENT LOCATION   YTL26760
*               FIELD 3 BLANK IF RESULT GOES INTO FIELD 1 LOCATION      YTL26770
*               OTHERWISE, RESULT GOES INTO LOCATION OF FIELD 3         YTL26780
*               FIELD 4 = NUMBER TO BE ADDED TO CONTENTS OF FIELD 1     YTL26790
*                         OR LOCATION OF NUMBER TO BE ADDED             YTL26800
*               FIELD 5 = 27                                            YTL26810
*                                                                       YTL26820
*               FIELD 3 IS NEGATIVE(OR CONTENTS IS NEGATIVE)            YTL26830
*               FOR A DECREMENT OF CELL IN FIELD 1                      YTL26840
*                                                                       YTL26850
 MMM27  CLA     FXADD                                                   YTL26860
        SLW     ADDSUB                                                  YTL26870
STRT27  CLA     YMAA                                                    YTL26880
        TSX     CHKRNG,4                                                YTL26890
        TRA     ZROER                                                   YTL26900
        TRA     CORLOC                                                  YTL26910
        TSX     RELADR,4                                                YTL26920
        STO     YMAA                                                    YTL26930
CORLOC  CLA     YMCC                                                    YTL26940
        TZE     NOCHK3                                                  YTL26950
        TSX     CHKRNG,4                                                YTL26960
```

```
        TRA     ZROER                                                   YTL26970
        TRA     NOCHK3                                                  YTL26980
        TSX     RELADR,4                                                YTL26990
        STO     YMCC                                                    YTL27000
NOCHK3  CLA     YMBB                                                    YTL27010
        TNZ     USEFLD                                                  YTL27020
        CLA     YMNC                                                    YTL27030
        CAS     PROGLO                                                  YTL27040
        TRA     *+3                                                     YTL27050
        TRA     *+2                                                     YTL27060
        TRA     USEFLD                                                  YTL27070
        CLA     YMNC                                                    YTL27080
        ADD     YSHIFT                                                  YTL27090
        STO     YMNC                                                    YTL27100
        CLA*    YMNC                                                    YTL27110
        TRA     *+2                                                     YTL27120
USEFLD  CLA     YMNC                                                    YTL27130
ADDSUB  PZE     0                                                       YTL27140
        ADD*    YMAA                                                    YTL27150
        ZET     YMCC                                                    YTL27160
        TRA     *+3                                                     YTL27170
        STO*    YMAA                                                    YTL27180
        TRA     DONE                                                    YTL27190
        STO*    YMCC                                                    YTL27200
        TRA     DONE                                                    YTL27210
RELADR  STO     SAVEF                                                   YTL27220
        CLA     KEY                                                     YTL27230
        TNZ     *+3                                                     YTL27240
        CLA     =24                                                     YTL27250
        TRA     MMME                                                    YTL27260
        CLA     SAVEF                                                   YTL27270
        SSP                                                             YTL27280
        SUB     TOPPOS                                                  YTL27290
        XCA                                                             YTL27300
        ZAC                                                             YTL27310
        DVP     =10                                                     YTL27320
        CAS     =6                                                      YTL27330
        TRA     FLDERR                                                  YTL27340
        TRA     *+1                                                     YTL27350
        STO     FLDSKP                                                  YTL27360
        MPY     =6                                                      YTL27370
        STQ     CELSKP                                                  YTL27380
        CLA     SAVEF                                                   YTL27390
        TMI     BACKSK                                                  YTL27400
        CLA     CELSKP                                                  YTL27410
        SUB     =7                                                      YTL27420
        ADD     FLDSKP                                                  YTL27430
        TRA     FINDCL                                                  YTL27440
BACKSK  CLA     CELSKP                                                  YTL27450
        ADD     =7                                                      YTL27460
        SUB     FLDSKP                                                  YTL27470
        SSM                                                             YTL27480
FINDCL  ADD     INST                                                    YTL27490
        TRA     1,4                                                     YTL27500
FLDERR  CLA     =33                                                     YTL27510
        TRA     MMME                                                    YTL27520
*                                                                       YTL27530
*               OPERATION CODE 28 - MERGE OF MATRIX A                   YTL27540
*               AND MATRIX A+1                                          YTL27550
*                                                                       YTL27560
```

```
*                                                                      YTL27570
*                                                                      YTL27580
*              FIELD 1 =POSITION NUMBER N.  MATRIX IN                  YTL27590
*                 POSITION N+1 MERGED WITH MATRIX IN POSITION          YTL27600
*                 N.  RESULTING MATRIX IN POSITION N                   YTL27610
*                                                                      YTL27620
*              FIELD 2 =BLANK                                          YTL27630
*                                                                      YTL27640
*              FIELD 3 = BLANK                                         YTL27650
*                                                                      YTL27660
*              FIELD 4 = MAME OF MERGED MATRIX                         YTL27670
*                                                                      YTL27680
*              FIELD 5 = 28                                            YTL27690
*                                                                      YTL27700
MMM28 LDQ     MA                                                       YTL27710
      MPY     NA                                                       YTL27720
      XCA                                                              YTL27730
      ADD     LA11                                                     YTL27740
      ADD     =1              L(MB)                                    YTL27750
      STA     GETMB                                                    YTL27760
      ADD     =1              L(NB)                                    YTL27770
      STA     GETNB                                                    YTL27780
GETNB CLA     **              NB                                       YTL27790
      STO     NB                                                       YTL27800
      SUB     NA                                                       YTL27810
      TNZ     MMMCF           COLUMN NUMBERS INCOMPATIBLE              YTL27820
GETMB CLA     **              MB                                       YTL27830
      STO     MB                                                       YTL27840
      ADD     MA                                                       YTL27850
      STO*    MMMA            NEW NUMBER OF ROWS                       YTL27860
      XCA                                                              YTL27870
      MPY     NA                                                       YTL27880
      XCA                                                              YTL27890
      ADD     LA11                                                     YTL27900
      STA     STMA                                                     YTL27910
      TSX     CKDM12,4                                                 YTL27920
      LDQ*    GETNB                                                    YTL27930
      MPY*    GETMB                                                    YTL27940
      XCA                                                              YTL27950
      PAX     0,1             NUMBER OF WORDS TO MOVE                  YTL27960
      ADD     GETNB                                                    YTL27970
      ADD     =1                                                       YTL27980
      STA     LDMB                                                     YTL27990
      CLA     GETNB                                                    YTL28000
      ADD     =1                                                       YTL28010
      STA     *+1                                                      YTL28020
      CLA     **              LB11                                     YTL28030
      SUB     WDMNL           IS POSITION N+1 NULL                     YTL28040
      TNZ     *+3             NO                                       YTL28050
      STZ*    *-3             CLEAR OUT CODE WORD                      YTL28060
      TRA     LDMB                                                     YTL28070
      CLA     ANULL           IS POSITION N NULL, N+1 NOT NULL         YTL28080
      TZE     LDMB            NO                                       YTL28090
      STZ*    LA11            CLEAR CODE WORD WHEN POS. N NULL, N+1 NOT NYTL28100
LDMB  CLA     **,1            LB11+MB*NB                               YTL28110
STMA  STO     Q*,1            LA11+(MA+MB)*NA                          YTL28120
      TIX     LDMB,1,1                                                 YTL28130
      AXT     3,1                                                      YTL28140
      STZ*    LDMB                                                     YTL28150
      TIX     *-1,1,1                                                  YTL28160
```

```
        CLA     YMNC                                                    YTL28170
        STO*    YMAA                                                    YTL28180
        TRA     DONE                                                    YTL28190
*                                                                       YTL28200
*               MMM29-CLEAR CORE                                        YTL28210
*                                                                       YTL28220
*               FIELD 1 IS THE STARTING ADDRESS OF CLEAR                YTL28230
*               FIELD 4 IS THE NUMBER OF LOCATIONS                      YTL28240
*                       IF ZERO,CORE IS CLEARED TO 32563                YTL28250
MMM29   CLA     YMAA                                                    YTL28260
        TSX     CHKRNG,4                                                YTL28270
        TRA     ZROER                                                   YTL28280
        TRA     *+2                                                     YTL28290
        TSX     RELADR,4                                                YTL28300
        ADD     YMNC            INCREMENT,IF ANY                        YTL28310
        CAS     HICORE                                                  YTL28320
        TRA     *+3                                                     YTL28330
        TRA     LIMOK                                                   YTL28340
        TRA     LIMOK                                                   YTL28350
        SUB     YSHIFT                                                  YTL28360
        CAS     HICORE                                                  YTL28370
        TRA     TEST2+3                                                 YTL28380
        TRA     *+1                                                     YTL28390
LIMOK   STA     ZROSRT                                                  YTL28400
        CLA     YMNC            INCREMENT                               YTL28410
        PAX     0,1                                                     YTL28420
        TZE     *+4                                                     YTL28430
        TPL     ZROSRT                                                  YTL28440
        CLA     =30             NEGATIVE INCREMENT                      YTL28450
        TRA     MMME                                                    YTL28460
        CLA     HICORE                                                  YTL28470
        ADD     =1                                                      YTL28480
        STA     ZROSRT                                                  YTL28490
        SUB     YMAA                                                    YTL28500
        PAX     0,1                                                     YTL28510
ZROSRT  STZ     **,1                                                    YTL28520
        TIX     *-1,1,1                                                 YTL28530
        TRA     DONE                                                    YTL28540
ZROER   CLA     =29                                                     YTL28550
        TRA     MMME                                                    YTL28560
*                                                                       YTL28570
*               MMM30,MMM31,MMM32,ARE THE MULTIPLY                      YTL28580
*               AND ADD ROUTINES CORRESPONDING TO MMM6,                 YTL28590
*               MMM7,ANDMMM8.A CHECK IS MADE TO SEE                     YTL28600
*               THAT THERE IS A CONFORMABLE MATRIX IN                   YTL28610
*               THE POSITION OF FIELD 3,AND THEN IT                     YTL28620
*               TRANSFERS CONTROL TO MMM6,MMM7,OR                       YTL28630
*               MMM8. THE INITIALIZING OF THE SUMMING                   YTL28640
*               MATRIX IS LEFT UP TO THE USER                           YTL28650
*                                                                       YTL28660
 MMM30  CLA     =1                                                      YTL28670
        STO     MAD                                                     YTL28680
        CLA     MA                                                      YTL28690
        SUB*    LOCMC                                                   YTL28700
        TZE     *+3                                                     YTL28710
NCNFBS  CLA     =31             SUMMING MATRIX NON-CONFORMABLE          YTL28720
        TRA     MMME                                                    YTL28730
        CLA     NB                                                      YTL28740
        SUB*    LOCNC                                                   YTL28750
        TNZ     NCNFBS          NON-CONFORMABLE                         YTL28760
```

```
        CLA     WDMNL                                                   YTL28770
        SUB*    LC11                                                    YTL28780
        TNZ     *+2                                                     YTL28790
        STZ     MAD                                                     YTL28800
        TRA     MMM6                                                    YTL28810
*                                                                       YTL28820
 MMM31  CLA     =1                                                      YTL28830
        STO     MAD                                                     YTL28840
        CLA     MA                                                      YTL28850
        SUB*    LOCMC                                                   YTL28860
        TNZ     NCNFBS          NON-CONFORMABLE                         YTL28870
        CLA     MB                                                      YTL28880
        SUB*    LOCNC                                                   YTL28890
        TNZ     NCNFBS          NON-CONFORMABLE                         YTL28900
        CLA     WDMNL                                                   YTL28910
        SUB*    LC11                                                    YTL28920
        TNZ     *+2                                                     YTL28930
        STZ     MAD                                                     YTL28940
        TRA     MMM7                                                    YTL28950
*                                                                       YTL28960
 MMM32  CLA     =1                                                      YTL28970
        STO     MAD                                                     YTL28980
        CLA     NA                                                      YTL28990
        SUB*    LOCMC                                                   YTL29000
        TNZ     NCNFBS          NON-CONFORMABLE                         YTL29010
        CLA     NB                                                      YTL29020
        SUB*    LOCNC                                                   YTL29030
        TNZ     NCNFBS          NON-CONFORMABLE                         YTL29040
        CLA     WDMNL                                                   YTL29050
        SUB*    LC11                                                    YTL29060
        TNZ     *+2                                                     YTL29070
        STZ     MAD                                                     YTL29080
        TRA     MMM8                                                    YTL29090
*                                                                       YTL29100
*               AS MMM 27 BUT SUB IS USED                               YTL29110
*                                                                       YTL29120
*                                                                       YTL29130
 MMM33  CLA     FXSUB                                                   YTL29140
        SLW     ADDSUB                                                  YTL29150
        TRA     STRT27                                                  YTL29160
*                                                                       YTL29170
*               FOLLOWING ARE ALL YTL01 CONSTANTS AND FORMATS           YTL29180
*                                                                       YTL29190
*                                                                       YTL29200
*               THE FOLLOWING IS A SET OF CELLS THAT                    YTL29210
*               ARE SET UP TO FACILITATE DUMPING IN                     YTL29220
*               CHARET.                                                 YTL29230
ERRCDE  PZE     0               ERROR CODE                              YTL29240
        BCI     1,ERRCDE                                                YTL29250
  TAPE  BSS     3               TAPE NO,ALSO FILE,MATRIX SPACING        YTL29260
        BCI     1,TAPE                                                  YTL29270
*               LAST INS  .       TAPE+1          .       TAPE+2        YTL29280
*                         .                         .                   YTL29290
*                  FSF    .FILE COUNT    IN DECR.          0           YTL29300
*                  BSF    .FILE COUNT -1 IN DECR..         0           YTL29310
*                  FSR    .2*MATRIX COUNT IN DECR.         0           YTL29320
*                FSF,FSR  .2*MATRIX COUNT IN DECR.  FILE COUNT IN DECR  YTL29330
*                                                                       YTL29340
  SAVE  BSS     16              MATRIX ID                               YTL29350
MATID   EQU     SAVE                                                    YTL29360
```

```
        BCI     1,MATID                                                         YTL29370
                                                                                YTL29380
 KEY    PZE                                                                     YTL29390
CORPRG  EQU     KEY                                                             YTL29400
        BCI     1,CORPRG                                                        YTL29410
*               THESE ARE CELLS WHICH MAY BE USEFUL                             YTL29420
PHYSCT  PZE     0               PHYSICAL COUNT OF INSTRUCTIONS                  YTL29430
INSCNT  PZE     0               COUNT OF EXECUTED INSTRUCTIONS                  YTL29440
 XMODE  PZE                     PARTITION ADDRESS                               YTL29450
 INST   PZE                     ADDRESS OF NEXT CORE EXECUTED INSTRUCTION       YTL29460
SAVKEY  PZE                                                                     YTL29470
MCROCT  PZE                                                                     YTL29480
RETURN  BSS     5                                                               YTL29490
PRGLOC  BSS     5                                                               YTL29500
 MA                             PARAMETERS                                      YTL29510
 NA                                                                             YTL29520
 LA11                                                                           YTL29530
 MB                                                                             YTL29540
 NB                                                                             YTL29550
 LB11                                                                           YTL29560
 MC                                                                             YTL29570
 NC                                                                             YTL29580
 LC11                                                                           YTL29590
TEMPAC  PZE     0                                                               YTL29600
TEMPMQ  PZE     0               SAME FOR MQ                                     YTL29610
 CELL0  PZE     0               SAVE OF LAST NON-ZERO CELL 0                    YTL29620
*               MISCELLANEOUS CONSTANTS                                         YTL29630
WHCHFT  PZE     0               FORMAT INDICATOR FOR 20                         YTL29640
TAP1IB  PZE     0               TAPE NUMBER FOR TITLE IN 20 AND 23              YTL29650
TAPIB   PZE     0                                                               YTL29660
READ    BSS     8               CALLING PARAMETERS FOR OP. CODE 19              YTL29670
KRDP1   EQU     READ+1          STARTING ADDRESS                                YTL29680
KRDP4   EQU     READ+4          NUMBER 4F FIELDS                                YTL29690
KRDP5   EQU     READ+5          NUMBER OF DIGITS PER FIELD                      YTL29700
KRDP7   EQU     READ+7          TAPE NO.                                        YTL29710
TAPEIB  PZE     0               TAPE UNIT SPECIFICATION FOR IOCS                YTL29720
ADRSDA  PZE     DATA            ADD. FOR LOWER LIMIT OF AVAILABLE CORE          YTL29730
  SEPE  PZE     0               TAPE NOTFOR LOAD PROGRAM                        YTL29740
   LOC  BSS     5               THESE ARE THE 5 OPERATION FIELDS                YTL29750
   TAP  PZE     0               TAPE UNIT FOR PROGRAM CARDS                     YTL29760
SQROOT  PZE     0               STORAGE FOR MMM25                               YTL29770
 TAP1   PZE                                                                     YTL29780
FPTTRA  TRA     FPSPIL          TRA TO GO INTO CELL 8                           YTL29790
TMPFPT  PZE     0               SAVE CELL FOR LOCATION 8                        YTL29800
 XR1    PZE                     XR STORAGE FOR YTL01                            YTL29810
 XR2    PZE                                                                     YTL29820
 XR4    PZE                                                                     YTL29830
YSHIFT  PZE                     RELOCATION CONSTANT                             YTL29840
 IRROR  PZE                     KRD ERROR CODE FOR ALL KRD CALLS                YTL29850
 INS    PZE     **,2            ADDRESS OF ERROR CODE CELLS                     YTL29860
*                                                                               YTL29870
*               XR STORAGE FOR MMM                                              YTL29880
 IR1    PZE                                                                     YTL29890
 IR2    PZE                                                                     YTL29900
   IR3  PZE     0                                                               YTL29910
 IR4    PZE                                                                     YTL29920
   IR5  PZE     0                                                               YTL29930
   IR6  PZE     0                                                               YTL29940
   IR7  PZE     0                                                               YTL29950
PARTON  OCT     777777777777    SET TO ZERO AFTER A PARTITION IS FORMED         YTL29960
 FIRST  PZE     1
```

```
*              CONSTANTS FOR VARIOUS DIAGNOSTICS                          YTL29970
PROGLO PZE                    LOWEST CORELOC. FROM CARDS NORMALLY 8000    YTL29980
SMLTAP DEC     1000000        SMALLEST TAPE NUMBER                        YTL29990
LOCORE PZE                    LOWEST LEGAL CORE ADDRESS                   YTL30000
HICORE PZE                    HIGEST POSSIBLE CORE ADDRESS                YTL30010
TOPPOS PZE     1000           LARGER THAN ANY LEGAL POSITION NUMBER       YTL30020
MXDATA PZE                    MAXIMUM SIZE OF ANY MATRIX                  YTL30030
HIP-OG PZE                    MAXIMUM LOAD ADDRESS OF A PROGRAM           YTL30040
TSTMAC EQU     7999           TEST CONSTANT FOR MACRO INSTRUCTION         YTL30050
MXTAPS EQU     19             MAXIMUM NUMBER OF TAPES                     YTL30060
*              THESE TWO CONSTANTS ARE EQUATED TO ABOVE ONES,              YTL30070
*              AND MUSTBE PRESERVED IF SMLTAP,TOPPOS,CHANGE                YTL30080
*                                                                          YTL30090
THOUSN EQU     TOPPOS         MUST BE 1000(DECIMAL)                       YTL30100
MILYN  EQU     SMLTAP         MUST BE 1000000(DECIMAL)                    YTL30110
*                                                                          YTL30120
 SIGN  PZE                    SET TO NON-ZERO IF 32564 HAS BEEN STUFFED   YTL30130
COREND BCI     1,COREND       CODE WORD PUT IN 32564                      YTL30140
*              XR4 STORAGE FOR MMM90                                       YTL30150
 SPOT4                                                                    YTL30160
*              PARAMATERS FOR 3 WORKING MATRICES                           YTL30170
*              CELLS FOR READ/WRITE BINARY TAPE                            YTL30180
 MTN   PZE                    M*N                                         YTL30190
 TESTC PZE                    TEMPORARY                                   YTL30200
 WDMNL BCI     1,M=NULL                                                   YTL30210
SPARSE BCI     1,SPARSE                                                   YTL30220
SPMTC1 PZE                    TEST CELL FOR SPARSE MATRIX                 YTL30230
*      WDWC IS THE NUMBER OF ELEMENTS AND CONTROL                         YTL30240
*      WORDS NEEDED TO WRITE A SPARSE MATRIX. NEEDED                      YTL30250
*      TO INSURE A MINIMUM RECORD ON TAPE OF 16 WORDS.                    YTL30260
 WDWC  PZE                                                                YTL30270
 MNP3  PZE                    M*N+3 +LA11                                 YTL30280
RSCKSM PZE                    USED FOR SPARSE CHECKSUM TEMP               YTL30290
READTP PZE                    TEMP FOR WORD FROM TAPE IN SPARSE READ      YTL30300
 ERR   PZE                    ERROR CODE FOR BSF,FSR,FSF                  YTL30310
*              4 TEMPORARY STORAGE CELLS                                   YTL30320
       BSS     3                                                          YTL30330
 TEMP  PZE                                                                YTL30340
*              NULL MATRIX COTEMP AC FORFPT ANALYSIS                       YTL30350
 ANULL                                                                    YTL30360
 BNULL                                                                    YTL30370
 FILL5 CLA     0,1                                                        YTL30380
 FILL6 CLS     0,1                                                        YTL30390
 FILL7 CLA     =0                                                         YTL30400
*                                                                         YTL30410
 FILLO NOP                    FOR MATRIX MOVE                             YTL30420
 FILL1 FAD 0,1                FOR MMM1                                    YTL30430
 FILL2 FSB 0,1                FOR MMM2                                    YTL30440
 FILL3 FAD 0,2                FOR MMM3                                    YTL30450
 FILL4 FSB 0,2                FOR MMM4                                    YTL30460
DOTPRD BSS     3              CELLS USED FOR DOUBLE PRECISION             YTL30470
*                             DOT PRODUCT - NEED 3 TO INSURE              YTL30480
*                             AN EVEN LOCATION ALWAYS                     YTL30490
  TOTA PZE     0              MA * NA                                     YTL30500
  TOTB PZE     0              MB*NB                                       YTL30510
INCRMA PZE     0              STORAGE INCREMENT FOR A                     YTL30520
INCRMB PZE     0              STORAGE INCREMENT FOR B                     YTL30530
MADYES FAD*    ANSWER         USED FOR MMM30,31,32                        YTL30540
MADNO  TRA     ANSWER-1       USED FOR MMM6,7,8                           YTL30550
 FIL10 FAD 0,2                FOR MMM10                                   YTL30560
```

```
 FIL11 FSB 0,2                  FOR MMM11                                    YTL30570
*              CONSTANT STORAGE FOR MMM18                                   YTL30580
  IRR1 PZE     0                ERROR CODE 1                                YTL30590
  IRR2 PZE     0                ERROR CODE 2                                YTL30600
 SCALE PZE     0                CELL FOR SCALE FACTOR                       YTL30610
   DET PZE     0                FRACTIONAL PART OF DETERMINANT              YTL30620
       PZE     0                                                            YTL30630
NDETXP PZE     0                EXPONENT PART OF DETERMINANT                YTL30640
   NSQ PZE     0                N**2                                        YTL30650
TWONSQ PZE     0                2*N**2                                      YTL30660
*              MMM20 CONSTANTS                                              YTL30670
 TITLE BSS     14               TITLE STORAGE                               YTL30680
CHKSUM PZE     0                PRINT CHECKSUM                              YTL30690
  LINE PZE     0                LINE COUNT                                  YTL30700
 SPACE PZE     0                TEST CELL FOR INITIAL SPACING TO NEW PAGE   YTL30710
  TTLE PZE     0                FIELD 4 INDICATOR                           YTL30720
   MAP PZE     0                TEST CELL FOR MAP ROUTINE                   YTL30730
     I PZE     0                ROW COUNT                                   YTL30740
 CELL1 PZE     0                SCRATCH FOR MMM22                           YTL30750
*              MMM23 CONSTANTS                                              YTL30760
 LEVEL PZE     0                FL.PT. NUMBER FOR TESTING                   YTL30770
 TESTI PZE     0                COUNT FOR FINDING EVERY 10TH ROW            YTL30780
  ROW  BSS     19               STORAGE FOR 1 PRINTED ROW                   YTL30790
SVCLPT PZE     0                COUNT FOR COLUMN ID                         YTL30800
 NCLPT PZE     0                FLAG FOR NEW PAGE COLUMN HEADING            YTL30810
*              MAP PARAMETERS                                               YTL30820
  ZERO OCT     007777777777                                                 YTL30830
       OCT     770077777777                                                 YTL30840
       OCT     777700777777                                                 YTL30850
       OCT     777777007777                                                 YTL30860
       OCT     777777770077                                                 YTL30870
       OCT     777777777700                                                 YTL30880
DOLSGN OCT     530000000000                                                 YTL30890
       OCT     005300000000                                                 YTL30900
       OCT     000053000000                                                 YTL30910
       OCT     000000530000                                                 YTL30920
       OCT     000000005300                                                 YTL30930
       OCT     000000000053                                                 YTL30940
 POINT OCT     330000000000                                                 YTL30950
       OCT     003300000000                                                 YTL30960
       OCT     000033000000                                                 YTL30970
       OCT     000000330000                                                 YTL30980
       OCT     000000003300                                                 YTL30990
       OCT     000000000033                                                 YTL31000
FLDSKP PZE     0                COUNT OF FIELDS -MMM27                      YTL31010
CELSKP PZE     0                COUNT OF CELLS FOR CARDS TO SKIP            YTL31020
SAVEF  PZE     0                                                            YTL31030
FXSUB  CHS                                                                  YTL31040
FXADD  TRA     ADDSUB+1                                                     YTL31050
   MAD PZE     0                TEST CELL FOR MMM30,31,32                   YTL31060
SAVINS PZE                                                                  YTL31070
*              NON ZERO FOR 30,31,32,OTHERWISE 0                            YTL31080
*                                                                           YTL31090
*              FORMAT FOR ERROR PRINT IN CONTROL SECTION                    YTL31100
CORERR BCI     8,(1H ,10X,51HUNABLE TO PRINT CURRENT CORE PROGRAM           YTL31110
       BCI     4, - ERROR CODE = I10)                                       YTL31120
ERRPNT BCI     7,(1H1,10X,43HAN ERROR HAS OCCURRED IN TL-01                 YTL31130
       BCI     7, - THE CODE = I10/1H0,9X,11HA TOTAL OF I6,                 YTL31140
       BCI     7,60H INSTRUCTIONS WERE SUCCESSFULLY EXECUTE                 YTL31150
       BCI     5,D PRIOR TO THE ERROR./1H )                                 YTL31160
```

```
PROGER BCI     6,(11X,58HTHE ABOVE COUNT WAS STARTED                          YTL31170
       BCI     6,WITH THE CORE PR-GRAM BELOW.  /1H0,                          YTL31180
       BCI     6,10X,49HTHE ERROR OCCURRED AT PHYSICA                         YTL31190
       BCI     5,L INSTRUCTION NUMBER I6/1H0)                                 YTL31200
*                                                                             YTL31210
*              INVERSION FORMATS                                              YTL31220
INVFMT BCI     9,(1H1,10X,26HINVERSION ERROR - CODE1 = I5,9H CODE2 =          YTL31230
       BCI     5,I5,16H SCALE FACTOR = E18.8)                                 YTL31240
DETPNT BCI     9,(1H1/1H0/1H0,4X,4HTHE I5,4H BY I5,15H MATRIX NUMBER          YTL31250
       BCI     6,I12,22H HAS A DETERMINANT OF F10.7,                          YTL31260
       BCI     6,17H TIMES 10 TO THE I12,7H POWER.)                           YTL31270
*                                                                             YTL31280
*              MMM20 FORMATS                                                  YTL31290
*                                                                             YTL31300
 NPAGE BCI     3,(1H1/1H0/1H )                                                YTL31310
 TLEIN BCI     2,(13A6,A2)                                                    YTL31320
TLEOUT BCI     4,(1H ,10X,13A6,A2/1H )                                        YTL31330
 MATNO BCI     6,(1H ,10X,14HMATRIX NUMBER I11/1H )                           YTL31340
ORDSUM BCI     6,(1H ,10X,6HORDER I4,4H BY I4/1H0,10X                         YTL31350
       BCI     4,,12HCHECK SUM = O12/1H )                                     YTL31360
NWPGE1 BCI     6,(1H1/1H0,10X,14HMATRIX NUMBER I11/1H                         YTL31370
       BCI     1,0/1H )                                                       YTL31380
NWPGE2 BCI     5,(1H1/1H0,10X,13A6,A2/1H0/1H )                                YTL31390
 FINAL BCI     6,(1H0/1H0,10X,20HEND OF MATRIX PRINT.                         YTL31400
       BCI     2,/1H0/1H0)                                                    YTL31410
INTPNT BCI     6,(1H0,4X,I4,2X,6(I12,1HB,3X),I12,1HB/                         YTL31420
       BCI     7,(         I23,1HB,3X,I12,1HB,3X,I12,1HB,3X,                  YTL31430
       BCI     7,I12,1HB,3X,I12,1HB,3X,I12,1HB,3X,I12,1HB))                   YTL31440
FLPPNT BCI     6,(1H0,4X,I4,2X,6(1PE13.6,3X),E13.6/(                          YTL31450
       BCI     4,        E24.6,6E16.6))                                       YTL31460
CSMPT  BCI     8,(1H0/1H0,10X,24HREQUESTED CHECKSUM ONLY./1H0                  YTL31470
       BCI     1,/1H0)                                                        YTL31480
NULMAT BCI     6,(1H ,10X,18HTHE MATRIX IS NULL)                              YTL31490
*              MMM23 FORMATS                                                  YTL31500
*                                                                             YTL31510
 LVLPT BCI     7,(11X,53HMATRIX ELEMENTS WITH MAGNITUDE LES                   YTL31520
       BCI     6,S THAN OR EQUAL TO 1PE16.6,31H ARE M                         YTL31530
       BCI     5,APPED AS A DECIMAL POINT./1H )                               YTL31540
 COLPT BCI     4,(1H ,16X,I3,10(7X,I3))                                       YTL31550
COLPT1 BCI     4,(1H ,18X,A1,10(9X,A1))                                       YTL31560
RWPRNT BCI     3,(1H ,9X,18A6,A2)                                             YTL31570
RWIDPT BCI     4,(1H ,5X,I3,1H+,18A6,A2)                                      YTL31580
FINLPT BCI     8,(1H0/1H0,10X,18HEND OF MATRIX MAP./1H0/1H0)                  YTL31590
*                                                                             YTL31600
       END                                                                    YTL99990
```

SUBROUTINE INV4DS

```
$IBMAP INVERT  650,DECK,M94/2                                          INV00000
*INV4DS        7094 DOUBLE PRECISION INVERSION ROUTINE                 INV00010
*                                                                      INV00020
*              CALL INV4DS,A,N,ERR1,ERR2,SCALE,DET,NDETXP              INV00030
*                   A = STARTING ADDRESS OF MATRIX - MUST BE EVEN      INV00040
*                   N = NUMBER OF ROWS (COLUMNS) OF MATRIX             INV00050
*                       IF NEGATIVE, PIVOT TERMS ARE PRINTED           INV00060
*                ERR1 = 0 IF INVERSION SUCCESSFUL                      INV00070
*                     = 1 IF OVERFLOW OCCURS                           INV00080
*                     = 2 IF MATRIX IS SINGULAR                        INV00090
*                     = 3 IF SCALED INVERSE CANNOT BE RE-SCALED        INV00100
*                     = 4 IF ROWS AND COLUMNS CANNOT BE RE-ARRANGED    INV00110
*                         THIS IS BASICALLY A MACHINE ERROR            INV00120
*                     = 5 IF (1,1) ELEMENT IS IN AN ODD LOCATION       INV00130
*                     = 6  IF ODD STORAGE TRAP DURING REDUCTION        INV00140
*                          THIS MUST BE A MACHINE ERROR                INV00150
*                     = 10 + I IF ERROR CODE 3 OCCURED SUBSEQUENT      INV00160
*                              TO ERROR CODE I                         INV00170
*                     = 20 + I IF ERROR CODE 4 OCCURED SUBSEQUENT      INV00180
*                              TO ERROR CODE I                         INV00190
*                ERR2 = CURRENT REDUCTION STAGE IF ERR1=1              INV00200
*                     = RANK OF MATRIX IF ERR1 =2                      INV00210
*                     = 0 OTHERWISE                                    INV00220
*               SCALE = 0 IF ERR1 NOT = 3                              INV00230
*                     = SCALING FACTOR IF ERR1=3                       INV00240
*                 DET= DECIMAL PART OF DETERMINANT IF ERR1.            INV00250
*                      NOT = 2 - A NUMBER GT OR E TO 1. AND LT 10.     INV00260
*                     = 0 IF ERR1=2                                    INV00270
*               DETXP = EXPONENT PART OF DETERMINANT IF ERR1           INV00280
*                      NOT = 2 - AN ADDRESS INTEGER GIVING             INV00290
*                      POWER OF TEN FOR DETERMINANT                    INV00300
*                      DETERMINANT = DET*10.**DETXP                    INV00310
       ENTRY   INV4DS                                                  INV00320
*                                                                      INV00330
*              SET UP CONSTANTS AND INITIALIZE CELLS                   INV00340
*                                                                      INV00350
INV4DS LMTM                                                            INV00360
       EFTM                                                            INV00370
       SXA     XR1,1                                                   INV00380
       SXA     XR2,2                                                   INV00390
       SXA     XR3,3                                                   INV00400
       SXA     XR4,4                                                   INV00410
       SXA     XR5,5                                                   INV00420
       SXA     XR6,6                                                   INV00430
       SXA     XR7,7                                                   INV00440
       STZ     0             CLEAR CELL 0 FOR FPT                      INV00450
       STZ     CHRCNG        CLEAR CELL FOR CHARACTERISTIC SCALING     INV00460
       STZ     ERRCD1        CLEAR ERROR CODE CELLS                    INV00470
       STZ     ERRCD2                                                  INV00480
       CAL     8             SAVE CELL 8 FOR (FPT)                     INV00490
       SLW     TEMP8                                                   INV00500
       CAL     FPTTRA        STORE TRANSFER FOR FPT ANALYSIS.          INV00510
       SLW     8                                                       INV00520
       STZ*    7,4           ZERO SCALING CELL                         INV00530
       STZ     DETXP         ZERO DETERMINANT EXPONENT                 INV00540
       CLA*    4,4                                                     INV00550
       STO     PRINT         SET INDICATOR FOR PRINT                   INV00560
```

```
        SSP                                                          INV00570
        XCA                                                          INV00580
        STQ     N                                                    INV00590
        MPY     N                                                    INV00600
        XCA                                                          INV00610
        ALS     1                                                    INV00620
        STO     TWONSQ          2*N**2                               INV00630
        ACL     3,4                                                  INV00640
        STA     AP2NSQ          A+2*N**2                             INV00650
        ACL     N                                                    INV00660
        STA     ADRINT          ADDRESS OF ROW INTEGERS +N           INV00670
        ACL     N                                                    INV00680
        STA     ADCINT          ADDRESS OF COLUMN INTEGERS +N        INV00690
        STA     MULTPL          ADDRESS OF REDUCTION FACTOR          INV00700
        ADD     =2                                                   INV00710
        STA     DET             ADDRESS OF DETERMINANT               INV00720
        ADD     =1                                                   INV00730
        STA     DETP1           ADDRESS OF DETERMINANT LEAST SIG.    INV00740
        ADD     =1                                                   INV00750
        STA     TEN             ADDRESS OF FLOATING POINT 10         INV00760
        CLA     3,4                                                  INV00770
        LBT                     TEST TO SEE IF STORAGE IS EVEN       INV00780
        TRA     *+2             EVEN STORAGE                         INV00790
        TRA     ERR5            ODD STORAGE - ERROR                  INV00800
        STA     ADA             ADDRESS OF A                         INV00810
        STA     ADPVEL          ADDRESS OF CURRENT PIVOT ELEMENT     INV00820
        ACL     N                                                    INV00830
        ACL     N                                                    INV00840
        STA     ADPVRW          ADDRESS OF CURRENT PIVOT ROW + 2*N   INV00850
        SUB     ADA                                                  INV00860
        STA     TWON            2*N                                  INV00870
        PAX     0,1                                                  INV00880
        SXD     REDUC4,1        STUFF LOOP CONTROL FOR REDUCE        INV00890
        CLA     =1.                                                  INV00900
        LDQ     =0                                                   INV00910
        DST*    DET             START DET OFF AT 1.                  INV00920
        CLA     =10.                                                 INV00930
        DST*    TEN             LOAD 10 IN DOUBLE PRECISION CELL     INV00940
        CLA     PRINT                                                INV00950
        TPL     NPRNT1                                               INV00960
        CALL    .FWRD.(.UN06..TITLE)                                 INV00970
NPRNT1  LXA     N,7                                                  INV00980
        SXD     RDLPED,7        TEST AT END OF REDUCTION LOOP        INV00990
        SXD     STOINT,7        PUT INTEGERS INTO ROW AND COLUMN     INV01000
        PXA     0,0                                                  INV01010
        AXT     1,1             INTERCHANGE ARRAYS                   INV01020
        PXA     0,1                                                  INV01030
        STO*    ADCINT          ADCINT HAS A TAG OF 1                INV01040
        STO*    ADRINT          ADRINT HAS A TAG OF 1                INV01050
        TXI     *+1,1,1         INTEGERS ARE STORRED BACKWARDS       INV01060
STOINT  TXL     *-4,1,**        N IN DECREMENT                       INV01070
*                                                                    INV01080
*               START OF REDUCTION LOOP                              INV01090
*               REDUCTION STAGE IS KEPT IN XR1                       INV01100
*               AT THE END OF THIS LOOP ALL INVERSION                INV01110
*                  ARITHMETIC IS DONE                                INV01120
*                                                                    INV01130
        AXT     1,1             INITIALIZE REDUCTION STAGE           INV01140
RDLPST  TSX     FINDLE,4        FIND LARGEST ELEMENT IN N-XR1+1 SUB MATRIX INV01150
        SXA     TEMPXR,1                                             INV01160
```

```
        PXA     0,1                                                     INV01170
        SUB     LGSTRW                                                  INV01180
        TZE     COLTST                                                  INV01190
        CLA*    DET                                                     INV01200
        CHS                                                             INV01210
        STO*    DET                                                     INV01220
        TSX     INRCHR,4                                                INV01230
        TSX     TEMPXR                                                  INV01240
        TSX     LGSTRW                                                  INV01250
        CLA     ADRINT                                                  INV01260
        SUB     LGSTRW                                                  INV01270
        STA     *+2                                                     INV01280
        STA     *+3                                                     INV01290
        CLA     **                                                      INV01300
        LDQ*    ADRINT                                                  INV01310
        STQ     **                                                      INV01320
        STO*    ADRINT                                                  INV01330
COLTST  PXA     0,1                                                     INV01340
        SUB     LGSTCL                                                  INV01350
        TZE     PRNTPV                                                  INV01360
        CLA*    DET                                                     INV01370
        CHS                                                             INV01380
        STO*    DET                                                     INV01390
        TSX     INRCHC,4                                                INV01400
        TSX     TEMPXR                                                  INV01410
        TSX     LGSTCL                                                  INV01420
        CLA     ADCINT                                                  INV01430
        SUB     LGSTCL                                                  INV01440
        STA     *+2                                                     INV01450
        STA     *+3                                                     INV01460
        CLA     **                                                      INV01470
        LDQ*    ADCINT                                                  INV01480
        STQ     **                                                      INV01490
        STO*    ADCINT                                                  INV01500
PRNTPV  CLA     PRINT                                                   INV01510
        TPL     STRTSC                                                  INV01520
        LFTM                                                            INV01530
        PXA     0,1                                                     INV01540
        TSX     .FCNV.,4                                                INV01550
        CLS     CHRCNG                                                  INV01560
        ADM*    ADPVEL                                                  INV01570
        LDQ     ADPVEL                                                  INV01580
        LLS     0                                                       INV01590
        TSX     .FCNV.,4                                                INV01600
        CLA*    ADRINT                                                  INV01610
        TSX     .FCNV.,4                                                INV01620
        CLA*    ADCINT                                                  INV01630
        TSX     .FCNV.,4                                                INV01640
        EFTM                                                            INV01650
*                                                                       INV01660
*                                                                       INV01670
STRTSC  TXH     SCLOOP+1,1,1                                            INV01680
        CAL*    ADPVEL          LARGEST ELEMENT = PIVOT ELEMENT         INV01690
        ANA     =0377000000000    MASK OUT ALL BUT CHARACTERISTIC       INV01700
        SUB     =0177000000000    FIND DIFFERENCE FROM 177              INV01710
        CHS                                                             INV01720
        STO     CHRCNG          SAVE CHARACTERISTIC SCALING             INV01730
        TZE     SCLOOP+1        BY-PASS SCALING IF NOT NEEDED           INV01740
        LXA     TWONSQ,7                                                INV01750
FSTSCL  NZT*    AP2NSQ          AP2NSQ HAS A TAG OF 7                   INV01760
```

```
        TRA     SCLOOP          NUMBER IS ZERO, DO NOT SCALE            INV01770
        CLA     CHRCNG          SCALE NUMBER                            INV01780
        ADM*    AP2NSQ                                                  INV01790
        TPL     *+3                                                     INV01800
        STZ*    AP2NSQ          IF NEGATIVE, UNDERFLOW OCCURRED         INV01810
        TRA     SCLOOP                                                  INV01820
        LDQ*    AP2NSQ          SCALING SUCCESSFUL, RESTORE SIGN        INV01830
        LLS     0                                                       INV01840
        STO*    AP2NSQ          STORE IN ORIGINAL LOCATION              INV01850
SCLOOP  TIX     FSTSCL,7,1      END OF SCALING LOOP                     INV01860
        TSX     REDUCE,4        PERFORM CURRENT REDUCTION STAGE         INV01870
        DLD*    ADPVEL                                                  INV01880
        TSX     SCALE,4         SCALE PIVOT ELEMENT                     INV01890
        DFMP*   DET                                                     INV01900
        TSX     SCALE,4         SCALE PRODUCT                           INV01910
        DST*    DET             RUNNING PRODUCT OF PIVOT ELEMENTS       INV01920
        LXA     TWON,7                                                  INV01930
        DLD*    ADPVEL                                                  INV01940
        DST*    MULTPL          KEEP PIVOT ELEMENT TEMP. IN MULTPL      INV01950
ADJUST  DLD*    ADPVRW          ADPVRW HAS TAG OF 7                     INV01960
        DFDP*   MULTPL                                                  INV01970
        DST*    ADPVRW          DIVIDE PIVOT ROW BY PIVOT ELEMENT       INV01980
        TIX     ADJUST,7,2                                              INV01990
        CLA     =1.                                                     INV02000
        LDQ     =0                                                      INV02010
        DFDP*   MULTPL                                                  INV02020
        DST*    ADPVEL          PUT 1/(PIV. ELEM) INTO PIVOT POSITION   INV02030
        TXI     *+1,1,1         INCREMENT TO NEW REDUCTION STAGE        INV02040
RDLPED  TXH     REDOVR,1,**     N IN DECREMENT-LOOP EXIT                INV02050
        CLA     ADPVEL                                                  INV02060
        ADD     =2                                                      INV02070
        ADD     TWON                                                    INV02080
        STA     ADPVEL          INCREMENT BY 2*N+2                      INV02090
        CLA     ADPVRW                                                  INV02100
        ADD     TWON                                                    INV02110
        STA     ADPVRW          INCREMENT BY 2*N                        INV02120
        TRA     RDLPST          ANOTHER REDUCTION STAGE                 INV02130
*                                                                       INV02140
*               ALL ARITHMETIC IS OVER NOW REARANGE ROWS                INV02150
*               AND COLUMNS                                             INV02160
*                                                                       INV02170
*               FIRST, INTERCHANGE COLUMNS ACCORDING TO ROW TABLE       INV02180
*                                                                       INV02190
REDOVR  LXA     N,7                                                     INV02200
        STZ     TEMP1           CLEAR FOR XR USE                        INV02210
        PXA     0,7                                                     INV02220
        PAX     0,1                                                     INV02230
RWSRCH  PXA     0,7                                                     INV02240
        SUB*    ADRINT          ADRINT HAS A TAG OF 1                   INV02250
        TZE     FOUNDR                                                  INV02260
        TIX     RWSRCH,1,1                                              INV02270
        TRA     ERR4                                                    INV02280
FOUNDR  PXA     0,7                                                     INV02290
        SXA     TEMP1,1                                                 INV02300
        SUB     TEMP1                                                   INV02310
        TZE     RWLPED          COLUMN IS IN CORRECT PLACE              INV02320
        SXA     TEMPXR,7                                                INV02330
        TSX     INRCHC,4                                                INV02340
        TSX     TEMPXR          ROW INTEGER FOUND                       INV02350
        TSX     TEMP1           LOCATION OF ROW INTEGER                 INV02360
```

```
*                INTERCHANGE ROW INTEGERS.                              INV02370
                                                                        INV02380
         LXA     TEMPXR,7                                               INV02390
         CLA     ADRINT                                                 INV02400
         SUB     TEMPXR                                                 INV02410
         STA     *+1                                                    INV02420
         CLA     **                                                     INV02430
         STO*    ADRINT          ADRINT HAS A TAG OF 1                  INV02440
         CLA     TEMPXR                                                 INV02450
         STO*    *-3                                                    INV02460
RWLPED   TIX     RWSRCH-2,7,1    LOOP FOR N ROWS.                       INV02470
*                                                                       INV02480
*                NOW INTERCHANCE ROWS ACCORDING TO COLUMN TABLE         INV02490
*                                                                       INV02500
         LXA     N,7                                                    INV02510
         PXA     0,7                                                    INV02520
         PAX     0,1                                                    INV02530
CLSRCH   PXA     0,7                                                    INV02540
         SUB*    ADCINT          ADCINT HAS A TAG OF 1                  INV02550
         TZE     FOUNDC                                                 INV02560
         TIX     CLSRCH,1,1                                             INV02570
         TRA     ERR4                                                   INV02580
FOUNDC   PXA     0,7                                                    INV02590
         SXA     TEMP1,1                                                INV02600
         SUB     TEMP1                                                  INV02610
         TZE     CLLPED          ROW IS IN CORRECT PLACE                INV02620
         SXA     TEMPXR,7                                               INV02630
         TSX     INRCHR,4                                               INV02640
         TSX     TEMPXR                                                 INV02650
         TSX     TEMP1                                                  INV02660
*                INTERCHANGE COLUMN INTEGERS                            INV02670
         LXA     TEMPXR,7                                               INV02680
         CLA     ADCINT                                                 INV02690
         SUB     TEMPXR                                                 INV02700
         STA     *+1                                                    INV02710
         CLA     **                                                     INV02720
         STO*    ADCINT          ADCINT HAS A TAG OF 1                  INV02730
         CLA     TEMPXR                                                 INV02740
         STO*    *-3                                                    INV02750
CLLPED   TIX     CLSRCH-2,7,1    LOOP FOR N COLUMNS                     INV02760
*                                                                       INV02770
*                INVERSE IS NOW IN THE CORRECT FORM - TIME TO           INV02780
*                RESCALE THE SCALED INVERSE                             INV02790
*                                                                       INV02800
         CLA     CHRCNG                                                 INV02810
         TZE     DSLPED+1        IF CHRCNG=0, RESCALING UNNECESSARY     INV02820
         TMI     RESCAL          IF NEGATIVE, NO OVERFLOW PROBLEMS      INV02830
         AXT     1,1             SET UP XR1 FOR FINDLE                  INV02840
         CLA     ADA             SET UP TO PIVOT ROW 1 FOR FINDLE       INV02850
         ADD     TWON                                                   INV02860
         STA     ADPVRW                                                 INV02870
         TSX     FINDLE,4        FIND LARGEST ELEMENT IN SCALED INVERSE INV02880
         CLA     CHRCNG                                                 INV02890
         ADM     TEMP1           FINDLE STORES LARGEST IN TEMP1         INV02900
         PBT                     WAS THERE OVERFLOW                     INV02910
         TRA     *+2             NO                                     INV02920
         TRA     ERR3            YES                                    INV02930
RESCAL   LXA     TWONSQ,7                                               INV02940
         CLA     CHRCNG                                                 INV02950
         ADM*    AP2NSQ          AP2NSQ HAS A TAG OF 7                  INV02960
         ZET*    AP2NSQ
```

```
        TPL     *+3             NO UNDERFLOW IF POSITIVE                INV02970
        STZ*    AP2NSQ          UNDERFLOW, STORE ZERO                   INV02980
        TRA     RSCLED                                                  INV02990
        LDQ*    AP2NSQ                                                  INV03000
        LLS     0                                                       INV03010
        STO*    AP2NSQ                                                  INV03020
RSCLED  TIX     RESCAL+1,7,1    LOOP FOR 2*N**2 NUMBERS                 INV03030
*                                                                       INV03040
*               NOW RESCALE THE DETERMINANT                             INV03050
*                                                                       INV03060
        NZT*    DET             IS MATRIX SINGULAR                      INV03070
        TRA     DSLPED+1        IF SINGULAR, RETURN                     INV03080
        CLA     CHRCNG                                                  INV03090
        CHS                                                             INV03100
        STO     CHRCNG                                                  INV03110
        LXA     N,7                                                     INV03120
STDSCL  CLA     CHRCNG                                                  INV03130
        ADM*    DET                                                     INV03140
        PBT                                                             INV03150
        TRA     *+2             NO OVERFLOW                             INV03160
        TRA     DIVDET          OVERFLOW, DIVIDE BY 10                  INV03170
        TMI     MLTDET          UNDERFLOW, MULTIPLY BY 10               INV03180
        LDQ*    DET                                                     INV03190
        LLS     0                                                       INV03200
        STO*    DET                                                     INV03210
        CLA     CHRCNG          SCALE LEAST HALF                        INV03220
        ADM*    DETP1                                                   INV03230
        TPL     *+2             POSITIVE IF NO UNDERFLOW                INV03240
        PXA     0,0             ZERO LEAST HALF IF UNDERFLOW            INV03250
        LLS     0               RESTORE SIGN TO LEAST HALF              INV03260
        STO*    DETP1                                                   INV03270
        DLD*    DET                                                     INV03280
        TSX     SCALE,4                                                 INV03290
        DST*    DET                                                     INV03300
        TRA     DSLPED          GO TO END OF LOOP                       INV03310
DIVDET  DLD*    DET             DIVIDE DET BY TEN                       INV03320
        DFDP*   TEN                                                     INV03330
        DST*    DET                                                     INV03340
        TRA     STDSCL                                                  INV03350
MLTDET  DLD*    DET             MULTIPLY DET BY TEN                     INV03360
        DFMP*   TEN                                                     INV03370
        DST*    DET                                                     INV03380
        TRA     STDSCL                                                  INV03390
DSLPED  TIX     STDSCL,7,1      LOOP N TIMES FOR COMPLETE SCALING       INV03400
*                                                                       INV03410
*               END OF ALL ARITHMETIC NOW TIME TO ARRANGE               INV03420
*               ERROR CODES AND SUCH THEN RETURN                        INV03430
        CLA     PRINT           DID WE P-INT                            INV03440
        TPL     XR1                                                     INV03450
        CALL    .FFIL.                                                  INV03460
   XR1  AXT     **,1                                                    INV03470
   XR2  AXT     **,2                                                    INV03480
   XR3  AXT     **,3                                                    INV03490
   XR4  AXT     **,4                                                    INV03500
   XR5  AXT     **,5                                                    INV03510
   XR6  AXT     **,6                                                    INV03520
   XR7  AXT     **,7                                                    INV03530
        CAL     TEMP8           RESTORE CELL 8                          INV03540
        SLW     8                                                       INV03550
        DLD*    DET                                                     INV03560
```

```
        DST*    8,4                                                     INV03570
        CLA     DETXP                                                   INV03580
        STO*    9,4                                                     INV03590
        CLA     ERRCD1                                                  INV03600
        STO*    5,4                                                     INV03610
        CLA     ERRCD2                                                  INV03620
        STO*    6,4                                                     INV03630
        TRA     10,4                                                    INV03640
*                                                                       INV03650
*               FOLLOWING ARE ALL THE ERROR STOPS                       INV03660
*                                                                       INV03670
  ERR1  CLA     =1              OVERFLOW RETURN                         INV03680
        STO     ERRCD1                                                  INV03690
        SXA     ERRCD2,1        SAVE REDUCTION STAGE                    INV03700
        STZ*    DET             ZERO DETERMINANT                        INV03710
        STZ*    DETP1                                                   INV03720
        STZ     DETXP                                                   INV03730
        TRA     REDOVR          TRANSFER BACK TO INTERCHANGE ROWS       INV03740
*                               AND COLUMNS, AND RESCALE                INV03750
  ERR2  CLA     =2                                                      INV03760
        STO     ERRCD1                                                  INV03770
        PXA     0,1                                                     INV03780
        SUB     =1                                                      INV03790
        STO     ERRCD2          RANK OF MATRIX =XR1-1                   INV03800
        TRA     ERR1+3          TRANSFER BACK TO INTERCHANGE            INV03810
*                               ROWS AND COLUMNS, AND RESCALE           INV03820
  ERR3  CLA     ERRCD1                                                  INV03830
        TNZ     ERR3PL                                                  INV03840
        CLA     =3                                                      INV03850
        STO     ERRCD1                                                  INV03860
        CLA     =1.             =0201400000000                          INV03870
        ADD     CHRCNG                                                  INV03880
        LXA     XR4,4                                                   INV03890
        STO*    7,4             STORE SCALING FACTOR                    INV03900
        TRA     RSCLED+1        GO BACK AND SCALE DETERMINANT           INV03910
  ERR4  CLA     ERRCD1          CANNOT RE-ARRANGE ROWS,COLUMNS -MACH. ERRORINV03920
        TNZ     ERR4PL                                                  INV03930
        CLA     =4                                                      INV03940
        STO     ERRCD1          CANNOT RE-ORDER ROWS AND COLUMNS        INV03950
        TRA     DSLPED+1        RETURN IMMEDIATELY                      INV03960
  ERR5  CLA     =5              A(1,1) NOT IN EVEN CELL                 INV03970
        STO     ERRCD1                                                  INV03980
        TRA     DSLPED+1        RETURN IMMEDIATELY                      INV03990
  ERR6  CLA     =6                                                      INV04000
        TRA     ERR5+1                                                  INV04010
ERR3PL  ADD     =10                                                     INV04020
        TRA     ERR3+3                                                  INV04030
ERR4PL  ADD     =20                                                     INV04040
        TRA     ERR4+3                                                  INV04050
*                                                                       INV04060
*               STORAGE FOR PROGRAM CONSTANTS                           INV04070
*                                                                       INV04080
 TEMP8  PZE     0               STORAGE FOR CORE LOCATION 8 (DEC)       INV04090
FPTTRA  TRA     FLPSPL                                                  INV04100
 CELL0  PZE     0               STORAGE FOR FLPSPL CODE                 INV04110
   DET  PZE     0               ADDRESS OF DETERMINANT                  INV04120
 DETP1  PZE     0               ADDRESS OF LEAST SIG. PART OF DET.      INV04130
MULTPL  PZE     0               ADDRESS OF REDUCTION FACTOR             INV04140
   TEN  PZE     0               ADDRESS OF FLOATING POINT 10.           INV04150
 DETXP  PZE     0               CELL FOR DETERMINANT EXPONENT           INV04160
```

```
      N PZE     0                                                        INV04170
    TWON PZE    0              2*N                                       INV04180
  TWONSQ PZE    0              2*N**2                                    INV04190
     ADA PZE    0              ADDRESS OF MATRIX                         INV04200
  ADPVEL PZE    0              ADDRESS OF CURRENT PIVOT ELEMENT          INV04210
  ADPVRW PZE    0,7            ADDRESS OF CURRENT PIVOT ROW +2N          INV04220
  ADOPEL PZE    0,6            ADDRESS OF ELEMENT BEING REDUCED          INV04230
  ADOPRW PZE    0,7            ROW OF ADOPEL                             INV04240
  AP2NSQ PZE    0,7            ADDRESS OF 1ST CELL BEYOND MATRIX         INV04250
  ADRINT PZE    0,1            ADDRESS OF ROW INTEGERS + N               INV04260
  ADCINT PZE    0,1            ADDRESS OF COLUMN INTEGERS + N            INV04270
  LGSTRW PZE    0              ROW INDEX OF LARGEST ELEMENT IN SUB-MATRIX INV04280
  LGSTCL PZE    0              COLUMN INDEX OF LARGEST ELEM. IN SUB-MATRIXINV04290
  CHRCNG PZE    0              CHARACTERISTIC SCALE FACTOR               INV04300
  ERRCD1 PZE    0              ERROR CODE 1                              INV04310
  ERRCD2 PZE    0              ERROR CODE 2                              INV04320
   TEMP1 PZE    0                                                        INV04330
   TEMP2 PZE    0                                                        INV04340
  TEMPXR PZE    0              TEMPORARY FOR ADDRESS ONLY                INV04350
  TEMPAC PZE    0              TEMPORARY STORAGE FOR FLPSPL              INV04360
  TITLE  BCI    6,(1H1,10X,93HBELOW ARE THE PIVOT TERM                   INV04370
         BCI    7,S DERIVED DURING INVERSION AND THEIR ORIGI             INV04380
         BCI    6,NAL ROW - COLUMN LOCATIONS./1H0,22X,                   INV04390
         BCI    6,9HREDUCTION,8X,5HPIVOT,9X,3HROW,5X,                    INV04400
         BCI    6,6HCOLUMN/24X,5HSTAGE,11X,4HTERM,9X,                    INV04410
         BCI    6,6HNUMBER,3X,6HNUMBER/(1H0,24X,I3,6X,                   INV04420
         BCI    4,1PE14.6,6X,I3,6X,I3))                                  INV04430
   PRINT PZE    0                                                        INV04440
*               CLOSED SUBROUTINE TO INTERCHANGE ROWS                    INV04450
*               TWO ARGUMENTS -I,J=ROWS TO BE MOVED                      INV04460
*                                                                        INV04470
*                                                                        INV04480
INRCHR LDQ*     1,4                                                      INV04490
         MPY    TWON                                                     INV04500
         XCA                                                             INV04510
         ADD    ADA             ADDRESS OF ROW I + 2N                    INV04520
         STA    INR1                                                     INV04530
         STA    INR1+3                                                   INV04540
         LDQ*   2,4                                                      INV04550
         MPY    TWON                                                     INV04560
         XCA                                                             INV04570
         ADD    ADA             ADDRESS OF ROW I + 2 N                   INV04580
         STA    INR1+1                                                   INV04590
         STA    INR1+2                                                   INV04600
         LXA    TWON,7                                                   INV04610
   INR1  CLA    **,7            ROW I                                    INV04620
         LDQ    **,7            ROW J                                    INV04630
         STO    **,7            ROW J                                    INV04640
         STQ    **,7            ROW I                                    INV04650
         TIX    INR1,7,1                                                 INV04660
         TRA    3,4             DONE                                     INV04670
*                                                                        INV04680
*               CLOSED SUBROUTINE TO INTERCHANGE COLUMNS                 INV04690
*               TWO ARGUMENTS -I,J = COLUMNS TO BE MOVED                 INV04700
*                                                                        INV04710
INRCHC CLA      ADA                                                      INV04720
         SUB    =1                                                       INV04730
         ADD    TWONSQ                                                   INV04740
         STO    TEMP2           A-1+2*N**2                               INV04750
         CLA*   1,4                                                      INV04760
```

```
         ALS     1                                                    INV04770
         ADD     TEMP2           ADDRESS OF COLUMN I+2N**2+1          INV04780
         STA     INC1                                                 INV04790
         STA     INC1+3                                               INV04800
         CLA*    2,4                                                  INV04810
         ALS     1                                                    INV04820
         ADD     TEMP2           ADDRESS OF COLUMN J + 2N**2+1        INV04830
         STA     INC1+1                                               INV04840
         STA     INC1+2                                               INV04850
         LXA     TWONSQ,7                                             INV04860
         LXA     TWON,6                                               INV04870
         SXD     INC2,6                                               INV04880
         LXA     =2,6                                                 INV04890
  INC1   CLA     **,7            COLUMN I                             INV04900
         LDQ     **,7            COLUMN J                             INV04910
         STO     **,7            COLUMN J                             INV04920
         STQ     **,7            COLUMN I                             INV04930
         TXI     *+1,7,1         SMALL LOOP TO EXCHANGE LEAS. SIG.    INV04940
         TIX     INC1,6,1                                             INV04950
         TXI     *+1,7,-2                                             INV04960
  INC2   TIX     INC1-1,7,0      MAIN LOOP -2N IN DECREMENT           INV04970
         TRA     3,4             DONE                                 INV04980
*                                                                     INV04990
*        CLOSED SUBROUTINE TO FIND THE LARGEST                        INV05000
*        MAGNITUDE IN THE N-XR1+1 SQUARE SUB-MATRIX                   INV05010
*        CURRENT REDUCTION STAGE IS IN XR1                            INV05020
*        ROW LOCATION OF L.E. = LGSTRW                                INV05030
*        COLUMN LOCATION OF L.E. = LGSTCL                             INV05040
*                                                                     INV05050
FINDLE   PXA     0,1                                                  INV05060
         SUB     N                                                    INV05070
         SUB     =1                                                   INV05080
         PAX     0,3             COUNT OF ROWS IN XR3=N+1-XR1         INV05090
         ALS     1                                                    INV05100
         STA     TEMPXR          TEMPXR=2(N+1-XR1)-TO GO INTO XR7     INV05110
         SXA     LGSTRW,1        START LOCATION AT PIVOT ELEMENT      INV05120
         SXA     LGSTCL,1                                             INV05130
         CLA     ADPVRW                                               INV05140
         STA     FLE1+1                                               INV05150
         STZ     TEMP1           KEEP LARGEST IN TEMP1                INV05160
         LXA     TEMPXR,7                                             INV05170
  FLE1   PXA     0,0                                                  INV05180
         ADM     **,7                                                 INV05190
         CAS     TEMP1                                                INV05200
         TRA     *+3             GREATER - EXCHANGE                   INV05210
         TRA     *+5                                                  INV05220
         TRA     *+4                                                  INV05230
         STO     TEMP1                                                INV05240
         SXA     LGSTRW,3                                             INV05250
         SXA     LGSTCL,7                                             INV05260
         TIX     FLE1,7,2        LOOP BACK IF ROW NOT DONE            INV05270
         CAL     FLE1+1                                               INV05280
         ADD     TWON                                                 INV05290
         STA     FLE1+1          INCREMENT TO NEW ROW                 INV05300
         TIX     FLE1-1,3,1      LOOP BACK IF MORE ROWS               INV05310
         NZT     TEMP1           DONE, IS LARGEST ELEMENT = 0         INV05320
         TRA     ERR2            SINGULAR MATRIX ERROR RETURN         INV05330
*                                 ADJUST LGSTRW, CL                   INV05340
         CLS     LGSTRW                                               INV05350
         ADD     N                                                    INV05360
```

```
        ADD     =1                                                      INV05370
        STA     LGSTRW                                                  INV05380
        CLS     LGSTCL                                                  INV05390
        ARS     1                                                       INV05400
        ADD     N                                                       INV05410
        ADD     =1                                                      INV05420
        STA     LGSTCL                                                  INV05430
        TRA     1,4                                                     INV05440
*                                                                       INV05450
*               CLOSED SUBROUTINE TO PERFORM ONE REDUCTION              INV05460
*               STAGE - ONE STAGE CONSISTS OF PRODUCING                 INV05470
*               ZEROS ON THE PIVOT COLUMN AND A 1 IN THE                INV05480
*               PIVOT POSITION- ZEROS AND 1 ARE ONLY UNDERSTOOD         INV05490
*               DUE TO STORAGE OVER ITSELF                              INV05500
*               XR1=REDUCTION STAGE                                     INV05510
*                                                                       INV05520
REDUCE  LXA     TWONSQ,6                                                INV05530
        SXA     TEMPXR,1                                                INV05540
        CLA     ADA                                                     INV05550
        ADD     TWON                                                    INV05560
        STA     ADOPRW          2N + ADDRESS OF ROW BEING REDUCED       INV05570
        SUB     TWON                                                    INV05580
        ADD     TEMPXR                                                  INV05590
        ADD     TEMPXR                                                  INV05600
        SUB     =2                                                      INV05610
        ADD     TWONSQ          2N**2 + ADDRESS OF ELEMENT BEING ZEROED INV05620
        STA     ADOPEL                                                  INV05630
        AXT     1,5             COUNT OF ROWS BEING REDUCED             INV05640
REDUC1  PXA     0,5                                                     INV05650
        SUB     TEMPXR                                                  INV05660
        TZE     REDUC3          DO NOT OPERATE ON PIVOT ROW             INV05670
        DLD*    ADOPEL          ADOPEL HAS A TAG OF 6                   INV05680
        TZE     REDUC3          IF ALREADY ZERO, NO REDUCTION NECESSARY INV05690
        DFDP*   ADPVEL                                                  INV05700
        CHS                     USE NEGATIVE RATIO                      INV05710
        DST*    MULTPL          REDUCTION FACTOR                        INV05720
        LXA     TWON,7                                                  INV05730
REDUC2  DLD*    ADPVRW          ADPVRW HAS A TAG OF 7                   INV05740
        DFMP*   MULTPL                                                  INV05750
        DFAD*   ADOPRW          ADOPRW HAS A TAG OF 7                   INV05760
        DST*    ADOPRW                                                  INV05770
        TIX     REDUC2,7,2                                              INV05780
        DLD*    MULTPL          PUT REDUCTION FACTOR WHERE              INV05790
        DST*    ADOPEL          ZERO WAS PRODUCED                       INV05800
REDUC3  CLA     ADOPRW                                                  INV05810
        ADD     TWON                                                    INV05820
        STA     ADOPRW          GO TO NEXT ROW                          INV05830
        TXI     *+1,5,1         XR5 = ROW ABOUT TO BE REDUCED           INV05840
REDUC4  TIX     REDUC1,6T**     2N IN DECREMENT                         INV05850
        TRA     1,4             DONE                                    INV05860
*                                                                       INV05870
*               CLOSED SUBROUTINE TO SCALE AN INCOMING                  INV05880
*               NUMBER SO THAT IT IS LESS THAN 10. AND                  INV05890
*               GREATER THAN OR EQUAL TO 1. DIVISIONS AND               INV05900
*               MULTIPLICATIONS KEPT TRACK OF IN DETXP.                 INV05910
*               NUMBER EXPECTED IN AC AND MQ-RESULT LEFT THERE          INV05920
*                                                                       INV05930
 SCALE  STO     TEMPAC                                                  INV05940
        SSP                                                             INV05950
        LRS     0                                                       INV05960
```

```
SCALE1 CAS*    TEN                                                    INV05970
       TRA     DIVIDE                                                 INV05980
       TRA     DIVIDE                                                 INV05990
TSTONE CAS     =1.                                                    INV06000
       TRA     CHKSGN                                                 INV06010
       TRA     CHKSGN                                                 INV06020
       DFMP*   TEN                                                    INV06030
       DST*    MULTPL          TEMPORARY STORAGE                      INV06040
       CLA     DETXP                                                  INV06050
       SUB     =1                                                     INV06060
       STO     DETXP                                                  INV06070
       DLD*    MULTPL                                                 INV06080
       TRA     TSTONE                                                 INV06090
DIVIDE DFDP*   TEN                                                    INV06100
       DST*    MULTPL                                                 INV06110
       CLA     DETXP                                                  INV06120
       ADD     =1                                                     INV06130
       STO     DETXP                                                  INV06140
       DLD*    MULTPL                                                 INV06150
       TRA     SCALE1                                                 INV06160
CHKSGN DST*    MULTPL                                                 INV06170
       CLA     TEMPAC                                                 INV06180
       LRS     0                                                      INV06190
       CLA*    MULTPL                                                 INV06200
       LLS     0                                                      INV06210
       TRA     1,4                                                    INV06220
*                                                                     INV06230
*              FLOATING POINT SPILL ROUTINE TO ANALYZE                INV06240
*              OVER/UNDER FLOW DURING DOUBLE PRECISION                INV06250
*              OPERATIONS                                             INV06260
*                                                                     INV06270
FLPSPL STO     TEMPAC                                                 INV06280
       CLA     0                                                      INV06290
       STD     CELL0                                                  INV06300
       CLA     CELL0                                                  INV06310
       CAS     =3B17                                                  INV06320
       TRA     TSTSTO          CHECK TO BE SURE IT IS NOT A STORAGE TRAP INV06330
       TRA     *+4             UNDERFLOW IN AC AND MQ                 INV06340
       CLA     TEMPAC          UNDERFLOW IN MQ ONLY                   INV06350
       LDQ     =0                                                     INV06360
       TRA*    0               RETURN                                 INV06370
       PXA     0,0                                                    INV06380
       TRA     *-3                                                    INV06390
TSTSTO CAS     =7B17                                                  INV06400
       TRA     ERR6            ODD STORAGE TRAP                       INV06410
       TRA     ERR1            OVERFLOW IN AC OR AC AND MQ            INV06420
       TRA     ERR1                                                   INV06430
*                                                                     INV06440
       END                                                            INV99990
```

SUBROUTINE DATASB

```
$IBMAP  DATASB   4,DECK,M94/2                                  DAT00000
        ENTRY    DATA                                          DAT00010
 DATA   PZE      12000                                         DAT00020
        PZE      8000                                          DAT00030
        BSS      11999                                         DAT00040
        END                                                    DAT99990
```

SUBROUTINE KRD

```
$IBMAP FKRD    100,DECK,XR7,M94                                                KRD00010
*KRD   7090 FORTRAN LIBRARY / BCD TAPE INPUT ROUTINE                           KRD00020
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * KRD00030
*                                                                              KRD00040
*     M = KRD (A,NZ,IRROR,NFC,NCOL,NWT,LOGIC)                                 KRD00050
*                                                                              KRD00060
*           A - BEGINNING STORAGE LOCATION.                                    KRD00070
*          NZ - IF ZERO, WILL EXIT AFTER EACH CARD.                            KRD00080
*               IF ONE, WILL EXIT AFTER END-FILE-CODE(12 PCH-COL 72)           KRD00090
*       IRROR - LOCATION WHERE ERROR CODES ARE STORED.                         KRD00100
*               0  SUCCESSFUL.                                                 KRD00110
*               1  PHYISCAL END-OF-FILE ON TAPE.                               KRD00120
*               2  NFC*NCOL GREATER THAN 72.                                   KRD00130
*               3  IMPROPER NUMBER FOLLOWING B.                               KRD00140
*               4  ILLEGAL CHARACTER IN DATA FIELD.                           KRD00150
*               5  DIVIDE CHECK IN CONVERTING TO FX PT.                       KRD00160
*               6  OVERFLOW CONVERTING TO FLOATING BINARY.                    KRD00170
*               7  EFFECTIVE POWER GREATER THAN 38.                           KRD00180
*               8  FIELD VALUE GREATER THAN 34,359,738,367.                   KRD00190
*               9  ILLEGAL CHARACTER IN COLUMN 72.                            KRD00200
*              10  MACHINE FAILURE.                                           KRD00210
*              11  END OF BUFFER ERROR READING                                KRD00220
*         NFC - NUMBER OF FIELDS TO CONVERT.                                   KRD00230
*        NCOL - NUMBER OF COLUMNS PER FIELD.                                   KRD00240
*         NWT - NUMBER OF WORDS FOLLOWING THE CONVERSION FIELDS TO             KRD00250
*                 TRANSFER AS UNCONVERTED BCD WORDS.                           KRD00260
*       LOGIC - LOGICAL TAPE NUMBER                                            KRD00270
*           M - NUMBER OF CARDS SUCCESSFULLY CONVERTED.                        KRD00271
*                                                                              KRD00272
* CONVERSION OF BCD CARD IMAGES IN CORE                                        KRD00273
*     CALL KRDG (A,B,IRROR,NFC,NCOL)                                           KRD00274
*                                                                              KRD00275
*               B - LOCATION OF 1ST WORD OF BCD FIELDS                         KRD00276
*                     TO BE CONVERTED.                                         KRD00277
*                     2ND WORD AT B+1, ETC.                                    KRD00280
*                                                                              KRD00290
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * KRD00300
       REM                                                                     KRD00310
       ENTRY   KRD                                                             KRD00315
       ENTRY   KRDG                                                            KRD00320
       LDIR                                                                    KRD00330
KRD    STZ     CLAUDE                                                          KRD00331
       TRA     CART                                                            KRD00332
KRDG   STL     CLAUDE                                                          KRD00333
CART   SAVE    4,1,2,I                                                         KRD00340
PRE    TXI     *+1,4,-2                                                        KRD00350
       SXA     IR4,4                                                           KRD00360
       SXA     IR2,2                                                           KRD00370
       SXA     IR1,1                                                           KRD00380
       STZ     CDCT                                                            KRD00381
       NZT     CLAUDE                                                          KRD00382
       TRA     *+3                                                             KRD00383
       CLA     2,4                                                             KRD00384
       TRA     LOCATE+5                                                        KRD00390
       CLA*    7,4                                                             KRD00400
       STO     ALPHA+4                                                         KRD00410
       CALL    .FVIO.(ALPHA+4,ALPHA+5)
```

```
        CLA     ALPHA+5                                 KRD00420
        STA     ROPEN+1                                 KRD00430
        STA     LOCATE+1                                KRD00440
        STA     LOCTWO+1                                KRD00450
        PAC     0,4                                     KRD00460
        LDI     1,4                                     KRD00470
        LFT     040000                                  KRD00480
        TRA     LOCATE                                  KRD00490
ROPEN   TSX     .OPEN,4                                 KRD00500
        MZE     **                                      KRD00510
LOCATE  TSX     .READ,4                                 KRD00520
        PZE     **,,ER11                                KRD00530
        PZE     ER1,,ER10                               KRD00540
        IORTN   **,,**                                  KRD00550
        CLA     *-1                                     KRD00560
        STA     LDQ                                     KRD00570
        STA     ALOOP                                   KRD00580
        ADD     =11                                     KRD00590
        STA     SETAD                                   KRD00600
        STA     LAST-2                                  KRD00610
        LXA     IR4,4                                   KRD00620
        CLA*    2,4                                     KRD00630
        STO     TEST                                    KRD00640
        CLA     BFILL                                   KRD00650
        STD     BDEC                                    KRD00660
        CLA     SWONE                                   KRD00670
KRDA    STA     SW                                      KRD00680
        AXT     10,1                                    KRD00690
SWALT   PXA     0,1                                     KRD00700
        STA*    3,4                                     KRD00710
        CLA     1,4                                     KRD00720
        STA     STORE                                   KRD00730
        CLA*    4,4                                     KRD00740
        STA     NFC                                     KRD00750
        XCA                                             KRD00760
        CLA*    5,4                                     KRD00770
        STO     NCOL                                    KRD00780
        MPY     NCOL                                    KRD00790
        XCA                                             KRD00800
        LXA     ZERO,4                                  KRD00810
        CAS     =72                                     KRD00820
        TRA     ER2                                     KRD00830
        TXI     *+1,4,-1                                KRD00840
        SXA     LAST,4                                  KRD00850
AGAIN   LXA     =077777,4                               KRD00860
        TOV     *+1                                     KRD00870
        LXA     NCOL,2                                  KRD00880
        SXD     NFCC,4                                  KRD00890
        CAL     =017                                    KRD00900
        STD     WDCT                                    KRD00910
        STZ     PWR                                     KRD00920
        ANA     **                                      KRD00930
        LAC     NFC,4                                   KRD00940
LAST    AXT     **,1                                    KRD00950
        TZE     *+3                                     KRD00960
        AXT     0,1                                     KRD00970
        PAC     **,4                                    KRD00980
        TXL     IR4,4,0                                 KRD00990
        SXD     ALL-1,1                                 KRD01000
        SXD     NFT,4                                   KRD01010
```

```
        TXL     *+2,4,-2                                        KRD01020
        STD     NFCC                                            KRD01030
BACK    CLA     *                                               KRD01040
        STD     STAR1                                           KRD01050
CLS     CLS     =0                                              KRD01060
        STO     ALPHA+1                                         KRD01070
        STO     ALPHA+2                                         KRD01080
LOOP    LXD     WDCT,4                                          KRD01090
LDQ     LDQ     **,4                                            KRD01100
        AXT     6,1                                             KRD01110
BDEC    TXI     *+1,4,-1                                        KRD01120
        SXD     WDCT,4                                          KRD01130
ZERO    PXD     0,0                                             KRD01140
        LGL     6                                               KRD01150
SWONE   PAX     SW+1,4                                          KRD01160
        TXL     NUM,4,9                                         KRD01170
        TXL     ER4,4,15                                        KRD01180
        TXL     NEXT,4,16                                       KRD01190
        TXL     ER4,4,17                                        KRD01200
        TXL     FIXED,4,18                                      KRD01210
        TXL     ER4,4,20                                        KRD01220
        TXL     FLOAT,4,21                                      KRD01230
        TXL     ER4,4,26                                        KRD01240
        TXL     POINT,4,27                                      KRD01250
        TXL     ER4,4,31                                        KRD01260
        TXL     MINUS,4,42                                      KRD01270
        TXL     ER4,4,47                                        KRD01280
        TXL     NEXT,4,48                                       KRD01290
ER4     AXT     4,4             ILLEGAL CHARACTER               KRD01300
        TRA     SCRAM                                           KRD01310
ER1     AXT     1,4             END OF FILE                     KRD01320
        TRA     SCRAM                                           KRD01330
ER2     AXT     2,4             CF 772                          KRD01340
        TRA     SCRAM                                           KRD01350
ER5     AXT     5,4             SCALING TOO LARGE               KRD01360
        TRA     SCRAM                                           KRD01370
ER6     AXT     6,4             FL. PT. OVERFLOW                KRD01380
        TRA     SCRAM                                           KRD01390
ER7     AXT     7,4             FL. EXP. OUT OF RANGE           KRD01400
        TRA     SCRAM                                           KRD01410
ER9     AXT     9,4             ILLEGAL CHAR. COL. 72           KRD01420
        TRA     SCRAM                                           KRD01430
ER10    AXT     10,4                                            KRD01440
        TRA     SCRAM                                           KRD01450
ER11    AXT     11,4                                            KRD01460
SCRAM   PXA     0,4                                             KRD01470
        LXA     IR4,4                                           KRD01480
        STO*    3,4                                             KRD01490
        PXD     0,0                                             KRD01500
        TRA     IR2                                             KRD01510
POINT   STZ     ALPHA+2                                         KRD01520
        TRA     NEXT                                            KRD01530
MINUS   TXL     NOTL,4,34                                       KRD01540
        TXH     NOTL,4,35                                       KRD01550
        TXH     NOTL,2,1                                        KRD01560
        LXA     IR4,4                                           KRD01570
        CLA     1,4                                             KRD01580
        ADD     =1B17                                           KRD01590
        ADD     ALPHA+1                                         KRD01600
        STA     STORE                                           KRD01610
```

```
        TRA     NF                                                  KRD01620
NOTL    SUB     =040                                                KRD01630
        TXL     *+2,4,42                                            KRD01640
        SUB     =10                                                 KRD01650
        LXD     CLS,4                                               KRD01660
        SXD     STAR1,4                                             KRD01670
NUM     STO     ALPHA                                               KRD01680
        CLA     ALPHA+1                                             KRD01690
        ALS     3                                                   KRD01700
        ADD     ALPHA+1                                             KRD01710
        ADD     ALPHA+1                                             KRD01720
        ADD     ALPHA                                               KRD01730
        SSP                                                         KRD01740
        TNO     *+3                                                 KRD01750
        AXT     8,4             CONVERSION OVERFLOW                 KRD01760
        TRA     SCRAM                                               KRD01770
        STO     ALPHA+1                                             KRD01780
        CLA     ALPHA+2                                             KRD01790
        TMI     NEXT                                                KRD01800
        CLA     PWR                                                 KRD01810
        SUB     =1                                                  KRD01820
        STO     PWR                                                 KRD01830
NEXT    TNX     ENDFD,2,1                                           KRD01840
        TIX     ZERO,1,1                                            KRD01850
        TRA     LOOP                                                KRD01860
FIXED   BSS     0                                                   KRD01870
FLOAT   SSM                                                         KRD01880
        STO     ALPHA+2                                             KRD01890
        XEC     STAR1                                               KRD01900
        STO     ALPHA+3                                             KRD01910
        CLA     PT2                                                 KRD01920
        STA     NEXT                                                KRD01930
        CLA     *                                                   KRD01940
        STD     STAR1                                               KRD01950
        CLS     =0                                                  KRD01960
        STO     ALPHA+1                                             KRD01970
        TRA     NEXT                                                KRD01980
PT2     PZE     ENDEX                                               KRD01990
PT1     PZE     ENDFD                                               KRD02000
ENDEX   CLA     PT1                                                 KRD02010
        STA     NEXT                                                KRD02020
        LXA     ALPHA+2,4                                           KRD02030
        XEC     STAR1                                               KRD02040
        TIX     ERTN,4,19                                           KRD02050
BRTN    TNZ     *+3                                                 KRD02060
        TPL     *+2                                                 KRD02070
        CLA     =35                                                 KRD02080
        SUB     =35                                                 KRD02090
        TMI     *+4                                                 KRD02100
        TZE     *+3                                                 KRD02110
ER3     AXT     3,4             NEGATIVE SCALING                    KRD02120
        TRA     SCRAM                                               KRD02130
        STA     SHIFT                                               KRD02140
        STQ     ALPHA+2                                             KRD02150
        LDQ     ALPHA+3                                             KRD02160
        PXD     0,0                                                 KRD02170
SHIFT   LLS     **                                                  KRD02180
        LXA     PWR,4                                               KRD02190
        DCT                                                         KRD02200
        TRA     *+1                                                 KRD02210
```

```
        DVP     10T,4                                   KRD02220
        DCT                                             KRD02230
        TRA     ER5                                     KRD02240
        TOV     ER3                                     KRD02250
        ALS     1                                       KRD02260
        STQ     ALPHA                                   KRD02270
        LDQ     *                                       KRD02280
        LAS     10T,4                                   KRD02290
        NOP                                             KRD02300
        LDQ     MINUS                                   KRD02310
        CLA     ALPHA                                   KRD02320
        RND                                             KRD02330
        TRA     STORE                                   KRD02340
ERTN    ADD     PWR                                     KRD02350
        STO     PWR                                     KRD02360
        CLA     ALPHA+3                                 KRD02370
        TRA     *+2                                     KRD02380
ENDFD   BSS     0                                       KRD02390
STAR1   CLA     ALPHA+1                                 KRD02400
        STQ     ALPHA+2                                 KRD02410
        TZE     STORE                                   KRD02420
        LFTM                                            KRD02430
        LRS     8                                       KRD02440
        TNZ     *+3                                     KRD02450
        ORA     =0232000000000                          KRD02460
        TRA     *+2                                     KRD02470
        ORA     =0243000000000                          KRD02480
        STO     ALPHA                                   KRD02490
        CLA     =0466000000                             KRD02500
        LLS     8                                       KRD02510
        FAD     ALPHA                                   KRD02520
        FRN                                             KRD02530
        LXA     PWR,4                                   KRD02540
        TXL     SAVE,4,0                                KRD02550
        XCA                                             KRD02560
        CLA     PWR                                     KRD02570
        TXH     ER7,4,38                                KRD02580
        TMI     DIV                                     KRD02590
        FMP     PWR10,4                                 KRD02600
        TRA     OUT                                     KRD02610
DIV     XCA                                             KRD02620
        FDP     PWR10,4                                 KRD02630
        STQ     ALPHA                                   KRD02640
        FDH     PWR10,4                                 KRD02650
        PXA     0,0                                     KRD02660
        TQO     *+2                                     KRD02670
        XCA                                             KRD02680
        FAD     ALPHA                                   KRD02690
OUT     FRN                                             KRD02700
        TOV     ER6                                     KRD02710
SAVE    EFTM                                            KRD02720
STORE   STO     **                                      KRD02730
        CLA     STORE                                   KRD02740
        ADD     INCR                                    KRD02750
        STA     STORE                                   KRD02760
        LDQ     ALPHA+2                                 KRD02770
NF      LXD     NFCC,4                                  KRD02780
        TNX     ALL,4,1                                 KRD02790
        SXD     NFCC,4                                  KRD02800
        CLA     *                                       KRD02810
```

```
        STD     STAR1                                                   KRD02820
        CLS     =0                                                      KRD02830
        STO     ALPHA+1                                                 KRD02840
        STO     ALPHA+2                                                 KRD02850
        STZ     PWR                                                     KRD02860
        LXA     NCOL,2                                                  KRD02870
 NFT    TXH     NEXT+1,4,**                                             KRD02880
        STD     NFCC                                                    KRD02890
        TXI     NEXT+1,2,-1                                             KRD02900
 ALL    BSS     0                                                       KRD02910
 IR4    AXT     **,4                                                    KRD02920
 SW     TRA     **                                                      KRD02930
        CLA     CDCT                                                    KRD02940
        ADD     =1                                                      KRD02950
        STO     CDCT                                                    KRD02960
        ZET     CLAUDE                                                  KRD02961
        TRA     EXIT                                                    KRD02962
        CLA*    6,4                                                     KRD02970
        PAX     0,1                                                     KRD02980
        TXL     BLOOP,1,0                                               KRD02990
        LXD     WDCT,2                                                  KRD03000
 ALOOP  CLA     0,2                                                     KRD03010
        XEC     STORE                                                   KRD03020
        CLA     STORE                                                   KRD03030
        ADD     INCR                                                    KRD03040
        STA     STORE                                                   KRD03050
 BFILL  TXI     *+1,2,-1                                                KRD03060
        TIX     ALOOP,1,1                                               KRD03070
 BLOOP  CLA     TEST                                                    KRD03080
        TZE     EXIT                                                    KRD03090
 SETAD  CLA     0                                                       KRD03100
        ALS     9                                                       KRD03110
        PAX     0,2                                                     KRD03120
        TXH     MTEST,2,8191                                            KRD03130
LOCTWO  TSX     .READ,4                                                 KRD03140
        PZE     **,,ER11                                                KRD03150
        PZE     ER1,,ER10                                               KRD03160
        IORTN   **,,**                                                  KRD03170
        CLA     *-1                                                     KRD03180
        STA     LDQ                                                     KRD03190
        STA     ALOOP                                                   KRD03200
        ADD     =11                                                     KRD03210
        STA     SETAD                                                   KRD03220
        STA     LAST-2                                                  KRD03230
        TRA     AGAIN                                                   KRD03240
 MTEST  TXH     ER9,2,24576                                             KRD03250
        TXH     LOCTWO,2,24575                                          KRD03260
        TXH     ER9,2,13312                                             KRD03270
 EXIT   PXD     0,0                                                     KRD03280
        STO*    3,4                                                     KRD03290
 IR2    AXT     **,2                                                    KRD03300
 IR1    AXT     **,1                                                    KRD03310
        CLA     CDCT                                                    KRD03320
        TRA     CART+1                                                  KRD03330
*                                                                       KRD03340
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * KRD03350
*                                                                       KRD03360
 INCR   PZE     1                                                       KRD03370
ALPHA   BSS     6                                                       KRD03380
CLAUDE  DEC     0                                                       KRD03381
```

```
 TEST   BSS      1                                                              KRD03390
 NFC    BSS      0                                                              KRD03400
 NFCC   BSS      1                                                              KRD03410
 NCOL   BSS      0                                                              KRD03420
 WDCT   BSS      1                                                              KRD03430
 CDCT   BSS      1                                                              KRD03440
 PWR    BSS      1                                                              KRD03450
        DEC      1000000000 ,10000*000*,100000000,10000000                      KRD03460
        DEC      1000000,100000,10000,1000,100,10                               KRD03470
 10T    PZE      1                                                              KRD03480
ZPWR10 OCT 377454732313,373741367021,370601137164                               KRD03490
        OCT 365464114135,361755023373,356612334311                              KRD03500
        OCT 353473426555,347770675742,344623713116                              KRD03510
        OCT 341503074077,336402374714,332635456171                              KRD03520
        OCT 327512676456,324410545213,320647410336                              KRD03530
        OCT 315522640262,312417031702,306661534466                              KRD03540
        OCT 303532743536,300425434430,274674055532                              KRD03550
        OCT 271543212741,266434157116,262706576512                              KRD03560
        OCT 257553630410,254443023471,250721522450                              KRD03570
        OCT 245564416672,242452013710,236734654500                              KRD03580
        OCT 233575360400,230461132000,224750220000                              KRD03590
        OCT 221606500000,216470400000,212764000000                              KRD03600
        OCT 207620000000,204500000000,201400000000                              KRD03610
 END    BSS      0                                                              KRD03620
 PWR10  SYN      END-1                                                          KRD03630
                                                                                KRD03640
*                                                                               
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * KRD03650
        REM                                                                     KRD03660
        END                                                                     KRD03670
```

SUBROUTINE BSF

```
$IBMAP FBSF    100,LIST,DECK,M94                                          BSF00010
*      BACK-SPACE FILE SUBROUTINE / BSF                                   BSF00020
*                                                                         BSF00050
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * BSF00060
*                                                                         BSF00070
*      CALL BSF (NFILE,LTAPE,LERROR)                                      BSF00080
*                                                                         BSF00090
*           NFILE  = NUMBER OF FILES TO BE SPACED.  THE TAPE IS ALWAYS    BSF00100
*                    LEFT POSITIONED AT THE BEGINNING OF SOME FILE.       BSF00110
*                    0  MEANS BACKSPACE TO BEGINNING OF CURRENT FILE.     BSF00120
*                    1  MEANS BACKSPACE TO BEGINNING OF PRECEDING FILE,   BSF00130
*                       ETC.                                              BSF00140
*           LTAPE  = LOGICAL TAPE UNIT TO BE USED.                        BSF00150
*           LERROR = LOCATION IN WHICH ERROR CODE IS TO BE STORED.        BSF00160
*                    0  INDICATES SUCCESS.                                BSF00170
*                    N  INDICATES NUMBER OF FILES NOT SPACED WHEN         BSF00180
*                       BEGINNING OF TAPE WAS REACHED.                    BSF00190
*                                                                         BSF00200
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * BSF00210
       REM                                                               BSF00220
       LDIR                                                              BSF00230
       REM                                                               BSF00240
       REM                                                               BSF00250
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * BSF00260
 BSF   SAVE    4,2,1,I                                                   BSF00270
       CLA*    4,4                                                       BSF00280
       STO     TEMP                                                      BSF00290
       CALL    .FVIO.(TEMP,TEMP+1)                                       BSF00300
       CLA     TEMP+1                                                    BSF00310
       STA     SETA                                                      BSF00320
       STA     SETB                                                      BSF00330
       LXA     ..0001,4                                                  BSF00340
       CLA*    3,4                                                       BSF00350
       PAX     0,1                                                       BSF00360
       TXI     *+1,1,1                                                   BSF00370
       LAC     TEMP+1,2                                                  BSF00380
       LDI     1,2                                                       BSF00390
       LNT     040000                                                    BSF00400
       TRA     SETA+1                                                    BSF00410
       TSX     .CLOSE,4                                                  BSF00420
 SETA  MON     **                                                        BSF00430
 CONT  TSX     .NDSEL,4                                                  BSF00440
       PZE     0,2,6                                                     BSF00450
       AXC     2,4                                                       BSF00460
       NZT*    0,2                                                       BSF00470
       TRA     SETN                                                      BSF00480
       TIX     CONT,1,1                                                  BSF00490
       AXT     WAIT,4                                                    BSF00500
       SXA     WAIT,4                                                    BSF00510
       CLA     0,2                                                       BSF00520
       PAC     0,1                                                       BSF00530
       ZET     1,1                                                       BSF00540
       TRA     *-1                                                       BSF00550
       LDQ     SETWD                                                     BSF00560
       STQ     1,1                                                       BSF00570
       TSX     .ACTV,4                                                   BSF00580
       MZE     0,2                                                       BSF00590
```

```
WAIT   TRA     *                                                    BSF00600
SETWD  TWO     COM,,SELRT                                           BSF00610
COM    IORT    TEMP+4,,10                                           BSF00620
MPC    LXA     ..0001,4                                             BSF00630
       STZ*    5,4                                                  BSF00640
MPCA   TSX     .OPEN,4                                              BSF00650
SETB   MON     **                                                   BSF00660
       RETURN  BSF                                                  BSF00670
SETN   TXI     *+1,1,-1                                             BSF00680
       PXA     0,1                                                  BSF00690
       LXA     ..0001,4                                             BSF00700
       STO*    5,4                                                  BSF00710
       TRA     MPCA                                                 BSF00720
SELRT  TPL     SELPL                                                BSF00730
       AXT     MPC,2                                                BSF00740
       SXA     WAIT,2                                               BSF00750
       PAC     0,2                                                  BSF00760
       STZ     1,2                                                  BSF00770
       TRA     1,4                                                  BSF00780
SELPL  PAC     0,2                                                  BSF00790
       CLA     1,2                                                  BSF00800
       STA*    .RCHX                                                BSF00810
       TRA     1,4                                                  BSF00870
       REM                                                          BSF00880
       LORG                                                         BSF00890
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * BSF00900
*                                                                   BSF00910
ERASE  CONTRL  ER                                                   BSF00920
       USE     ER                                                   BSF00930
TEMP   BSS     20                                                   BSF00940
       END                                                          BSF00950
```

SUBROUTINE FSF

```
$IBMAP FFSF    50,DECK,M94                                                    FFSF0010
*FSF     709/90 FORTRAN LIBRARY / FORWARD SPACE FILE                          FFSF0020
*                                                                             FFSF0050
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FFSF0060
*                                                                             FFSF0070
*     CALL FSF (NFILE,LTAPE,LERROR)                                           FFSF0080
*                                                                             FFSF0090
*          NFILE  - NUMBER OF END-OF-FILE MARKS TO SPACE PASS.                FFSF0100
*                   ZERO IS EQUIVALENT TO ONE.                                FFSF0110
*          LTAPE  - LOGICAL TAPE NUMBER.                                      FFSF0120
*          LERROR - LOCATION IN WHICH THE ERROR CODE IS STORED.               FFSF0130
*                   0  INDICATES SUCCESS.                                     FFSF0140
*                   N  INDICATES NUMBER OF UNSPACED FILES WHEN                FFSF0150
*                      END-OF-TAPE SIGNAL WAS REACHED (SEE MM-23).            FFSF0160
*                                                                             FFSF0170
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FFSF0180
       REM                                                                    FFSF0190
       LDIR                                                                   FFSF0200
       REM                                                                    FFSF0210
       REM                                                                    FFSF0220
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FFSF0230
 FSF   SAVE    (4,2,1)I                                                       FFSF0240
       CLA*    4,4                                                            FFSF0250
       STO     TEMP           LOGICAL UNIT                                    FFSF0260
       CLA*    3,4                                                            FFSF0270
       PAX     0,1            CTR TO XR1                                      FFSF0280
       CALL    .FVIO.(TEMP,TEMP+1)                                            FFSF0290
       LAC     TEMP+1,2                                                       FFSF0300
       LDI     1,2                                                            FFSF0310
       CLA     TEMP+1                                                         FFSF0320
       STA     SETA                                                           FFSF0330
       STA     SETX                                                           FFSF0340
       LFT     040000                                                         FFSF0350
       TRA     LOOP                                                           FFSF0360
       TSX     .OPEN,4                                                        FFSF0370
 SETX  MON     **                                                             FFSF0380
 LOOP  TSX     .READ,4        READ TO EOF                                     FFSF0390
 SETA  PZE     **,,0                                                          FFSF0400
       PZE     EOF,,ERR                                                       FFSF0410
       IOCP    SKIM,0,2                                                       FFSF0420
       IORTN   **,0,-1                                                        FFSF0430
 ERR   CAL     SKIM+1                                                         FFSF0440
       LAS     CODE           IF TAPEND                                       FFSF0450
       TRA     LOOP                                                           FFSF0460
       TRA     ERRA                                                           FFSF0470
       TRA     LOOP                                                           FFSF0480
 EOF   TIX     LOOP,1,1                                                       FFSF0490
       LXA     ..0001,4                                                       FFSF0500
       STZ*    5,4            SET N=0                                         FFSF0510
       RETURN  FSF                                                           FFSF0520
 ERRA  PXA     0,1            SET N TO CTR OF                                 FFSF0530
       LXA     ..0001,4         UNSKIPPED FILES                               FFSF0540
       STO*    5,4                                                           FFSF0550
       TRA     ERRA-1                                                        FFSF0560
 CODE  BCI     1,TAPEND                                                      FFSF0570
       LORG                                                                  FFSF0580
 ERASE CONTRL  ER                                                            FFSF0590
```

SUBROUTINE FSR

```
$IBMAP FFSR    50,LIST,DECK,M94                                                FSR00010
*      FORWARD SPACE RECORD / FSR                                              FSR00020
*                                                                              FSR00060
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  FSR00070
*                                                                              FSR00080
*      CALL FSR (NREC,LTAPE,LERROR)                                            FSR00090
*                                                                              FSR00100
*           NREC - NUMBER OF FORTRAN LOGICAL RECORDS TO SPACE FORWARD.         FSR00110
*          LTAPE - LOGICAL TAPE NUMBER.                                        FSR00120
*         LERROR - LOCATION IN WHICH THE ERROR CODE IS STORED.                 FSR00130
*                  0  INDICATES SUCCESS.                                       FSR00140
*                  N  INDICATES NUMBER OF UNSPACED RECORDS WHEN AN             FSR00150
*                     END-OF-FILE MARK WAS FOUND.  TAPE LEFT SPACED IN         FSR00160
*                     FRONT OF EOF.                                            FSR00170
*                                                                              FSR00180
*      CALL      FSPR(NPREC,LTAPE,LERROR)                                      FSR00190
*                                                                              FSR00200
*          NPREC - NUMBER PHYSICAL RECORDS TO SPACE FORWARD.                   FSR00210
*                                                                              FSR00220
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  FSR00230
                                                                               FSR00240
       REM                                                                     FSR00250
       ENTRY    FSR                                                        M3  FSR00260
       ENTRY    FSPR                                                        M3  FSR00270
       REM                                                                     FSR00280
       REM                                                                     FSR00290
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  FSR00300
       LDIR                                                                    FSR00310
FSPR   STZ      CODE                                                           FSR00320
       TRA      *+2                                                            FSR00330
FSR    STL      CODE                                                           FSR00340
TURN   SAVEN    4,2,1,I                                                        FSR00350
       CLA*     4,4                                                            FSR00360
       STO      TEMP                                                           FSR00370
       CLA*     3,4                                                            FSR00380
       PAX      0,1                                                            FSR00390
       TZE      WIND                                                           FSR00400
       CALL     .FVIO.(TEMP,TEMP+1)                                            FSR00410
       LAC      TEMP+1,2                                                       FSR00420
       SCA      SETA,2                                                         FSR00430
       SCA      SETB,2                                                         FSR00440
       LDI      1,2                                                            FSR00450
       LFT      040000                                                         FSR00460
       TRA      READL                                                          FSR00470
       TSX      .OPEN,4                                                        FSR00480
SETB   MON      **                                                             FSR00490
READL  TSX      .READ,4                                                        FSR00500
SETA   PZE      **,,0                                                          FSR00510
       PZE      EOF,,ERR                                                       FSR00520
       IOCP     SKIM,0,2                                                       FSR00530
       IORTN    **,0,-1                                                        FSR00540
ERR    LXA      SKIM,4                                                         FSR00550
       ZET      CODE                                                           FSR00560
       TXL      READL,4,0                                                      FSR00570
       TIX      READL,1,1                                                      FSR00580
       PXD      0,0                                                            FSR00590
DONE   LXA      ..0001,4                                                       FSR00600
       STO*     5,4
```

```
        RETURN  TURN                                                    FSR00610
 EOF    TSX     .NDSEL.,4                                               FSR00620
        PZE     0,2,5                                                   FSR00630
        NOP                                                             FSR00640
 WIND   PXA     0,1                                                     FSR00650
        TRA     DONE                                                    FSR00660
*                                                                       FSR00670
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FSR00680
*                                                                       FSR00690
 ERASE  CONTRL  ER                                                      FSR00700
        USE     ER                                                      FSR00710
 TEMP   BSS     2                                                       FSR00720
 SKIM   BSS     2                                                       FSR00730
 CODE   BSS     1                                                       FSR00740
        BSS     15                                                      FSR00750
*                                                                       FSR00760
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FSR00770
        REM                                                             FSR00780
        END                                                             FSR00790
```

APPENDIX II

SUBROUTINES READTP, WRTETP, FVIO, FILE, INRPRD, AND FRUN

1. SUBROUTINE READTP DESCRIPTION

Subroutine READTP reads a binary tape in the format of the 7094 matrix interpretive scheme (TL01).

Subroutine READTP has the following restrictions:

- The matrix on tape must be written by subroutine WRTETP, a TL01 program, or its equivalent.
- Tape spacing subroutines FSF, FSR, and BSF are used by READTP.
- Tape reading is done by FORTRAN IV I/O subroutines.
- Only one- or two-dimensional arrays can be written on tape.

a. Calling Sequence

The calling sequence and descriptions of the READTP arguments are as follows:

Parameter	Function
A	(1, 1) element of the matrix to be read from tape. If this is not A(1, 1), the designated submatrix is read. (This is a method of departitioning matrices as they are read from tape.)
K	Row DIMENSION statement entry for A. This must be 1 if A is a singular dimensioned variable.
NAME	Name of matrix as it is read from tape. If NAME is nonzero upon entry to READTP, the name coming from the tape is compared to the incoming NAME and, if there is no agreement, an error return occurs. If the two names are the same, the name from the tape is stored in NAME.
M	Number of rows in the matrix
N	Number of columns in the matrix
B	Words 5 through 16 of the 16_{10}-word matrix identification

Parameter	Function
NFILE	Number of file marks to be passed before reading starts. If 0, no file spacing takes place. If negative, backspacing occurs and NFILE end-of-file marks are passed. Then, the tape is spaced forward and past the last file mark encountered. The tape is always at the beginning of a file after any file spacing.
NMAT	Number of matrices to be passed before reading starts. If 0, no matrix spacing takes place. A negative value is illegal, because backward spacing of matrices is not allowed. All matrix spacing takes place after file spacing is complete.
NTAPE	Logical tape number
IRROR	0 if successfully read; 1 if file spacing error; 2 if matrix spacing is negative; 3 if matrix spacing error; 4 if checksum error; 5 if name on tape is wrong

b. Space Required

Subroutine READTP requires 330 cells. Also, 134 cells of storage are required for subroutines FSR, FSF, and BSF.

c. Tape Format

The matrix must be written in two FORTRAN logical records. This is automatically satisfied if the tape is written by WRTETP or TL01.

2. SUBROUTINE WRTETP DESCRIPTION

Subroutine WRTETP writes a matrix on binary tape in a format consistent with the 7094 matrix interpretive scheme (TL01).

This subroutine has the following restrictions:

- The matrix must be in the core in normal FORTRAN IV order.
- Tape spacing subroutines FSF, FSR, and BSF are used by subroutine WRTETP.
- Tape writing is done by FORTRAN IV I/O subroutines.
- Only one- or two-dimensional arrays can be written on tape.

a. Calling Sequence

The calling sequence and descriptions of the READTP arguments are as follows:

Parameter	Function
A	(1, 1) element of the matrix to be written on tape. If this is not A(1, 1), then the designated submatrix is put on tape.
K	Row DIMENSION statement entry for A. This is 1 if A is a singular dimensioned variable.
NAME	Name of the matrix; a fixed-point number
M	Number of rows in the matrix
N	Number of columns in the matrix
B	Words 11 through 16 of 16_{10} -word matrix identification
NFILE	Number of end-of-file marks to be passed before writing starts. If 0, no file spacing takes place. If negative, backspacing occurs and NFILE end-of-file marks will be passed. Then, the tape will be spaced forward and past the last file mark encountered. The tape is always at the beginning of a file after any file spacing.
NMAT	Number of matrices to be passed before writing occurs. If 0, no matrix spacing takes place. A negative value is illegal, because backward spacing of matrices is not allowed. All matrix spacing takes place after file spacing is complete.
NTAPE	Logical tape number
IRROR	0 if successfully written; 1 if error occurs during file spacing; 2 if matrix spacing is negative; 3 if error occurs during matrix spacing

b. Space Required

Subroutine WRTETP requires 431 cells. Also, 134 cells of storage are required for subroutines FSR, FSF, and BSF.

c. Tape Format

The matrix consists of two FORTRAN logical records. The first of these is the 16_{10} -word identification, and the second consists of the matrix elements. The matrix may be written in fixed point, sparse, or null forms.

3. SUBROUTINE LISTINGS

This section contains the following subroutine listings.

The following is a description of subroutines FVIO, FILE, INRPRD, and FRUN:

Subroutine	Function
FVIO	This subroutine is the input/output statement specifying variable-output units 1 through 17.
FILE	The FILE definitions of units 1 through 17 are computed by this subroutine.
INRPRD	This subroutine computes the interproducts of two matrices.
FRUN	This is the subroutine that rewinds and unloads tapes.

SUBROUTINE READTP

```
$IBMAP READTP  300,M94/2,DECK                                          00000000
*READTP        SUBROUTINE TO READ A TL-01 BINARY TAPE                  00000001
*              FROM A FORTRAN PROGRAM                                  00000002
*              CALL READTP (A,K,NAME,M,N,B,NFILE,NMAT,NTAPE,IRR)       00000003
*                          A = ADDRESS WHERE MATRIX IS TO BE STORED    00000004
*                          K = ROW DIMENSION STATEMENT ENTRY FOR A     00000005
*                          NAME = NAME OF MATRIX                       00000006
*                          M = ROW SIZE OF MATRIX                      00000007
*                          N = COLUMN SIZE OF MATRIX                   00000008
*                          B = WORDS 5  THROUGH 16 OF TAPE ID          00000009
*                          NFILE = NUMBER OF FILES TO BE SPACED        00000010
*                                  FROM CURRENT POSITION               00000011
*                                  POSITIVE CAUSES FORWARD SPACE       00000012
*                                  NEGATIVE CAUSES BACKWARD SPACE      00000013
*                          NMAT = NUMBER OF MATRICES TO BE SPACED      00000014
*                                  FROM CURRENT POSITION               00000015
*                          NTAPE = LOGICAL TAPE NUMBER                 00000016
*                          IRR = 0 IF READ IS SUCCESSFUL               00000017
*                              = 1 IF FILE SPACING ERROR               00000018
*                              = 2 IF MATRIX SPACING IS NEG.           00000019
*                              = 3 IF MATRIX SPACING ERROR             00000020
*                              = 4 IF CHECKSUM ERROR                   00000021
*                              = 5 IF NAME ON TAPE IS WRONG            00000022
*                                                                      00000023
READTP SAVE    (1,2,4)I                                                00000024
       SXA     XR4,4                                                   00000025
       STZ     TESTC2                                                  00000026
       CLA*    5,4                                                     00000027
       STO     TEMP1                                                   00000028
       CLA*    11,4                                                    00000029
       STO     TAPE                                                    00000030
       CLA*    9,4                                                     00000031
       TZE     RCDSP                                                   00000032
       TPL     FSFILE                                                  00000033
       SSP                      BACKSPACE                              00000034
       SUB     =1                                                      00000035
       STO     MFILE                                                   00000036
       CALL    BSF(MFILE,TAPE,IRR1)                                    00000037
       TRA     TESTFL                                                  00000038
FSFILE STO     MFILE                                                   00000039
       CALL    FSF(MFILE,TAPE,IRR1)                                    00000040
TESTFL ZET     IRR1                                                    00000041
       TRA     FILEER           FILE SPACING ERROR                     00000042
RCDSP  LXA     XR4,4                                                   00000043
       CLA*    10,4                                                    00000044
       TZE     READID                                                  00000045
       TMI     RCDER1                                                  00000046
       ALS     1                                                       00000047
       STO     MMAT             NUMBER OF LOGICAL RECORDS.             00000048
       CALL    FSR(MMAT,TAPE,IRR1)                                     00000049
       ZET     IRR1                                                    00000050
       TRA     RCDER2           RECORD SPACING ERROR                   00000051
READID CALL    .FVIO.(TAPE,TAPIB)                                      00000052
       CALL    .FRDB.(TAPIB)                                           00000053
       TSX     .FBLT.,4                                                00000054
       STO     CNAME                                                   00000055
       LXA     XR4,2                                                   00000056
```

```
        STO*    5,2             NAME                                    00000057
        TSX     .FBLT.,4                                                00000058
        STO     CM                                                      00000059
        STO*    6,2             M                                       00000060
        TSX     .FBLT.,4                                                00000061
        STO     CN                                                      00000062
        STO*    7,2             N                                       00000063
        TSX     .FBLT.,4                                                00000064
        STO     CSUM            CHECKSUM                                00000065
        AXT     12,1                                                    00000066
        CLA     8,2             B                                       00000067
        ADD     =12                                                     00000068
        STA     *+2                                                     00000069
PNT1    TSX     .FBLT.,4                                                00000070
        STO     **,1            LAST 12 NOS. INTO B                     00000071
        TIX     PNT1,1,1                                                00000072
        LDQ     CM                                                      00000073
        MPY     CN                                                      00000074
        STQ     MTN                                                     00000075
        CALL    .FRLR.                                                  00000076
        CLA     TEMP1           NAME SUPPLIED BY CALLING PROGRAM        00000077
        TZE     NULOSP          IF IT IS ZERO, DO NOT CHECK             00000078
        SUB*    5,2             DO THE NAMES AGREE                      00000079
        TNZ     NAMEER          NO                                      00000080
NULOSP  CLA*    8,2                                                     00000081
        CAS     WDMSP                                                   00000082
        TRA     *+2                                                     00000083
        STO     TESTC2          MATRIX SPARSE                           00000084
READ    CALL    .FRDB.(TAPIB)                                           00000085
        AXT     1,2                                                     00000086
        LXA     XR4,4                                                   00000087
        CLA     3,4             A                                       00000088
        STA     REDE                                                    00000089
        CLA*    6,4                                                     00000090
        ALS     18                                                      00000091
        STD     TEST2           M                                       00000092
        STD     TEST4                                                   00000093
        STD     TEST6                                                   00000094
        CLA*    4,4             K                                       00000095
        PAC     0,1                                                     00000096
        SXD     TEST1-1,1                                               00000097
        SXD     TEST3-1,1                                               00000098
        SXD     TEST5-1,1                                               00000099
        XCA                                                             00000100
        MPY*    7,4             N                                       00000101
        XCA                                                             00000102
        PAC     0,1                                                     00000103
        SXD     TEST1,1         -K*N                                    00000104
        SXD     TEST3,1                                                 00000105
        SXD     TEST5,1                                                 00000106
        AXT     0,1                                                     00000107
        PXA     0,0                                                     00000108
*               START CHECKSUM OFF WITH NAME AND DIMENSIONS             00000109
        ACL     CNAME           NAME IN ADDRESS                         00000110
        ACL     CM              M IN ADDRESS                            00000111
        ACL     CN              N IN ADDRESS                            00000112
        SLW     TEMP            STORE SUM IN CELL FOR UPCOMING CHECKSUM 00000113
        NZT     TESTC2                                                  00000114
        TRA     REDEN                                                   00000115
        ACL     WDMSP                                                   00000116
```

```
        SLW     TEMP                                                    00000117
REDES   TSX     .FBLT.,4        SPARSE MATRIX                           00000118
        STO*    REDE                                                    00000119
        PDX     0,4                                                     00000120
        XCA                     GET LOGICAL VERSION FOR CKSUM           00000121
        XCL                                                             00000122
        ACL     TEMP                                                    00000123
        SLW     TEMP                                                    00000124
        TXL     *+3,4,0                                                 00000125
        AXT     0,4             STORE NO ZEROS                          00000126
        TRA     TEST3-1                                                 00000127
        CLA*    REDE                                                    00000128
        PAX     0,4                                                     00000129
STZRO   STZ*    REDE            STORE ZEROS                             00000130
        TXI     *+1,1,**        -K                                      00000131
TEST3   TXH     TIR4,1,**       -K*N                                    00000132
        PXD     0,2                                                     00000133
        PDC     0,1                                                     00000134
        TXI     *+1,2,1                                                 00000135
TEST4   TXL     TIR4,2,**       M                                       00000136
        TRA     RLR             TR TO RLR                               00000137
TIR4    TIX     STZRO,4,1                                               00000138
        TRA     REDES           READ NEXT WORD                          00000139
REDEN   TSX     .FBLT.,4        TEST FOR NULL MATRIX                    00000140
        NZT     MTN                                                     00000141
        TRA     RLR                                                     00000142
        STO*    REDE                                                    00000143
        LDQ*    REDE                                                    00000144
        SUB     WDMNL                                                   00000145
        TZE     *+3             NULL                                    00000146
        XCL                     NOT NULL                                00000147
        TRA     REDE1                                                   00000148
        XCL                                                             00000149
        ACL     TEMP                                                    00000150
        SLW     TEMP                                                    00000151
STZN    STZ*    REDE                                                    00000152
        TXI     *+1,1,**        -K                                      00000153
TEST5   TXH     STZN,1,**       -K*N                                    00000154
        PXD     0,2                                                     00000155
        PDC     0,1                                                     00000156
        TXI     *+1,2,1                                                 00000157
TEST6   TXL     STZN,2,**       M                                       00000158
        TRA     RLR                                                     00000159
        TSX     .FBLT.,4                                                00000160
REDE    STO     **,1            A                                       00000161
        XCA                     GET LEGICAL VERSION FOR CKSUM           00000162
        XCL                                                             00000163
REDE1   ACL     TEMP                                                    00000164
        SLW     TEMP                                                    00000165
        TXI     *+1,1,**        -K                                      00000166
TEST1   TXH     REDE-1,1,**     -K*N                                    00000167
        SXD     TEMP1,2                                                 00000168
        LDC     TEMP1,1                                                 00000169
        TXI     *+1,2,1                                                 00000170
TEST2   TXL     REDE-1,2,**     M                                       00000171
 RLR    CALL    .FRLR.                                                  00000172
        CLA     CSUM                                                    00000173
        TZE     XR4             IF CSUM=0, DO NOT CHECK                 00000174
        SUB     =1.                                                     00000175
        TZE     XR4                                                     00000176
```

```
        CAL     CSUM                                            00000177
        ERA     TEMP                                            00000178
        TNZ     CSUMER                                          00000179
        AXT     **,4                                            00000180
XR4     STZ*    12,4            ZERO ERROR CODE                 00000181
        RETURN  READTP                                          00000182
FILEER  CLA     =1              FILE SPACING ERROR              00000183
        LXA     XR4,4                                           00000184
        STO*    12,4                                            00000185
        RETURN  READTP                                          00000186
RCDER1  CLA     =2              NEGATIVE MATRIX SPACING         00000187
        TRA     FILEER+1                                        00000188
RCDER2  CLA     =3              MATRIX SPACING ERROR            00000189
        TRA     FILEER+1                                        00000190
CSUMER  CLA     =4              CHECKSUMS DO NOT AGREE          00000191
        TRA     FILEER+1                                        00000192
NAMEER  CLA     =5              NAMES DO NOT AGREE              00000193
        TRA     FILEER+1                                        00000194
 WDMNL  BCI     1,M=NULL                                        00000195
 WDMSP  BCI     1,SPARSE                                        00000196
TESTC2  PZE                                                     00000197
 MTN    PZE                                                     00000198
  TAPE  HTR     0                                               00000199
  IRR1  HTR     0                                               00000200
  TEMP  HTR     0                                               00000201
 TEMP1  HTR     0                                               00000202
 CSUM   HTR     0                                               00000203
 MFILE  HTR     0                                               00000204
 MMAT   HTR     0                                               00000205
 CNAME  PZE     0                                               00000206
    CM  PZE     0                                               00000207
    CN  PZE     0                                               00000208
 TAPIB  PZE             TAPE FOR IBSYS                          00000209
        END                                                     00000210
```

SUBROUTINE WRTETP

```
$IBMAP WRTETP  400,M94/2,DECK                                          00000000
*WRTETP        SUBROUTINE TO WRITE A TL-01 BINARY TAPE                 00000001
*              FROM A FORTRAN PROGRAM.                                 00000002
*              CALL WRTETP (A,K,NAME,M,N,B,NFILE,NMAT,NTAPE,IRR)       00000003
*                          A = ADDRESS OF MATRIX (1,1) ELEMENT         00000004
*                          K= ROW DIMENSION STATEMENT ENTRY FOR A      00000005
*                          NAME = NAME OF MATRIX                       00000006
*                          M = ROW SIZE OF MATRIX                      00000007
*                          N=COLUMN SIZE OF MATRIX                     00000008
*                          B=WORDS 11 THROUGH 16 OF ID                 00000009
*                          NFILE=NUMBER OF FILES TO BE SPACED          00000010
*                                FROM CURRENT POSITION                 00000011
*                                POSITIVE CAUSES FORWARD SPACE         00000012
*                                NEGATIVE CAUSES BACKWARD SPACE        00000013
*                          NMAT=NUMBER OF MATRICES TO BE SPACED        00000014
*                                FROM CURRENT POSITION                 00000015
*                          NTAPE=LOGICAL TAPE NUMBER                   00000016
*                          IRR= 0 IF SUCCESSFUL WRITE                  00000017
*                                = 1 IF ERROR ON FILE SPACE            00000018
*                                = 2 IF MATRIX SPACING IS NEGATIVE     00000019
*                                = 3 IF ERROR ON MATRIX SPACE          00000020
*                                                                      00000021
WRTETP SAVE    (1,2,4)I                                                00000022
       SXA     XR4,4                                                   00000023
       CLA*    11,4            NTAPE                                   00000024
       STO     TAPE                                                    00000025
       CLA*    9,4             NFILE                                   00000026
       TZE     RCDSP                                                   00000027
       TPL     FSFILE                                                  00000028
       SSP                     BACKSPACE                               00000029
       SUB     =1                                                      00000030
       STO     MFILE                                                   00000031
       CALL    BSF(MFILE,TAPE,IRR1)                                    00000032
       TRA     TESTFL                                                  00000033
FSFILE STO     MFILE                                                   00000034
       CALL    FSF(MFILE,TAPE,IRR1)                                    00000035
TESTFL ZET     IRR1                                                    00000036
       TRA     FILEER          FILE SPACING ERROR                      00000037
RCDSP  LXA     XR4,4                                                   00000038
       CLA*    10,4            NMAT                                    00000039
       TZE     CALCSM                                                  00000040
       TMI     RCDER1                                                  00000041
       ALS     1                                                       00000042
       STO     MMAT            NUMBER OF LOGICAL RECORDS.              00000043
       CALL    FSR(MMAT,TAPE,IRR1)                                     00000044
       ZET     IRR1                                                    00000045
       TRA     RCDER2          RECORD SPACING ERROR                    00000046
CALCSM LXA     XR4,4                                                   00000047
       STZ     TESTC1                                                  00000048
       STZ     TESTC2                                                  00000049
       CLA     WDSPSE                                                  00000050
       STO     TESTC3                                                  00000051
*              TESTC3=SPARSE,MATRIX SPARSE                             00000052
*              TESTC3=0,MATRIX NOT SPARSE                              00000053
       CLA*    3,4                                                     00000054
       CAS     WDMNL           TEST ELEMENT A(1,1)                     00000055
       TRA     *+2                                                     00000056
```

```
        STO     TESTC1          ELEMENT A(1,1) =M=NULL                          00000057
        CLA     3,4                                                             00000058
        STA     CSM1                                                            00000059
        CLA*    6,4                                                             00000060
        SLW     CM              SAVE M IN ADDRESS                               00000061
        ALS     18                                                              00000062
        STD     TST2                                                            00000063
        STD     TST4                                                            00000064
        STD     TST6                                                            00000065
        STD     TST8                                                            00000066
        CLA*    4,4             K                                               00000067
        PAC     0,1                                                             00000068
        SXD     TST1-1,1                                                        00000069
        SXD     TST3-1,1                                                        00000070
        SXD     TST5-1,1                                                        00000071
        SXD     TST7-1,1                                                        00000072
        XCA                                                                     00000073
        MPY*    7,4             K*N                                             00000074
        XCA                                                                     00000075
        PAC     0,1                                                             00000076
        SXD     TST1,1                                                          00000077
        SXD     TST3,1                                                          00000078
        SXD     TST5,1                                                          00000079
        SXD     TST7,1                                                          00000080
        AXT     0,1                                                             00000081
        AXT     1,2                                                             00000082
        CLA*    5,4             NAME                                            00000083
        STO     CNAME           NAME IN ADDRESS                                 00000084
        CAL*    7,4             N                                               00000085
        SLW     CN              N IN ADDRESS                                    00000086
        ACL     CNAME           START CHECKSUM OFF WITH NAME AND DIMENSIONS     00000087
        ACL     CM                                                              00000088
        SLW     TEMP                                                            00000089
        LDQ     CN                                                              00000090
        MPY     CM                                                              00000091
        STQ     MTN                                                             00000092
        NZT     MTN                                                             00000093
        TRA     PTWT1-1                                                         00000094
        ZET     TESTC1          IS A(1,1) = M=NULL                              00000095
        TRA     NULLCS          NULL MATRIX                                     00000096
        CAL     TEMP                                                            00000097
CSM1    ACL     **,1            A                                               00000098
        XCL                                                                     00000099
        CLA*    CSM1                                                            00000100
        TZE     GCKSMB                                                          00000101
        STO     TESTC2          MATRIX NOT NULL                                 00000102
        ANA     =0377400000000                                                  00000103
        TNZ     *+2                                                             00000104
        STZ     TESTC3          MATRIX NOT SPARSE                               00000105
GCKSMB  XCL                                                                     00000106
        TXI     *+1,1,**        -K                                              00000107
TST1    TXH     CSM1,1,**       -K*N                                            00000108
        SXD     TEMP,2                                                          00000109
        LDC     TEMP,1                                                          00000110
        TXI     *+1,2,1                                                         00000111
TST2    TXL     CSM1,2,**       M                                               00000112
        SLW     TEMP                                                            00000113
        CLA     TESTC2                                                          00000114
        TMI     PTWT                                                            00000115
        TNZ     PTWT                                                            00000116
```

```
NULLCS  CAL     WDMNL           NULL MATRIX                          00000117
        SLW     TESTC1                                               00000118
        ACL     CN                                                   00000119
        ACL     CM                                                   00000120
        ACL     CNAME                                                00000121
        SLW     TEMP            NULL MATRIX CHECK SUM                00000122
        STZ     TESTC3                                               00000123
        TRA     PTWT1                                                00000124
PTWT    NZT     TESTC3                                               00000125
        TRA     PTWT1                                                00000126
        AXT     0,1             SPARSE MATRIX                        00000127
        AXT     1,2                                                  00000128
        AXT     0,4             0 COUNT                              00000129
        PXA     0,0                                                  00000130
FIXSP   ZET*    GSM1                                                 00000131
        TRA     CKWC                                                 00000132
        TXI     *+1,4,1         ADD 1 TO ZERO COUNT                  00000133
        SXA     SPIR,1          ELEMENT LOC.                         00000134
        TXI     *+1,1,**        -K                                   00000135
TST5    TXH     FIXSP,1,**      -K*N                                 00000136
        SXD     TEMP,2                                               00000137
        LDC     TEMP,1                                               00000138
        TXI     *+1,2,1                                              00000139
TST6    TXL     FIXSP,2,**      M                                    00000140
        SLW     TEMP                                                 00000141
        PXA     0,4                                                  00000142
        TZE     *+3                                                  00000143
        LXA     SPIR,1                                               00000144
        STO*    CSM1            STORE 0 COUNT                        00000145
        ACL     TEMP            FORM SPARSE CHECK SUM                00000146
        ACL     CN                                                   00000147
        ACL     CNAME                                                00000148
        ACL     CM                                                   00000149
        ACL     WDSPSE                                               00000150
        SLW     TEMP                                                 00000151
        TRA     PTWT1                                                00000152
CKWC    ACL*    CSM1                                                 00000153
        TXH     *+2,4,0         ANY ZEROES YET                       00000154
        TRA     TST5-2                                               00000155
        SLW     TEMP                                                 00000156
        SXD     SPIR,1          LOC. NON-ZERO TERM AFTER COUNT       00000157
        LXA     SPIR,1          LAST ZERO ELEM. LOC.                 00000158
        PXA     0,4                                                  00000159
        STO*    CSM1            STORE CONTROL WORD                   00000160
        ACL     TEMP            UPDATE CHECKSUM                      00000161
        LXD     SPIR,1                                               00000162
        AXT     0,4             ZERO COUNT=0                         00000163
        TRA     TST5-2                                               00000164
        STZ     TESTC3                                               00000165
PTWT1   CALL    .FVIO.(TAPE,TAPIB)                                   00000166
        CALL    .FWRB.(TAPIB)                                        00000167
        CLA     CNAME                                                00000168
        TSX     .FBLT.,4        NAME                                 00000169
        CLA     CM                                                   00000170
        TSX     .FBLT.,4        M                                    00000171
        CLA     CN                                                   00000172
        TSX     .FBLT.,4        N                                    00000173
        CLA     TEMP                                                 00000174
        TSX     .FBLT.,4        CHECKSUM                             00000175
        CLA     TESTC3                                               00000176
```

```
          TSX     .FBLT.,4        SPARSE OR 0                          00000177
          AXT     5,1                                                  00000178
          CLA     =0                                                   00000179
          TSX     .FBLT.,4        5 CONSECUTIVE ZEROES                 00000180
          TIX     *-2,1,1                                              00000181
          LXA     XR4,4                                                00000182
          AXT     6,1                                                  00000183
          CLA     8,4             B                                    00000184
          ADD     =6                                                   00000185
          STA     *+1                                                  00000186
          CLA     **,1            ADDRESS B+6                          00000187
          TSX     .FBLT.,4        6 NUMBERS OF B                       00000188
          TIX     *-2,1,1                                              00000189
          CALL    .FWLR.                                               00000190
          CALL    .FWRB.(TAPIB)                                        00000191
          NZT     MTN                                                  00000192
          TRA     CKWZR                                                00000193
          NZT     TESTC1                                               00000194
          TRA     NWRTE           MATRIX NOT NULL                      00000195
          CLA     TESTC1          M=NULL                               00000196
          TSX     .FBLT.,4                                             00000197
          AXT     15,1                                                 00000198
          CLA     =0                                                   00000199
          TSX     .FBLT.,4                                             00000200
          TIX     *-2,1,1                                              00000201
          TRA     WLR                                                  00000202
NWRTE     AXT     0,1                                                  00000203
          AXT     1,2                                                  00000204
          NZT     TESTC3                                               00000205
          TRA     WRTE            MATRIX NOT SPARSE                    00000206
          AXT     0,4             NUMBER OF WORDS COUNT                00000207
SPWRTE    CLA*    CSM1                                                 00000208
          TZE     TST7-1                                               00000209
          ANA     =0377400000000                                       00000210
          LDQ*    CSM1                                                 00000211
          TNZ     *+2             NOT CONTROL WORD                     00000212
          STZ*    CSM1            ZERO OUT CONTROL WORD                00000213
          XCA                                                          00000214
          SXD     TEMP,4                                               00000215
          TSX     .FBLT.,4                                             00000216
          LXD     TEMP,4                                               00000217
          TXI     *+1,4,1                                              00000218
          TXI     *+1,1,**        -K                                   00000219
TST7      TXH     SPWRTE,1,**     -K*N                                 00000220
          SXD     TEMP,2                                               00000221
          LDC     TEMP,1                                               00000222
          TXI     *+1,2,1                                              00000223
TST8      TXL     SPWRTE,2,**     M                                    00000224
          PXA     0,4                                                  00000225
          SUB     =16                                                  00000226
          TPL     WLR             MORE THAN 16 WORDS                   00000227
          SSP                                                          00000228
          TRA     WRZR                                                 00000229
*                 WRITE FIXED POINT MATRIX                             00000230
WRTE      CLA*    CSM1            CHECKSUM CORRECT                     00000231
          TSX     .FBLT.,4                                             00000232
          TXI     *+1,1,**        -K                                   00000233
TST3      TXH     WRTE,1,**       -K*N                                 00000234
          SXD     TEMP,2                                               00000235
          LDC     TEMP,1                                               00000236
```

```
        TXI     *+1,2,1                                          00000237
  TST4  TXL     WRTE,2,**       M                                00000238
 CKWZR  CLA     =16                                              00000239
        SUB     MTN                                              00000240
        TMI     WLR                                              00000241
        TZE     WLR                                              00000242
 WRZR   PAX     0,1             M*N LESS THAN 16                 00000243
        CLA     =0                                               00000244
        TSX     .FBLT.,4                                         00000245
        TIX     *-2,1,1                                          00000246
 WLR    CALL    .FWLR.                                           00000247
   XR4  AXT     **,4                                             00000248
        STZ*    12,4                                             00000249
        RETURN  WRTETP                                           00000250
FILEER  CLA     =1              FILE SPACING ERROR               00000251
        LXA     XR4,4                                            00000252
        STO*    12,4                                             00000253
        RETURN  WRTETP                                           00000254
RCDER1  CLA     =2              NEGATIVE MATRIX SPACING          00000255
        TRA     FILEER+1                                         00000256
RCDER2  CLA     =3              MATRIX SPACING ERROR             00000257
        TRA     FILEER+1                                         00000258
 WDMNL  BCI     1,M=NULL                                         00000259
WDSPSE  BCI     1,SPARSE                                         00000260
TESTC1  PZE                                                      00000261
TESTC2  PZE                                                      00000262
TESTC3  PZE                                                      00000263
SPIR    PZE                                                      00000264
 MTN    PZE                                                      00000265
  TEMP  HTR     0                                                00000266
 MFILE  HTR     0                                                00000267
 MMAT   HTR     0                                                00000268
  IRR1  HTR     0                                                00000269
  TAPE  HTR     0                                                00000270
 CNAME  PZE     0                                                00000271
    CM  PZE     0                                                00000272
    CN  PZE     0                                                00000273
TAPIB   PZE             TAPE FOR IOCS                           00000274
        END                                                      00000275
```

SUBROUTINE FVIO

```
$IBMAP  FVIO    50,()OK,DECK
        TTL     FVIO - FORTRAN VARIABLE I/O LOGICAL UNIT                    3FA00020
        REM                                                                 3FA00030
        REM     CALLING SEQUENCE IS CALL .FVIO.(LN,ERAS) WHERE LN IS        3FA00040
        REM     LOCATION OF VARIABLE LOGICAL UNIT AND ERAS WILL CONTAIN     3FA00050
        REM     CONTENTS OF APPROPRIATE .UNXX.  (.UNXX. CONTAINS            3FA00060
        REM     PZE UNITXX WHERE XX CORRESPONDS TO LOGICAL UNIT N).         3FA00070
        REM     FVIO IS CALLED FOR ANY I/O STATEMENT SPECIFYING             3FA00080
        REM     A VARIABLE LOGICAL UNIT.                                    3FA00090
        REM                                                                 3FA00100
.FVIO.  SAVE    (2)                                                     (1)3FA00110
        CLA*    3,4                     PICK UP LOGICAL UNIT NUMBER     (1)3FA00120
        PAC     ,2                                                      (1)3FA00130
        TXL     ERROR,2,-NUNITS-1       IS UNIT ZERO, OR TOO LARGE      (1)3FA00140
        CLA*    TABLE,2                 SAVE ADDRESS OF                 (1)3FA00150
        STO*    4,4                     FILE CONTROL BLOCK              (1)3FA00160
        RETURN  .FVIO.                                                  (1)3FA00170
ERROR   CLA*    3,4                     LOGICAL UNIT IN ERROR           (1)3FA00250
        ANA     ADMSK                   DEFINED FOR THIS UNIT VALUE.        3FA00260
        XCA                             CONVERT THIS ILLEGAL VALUE          3FA00270
        AXT     0,4                     TO DECIMAL FOR ERROR MESSAGE.       3FA00280
        STZ     TEMP                                                        3FA00290
CNVT    PXA     0,0                                                         3FA00300
        DVP     L(10)                                                       3FA00310
        ALS     0,4                                                         3FA00320
        ORS     TEMP                                                        3FA00330
        CLA     =1                                                          3FA00340
        TLQ     *+2                                                         3FA00350
        TXI     CNVT,4,-6                                                   3FA00360
        CAL     BLANKS                                                      3FA00370
        ALS     6,4                                                         3FA00380
        ORA     TEMP                                                        3FA00390
        SLW     E47MES+6                                                    3FA00400
UNERR   CALL    .FXEM.(CODE)            EXIT FOR EXECUTION ERROR.           3FA00410
        TRA     .LXERR                  NO OPTIONAL EXIT.                   3FA00420
CODE    PZE     47                                                          3FA00430
ER47MS  PZE     E47MES,,7                                                   3FA00440
ER47OP  PZE     NOOPXT,,7                                                   3FA00450
E47MES  BCI     7,0LOGICAL UNIT NOT DEFINED FOR VALUE                       3FA00460
NOOPXT  BCI     7,0NO OPTIONAL EXIT - EXECUTION TERMINATED                  3FA00470
ADMSK   OCT     000000077777                                                3FA00480
TEMP    PZE     **                                                          3FA00490
L(10)   DEC     10                                                          3FA00500
BLANKS  BCI     1,                                                          3FA00510
TABLE   PZE     NUNITS                                                      3FA00550
        PZE     .UN01.                                                      3FA00560
        PZE     .UN02.                                                      3FA00570
        PZE     .UN03.                                                      3FA00580
        PZE     .UN04.                                                      3FA00590
        PZE     .UN05.                                                      3FA00600
        PZE     .UN06.                                                      3FA00610
        PZE     .UN07.                                                      3FA00620
        PZE     .UN08.                                                      3FA00630
        PZE     .UN09.
        PZE     .UN10.
        PZE     .UN11.
        PZE     .UN12.
```

```
        PZE     .UN13.
        PZE     .UN14.
        PZE     .UN15.
        PZE     .UN16.
        PZE     .UN17.
*       ADDITIONAL UNITS MAY BE INSERTED HERE. FOR EACH UNIT          3FA00640
*       INSERTED, A CORRESPONDING ROUTINE MUST BE INSERTED TO         3FA00650
*       PRODUCE A $FILE CARD FOR THE ADDITIONAL UNIT.                 3FA00660
NUNITS  EQU     *-TABLE-1                                             3FA00670
        END                                                           3FA00680
```

SUBROUTINE FILE

```
$IBMAP  FILE     DECK
        ENTRY    .UN01.
        ENTRY    .UN02.
        ENTRY    .UN03.
        ENTRY    .UN04.
        ENTRY    .UN07.
        ENTRY    .UN08.
        ENTRY    .UN09.
        ENTRY    .UN10.
        ENTRY    .UN11.
        ENTRY    .UN12.
        ENTRY    .UN13.
        ENTRY    .UN14.
        ENTRY    .UN15.
        ENTRY    .UN16.
        ENTRY    .UN17.
.UN01.  PZE      UNIT01
.UN02.  PZE      UNIT02
.UN03.  PZE      UNIT03
.UN04.  PZE      UNIT04
.UN07.  PZE      UNIT07
.UN08.  PZE      UNIT08
.UN09.  PZE      UNIT09
.UN10.  PZE      UNIT10
.UN11.  PZE      UNIT11
.UN12.  PZE      UNIT12
.UN13.  PZE      UNIT13
.UN14.  PZE      UNIT14
.UN15.  PZE      UNIT15
.UN16.  PZE      UNIT16
.UN17.  PZE      UNIT17
UNIT01  FILE     ,UT1,READY,INOUT,BIN,SCRATCH
UNIT02  FILE     ,UT2,READY,INOUT,BIN,SCRATCH
UNIT03  FILE     ,UT3,READY,INOUT,BIN,SCRATCH
UNIT04  FILE     ,UT4,READY,INOUT,BIN,SCRATCH
UNIT07  FILE     ,PP1,READY,OUTPUT,BLK=14,BCD,PUNCH
UNIT08  FILE     ,C(1),READY,INOUT,BIN,SCRATCH
UNIT09  FILE     ,LB4,READY,INOUT,MXBCD,SCRATCH
UNIT10  FILE     ,C(2),READY,INOUT,BIN,SCRATCH
UNIT11  FILE     ,C(3),READY,INOUT,BIN,SCRATCH
UNIT12  FILE     ,C(4),READY,INOUT,BIN,SCRATCH
UNIT13  FILE     ,C(5),READY,INOUT,BIN,SCRATCH
UNIT14  FILE     ,B(1),READY,INOUT,BIN,SCRATCH
UNIT15  FILE     ,B(2),READY,INOUT,BIN,SCRATCH
UNIT16  FILE     ,A(1),READY,INOUT,BIN,SCRATCH
UNIT17  FILE     ,A(2),READY,INOUT,BIN,SCRATCH
        END
```

SUBROUTINE INRPRD

```
SIBMAP INRPRD  60,M94,DECK                                                    00000000
*INRPRD        SUBROUTINE TO COMPUTE AN INNER PRODUCT      (7094/2 ONLY)00000001
*                                                                           00000002
*              CALL INRPRD (A,INCRMA,B,INCRMB,PROD,N)                       00000003
*              A = STARTING ADDRESS OF ROW                                  00000004
*              INCRMA = STORAGE INCREMENT FOR ELEMENTS OF A                 00000005
*              B = STARTING ADDRESS OF COLUMN                               00000006
*              INCRMB = STORAGE INCREMENT FOR ELEMENTS OF B                 00000007
*              PROD = LOCATION WHERE PRODUCT IS TO BE STORED                00000008
*              PRDSUM ENTRY ADDS THE RESULT TO THE PROD CELL                00000009
*              N = NUMBER OF ELEMENTS IN ROW OR COLUMN                      00000010
*                                                                           00000011
       ENTRY   PRDSUM                                                       00000012
*                                                                           00000013
INRPRD SAVE    (1,2,4)I                                                     00000014
       CAL     IN             FIND OUT WHERE TO GO                         00000015
       TNZ     PRDSUM+2       IF NON-ZERO, ENTRY WAS AT PRDSUM             00000016
       TRA     BEGIN-2                                                     00000017
PRDSUM STL     IN             SAVE ENTRY LOCATION                          00000018
       TRA     INRPRD                                                      00000019
       CAL     ADD            STUFF FOR SUMMING WITH RESULT CELL           00000020
       SLW     SUM1                                                        00000021
       TRA     BEGIN                                                       00000022
       CAL     NOADD          STUFF FOR SIMPLE INNER PRODUCT               00000023
       SLW     SUM1                                                        00000024
BEGIN  SXA     XR4,4                                                       00000025
       CAL     3,4            PRE-MULT                                     00000026
       STA     OVER                                                        00000027
       CAL     5,4            POST-MULT                                    00000028
       STA     OVER+1                                                      00000029
       CLA*    4,4            PRE-MULT. INCREMENT                          00000030
       PAC     0,2             NEGATIVE OF INCREMENT FOR A                 00000031
       SXD     LOOPND-2,2                                                  00000032
       CLA*    6,4            POST-MULT. INCREMENT                         00000033
       PAC     0,2             NEGATIVE OF INCREMENT FOR B                 00000034
       SXD     LOOPND-1,2                                                  00000035
       AXT     0,1            CONTROL FOR A                                00000036
       AXT     0,2            CONTROL FOR B                                00000037
       CLA*    8,4            N                                            00000038
       PAX     0,4            CONTROL FOR NUMBER OF ELEMENTS               00000039
START  STZ     TEMPAD         CLEAR SUMMING CELL                           00000040
       STZ     TEMPAD+1                                                    00000041
  OVER LDQ     **,1           A                                            00000042
       FMP     **,2           B                                            00000043
       DFAD    TEMPAD                                                      00000044
       DST     TEMPAD                                                      00000045
       TXI     *+1,1,**       -INCRMA IN DECREMENT                         00000046
       TXI     *+1,2,**       -INCRMB IN DECREMENT                         00000047
LOOPND TIX     OVER,4,1                                                    00000048
   XR4 AXT     **,4                                                        00000049
  SUM1 HTR     9,4            THIS SHOULD ALWAYS BE STUFFED                00000050
       FRN                                                                 00000051
       STO*    7,4            INNER PRODUCT                                00000052
       STZ     IN                                                          00000053
       RETURN  INRPRD                                                      00000054
IN     PZE     0              FLAG FOR ENTRY POINT                         00000055
TEMPAD BSS     2              TEMPORARY STORAGE FOR D.P. SUMMING           00000056
```

```
NOADD TRA     SUM1+1        USED FOR INRPRD ENTRY                    00000057
  ADD FAD*    7,4           USED FOR PRDSUM ENTRY                    00000058
      END                                                            00000059
```

SUBROUTINE FRUN

```
$IBMAP FRUN    50,LIST,DECK,M94/2                                          RUN00010
*                                                                          RUN00040
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * RUN00050
*                                                                          RUN00060
*       CALL UNLOAD (LTAPE)                                                RUN00070
*            LTAPE = LOGICAL TAPE NUMBER                                   RUN00080
*                                                                          RUN00090
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * RUN00100
*                                                                          RUN00110
       LDIR                                                               RUN00120
UNLOAD SAVE    4,I                                                        RUN00130
       CLA*    3,4            LOGICAL NUMBER                              RUN00140
       STO     TEMP                                                       RUN00150
       CALL    .FVIO.(TEMP,TEMP+1)                                        RUN00160
       LAC     TEMP+1,4                                                   RUN00170
       SCA     SETB,4                                                     RUN00190
 RUN   TSX     .CLOSE,4                                                   RUN00250
 SETB  PZE     **                                                         RUN00260
 NOP   RETURN  UNLOAD                                                     RUN00270
       LORG                                                               RUN00280
 ERASE CONTRL  ER                                                         RUN00290
       USE     ER                                                         RUN00300
 TEMP  BSS     2                                                          RUN00310
       BSS     18                                                         RUN00320
       END                                                                RUN00330
```

APPENDIX III

PHASE I PROGRAM LISTING

This appendix contains the following listings:

PHASE I PROGRAM LISTING
PHASE I – MAIN PROGRAM

SUBROUTINE PHASE1

```
$IBFTC PHASE1  DECK
C           **************************************************
C           *                                                *
C          *RANDOM VIBRATION ANALYSIS SYSTEM FOR COMPLEX STRUCTURES*
C           *                                                *
C            *                ( R A N V I B )                *
C             **********************************************
C
C                           P H A S E    1
C
C    MAIN PROGRAM - THE EIGENVALUE EIGENVECTORS (FREMOD ROUTINE) AND THE
C                   STRUCTURAL ANALYSIS PROGRAM (MAST) IS CALLED IN THIS
C                   PROGRAM.
C
C
C
C                ***TAPE USAGE***
C                                NTAP10-OUTPUT TAPE. THE FOLLOWING ITEMS
C                                       ARE STORED ON THIS TAPE.
C                                     * * * * * * * * * * *
C                                     * STIFFNESS         *
C                                     * FLEXIBILITY       *
C                                     *             *EOF  *
C                                     * STRESS(PLATES)    *
C                                     *                   *
C                                     * STRESS(BEAMS)     *
C                                     *             *EOF  *
C                                     * FREQUENCIES       *
C                                     * MODE SHAPES       *
C                                     * GENERALIZED MASS  *
C                                     *             *EOF  *
C                                     * MASS(CARD INPUT)  *
C                                     *             *EOF  *
C                                     *                   *
C                                     *                   *
C                                     * * * * * * * * * * *
C
C
C
C                                NTAP2-  MAST OUTPUT. THE FOLLOWING
C                                       ITEMS ARE STORED ON THIS TAPE.
C                                     * * * * * * * * * * * * *
C                                     * PARAMETER MATRIX      *
C                                     *                 *EOF  *
C                                     * STIFFNESS MATRIX      *
C                                     *                 *EOF  *
C                                     * FLEXIBILITY MATRIX    *
C                                     *                 *EOF  *
C                                     * * * * * * * * * * * *
C
C
C                                NTAP8-  MAST OUTPUT. THE FOLLOWING
C                                       ITEMS ARE STORED ON THIS TAPE.
C                                     * * * * * * * * * * * * *
C                                     * PARAMETER MATRIX      *
C                                     *                 *EOF  *
C
```

```
C                                    * STRESS(BEAMS)          *
C                                    *                 *EOF   *
C                                    *                        *
C                                    * * * * * * * * * * * * *
C
C
C                                 NTAP12-  MAST OUTPUT. THE FOLLOWING
C                                       ITEMS ARE STORED ON THIS TAPE.
C                                     * * * * * * * * * * * * *
C                                     * PARAMETER MATRIX       *
C                                     *                 *EOF   *
C                                     * STRESS(PLATES)         *
C                                     *                 *EOF   *
C                                     *                        *
C                                     * * * * * * * * * * * * *
C
C
C                                IF FLG2=0  STRESSES ARE NOT CALCULATED
C                                       =1  STRESSES FOR PLATES ONLY
C                                       =2  STRESSES FOR BEAMS ONLY
C                                       =3  STRESSES FOR PLATES AND BEAMS
C
C                                IF FLG1=0       MAST AND FREMOD ROUTINES
C                                   FLG1 NOT 0   FREMOD ROUTINE ONLY
C
C
C
C
      COMMON/PRNT/NPRNTK,NSTRSP,NSTRSB
      INTEGER FLG1,FLG2
      NTAP1 = 10
      NTAP2 = 2
      NTAP8 = 8
      NTAP12 = 12
      REWIND NTAP1
      REWIND NTAP2
      REWIND NTAP8
      REWIND NTAP12
C
C*** READ IN THE CONTROL CARD
      READ(5,9000) FLG1,FLG2,NPRNTK,NSTRSP,NSTRSB
C
C*** IF FLG1 IS NOT EQUAL TO 0, THEN ONLY THE EIGENVALUE-EIGENVECTOR
C    ROUTINE IS CALLED.  OTHERWISE, THE MAST ROUTINE IS ALSO CALLED.
      IF( FLG1 .NE. 0 ) GO TO 50
C
C*** THE MAST PROGRAM CALCULATES THE STIFFNESS, FLEXIBILITY AND
C    STRESS MATRICES.
C     ****************************************************************
      CALL MAST
C
      REWIND NTAP1
C***                              MERGE THE STIFFNESS MATRIX
      NFILE = 1
C     ****************************************************************
      CALL AMERGE( NTAP2, NTAP1, NFILE )
C
C***                              MERGE THE FLEX MATRIX
```

```
      NFILE = 2
C     ******************************************************************
      CALL AMERGE( NTAP2, NTAP1, NFILE )
      END FILE NTAP1
C
C
      IF ( FLG2 .EQ. 0 ) GO TO 45
      GO TO (10,20,10), FLG2
C
C*** MERGE THE STRESS MARICES FOR PLATES AND REPARTITION TO 8XN
 10   ITYPE = 8
C     ******************************************************************
      CALL SMERGE( NTAP8,NTAP1,ITYPE )
      IF( FLG2 .EQ. 3 ) GO TO 20
      GO TO 45
C
C*** MERGE THE STRESS MATRICES FOR BEAMS AND REPARTITION TO 6XN
 20   ITYPE = 6
C     ******************************************************************
      CALL SMERGE( NTAP12,NTAP1,ITYPE )
C
C***FREMOD PROGRAM IS CALLED TO FIND THE EIGENVALUES(FREQUENCIES),
C                    EIGENVECTORS(MODE SHAPES) AND GENERALIZED MASSES.
C
C
 45   END FILE NTAP1
C
C     ******************************************************************
 50   CALL FREMOD
C
 9000 FORMAT(5I10)
      RETURN
      END
```

```
$IBFTC BLK      DECK
       BLOCK DATA                                                          021
       COMMON/TAPES/MT1,MT2,MT3,MT4,MT5,MT6,MT7,MT8,MT9,MT10,MT11,MT12,
      * MT13,MT14,MT15,MT16,MT17
       DATA MT1,MT2,MT3,MT4,MT5,MT6,MT7,MT8,MT9,MT10,MT11,MT12,MT13,MT14,
      * MT15,MT16,MT17/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17/
       COMMON/REDUC/NTEST,NTEST2
       DATA NTEST/6HAND ST/,NTEST2/6HONLY   /
       END                                                                 029
```

SUBROUTINE MAST

```
$IBFTC MAST*   DECK
      SUBROUTINE MAST                                                    041
C                                                                        043
C                                                                        067
C                                                                        068
C     PROGRAM CONTROL PARAMETERS                                         069
C     0  NO EXECUTION                                                    070
C     1  EXECUTION                                                       071
C     2  PRINT                                                           072
C                                                                        073
C     NP=0  (60*60 PARTITIONS)                                           074
C     IF NVIB EQUALS 1 PARTITIONS ARE PRINTED                            075
C                                                                        076
      COMMON/RENT/NRENT,KRENT
      COMMON/MAPSTR/IPTOT,IBTOT                                          077
      COMMON/CONT1/JPART(800)                                            078
      COMMON/LASTND/LN(200)                                              079
      COMMON/COMS/NSIZE(200)
      COMMON/TERMS/NBEAM,NPLATE,NNODE,NCOND,NPS,NTOL,NP,NOPT(4)
      COMMON/CONTRL/NDEFL,NKSP,NREX,NNF,NPSTR,NBSTR,NVIB,IF88            081
      COMMON/TAPES/MT1,MT2,MT3,MT4,MT5,MT6,MT7,MT8,MT9,MT10,MT11,MT12,
     * MT13,MT14,MT15,MT16,MT17
      COMMON/TITL/TITLE(13)
      COMMON/PAGE/NPAGE
      COMMON/REDUC/NTEST,NTEST2
      DIMENSION IERROR(5)
      INTEGER FRPR,SRP,SRB,FR
C                                                                        085
C SET ALL OPTIONS TO ZERO
      NDEFL = 0
      NKSP = 0
      NREX = 0
      NNF = 0
      NBSTR = 1
      NPSTR = 1
      IF88 = 0
      NCOND = 0
    1 NPS=0                                                              086
      NTOL=0                                                             087
      NPAGE=0
      DO 5 I=1,5
      IERROR(I) = 0
    5 CONTINUE
      REWIND MT2
      REWIND MT4
      REWIND MT8
      REWIND MT11
      REWIND MT12
      REWIND MT1
      REWIND MT3
      REWIND MT16
      IN = MT5
      IOUT = MT6
C                                                                        101
      READ(IN,9000)TITLE                                                 104
C
      READ (IN,9101) NOPT,NVIB,KRPR,FRPR,SRP,SRB,FR
```

```
      CALL PAGHED
      WRITE (IOUT,9001) NOPT
      READ(IN,9100)NBEAM,NPLATE,NNODE,NP
      WRITE(IOUT,9002)NBEAM,NPLATE,NNODE,NP
      IF (NOPT(3) .EQ. NTEST) GO TO 10
      IF (NOPT(3) .EQ. NTEST2) GO TO 10
      GO TO 99                                                          107
C
   10 CONTINUE
      CALL SUBM1                                                        108
                                                                        109
C
      CALL SOLN                                                         111

C
C***** THIS SECTION CONTROLS THE EXECUTION OF THE TL01 DATA PHASES.
C
      REWIND MT9
      AREA = 1.0

C SPACE TAPE PAST FIRST FILE CONTAINING FORTRAN GENERATION ROUTINES
      I=1
   50 CONTINUE
      CALL FSF(1,MT9,IER)
      IF(IER .NE. 0) GO TO 900

C EXECUTE A.TL01 PHASE (WHEN I=1, KR WILL BE COMPUTED)
      CALL TL01(MT9,0,IERROR)
      DO 55 ICELL=1,5
      IF(IERROR(ICELL) .NE. 0) GO TO 999
   55 CONTINUE

C TRANSFER TO APPROPRIATE TEST TO DETERMINE NEXT TL01 PHASE TO BE EXECUTED.
      GO TO (100,200,300,400,500,600,700,800),I
  100 CONTINUE

C IS PRINTING OF KR DESIRED
      IF(KRPR .EQ. 0) GO TO 150

C PRINT KR
      I = 2
      GO TO 50

C NO PRINT OF KR
  150 CONTINUE
      AREA = 2.0
      CALL FSF(1,MT9,IER)
      IF(IER .NE. 0) GO TO 900

C IS FR DESIRED
  200 CONTINUE
      IF(FR .NE. 0) GO TO 250
      I = 3
      GO TO 50

C NO FR DESIRED
  250 CONTINUE
      AREA = 3.0
      CALL FSF(2,MT9,IER)
      IF(IER .NE. 0) GO TO 900
      GO TO 400
```

```
C IS PRINT OF FR DESIRED
  300 CONTINUE
      IF(FRPR .EQ. 0) GO TO 350
      I = 4
      GO TO 50

C NO PRINT OF FR DESIRED
  350 CONTINUE
      AREA = 4.0
      CALL FSF(1,MT9,IER)
      IF(IER .NE. 0) GO TO 900

C ARE STRESSES DESIRED
  400 CONTINUE
      IF(NOPT(3) .NE. NTEST) GO TO 800
      IF(NPLATE .EQ. 0) GO TO 450
      I = 5
      GO TO 50

C NO PLATES
  450 CONTINUE
      AREA = 5.0
      CALL FSF(2,MT9,IER)
      IF(IER .NE. 0) GO TO 900
      GO TO 600

C IS PRINT OF PLATE STRESS DESIRED
  500 CONTINUE
      IF(SRP .EQ. 0) GO TO 550
      I = 6
      GO TO 50

C NO PRINT OF PLATE STRESSES
  550 CONTINUE
      AREA = 6.0
      CALL FSF(1,MT9,IER)
      IF(IER .NE. 0) GO TO 900

C ANY BEAMS
  600 CONTINUE
      IF(NBEAM .EQ. 0) GO TO 800
      I = 7
      GO TO 50

C IS PRINT OF BEAM STRESSES DESIRED
  700 CONTINUE
      IF(SRB .EQ. 0) GO TO 800
      I = 8
      GO TO 50
  800 CONTINUE
      REWIND MT9
      RETURN

C ERROR COMMENT
  900 CONTINUE
      WRITE(IOUT,9902) IER,MT9,AREA
      STOP
   99 CONTINUE
      WRITE (IOUT,9900) NOPT
```

```
      STOP
  999 CONTINUE
      WRITE (IOUT,9901) IERROR
      STOP
 9900 FORMAT(//49H ILLEGAL OPTION CONTROL CARD. CARD READ WAS.......,4A6,
     */104H IT SHOULD HAVE BEEN ONE OF THE FOLLOWING.....DEFLECTIONS ONL
     *Y........OR........DEFLECTIONS AND STRESSES)
 9901 FORMAT (//54H TL01 ERROR WHILE TRYING TO EXECUTE THE FOLLOWING CAR
     *D/5(5X,I10))
 9902 FORMAT(//31H FSF ERROR IN MAIN. THE CODE = I5,5X,7HTAPE = I5,5X,
     * 7HAREA = F2.0)                                                    112
C                                                                        
 9000 FORMAT(13A6)
 9001 FORMAT(//11H CALCULATE ,4A6)
 9002 FORMAT(//15H STRUCTURE SIZE 7X,5HBEAMS 5X,6HPLATES 4X,
     1 5HNODES 3X,10HPARTITIONS/16X,4I10)
 9100 FORMAT(3I4,4X,I4)
 9101 FORMAT(4A6,3X,I1,2X,I1,2X,I1,2X,I1,2X,I1,2X,I1)                    116
C                                                                        
      END                                                                118
```

SUBROUTINE PAGHED

```
$IBFTC PAGH*    DECK
       SUBROUTINE PAGHED
       COMMON /TITL/TITLE(13)
       COMMON/TAPES/MT1,MT2,MT3,MT4,MT5,MT6,MT7,MT8,MT9,MT10,MT11,MT12,
      * MT13,MT14,MT15,MT16,MT17
       COMMON/PAGE/NPAGE
       IOUT = MT6
       NPAGE=NPAGE+1
       WRITE(IOUT,100)TITLE,NPAGE
  100  FORMAT(1H1,23X,13A6,5X,5HPAGE ,I6)
       RETURN
       END
```

SUBROUTINE UNPACK

```
$IBFTC UNPAC*  DECK
      SUBROUTINE UNPACK(IJKLMN,I,J,K,L,M,N)                              120
C                                                                        121
C     ROUTINE UNPACKS BOUNDARY CONDITIONS                                122
C                                                                        123
      I=IJKLMN/100000                                                    124
      JP=IJKLMN-100000*I                                                 125
      J=JP/10000                                                         126
      KP=JP-10000*J                                                      127
      K=KP/1000                                                          128
      LP=KP-1000*K                                                       129
      L=LP/100                                                           130
      MP=LP-100*L                                                        131
      M=MP/10                                                            132
      N=MP-10*M                                                          133
C                                                                        134
      RETURN                                                             135
      END                                                                136
```

SUBROUTINE PRINT

```
$IBFTC PRINT*  DECK
      SUBROUTINE PRINT (A,NROW,NCOL,CASE,TITLE,PAGE,NIZ)                    171
C                                                                           172
C                                                                           173
C     PRINTS ONE OR TWO DIMENSIONAL ARRAYS                                  174
C     LIST OF ARGUMENTS FOR PRINT ROUTINE                                   175
C     NROW(I5),NCOL(I5),CASE(I6),TITLE(A6),PAGE(A1)                         176
C                                                                           177
      DIMENSION A(NIZ,1),FMT (4)                                            178
      INTEGER FMT,H8                                                        179
      DATA IOUT/6/,NPRNT/8/,H8/1H8/                                         180
      DATA FMT(1)/24H(1X2I3,1P          E14.5)/                             181
C                                                                           182
      IF(NROW.LE.24) WRITE(IOUT,9101)PAGE                                   183
      N2=0                                                                  184
      IF(NCOL.LT.NPRNT) GO TO 160                                           185
      M=1                                                                   186
   95 FMT(3)=H8                                                             187
  100 N1=N2+1                                                               188
      N2=N2+NPRNT                                                           189
      IF(NROW.GT.24) GO TO 120                                              190
      IF(M*(NROW+6    ).LT.60) GO TO 130                                    191
  120 M=1                                                                   192
      WRITE(IOUT,9102)                                                      193
  130 IF(N2.GT.NCOL) GO TO 170                                              194
  150 WRITE(IOUT,9103)TITLE,CASE,NROW,NCOL,N1,N2                            195
      WRITE(IOUT,FMT)(I,N1,(A(I,J),J=N1,N2),I=1,NROW)                       196
      M=M+1                                                                 197
      IF(N2.NE.NCOL) GO TO 100                                              198
C                                                                           199
      RETURN                                                                200
C                                                                           201
  160 N1=1                                                                  202
  170 N2=NCOL                                                               203
      FMT(3)=N2-N1+1                                                        204
      GO TO 150                                                             205
C                                                                           206
 9101 FORMAT(A1)                                                            207
 9102 FORMAT(1H1)                                                           208
 9103 FORMAT(//1X,7HMATRIX A6,10H IDENT NO.I6,I5,6H ROWS                    209
     1 I5,6H COLS 3X,7H1ST COL,I4,3X,8HLAST COL,I4//)                       210
C                                                                           211
      END                                                                   212
```

SUBROUTINE SUBM1

```
$IBFTC SUBM1*  DECK
      SUBROUTINE SUBM1                                                  140
C                                                                       141
C     ROUTINE CALLS FOR GENERATION AND MERGER                           142
C     INER=1 IF A QUADRILATERAL PLATE FAILS                             143
C     COPLANARITY TEST (SEE COPLAN SUBROUTINE)                          144
C                                                                       145
      COMMON/CONT1/JPART(800)                                           146
      COMMON/CONT2/KPART(800)                                           147
      COMMON/CONT3/LPART(800)                                           148
      COMMON/RENT/NRENT,KRENT                                           149
      COMMON/LASTND/LN(200)                                             150
      COMMON/COMS/NSIZE(200)                                            151
      COMMON/TERMS/NBEAM,NPLATE,NNODE,NCOND,NPS,NTOL,NP                 152
      COMMON/CONTRL/NDEFL,NKSP,NREX,NNF,NPSTR,NBSTR,NVIB                153
      COMMON/SKIP/NBSP,NBSB,NBSPI,NBSBI                                 154
      COMMON/NOMERG/INER                                                155
C                                                                       159
      INER=0                                                            160
      NRENT=0                                                           161
      CALL GENRAT                                                       162
C                                                                       163
      IF(NRENT.NE.0)GO TO 1000                                          164
      IF(INER.EQ.1)GO TO 1000                                           165
      CALL MERGE                                                        166
C                                                                       167
 1000 RETURN                                                            168
      END                                                               169
```

SUBROUTINE GENRAT

```
$IBFTC GENRA*  DECK
      SUBROUTINE GENRAT                                                  215
                                                                         216
C                                                                        217
C     CONTROL FOR GENERATION OF ELEMENT MATRICES                         218
C                                                                        219
      COMMON/CONT1/JPART(800)                                            220
      COMMON/CONT2/KPART(800)                                            221
      COMMON/CONT3/LPART(800)                                            222
      COMMON/LASTND/LN(200)                                              223
      COMMON/CORD/XN(2000),YN(2000),ZN(2000)                             224
      COMMON/CONTRL/NDEFL,NKSP,NREX,NNF,NPSTR,NBSTR,NVIB                 225
      COMMON/TERMS/NBEAM,NPLATE,NNODE,NCOND,NPS,NTOL,NP                  226
      COMMON/ADPRO/EM,G,RC,ALFA,DARC,DL                                  227
      COMMON/SSTR/EMM,GG                                                 228
      COMMON/CHECK/ACPT,GROSS
      COMMON/TAPES/MT1,MT2,MT3,MT4,MT5,MT6,MT7,MT8,MT9,MT10,MT11,MT12,
     * MT13,MT14,MT15,MT16,MT17
      COMMON/FLAG/NFLAG                                                  232
      COMMON/TITL/TITLE(13)
                                                                         233
C
      IN = MT5
      IOUT = MT6
                                                                         241
C                                                                        242
      NFLAG=0                                                            243
      CALL INFO                                                          244
C                                                                        245
C                                                                        246
      IF(NPLATE.EQ.0)GO TO 100
      ACPT=0.1
      GROSS=1.0
      READ(IN,9000) EMM,GG
      CALL PAGHED
      CALL PLATE                                                         248
  100 CONTINUE                                                           249
                                                                         250
C                                                                        251
      IF(NBEAM.EQ.0)GO TO 200                                            252
      READ(IN,9000)EMM,GG
      CALL PAGHED
      CALL BEAM                                                          253
  200 CONTINUE                                                           254
                                                                         255
C                                                                        263
      IF(NFLAG.NE.0)STOP                                                 264
C                                                                        265
 9000 FORMAT(6E12.4)                                                     266
      RETURN                                                             267
      END
```

SUBROUTINE REDUCE

```
$IBFTC REDUC*  DECK
      SUBROUTINE REDUCE(A,N,K)                                          273
C
      DIMENSION A(N,N)                                                  275
C                                                                       276
      DO 100 I=1,N                                                      277
      DO 100 J=1,N                                                      278
      IF((I.EQ.K).OR.(J.EQ.K))GO TO 100                                 279
      A(I,J)=A(I,J)-A(K,J)*A(I,K)/A(K,K)                                280
  100 CONTINUE                                                          281
C                                                                       305
      DO 500 L=1,N                                                      306
      A(L,K)=0.0                                                        307
  500 A(K,L)=0.0                                                        308
C                                                                       309
      RETURN                                                            310
      END
```

SUBROUTINE INFO

```
$IBFTC INFO*   DECK
      SUBROUTINE INFO                                                    314
C                                                                        315
C     ROUTINE READS IN BOUNDARY CONDITIONS                               316
C     AND WRITES BOUNDARY DATA ON TWO TAPES                              317
C     TAPE IKDF CONTAINS SPECIFIED DEFLECTIONS                           318
C     TAPE IKBC CONTAINS NODAL,B.C. AND SPRING DATA                      319
C     N18 = TAPE CONTAINING REDUCTION INFORMATION
C                                                                        320
      COMMON/CONT1/JPART(800)                                            321
      COMMON/CONT2/KPART(800)                                            322
      COMMON/CONT3/LPART(800)                                            323
      COMMON/LASTND/LN(200)                                              324
      COMMON/CORD/XN(2000),YN(2000),ZN(2000)                             325
      COMMON/CONTRL/NDEFL,NKSP,NREX,NNF,NPSTR,NBSTR,NVIB                 326
      COMMON/TERMS/NBEAM,NPLATE,NNODE,NCOND,NUM1,IUM2,NP
      COMMON/TAPES/MT1,MT2,MT3,MT4,MT5,MT6,MT7,MT8,MT9,MT10,MT11,MT12,
     * MT13,MT14,MT15,MT16,MT17
      COMMON/TITL/TITLE(13)
      COMMON/REDUC/NTEST
      DIMENSION FKK(6),JKK(6)                                            331
      DIMENSION KFIX(6)                                                  333
      DIMENSIONMAPD(200)
      DIMENSION IRETAN(6)
      EQUIVALENCE(MAPD,JPART)


      IN = MT5
      IOUT = MT6
      IKDF = MT4
      IKBC = MT3
      N18 = MT1
      DO 30 I=1,200                                                      334
   30 LN(I)=0                                                            335
C                                                                        336
      DO 40 I=1,800                                                      337
      JPART(I)=0                                                         338
      KPART(I)=0                                                         339
   40 LPART(I)=0                                                         340
C                                                                        341
```

```
C****** SET UP PARTITIONS                                                    INFO
      IF(NP.NE.0)GO TO 100                                                   342
C                                                                            343
      NP=NNODE/10                                                            344
      DO 20 I=1,NP                                                           345
   20 LN(I)=10*I                                                             346
      IF((NNODE-NP*10).EQ.0)GO TO 99                                         347
      NP=NP+1                                                                348
      LN(NP)=NNODE                                                           349
   99 CONTINUE                                                               350
      NUM1=NP                                                                351
      GO TO 101                                                              352
  100 NUM1=NP                                                                353
      READ(IN,9080)(LN(I),I=1,NP)                                            354
  101 CONTINUE                                                               355
C                                                                            356
      WRITE(IOUT,9000)(I,LN(I),I=1,NP)
      CALL PAGHED
      LINE=0
      WRITE(IOUT,9001)
C                                                                            361
C****** LOOP FOR EACH NODE
      DO 500 I=1,NNODE                                                       362
C                                                                            363
C****** DETERMINE WHICH PARTITION THIS NODE IS IN
      DO 250 K=1,NP                                                          364
  210 IF(I.GT.LN(K))GO TO 250                                                365
      NI=K                                                                   366
      GO TO 251                                                              367
  250 CONTINUE                                                               368
  251 CONTINUE                                                               369

C****** READ IN NODAL DATA
   45 READ(IN,9021) M,JKK,XN(M),YN(M),ZN(M),(IRETAN(IO),IO=1,6)
      DO 47 IO=1,6
   47 IRETAN(IO) = IABS(IRETAN(IO))
      WRITE(N18) M
      WRITE(N18) (IRETAN(IO),IO=1,6)
   46 CONTINUE
      IF(I.NE.M)WRITE(IOUT,9070)M                                            371
      DO 51 K=1,6
   51 JKK(K)=IABS(JKK(K))
      IJKLMN=100000*JKK(1)+10000*JKK(2)+1000*JKK(3)+100*JKK(4)+10*JKK(5)     372
     1+JKK(6)                                                                373
      WRITE(IKBC)M,IJKLMN                                                    374
      IF(LINE.LT.50)GO TO 1
      LINE=0
      CALL PAGHED
      WRITE(IOUT,9001)
    1 CONTINUE
      LINE=LINE+1
   60 WRITE(IOUT,9012) M,JKK,XN(M),YN(M),ZN(M),(IRETAN(IO),IO=1,6)
C                                                                            375
   65 K=0                                                                    376
      DO 300 N=1,6                                                           377
  300 IF(JKK(N).EQ.3)K=1                                                     378
      IF(K.NE.1)GO TO 350                                                    379
      READ(IN,9030)(FKK(JJ),JJ=1,6)                                          380
      WRITE(IKBC)(FKK(JJ),JJ=1,6)                                            381
C****** READ SPRING DATA
```

```
      WRITE(IOUT,9003)(FKK(JJ),JJ=1,6)
      LINE=LINE+1
      IF(LINE.LT.50)GO TO 2
      CALL PAGHED
      LINE=0
      WRITE(IOUT,9001)
    2 CONTINUE
  350 CONTINUE                                                          382
C                                                                       383
C****** COUNT FIXED DEGREES OF FREEDOM
      K=0                                                               385
      DO 102 N=1,6                                                      386
      IF((JKK(N).EQ.0).OR.(JKK(N).EQ.3))GO TO 102                       387
      K=K+1                                                             388
      KFIX(K)=N                                                         389
  102 CONTINUE                                                          392
C                                                                       417
  500 CONTINUE                                                          418
C                                                                       419
C                                                                       420
 9000 FORMAT(//25H LAST NODES IN PARTITIONS/(1X,10(I3,1H,I4,5X)))
 9001 FORMAT(//41X,10HNODAL DATA//6X,4HNODE,9X,2HBC,14X,1HX,18X,1HY,18X,
     1  1HZ,7X,17HRETAINED FREEDOMS//)
 9002 FORMAT(6X,I4,7X,6I1,6X,1PE12.4,7X,1PE12.4,7X,1PE12.4)
 9003 FORMAT(9H SPRINGS 6E12.4)
 9012 FORMAT(6X,I4,7X,6I1,6X,1PE12.4,7X,1PE12.4,7X,1PE12.4,7X,6I1)
 9020 FORMAT(I4,6X,6I1,8X,3E12.4)
 9021 FORMAT(I4,6X,6I1,8X,3E12.4,4X,6I1)
 9030 FORMAT(6E12.4)                                                    422
 9040 FORMAT(I4)
 9050 FORMAT(6E12.4)
 9070 FORMAT(6H NODE ,I4,16H OUT OF SEQUENCE )                          424
 9080 FORMAT(18I4)                                                      425
      REWIND N18
      RETURN                                                            426
      END                                                               427
```

SUBROUTINE PLATE

```
$IBFTC PLATE*  DECK
      SUBROUTINE PLATE                                                    431
C                                                                         432
      COMMON/RENT/NRENT,KRENT                                             433
      COMMON/CONT1/JPART(800)                                             434
      COMMON/CONT3/LPART(800)                                             435
      COMMON/LASTND/LN(200)                                               436
      COMMON/TAPES/MT1,MT2,MT3,MT4,MT5,MT6,MT7,MT8,MT9,MT10,MT11,MT12,
     * MT13,MT14,MT15,MT16,MT17
      COMMON/CORD/XN(2000),YN(2000),ZN(2000)                              440
      COMMON/TERMS/NBEAM,NPLATE,NNODE,NCOND,IPT,NPS ,IUM2                 441
C                                                                         442
      COMMON/ADPRO/EM,G,RC,ALFA,DARC,DL                                   443
      COMMON/SSTR/EMM,GG                                                  444
      COMMON/PSTIF2/H,TS                                                  445
      COMMON/PSTIF1/SK(18,18)                                             446
      COMMON/PSTIF8/SKQ(30,30)                                            447
      COMMON/PSTIF9/NP,N5                                                 448
      COMMON/PSTIFC/TIBX,TIBY,TIBS                                        449
      COMMON/PSTIFD/TX,TY,TO                                              450
      COMMON/PSTIFG/XS1,YS1,ZS1,XS2,YS2,ZS2,XS3,YS3,ZS3,XS4,YS4,ZS4       451
      COMMON/PMTR1/PT(3,3)
      COMMON/STRAN/GT(8,24)                                               453
      COMMON/FLAG/NFLAG                                                   454
      COMMON/PSTIFL/ISWAP                                                 455
      COMMON/SWAP/KSWAP                                                   456
      COMMON/PSTIFH/XL1,YL1,XL2,YL2,XL3,YL3,XL4,YL4                       457
      COMMON/TITL/TITLE(13)
      COMMON/BUCK/IBUC
      COMMON/THRST/TRS1,TRS2,TRS3,TRS4,TRS5,TRS6,TRS7
      COMMON/PSTRSS/SKQSS(30,30)
C                                                                         459
      DIMENSION NN(10),F(24,24),PI(8,40)
      EQUIVALENCE(F,SK)
C                                                                         463
C                                                                         464
C  N4 - NODE NUMBER FOR NODE 4 ON QUADRILATERAL PLATE                     466
C  IO - OUTPUT PARAMETER (1  PRINT)                                       467
C  TIBX - MOMENT OF INERTIA FOR OUT OF PLANE STIFFNESS IN X-DIRECTION     468
C  NP - PLATE NUMBER                                                      465
C  TIBY - MOMENT OF INERTIA FOR OUT OF PLANE STIFFNESS IN Y-DIRECTION     469
C  TX - THICKNESS FOR IN-PLANE LOADING IN X-DIRECTION                     471
C  TIBS - MOMENT OF INERTIA FOR OUT OF PLANE STIFFNESS IN SKIN DIRECTION  470
C  TY - THICKNESS FOR IN-PLANE LOADING IN Y-DIRECTION                     472
C  TO - THICKNESS FOR IN-PLANE SHEAR                                      473
C  TS - THICKNESS FOR SKIN                                                474
C  E - YOUNGS MODULUS                                                     475
C  GAMMA - POISSON,S RATIO                                                476
C  SK66 - STIFFNESS MATRIX FOR (INPST) AND (OUTPM)                        477
C  SK99 - STIFFNESS MATRIX FOR (INP) AND (OUTP)                           478
C XS1,YS1,ZS1 - STRUCTURAL COORDINATE AT NODE 1 TRIANULAR OR QUADRILATER  479
C XS2,YS2,ZS2 - STRUCTURAL COORDINATE AT NODE 2 TRIANULAR OR QUADRILATER  480
C XS3,YS3,ZS3 - STRUCTURAL COORDINATE AT NODE 3 TRIANULAR OR QUADRILATER  481
C XS4,YS4,ZS4 - STRUCTURAL COORDINATE AT NODE 4 QUADRILATERAL PLATE       482
C  XL1,YL1 - LOCAL COORDINATES AT NODE 1 TRIANGULAR OR QUADRILATERAL PLA  483
C  XL2,YL2 - LOCAL COORDINATES AT NODE 2 TRIANGULAR OR QUADRILATERAL PLA  484
C  XL3,YL3 - LOCAL COORDINATES AT NODE 3 TRIANGULAR OR QUADRILATERAL PLA  485
C  XL4,YL4 - LOCAL COORDINATES AT NODE 4 QUADRILATERAL PLATE              486
C                                                                         487
C                                                                         488
C                                                                         489
```

```
C                                                                       490
      ICON(I,J)=1000*MAX0(I,J)+MIN0(I,J)                                491
      IN = MT5
      IOUT = MT6
      KSOLN = MT8
      ISTRS = MT16
      K1=1                                                              492
      ICC=0                                                             493
      LINE=0
      WRITE(IOUT,9000)(II,II=1,4)
 1000 NT=0                                                              494
 1001 NT=NT+1                                                           495
C     READ PLATE DATA                                                   496
      READ(IN,9500)NP,N1,N2,N3,N4,IIN,IO,IBUC,T,EM,G
      IF(IBUC.NE.0)READ(IN,9502)TRS1,TRS2,TRS3,TRS4,TRS5,TRS6,TRS7
      IF(EM)1060,1070,1080                                              499
 1060 WRITE(IOUT,9560)NP                                                500
      EM=ABS(EM)                                                        501
      GO TO 1080                                                        502
 1070 EM=EMM                                                            503
 1080 IF(G)1090,1100,1110                                               504
 1090 WRITE(IOUT,9570)NP                                                505
      G=ABS(G)                                                          506
      GO TO 1110                                                        507
 1100 G=GG                                                              508
 1110 IF(NP-NT)1120,1125,1120                                           509
 1120 WRITE(IOUT,9580)NP,NT                                             510
      CALL EXIT                                                         511
C                                                                       512
 1125 CONTINUE                                                          513
      KSWAP=0                                                           514
      IF(IIN-1)1050,2,3                                                 515
    2 TO=T                                                              516
      TS=T                                                              517
      TIBS=T**3/12.                                                     518
      TX=0.                                                             519
      TY=0.                                                             520
      TIBX=0.                                                           521
      TIBY=0.                                                           522
      GO TO 1035                                                        523
    3 READ(IN,9501)TX,TIBX,TY,TIBY,TS,TIBS                              524
      TO=T                                                              525
      IF(TS.EQ.0.) TS=TO                                                526
      IF(TIBS.EQ.0.) TIBS=TO**3/12.                                     527
      IF(IIN.EQ.2) GO TO 4                                              528
      T=1.-G**2                                                         529
      TX=-TO*(1.-TX)*T                                                  530
      TY=-TO*(1.-TY)*T                                                  531
      TIBX=-TO*(1.-TIBX)*T                                              532
      TIBY=-TO*(1.-TIBY)*T                                              533
      GO TO 1035                                                        534
    4 CONTINUE                                                          535
C     CHECK VALIDITY OF PLATE INPUT                                     536
C     POSITIVE MOMENTS OF INERTIA AND THICKNESSES                       537
      IF(NNODE-N1)1020,1005,1005                                        538
 1005 IF(NNODE-N2)1020,1010,1010                                        539
 1010 IF(NNODE-N3)1020,1015,1015                                        540
 1015 IF(NNODE-N4)1020,1025,1025                                        541
 1020 WRITE(IOUT,9550)NP                                                542
      NFLAG=1                                                           543
 1025 CONTINUE                                                          544
```

```
      IF(TIBX.GE.0.)GO TO 1027                                          545
      WRITE(IOUT,9590)NP                                                546
      TIBX=ABS(TIBX)                                                    547
 1027 IF(TIBY.GE.0.)GO TO 1029                                          548
      WRITE(IOUT,9590)NP                                                549
      TIBY=ABS(TIBY)                                                    550
 1029 IF(TIBS.GE.0.)GO TO 1031                                          551
      WRITE(IOUT,9600)NP                                                552
      TIBS=ABS(TIBS)                                                    553
 1031 IF(TX.GE.0.)GO TO 1033                                            554
      WRITE(IOUT,9600)NP                                                555
      TX=ABS(TX)                                                        556
 1033 IF(TY.GE.0.)GO TO 1035                                            557
      WRITE(IOUT,9600)NP                                                558
      TY=ABS(TY)                                                        559
 1035 IF(TO.GE.0.)GO TO 1037                                            560
      WRITE(IOUT,9600)NP                                                561
      TO=ABS(TO)                                                        562
 1037 IF(TS.GE.0.)GO TO 1050                                            563
      WRITE(IOUT,9600)NP                                                564
      TS=ABS(TS)                                                        565
 1050 CONTINUE                                                          566
      WRITE(IOUT,9001)NP,N1,N2,N3,N4,TO,TX,TIBX,TY,TIBY,TS,TIBS,EM,G
      IF(IBUG .NE. 0) WRITE(IOUT,6000) TRS1,TRS2,TRS3
      LINE=LINE+2
      IF(LINE.LT.50)GO TO 10
      CALL PAGHED
      LINE=0
      WRITE(IOUT,9000)(II,II=1,4)
   10 CONTINUE
      M=N1                                                              567
      XS1=XN(M)                                                         568
      YS1=YN(M)                                                         569
      ZS1=ZN(M)                                                         570
C                                                                       571
      M=N2                                                              572
      XS2=XN(M)                                                         573
      YS2=YN(M)                                                         574
      ZS2=ZN(M)                                                         575
C                                                                       576
      M=N3                                                              577
      XS3=XN(M)                                                         578
      YS3=YN(M)                                                         579
      ZS3=ZN(M)                                                         580
C                                                                       581
      N5=N4                                                             582
      IF(N5.EQ.0)GO TO 1127                                             583
      XS4=XN(N5)                                                        584
      YS4=YN(N5)                                                        585
      ZS4=ZN(N5)                                                        586
 1127 CONTINUE                                                          587
C                                                                       588
C                                                                       589
      DO 1 I=1,30                                                       590
      DO 1 J=1,30                                                       591
      SKQSS(I,J)=0.0
    1 SKQ(I,J)=0.0                                                      592
C                                                                       593
C     CALL THE GENERATION OF THE STIFFNESS AND FIXED END FORCES         594
      CALL PSTIF(LINE)
  706 CONTINUE
```

```
      IF(LINE.LT.50)GO TO 21
      LINE=0
      CALL PAGHED
      WRITE(IOUT,9000)(II,II=1,4)
   21 CONTINUE
      IF(KRENT.NE.0) GO TO 1002                                          596
      IF(ISWAP.EQ.0) GO TO 1251                                          597
      NSAVE=N3                                                           598
      N3=N4                                                              599
      N4=NSAVE                                                           600
      KSWAP=1                                                            601
      GO TO 10
 1251 CONTINUE                                                           603
C                                                                        604
 1002 CONTINUE                                                           605
C     CONNECTIVITY INFO FOR PLATE                                        606
C                                                                        607
C                                                                        608
      MPT=IPT-1                                                          609
      DO 1129 I=1,MPT                                                    610
      IF((N1.LE.LN(I+1)).AND.(N1.GT.LN(I)))NI=I+1                        611
      IF((N2.LE.LN(I+1)).AND.(N2.GT.LN(I)))NJ=I+1                        612
      IF((N3.LE.LN(I+1)).AND.(N3.GT.LN(I)))NK=I+1                        613
      IF((N4.LE.LN(I+1)).AND.(N4.GT.LN(I)))NL=I+1                        614
      IF(N1.LE.LN(1))NI=1                                                615
      IF(N2.LE.LN(1))NJ=1                                                616
      IF(N3.LE.LN(1))NK=1                                                617
      IF(N4.LE.LN(1))NL=1                                                618
 1129 CONTINUE                                                           619
      LL=6                                                               620
      NN(1)=1001*NI                                                      621
      NN(2)=ICON(NJ,NI)                                                  622
      NN(3)=1001*NJ                                                      623
      NN(4)=ICON(NK,NI)                                                  624
      NN(5)=ICON(NK,NJ)                                                  625
      NN(6)=1001*NK                                                      626
      IF(N4.EQ.0)GO TO 1132                                              627
      NN(7)=ICON(NL,NI)                                                  628
      NN(8)=ICON(NL,NJ)                                                  629
      NN(9)=ICON(NL,NK)                                                  630
      NN(10)=1001*NL                                                     631
      LL=10                                                              632
 1132 CONTINUE                                                           633
      DO 1135 J=1,LL                                                     634
      DO 1134 I=1,K1                                                     635
      II=I                                                               636
      IF(JPART(I).EQ.NN(J))GO TO 1135                                    637
 1134 CONTINUE                                                           638
      II=K1                                                              639
      K1=K1+1                                                            640
      JPART(II)=NN(J)                                                    641
      LPART(II)=10001*NP                                                 642
 1135 LPART(II)=(LPART(II)/10000)*10000+NP                               643
      NPS=K1-1                                                           644
C                                                                        645
C     CALL FOR 6X6 TRANSFORMATION                                        646
      CALL PMTR                                                          647
C                                                                        648
C     CALL FOR STRESS TRANSFORMATION                                     649
      CALL SMTR                                                          650
C                                                                        651
```

```
C                                                                       652
      LIM=18                                                            667
      IF(N4.NE.0)LIM=24                                                 668
C     * * *          TRANSFORMATION- K(LAMBDA-T) * * *
C
      CALL KLAMT(SKQSS,PT)
C
C     * * * GENERATE STRESS MATRIX * * *
C
      DO 1275 I=1,8                                                     691
      DO 1275 J=1,24
 1275 PI(I,J)=0.0                                                       693
C                                                                       694
      CALL MUL1(GT,SKQSS,PI,1,8,LIM,LIM,8,30,8)
      IF(IBUC)1276,1277,1276
 1276 CALL KLAMT(SKQ,PT)
      GO TO 1279
 1277 DO 1278 I=1,30
      DO 1278 J=1,30
 1278 SKQ(I,J)=SKQSS(I,J)
C     * * *  COMPLETE COORD. TRANSFORMATION - LAMBDA(K*LAMDA-T) * * *
 1279 CALL LAMK(SKQ,PT)
C     WRITING STIFFNESS                                                 678
C                                                                       679
      WRITE(KSOLN)N1,N2,N3,N4                                           680
C                                                                       681
      WRITE(KSOLN)((SKQ (J,I),J=1,6),I=1,6),                            682
     1((SKQ(J,I),J= 1, 6),I= 7,12),((SKQ(J,I),J= 1, 6),I=13,18),        683
     2((SKQ(J,I),J= 1, 6),I=19,24),((SKQ(J,I),J= 7,12),I= 1, 6),        684
     3((SKQ(J,I),J= 7,12),I= 7,12),((SKQ(J,I),J= 7,12),I=13,18),        685
     4((SKQ(J,I),J= 7,12),I=19,24),((SKQ(J,I),J=13,18),I= 1, 6),        686
     5((SKQ(J,I),J=13,18),I= 7,12),((SKQ(J,I),J=13,18),I=13,18),        687
     6((SKQ(J,I),J=13,18),I=19,24),((SKQ(J,I),J=19,24),I= 1, 6),        688
     7((SKQ(J,I),J=19,24),I= 7,12),((SKQ(J,I),J=19,24),I=13,18),        689
     8((SKQ(J,I),J=19,24),I=19,24)                                      690
C     WRITING STRESS                                                    697
C                                                                       698
      WRITE(ISTRS)NP,N1,N2,N3,N4                                        699
      WRITE(ISTRS)((PI(I,J),I=1,8),J=1,24)                              700
C                                                                       701
C                                                                       702
      IF(IO.NE.1)GO TO 1280                                             703
C     PRINT OPTIONS FOR INDIVIDUAL PLATE                                704
      CALL PRINT(GT,8,LIM,1,4HSMTR,1,8)                                 706
      CALL PRINT(SKQ,LIM,LIM,1,4HSTIF,1,30 )                            707
      CALL PRINT(PI,8,LIM,1,4HSTRS,1,8)                                 708
 1280 CONTINUE                                                          709
C                                                                       710
C                                                                       711
C                                                                       763
C                                                                       808
C                                                                       809
C                                                                       810
      IF(NPLATE-NT)2003,2003,1001                                       811
C                                                                       812
C                                                                       813
 2003 CONTINUE                                                          814
C                                                                       815
 6000 FORMAT(15X,10HSIGMA-X = ,E12.5,3X,10HSIGMA-Y = ,E12.5,3X,9HTAU-XY
     *= ,E12.5)
 9000 FORMAT(/56X,10HPLATE DATA//
```

```
     1 1X,   6H PLATE,4(5H NODE ),4X,4HT(O),7X,4HT(X),7X,4HI(X),7X,
     2 4HT(Y),7X,4HI(Y),7X,4HT(S),7X,4HI(S),9X,1HE,7X,5HGAMMA//5X,4I5//)
 9001 FORMAT(1X,5I5,9(1PE11.3))
 9500 FORMAT(7I4,4X,I2,2X,3E12.4)
 9501 FORMAT(6E12.4)
 9502 FORMAT (7E10.4)
 9550 FORMAT(34H1INCORRECT NODE NUMBER, PLATE NO. I4)                          819
 9560 FORMAT(43H1NEGATIVE MODULUS OF ELASTICITY, PLATE NO. I4)                 820
 9570 FORMAT(35H1NEGATIVE POISSON RATIO, PLATE NO. I4)                         821
 9580 FORMAT(41H1INPUT NOT IN PROPER SEQUENCE, PLATE NO. I4,                   822
     111H SHOULD BE I4)                                                        823
 9590 FORMAT(38H1NEGATIVE MOMENT OF INTERTIA,PLATE NO. I4)                     824
 9600 FORMAT(31H1NEGATIVE THICKNESS, PLATE NO. I4)                             825
C                                                                              826
      RETURN                                                                   827
      END
```

SUBROUTINE MUL1

```
$IBFTC MUL1*   DECK
       SUBROUTINE MUL1(A,B,C,NT,N1,N2,N3,ID1,ID2,ID3)                    831
C                                                                        832
C      NT=1     C=A*B                                                    833
C      LIMITS OF MULTIPLY  A(N1,N2)  B(N2,N3)  C(N1,N3)                  834
C                                                                        835
       DIMENSION A(ID1,1),B(ID2,1),C(ID3,1)                              836
C                                                                        837
       DO 100 I=1,N1                                                     838
       DO 100 J=1,N3                                                     839
       C(I,J)=0.0                                                        840
       DO 100 K=1,N2                                                     841
  100  C(I,J)=C(I,J)+A(I,K)*B(K,J)                                       842
  400  RETURN                                                            843
       END                                                               844
```

SUBROUTINE MUL2

```
$IBFTC MUL2*    DECK
       SUBROUTINE MUL2(A,B,C,NT,N1,N2,N3,ID1,ID2,ID3)                    847
                                                                         848
C                                                                        849
C      NT=2     C=A(TRANSPOSE)*B                                         850
C      LIMITS OF MULTIPLY   A(N1,N2)   B(N2,N3)   C(N1,N3)              851
C                                                                        852
       DIMENSION A(ID1,1),B(ID2,1),C(ID3,1)                              853
C                                                                        854
       DO 300 I=1,N1                                                     855
       DO 300 J=1,N3                                                     856
       C(I,J)=0.0                                                        857
       DO 300 K=1,N2                                                     858
  300  C(I,J)=C(I,J)+A(K,I)*B(K,J)                                       859
C                                                                        860
  400  RETURN                                                            861
       END
```

SUBROUTINE PSTIF

```
$IBFTC PSTIF#  DECK
       SUBROUTINE PSTIF(LINE)
       COMMON/RENT/NRENT,KRENT                                          865
       COMMON/TPLN1/NERR                                                882
       COMMON/PSTIFL/ISWAP                                              886
       COMMON/PSTIF9/NP,N4
C                                                                       888
C                                                                       890
C FORM LOCAL COORDINATES FROM STRUCTURAL COORDINATES                    891
C FOR TRIANGULAR OR QUADRILATERAL PLATE                                 892
C                                                                       893
       CALL LOCAL                                                       894
       IF(KRENT.NE.0)GO TO 2                                            895
       IF(ISWAP.NE.0)GO TO 150                                          896
    2  CONTINUE                                                         897
C                                                                       905
C FOR ANALYSIS OF TRIANGULAR PLATE, N4=0                                906
C                                                                       907
       IF(N4.NE.0)GO TO 130                                             908
   10  CALL TRI                                                         909
       GO TO 150                                                        946
C                                                                       947
C FOR ANALYSIS OF QUADRILATERAL PLATE, N4 DOES NOT EQUAL 0              948
C                                                                       949
C CHECK COPLANARITY OF QUADRILATERAL PLATE                              950
C                                                                       951
  130  CALL COPLAN(NERR)                                                952
       CALL QUAD(LINE)
  150  RETURN                                                           954
       END                                                              957
```

SUBROUTINE QUAD

```
$IBFTC QUAD*   DECK
      SUBROUTINE QUAD(LINE)
C                                                                          961
C GENERATE STIFFNESS AND TEMPERATURE LOADS FOR QUADRILATERAL PLATE         962
C                                                                          963
      COMMON/PSTIF2/H,TS                                                   964
      COMMON/ADPRO/E,GAMMA,DUM1,DUM2,DUM3,DUM4                             965
      COMMON/PSTIF3/X1,Y1,X2,Y2,X3,Y3                                      966
      COMMON/PSTIF5/TLOAD(18),DEFL(18)                                     967
      COMMON/PSTIF8/SKQ(30,30)                                             968
      COMMON/PSTIFH/XL1,YL1,XL2,YL2,XL3,YL3,XL4,YL4                        969
      COMMON/PSTIFK/NQUA                                                   970
      COMMON/PSTIFL/ISWAP                                                  971
      COMMON/TEMP2/A(9,40)                                                 972
      COMMON/TPLN1/NERR                                                    973
      COMMON/RENT/NRENT                                                    977
      COMMON/SWAP/KSWAP                                                    978
      COMMON/LOAD/LDPT                                                     979
      COMMON/PSTRSS/SKQSS(30,30)
      COMMON/BUCK/IBUC
      EQUIVALENCE(XYL(1),XL1),(XY(1),X2)                                   980
      DIMENSION XYL(8),XY(4)                                               982
C                                                                          983
C FIND FIFTH NODE                                                          984
C                                                                          985
      XL23=XL2-XL3                                                         986
      P34=XL4*YL3+YL4*XL23                                                 987
      X1=(XL4+XL3+(XL2*YL4*XL23/P34))/3.0                                  988
      Y1=(YL4+YL3-(XL2*YL4*YL3/P34))/3.0                                   989
C                                                                          990
C ZERO THE ELEMENTS OF SKQ(30,30)                                          991
C                                                                          992
      DO 10 J=1,30                                                         993
      DO 10 I=1,30                                                         994
   10 SKQ(I,J)=0.0                                                         995
C                                                                          996
C SUBDIVIDED TRIANGULAR PLATES OF THE QUADRILATERAL PLATE                 1038
C                                                                         1039
   19 DO 240 NQUA=1,4                                                     1040
      DO 90 I=1,4                                                         1041
      GO TO(50,20,30,40),NQUA                                             1042
   20 GO TO(60,60,70,70),I                                                1043
   30 GO TO(80,80,60,60),I                                                1044
   40 GO TO(70,70,81,81),I                                                1045
   50 I1=I                                                                1046
      GO TO 85                                                            1047
   60 I1=I+2                                                              1048
      GO TO 85                                                            1049
   70 I1=I+4                                                              1050
      GO TO 85                                                            1051
   80 I1=I+6                                                              1052
      GO TO 85                                                            1053
   81 I1=I-2                                                              1054
   85 XY(I)=XYL(I1)                                                       1055
   90 CONTINUE                                                            1056
      CALL TRI                                                            1057
  240 CONTINUE                                                            1133
C                                                                         1134
C REDUCE OUT FIFTH NODE FROM STIFFNESS MATRIX SKQ(30,30)                  1135
C                                                                         1137
      DO 246 K=25,30
```

```
  246 CALL REDUCE(SKQSS,30,K)
      IF(IBUC.EQ.0)GO TO 280
      DO 245 K=25,30
  245 CALL REDUCE(SKQ,30,K)                                          1138
C
  280 RETURN                                                         1140
      END                                                            1141
                                                                     1144
```

SUBROUTINE LAMK

```
$IBFTC LAMK*   DECK
       SUBROUTINE LAMK(SK,PT)
       DIMENSION SK(3,300),PT(3,3),P(3)
       DO 50 I=1,300
       DO 25 IR=1,3
       P(IR)=0.
       DO 25 IC=1,3
       P(IR)=P(IR)+PT(IR,IC)*SK(IC,I)
    25 CONTINUE
       DO 50 IC=1,3
       SK(IC,I)=P(IC)
    50 CONTINUE
       RETURN
       END
```

SUBROUTINE KLAMT

```
$IBFTC KLAMT*  DECK
       SUBROUTINE KLAMT(SK,PT)
       DIMENSION SK(30,3,10),PT(3,3),P(3)
       DO 50 I=1,10
       DO 50 J=1,30
       DO 25 IR=1,3
       P(IR)=0.0
       DO 25 IC=1,3
       P(IR)=P(IR)+PT(IR,IC)*SK(J,IC,I)
    25 CONTINUE
       DO 50 IC=1,3
       SK(J,IC,I)=P(IC)
    50 CONTINUE
       RETURN
       END
```

SUBROUTINE TRI

```
$IBFTC TRI*    DECK
      SUBROUTINE TRI                                                    1147
C                                                                       1148
C TRIANGULAR PLATE STIFFNESS GENERATION SK(18,18)                       1149
C                                                                       1150
C                                                                       1151
      COMMON/PSTIF1/SK(18,18)                                           1152
      COMMON/PSTIF3/X1,Y1,X2,Y2,X3,Y3                                   1153
      COMMON/PSTIF7/X21,X31,X32,Y21,Y31,Y32                             1154
      COMMON/PSTIF9/NP,N4                                               1155
      COMMON/PSTIFH/XL1,YL1,XL2,YL2,XL3,YL3,XL4,YL4                     1156
      COMMON/PSTIFD/TX,TY,TO
      COMMON/BUCK/IBUC
      COMMON/THRST/TRS1,TRS2,TRS3,TRS4,TRS5,TRS6,TRS7
      COMMON/PREST/SKPRE(6)
C                                                                       1157
C                                                                       1158
C ************* DEFINITION OF ARGUMENTS *************                 1159
C                                                                       1160
C X1,Y1 - LOCAL COORDINATES AT NODE 1 OF TRIANGULAR PLATE               1161
C X2,Y2 - LOCAL COORDINATES AT NODE 2 OF TRIANGULAR PLATE               1162
C X3,Y3 - LOCAL COORDINATES AT NODE 3 OF TRIANGULAR PLATE               1163
C                                                                       1164
C                                                                       1165
C                                                                       1166
      EQUIVALENCE(XLOCAL,XL1)                                           1167
      DIMENSION XLOCAL(8)                                               1168
      T = TO
C                                                                       1169
      IF(N4.NE.0)GO TO 5                                                1170
    1 X1=XL1                                                            1171
      Y1=YL1                                                            1172
      X2=XL2                                                            1173
      Y2=YL2                                                            1174
      X3=XL3                                                            1175
      Y3=YL3                                                            1176
C                                                                       1177
C THE FOLLOWING PROJECTIONS, (LENGTHS), ARE USED IN ROUTINES            1178
C INPST,OUTSH,INPM AND PTEMP                                            1179
C                                                                       1180
    5 X21 = X2-X1                                                       1181
      X31 = X3-X1                                                       1182
      X32 = X3-X2                                                       1183
      Y21 = Y2-Y1                                                       1184
      Y31 = Y3-Y1                                                       1185
      Y32 = Y3-Y2                                                       1186
C                                                                       1187
C THE FOLLOWING ZERO'S THE SK ARRAY                                     1188
C                                                                       1189
      DO 10 J=1,18                                                      1190
      DO 10 I=1,18                                                      1191
   10 SK(I,J)=0.0                                                       1192


C GENERATE K1 FOR TRIANGULAR ELEMENT

      IF(IBUC.EQ.0)GO TO 20
      IF(N4.EQ.0) GO TO 40
      XT2=SQRT(X21**2 + Y21**2)
      XT3=(X21*X31 + Y21*Y31)/XT2
      YT3=SQRT(X31**2 + Y31**2 - XT3**2)
```

```
      XT32 = XT3 - XT2
      CES = X21/XT2
      SEN = Y21/XT2
      T4A=T/(2.*XT2*YT3)
      TS1T=T4A*TRS1
      TS2T=T4A*TRS2
      TS3T=T4A*TRS3
      TS1=((CES**2)*TS1T + (SEN**2)*TS2T + 2.*(SEN*CES)*TS3T)
      TS2=((SEN**2)*TS1T + (CES**2)*TS2T - 2.*(SEN*CES)*TS3T)
      TS3=((SEN*CES)*(TS2T-TS1T) + (CES**2-SEN**2)*TS3T)
      GO TO 19
   40 XT2=X2
      XT3=X3
      YT3=Y3
      XT32=X32
      T4A=T/(2.*XT2*YT3)
      TS1=T4A*TRS1
      TS2=T4A*TRS2
      TS3=T4A*TRS3
   19 SKPRE(1)=TS1*YT3*YT3+TS2*XT32*XT32-TS3*2.*XT32*YT3
      SKPRE(2)=-TS1*YT3*YT3-TS2*XT32*XT3+TS3*YT3*(XT3+XT32)
      SKPRE(3) =TS1*YT3*YT3+TS2*XT3*XT3-TS3*2.*XT3*YT3
      SKPRE(4) =TS2*XT32*XT2-TS3*XT2*YT3
      SKPRE(5) =-TS2*XT3*XT2+TS3*XT2*YT3
      SKPRE(6) =TS2*XT2*XT2
   20 CONTINUE
C                                                                       1193
C GENERATE STIFFNESS FOR IN-PLANE EFFECTS (INP)                         1194
C                                                                       1195
      CALL INP                                                          1196
      CALL STORE(2)                                                     1197
C                                                                       1198
C                                                                       1199
C                                                                       1200
C GENERATE STIFFNESS FOR OUT OF PLANE EFFECTS (OUTP)                    1201
C                                                                       1202
      CALL OUTP
      CALL STORE(0)
C                                                                       1204
                                                                        1205
  100 RETURN                                                            1206
      END                                                               1207
```

SUBROUTINE INP

```
$IBFTC INP*     DECK
       SUBROUTINE INP                                                    1210
       COMMON/PSTIFA/SK66(6,6)/PSTIFB/SK99(9,9)                          1211
C                                                                        1212
C ***************************************************************      1213
C GENERATE STIFFNESS FOR IN-PLANE EFFECTS (INP) FOR TRIANGULAR PLATE     1214
C ***************************************************************      1215
C                                                                        1216
C                                                                        1217
C GENERATE STIFFNESS FOR IN-PLANE STRETCHING (INPS)                      1218
C                                                                        1219
       CALL INPST(1)                                                     1220
C                                                                        1221
C GENERATE STIFFNESS FOR IN-PLANE MOMENTS (INPM)                         1222
C                                                                        1223
       CALL INPM                                                         1224
C                                                                        1225
C COMBINE (INPST) AND (INPM) STIFFNESS INTO (INP) STIFFNESS              1226
C                                                                        1227
       CALL COMBIN(1)                                                    1228
       RETURN                                                            1229
       END                                                               1230
```

SUBROUTINE INPM

```
$IBFTC INPM*   DECK
      SUBROUTINE INPM                                                    1233
C                                                                        1234
C GENERATE THE STIFFNESS CONTRIBUTION DUE TO                             1235
C IN-PLANE MOMENTS IN THE ELEMENTS OF UPPER TRIANGLE OF SK99             1236
C                                                                        1237
      COMMON/PSTIF2/H,TS                                                 1238
      COMMON/ADPRO/E,GAMMA,DUM1,DUM2,DUM3,DUM4                           1239
      COMMON/PSTIF7/X21,X31,X32,Y21,Y31,Y32                              1240
      COMMON/PSTIF9/NP,N4                                                1241
      COMMON/PSTIFB/SK99(9,9)                                            1242
      COMMON/PSTIFD/TX,TY,TO                                             1243
C                                                                        1244
      DIMENSION XY(3,2),D(3),SC(3,2),SC2(3,2),E2AI(3),                   1245
     * AD(3),ID(3),ID2(3),ID3(3),ID2SC(3,2),ID3SC2(3,2),ADSC2(3,2),AI(3) 1246
C                                                                        1247
      EQUIVALENCE (X21,XY(1,1))                                          1248
      REAL ID,ID2,ID3,ID2SC,ID3SC2                                       1249
      TXY(A)=0.5*((TX+TO)*(1.0+(A))+(TX+TO)*(1.0-(A)))*A2                1250
C                                                                        1251
      A2=(X21*Y31-Y21*X31)/6.0                                           1252
      TO1=(TO/12.0)*A2**3                                                1253
C                                                                        1254
      E2=E                                                               1255
      E4=2.*E                                                            1256
      E8=4.*E                                                            1257
      E12=6.*E                                                           1258
      E24=12.*E                                                          1259
C                                                                        1260
C                                                                        1261
C D(1) IS D12, ETC.                                                      1262
C                                                                        1263
      DO 10 I=1,3                                                        1264
   10 D(I)=SQRT(XY(I,1)**2+XY(I,2)**2)                                   1265
C                                                                        1266
C SC(1,1) IS SIN12,SC(1,2) IS COS12, ETC.                                1267
C                                                                        1268
      DO 20 J=1,2                                                        1269
      DO 20 I=1,3                                                        1270
   20 SC(I,J)=XY(I,J)/D(I)                                               1271
      SC(2,1)=-SC(2,1)                                                   1272
      SC(2,2)=-SC(2,2)                                                   1273
C                                                                        1274
C SC2(1,1) IS SIN12**2, ETC.                                             1275
C                                                                        1276
      DO 30 J=1,2                                                        1277
      DO 30 I=1,3                                                        1278
   30 SC2(I,J)=SC(I,J)**2                                                1279
C                                                                        1280
C AD(1) IS A12/D12, ETC.                                                 1281
C                                                                        1282
      DO 50 I=1,3                                                        1283
   50 AD(I)=0.                                                           1284
C                                                                        1285
C ID(1) IS I12/D12, ETC.                                                 1286
C                                                                        1287
      DO 60 I=1,3                                                        1288
```

```
   60 ID(I)=TO1/(D(I)**4)                                                1289
C                                                                        1290
C  ID2(1) IS I12/(D12**2), ETC.                                          1291
C                                                                        1292
      DO 70 I=1,3                                                        1293
   70 ID2(I)=ID(I)/D(I)                                                  1294
C                                                                        1295
C  ID3(1) IS I12/(D12**2), ETC.                                          1296
C                                                                        1297
      DO 80 I=1,3                                                        1298
   80 ID3(I)=ID2(I)/D(I)                                                 1299
C                                                                        1300
C  ID2SC(1,1) IS I12*SIN12/(D12**2), ETC.                                1301
C                                                                        1302
      DO 90 J=1,2                                                        1303
      DO 90 I=1,3                                                        1304
   90 ID2SC(I,J)=ID2(I)*SC(I,J)                                          1305
C                                                                        1306
C  ID3SC2(1,1) IS I12*SIN12**2/(D12**3), ETC.                            1307
C                                                                        1308
      DO 100 J=1,2                                                       1309
      DO 100 I=1,3                                                       1310
  100 ID3SC2(I,J)=ID3(I)*SC2(I,J)                                        1311
C                                                                        1312
C  ADSC2(1,1) IS A12*SIN12**2/D12, ETC.                                  1313
C                                                                        1314
      DO 110 J=1,2                                                       1315
      DO 110 I=1,3                                                       1316
  110 ADSC2(I,J)=AD(I)*SC2(I,J)                                          1317
C                                                                        1318
C  E2AI(1) IS 2*E*(SIN12*COS12*(A12/D12-I12/(D12**3))), ETC.             1319
C                                                                        1320
      DO 120 I=1,3                                                       1321
  120 E2AI(I)=SC(I,1)*SC(I,2)*(AD(I)-12.0*ID3(I))*E2                     1322
C                                                                        1323
C  ROW1                                                                  1324
C                                                                        1325
      SK99(1,1) =   E8*(ID(1) + ID(2))                                   1326
      SK99(1,2) = - E12*(ID2SC(1,2) - ID2SC(2,2))                        1327
      SK99(1,3) =   E12*(ID2SC(1,1) - ID2SC(2,1))                        1328
      SK99(1,4) =   E4*ID(1)                                             1329
      SK99(1,5) =   E12*ID2SC(1,2)                                       1330
      SK99(1,6) = - E12*ID2SC(1,1)                                       1331
      SK99(1,7) =   E4*ID(2)                                             1332
      SK99(1,8) = - E12*ID2SC(2,2)                                       1333
      SK99(1,9) =   E12*ID2SC(2,1)                                       1334
C                                                                        1335
C  ROW2                                                                  1336
C                                                                        1337
      SK99(2,2) =   E2*(ADSC2(1,1) + ADSC2(2,1)) +                       1338
     *              E24*(ID3SC2(1,2) + ID3SC2(2,2))                      1339
      SK99(2,3) =   E2AI(1) + E2AI(2)                                    1340
      SK99(2,4) = - SK99(1,5)                                            1341
      SK99(2,5) = - E2*(ADSC2(1,1) + 12.0*ID3SC2(1,2))                   1342
      SK99(2,6) = - E2AI(1)                                              1343
      SK99(2,7) = - SK99(1,8)                                            1344
      SK99(2,8) = - E2*(ADSC2(2,1) + 12.0*ID3SC2(2,2))                   1345
      SK99(2,9) = - E2AI(2)                                              1346
C                                                                        1347
C  ROW3                                                                  1348
```

```
C
      SK99(3,3) =    E2*(ADSC2(1,2) + ADSC2(2,2)) +
     *               E24*(ID3SC2(1,1) + ID3SC2(2,1))
      SK99(3,4) = - SK99(1,6)
      SK99(3,5) = - E2AI(1)
      SK99(3,6) = - E2*(ADSC2(1,2) + 12.0*ID3SC2(1,1))
      SK99(3,7) = - SK99(1,9)
      SK99(3,8) = - E2AI(2)
      SK99(3,9) = - E2*(ADSC2(2,2) + 12.0*ID3SC2(2,1))
C
C ROW4
C
      SK99(4,4) =    E8*(ID(1) + ID(3))
      SK99(4,5) = - E12*(ID2SC(3,2) - ID2SC(1,2))
      SK99(4,6) =    E12*(ID2SC(3,1) - ID2SC(1,1))
      SK99(4,7) =    E4*ID(3)
      SK99(4,8) =    E12*ID2SC(3,2)
      SK99(4,9) = - E12*ID2SC(3,1)
C
C ROW5
C
      SK99(5,5) =    E2*(ADSC2(1,1) + ADSC2(3,1)) +
     *               E24*(ID3SC2(1,2) + ID3SC2(3,2))
      SK99(5,6) =    E2AI(1) + E2AI(3)
      SK99(5,7) = - SK99(4,8)
      SK99(5,8) = - E2*(ADSC2(3,1) + 12.0*ID3SC2(3,2))
      SK99(5,9) = - E2AI(3)
C
C ROW6
C
      SK99(6,6) =    E2*(ADSC2(1,2) + ADSC2(3,2)) +
     *               E24*(ID3SC2(1,1) + ID3SC2(3,1))
      SK99(6,7) = - SK99(4,9)
      SK99(6,8) = - E2AI(3)
      SK99(6,9) = - E2*(ADSC2(3,2) + 12.0*ID3SC2(3,1))
C
C ROW7
C
      SK99(7,7) =    E8*(ID(2) + ID(3))
      SK99(7,8) = - E12*(ID2SC(2,2) - ID2SC(3,2))
      SK99(7,9) =  -E12*(ID2SC(3,1) - ID2SC(2,1))
C
C ROW8
C
      SK99(8,8) =    E2*(ADSC2(3,1) + ADSC2(2,1)) +
     *               E24*(ID3SC2(3,2) + ID3SC2(2,2))
      SK99(8,9) =    E2AI(2) + E2AI(3)
C
C ROW9
C
      SK99(9,9) = + E2*(ADSC2(2,2) + ADSC2(3,2)) +
     *               E24*(ID3SC2(2,1) + ID3SC2(3,1))
C
  200 RETURN
      END
```

SUBROUTINE INPST

```
$IBFTC INPST*  DECK                                                    1406
      SUBROUTINE INPST(N)                                              1407
C                                                                      1408
C GENERATE STIFFNESS FOR IN-PLANE STRETCHING IN UPPER TRIANGLE         1409
C OF SK66(I,J) ELEMENTS.                                               1410
C                                                                      1411
C  N=1, GENERATE (INPST)                                               1412
C  N=0, GENERATE (OUTPM)                                               1413
C                                                                      1414
      COMMON/PSTIF2/H,TS                                               1415
      COMMON/ADPRO/E,GAMMA,DUM1,DUM2,DUM3,DUM4                         1416
      COMMON/PSTIF7/X21,X31,X32,Y21,Y31,Y32                            1417
      COMMON/PSTIF9/NP,N4                                              1418
      COMMON/PSTIFA/SK66(6,6)                                          1419
      COMMON/PSTIFC/TIBX,TIBY,TIBS                                     1420
      COMMON/PSTIFD/TX,TY,TO                                           1421
C                                                                      1422
      PHI = E/(ABS(X21*Y31-Y21*X31)*(1.0-GAMMA**2)*2.0)                1423
      IF(N.EQ.0) GO TO 10                                              1424
      T1 = (0.5*TO*(1.0-GAMMA))*PHI                                    1425
      T2 = (0.5*TO*(1.0+GAMMA))*PHI                                    1426
      GAT = GAMMA*TO*PHI                                               1427
      TPX = (TO+TX*(1.0-GAMMA**2))*PHI                                 1428
      TPY = (TO+TY*(1.0-GAMMA**2))*PHI                                 1429
      GO TO 20                                                         1430
   10 TOX=TIBS                                                         1431
      T1 = (0.5*TOX*(1.0-GAMMA))*PHI                                   1432
      T2 = (0.5*TOX*(1.0+GAMMA))*PHI                                   1433
      GAT = GAMMA*TOX*PHI                                              1434
      TPX = (TOX+TIBX*(1.0-GAMMA**2))*PHI                              1435
      TPY = (TOX+TIBY*(1.0-GAMMA**2))*PHI                              1436
C                                                                      1437
C ROW 1                                                                1438
C                                                                      1439
   20 SK66(1,1) =   TPX*Y32**2  + T1*X32**2                            1440
      SK66(1,2) =               - T2*X32*Y32                           1441
      SK66(1,3) = - TPX*Y31*Y32 - T1*X31*X32                           1442
      SK66(1,4) =   GAT*X31*Y32 + T1*Y31*X32                           1443
      SK66(1,5) =   TPX*Y21*Y32 + T1*X21*X32                           1444
      SK66(1,6) = - GAT*X21*Y32 - T1*Y21*X32                           1445
C                                                                      1446
C ROW 2                                                                1447
C                                                                      1448
      SK66(2,2) =   TPY*X32**2  + T1*Y32**2                            1449
      SK66(2,3) =   GAT*Y31*X32 + T1*X31*Y32                           1450
      SK66(2,4) = - TPY*X31*X32 - T1*Y31*Y32                           1451
      SK66(2,5) = - GAT*Y21*X32 - T1*X21*Y32                           1452
      SK66(2,6) =   TPY*X21*X32 + T1*Y21*Y32                           1453
C                                                                      1454
C ROW 3                                                                1455
C                                                                      1456
      SK66(3,3) =   TPX*Y31**2  + T1*X31**2                            1457
      SK66(3,4) =               - T2*X31*Y31                           1458
      SK66(3,5) = - TPX*Y21*Y31 - T1*X21*X31                           1459
      SK66(3,6) =   GAT*X21*Y31 + T1*Y21*X31                           1460
C                                                                      1461
C ROW 4
```

```
C                                                                       1462
      SK66(4,4) =    TPY*X31**2  + T1*Y31**2                            1463
      SK66(4,5) =    GAT*X31*Y21 + T1*X21*Y31                           1464
      SK66(4,6) = - TPY*X21*X31 - T1*Y21*Y31                            1465
C                                                                       1466
C ROW 5                                                                 1467
C                                                                       1468
      SK66(5,5) =    TPX*Y21**2  + T1*X21**2                            1469
      SK66(5,6) =                 -T2*X21*Y21                           1470
C                                                                       1471
C ROW 6                                                                 1472
C                                                                       1473
      SK66(6,6) =    TPY*X21**2  + T1*Y21**2                            1474
C                                                                       1475
  200 RETURN                                                            1476
      END                                                               1477
```

SUBROUTINE OUTP

```
$IBFTC OUTP*   DECK                                                              1480
       SUBROUTINE OUTP                                                           1481
       COMMON/PSTIFA/SK66(6,6)/PSTIFB/SK99(9,9)                                  1482
C                                                                                1483
C ***************************************************************              1484
C GENERATE STIFFNESS FOR OUT OF PLANE EFFECTS (OUTP) FOR TRIANGULAR PLAT         1485
C ***************************************************************              1486
C                                                                                1487
C                                                                                1488
C GENERATE STIFFNESS FOR OUT OF PLANE MOMENTS (OUTPM)                            1489
C                                                                                1490
       CALL OUTPM                                                                1491
C                                                                                1492
C GENERATE STIFFNESS FOR OUT OF PLANE SHEAR (OUTPSH)                             1493
C                                                                                1494
       CALL OUTPSH                                                               1495
C                                                                                1496
C COMBINE (OUTPM) AND (OUTPSH) STIFFNESS INTO (OUTP) STIFFNESS                   1497
C                                                                                1498
       CALL COMBIN (0)                                                           1499
       RETURN                                                                    1500
       END
```

SUBROUTINE OUTPM

```
$IBFTC OUTPM*  DECK
      SUBROUTINE OUTPM                                                    1503
C                                                                         1504
C GENERATE STIFFNESS FOR OUT OF PLANE MOMENTS BY TRANSFORMATION           1505
C OF IN-PLANE STRETCHING STIFFNESS MATRIX, (SEE SUBROUTINE (INPST)),      1506
C OTHER CHANGES AS INDICATED BELOW                                        1507
C                                                                         1508
      COMMON/PSTIFA/SK66(6,6)                                             1509
      COMMON/PSTIFC/TIBX,TIBY,TIBS                                        1510
      COMMON/PSTIFD/TX,TY,TO                                              1511
C                                                                         1512
C                                                                         1513
      CALL INPST(0)                                                       1514
      I=1                                                                 1515
   10 SAVE=SK66(I,I)                                                      1516
      SK66(I,I)=SK66(I+1,I+1)                                             1517
      SK66(I+1,I+1)=SAVE                                                  1518
      SK66(I,I+1)=-SK66(I,I+1)                                            1519
      I=I+2                                                               1520
      IF(I.LE.5) GO TO 10                                                 1521
      J=3                                                                 1522
   20 I=1                                                                 1523
   30 SAVE=SK66(I,J)                                                      1524
      SK66(I,J)=SK66(I+1,J+1)                                             1525
      SK66(I+1,J+1)=SAVE                                                  1526
      SAVE=-SK66(I+1,J)                                                   1527
      SK66(I+1,J)=-SK66(I,J+1)                                            1528
      SK66(I,J+1)=SAVE                                                    1529
      I=I+2                                                               1530
      IF(I.LT.J)GO TO 30                                                  1531
      J=J+2                                                               1532
      IF(J.LE.5)GO TO 20                                                  1533
      RETURN                                                              1534
      END                                                                 1535
```

SUBROUTINE OUTPSH

```
$IBFTC OUTPS*  DECK                                                    1538
      SUBROUTINE OUTPSH                                                1539
C                                                                      1540
C GENERATE THE STIFFNESS CONTRIBUTION DUE TO OUT OF PLANE              1541
C SHEAR (OUTPSH) IN THE ELEMENTS OF SKQ(I,J,22),SKQ(I,J,23),           1542
C                                                                      1543
      COMMON/PSTIF2/H,TS                                               1544
      COMMON/ADPRO/E,GAMMA,DUM1,DUM2,DUM3,DUM4                         1545
      COMMON/PSTIF7/X21,X31,X32,Y21,Y31,Y32                            1546
      COMMON/PSTIFB/SK99(9,9)                                          1547
C                                                                      1548
C                                                                      1549
      G = E/(2.0*(1.0 + GAMMA))                                        1550
      ALP = (G*TS)/(8.0*ABS(Y21*X31-Y31*X21))                          1551
      ALP12 = ALP*ABS(Y31*Y32+X31*X32)                                 1552
      ALP13 = ALP*ABS(Y21*Y32+X21*X32)                                 1553
      ALP23 = ALP*ABS(Y21*Y31+X21*X31)                                 1554
C                                                                      1555
C ROW1                                                                 1556
C                                                                      1557
      SK99(1,1) =   ALP12*Y21**2 + ALP13*Y31**2                        1558
      SK99(1,2) = - (ALP12*X21*Y21 + ALP13*X31*Y31)                    1559
      SK99(1,3) =   2.0*(ALP12*Y21 + ALP13*Y31)                        1560
      SK99(1,4) =   ALP12*Y21**2                                       1561
      SK99(1,5) = - ALP12*X21*Y21                                      1562
      SK99(1,6) = - 2.0*ALP12*Y21                                      1563
      SK99(1,7) =   ALP13*Y31**2                                       1564
      SK99(1,8) = - ALP13*X31*Y31                                      1565
      SK99(1,9) = - 2.0*ALP13*Y31                                      1566
C                                                                      1567
C ROW2                                                                 1568
C                                                                      1569
      SK99(2,2) =   ALP12*X21**2 + ALP13*X31**2                        1570
      SK99(2,3) = - 2.0*(ALP12*X21 + ALP13*X31)                        1571
      SK99(2,4) = - ALP12*X21*Y21                                      1572
      SK99(2,5) =   ALP12*X21**2                                       1573
      SK99(2,6) =   2.0*ALP12*X21                                      1574
      SK99(2,7) = - ALP13*X31*Y31                                      1575
      SK99(2,8) =   ALP13*X31**2                                       1576
      SK99(2,9) =   2.0*ALP13*X31                                      1577
C                                                                      1578
C ROW3                                                                 1579
C                                                                      1580
      SK99(3,3) =   4.0*(ALP12+ALP13)                                  1581
      SK99(3,4) =   2.0*ALP12*Y21                                      1582
      SK99(3,5) = - 2.0*ALP12*X21                                      1583
      SK99(3,6) = - 4.0*ALP12                                          1584
      SK99(3,7) =   2.0*ALP13*Y31                                      1585
      SK99(3,8) = - 2.0*ALP13*X31                                      1586
      SK99(3,9) = - 4.0*ALP13                                          1587
C                                                                      1588
C ROW4                                                                 1589
C                                                                      1590
      SK99(4,4) =   ALP12*Y21**2 + ALP23*Y32**2                        1591
      SK99(4,5) = - (ALP12*X21*Y21 + ALP23*X32*Y32)                    1592
      SK99(4,6) = - 2.0*(ALP12*Y21-ALP23*Y32)                          1593
      SK99(4,7) =   ALP23*Y32**2
```

```
      SK99(4,8) = - ALP23*X32*Y32                                       1594
      SK99(4,9) = - 2.0*ALP23*Y32                                       1595
C                                                                       1596
C ROW5                                                                  1597
C                                                                       1598
      SK99(5,5) =   ALP12*X21**2+ALP23*X32**2                           1599
      SK99(5,6) =   2.0*(ALP12*X21-ALP23*X32)                           1600
      SK99(5,7) = - ALP23*X32*Y32                                       1601
      SK99(5,8) =   ALP23*X32**2                                        1602
      SK99(5,9) =   2.0*ALP23*X32                                       1603
C                                                                       1604
C ROW6                                                                  1605
C                                                                       1606
      SK99(6,6) =   4.0*(ALP12 + ALP23)                                 1607
      SK99(6,7) =   2.0*ALP23*Y32                                       1608
      SK99(6,8) = - 2.0*ALP23*X32                                       1609
      SK99(6,9) = - 4.0*ALP23                                           1610
C                                                                       1611
C ROW7                                                                  1612
C                                                                       1613
      SK99(7,7) =  ALP13*Y31**2+ALP23*Y32**2                            1614
      SK99(7,8) = -(ALP13*X31*Y31 + ALP23*X32*Y32)                      1615
      SK99(7,9) = -2.0*(ALP13*Y31 + ALP23*Y32)                          1616
C                                                                       1617
C ROW8                                                                  1618
C                                                                       1619
      SK99(8,8) =   ALP13*X31**2 + ALP23*X32**2                         1620
      SK99(8,9) =   2.0*(ALP13*X31 + ALP23*X32)                         1621
C                                                                       1622
C ROW9                                                                  1623
C                                                                       1624
      SK99(9,9) =  4.0*(ALP13 + ALP23)                                  1625
C                                                                       1626
      RETURN                                                            1627
      END                                                               1628
```

SUBROUTINE COMBIN

```
$IBFTC COMBI*  DECK
      SUBROUTINE COMBIN(N)                                                  1631
C                                                                           1632
C  N=1, WILL COMBINE IN-PLANE STRETCHING STIFFNESS (INPST) AND              1633
C       IN-PLANE MOMENT STIFFNESS (INPM) IN UPPER TRIANGLE OF SK99          1634
C  N=0, WILL COMBINE OUT OF PLANE MOMENT STIFFNESS (OUTPM) AND              1635
C       OUT OF PLANE SHEAR STIFFNESS                                        1636
C                                                                           1637
      COMMON/PSTIFA/SK66(6,6)                                               1638
      COMMON/PSTIFB/SK99(9,9)                                               1639
C                                                                           1640
C                                                                           1641
      DO 40 I=1,6                                                           1642
      I1=I+N                                                                1643
      GO TO (20,20,10,10,25,25),I                                           1644
   10 I1=I+N+1                                                              1645
      GO TO 20                                                              1646
   25 I1=I+N+2                                                              1647
   20 DO 40 J=I,6                                                           1648
      J1=J+N                                                                1649
      GO TO (40,40,30,30,35,35),J                                           1650
   30 J1=J+N+1                                                              1651
      GO TO 40                                                              1652
   35 J1=J+N+2                                                              1653
   40 SK99(I1,J1) = SK99(I1,J1) + SK66(I,J)                                 1654
C                                                                           1655
C PLACE UPPER TRIANGLE OF SK99 IN LOWER TRIANGLE OF SK99                    1656
C                                                                           1657
      DO 50 I=1,8                                                           1658
      K=I+1                                                                 1659
      DO 50 J=K,9                                                           1660
   50 SK99(J,I)=SK99(I,J)                                                   1661
      RETURN                                                                1662
      END                                                                   1663
```

SUBROUTINE STORE

```
$IBFTC STORE*  DECK
      SUBROUTINE STORE(NTEST)                                          1666
C TRANSFER ELEMENTS OF (INP) OR (OUTP) STIFFNESS IN THERE              1668
C INTERMEDIATE MATRIX TO THEIR PROPER ELEMENTS IN SK(18,18)            1669
C FOR THE QUADRILATERAL PLATE ANALYSIS, TRANSFER ELEMENTS              1670
C OF SK(18,18) TO PROPER ELEMENTS OF SKQ(30,30).                       1671
      COMMON/PSTIF1/SK(18,18)                                          1673
      COMMON/PSTIF8/SKQ(30,30)                                         1674
      COMMON/PSTIF9/NP,N4                                              1675
      COMMON/PSTIFB/SK99(9,9)                                          1676
      COMMON/PSTIFK/NQUA                                               1677
      COMMON/PSTRSS/SKQSS(30,30)
      COMMON/BUCK/IBUC
      COMMON/PREST/SKPRE(6)
C NTEST=2, FOR (INP)                                                   1680
C NTEST=0, FOR (OUTP)                                                  1681
      DO 70 I=1,9                                                      1683
      M=I/4                                                            1684
      IF(I.EQ.7)M=2                                                    1685
      I1=I+3*M+NTEST                                                   1686
      IF(NTEST)30,10,30                                                1687
   10 GO TO(30,30,20,30,30,20,30,30,20),I                              1688
   20 I1=I1+3                                                          1689
   30 DO 70 J=1,9                                                      1690
      N=J/4                                                            1691
      IF(J.EQ.7)N=2                                                    1692
      J1=J+3*N+NTEST                                                   1693
      IF(NTEST)60,40,60                                                1694
   40 GO TO(60,60,50,60,60,50,60,60,50),J                              1695
   50 J1=J1+3                                                          1696
   60 SK(I1,J1)=SK99(I,J)
   70 CONTINUE                                                         1698
      IF(NTEST.EQ.2) GO TO 200                                         1699
C TRANSFER OF ELEMENTS FOR QUADRILATERAL                               1701
      IF(N4.NE.0)GO TO 80                                              1703
      CALL MOVE(SK(1,1),SKQSS(1,1),18,18,18,30)
      GO TO 300
   80 CALL MOVE(SK(1,1),SKQSS(25,25),6,6,18,30)
      GO TO(90,110,130,150),NQUA                                       1708
C 1ST SUBDIVIDED PLATE OF QUADRILATERAL                                1710
   90 CALL MOVE(SK(7,7),SKQSS(1,1),12,12,18,30)
      CALL MOVE(SK(1,7),SKQSS(25,1),6,12,18,30)
      CALL MOVE(SK(7,1),SKQSS(1,25),12,6,18,30)
      GO TO 300
C 2ND SUBDIVIDED PLATE OF QUADRILATERAL                                1717
  110 CALL MOVE(SK( 7, 7),SKQSS( 7, 7),6,6,18,30)
      CALL MOVE(SK(13, 7),SKQSS(19, 7),6,6,18,30)
      CALL MOVE(SK( 1, 7),SKQSS(25, 7),6,6,18,30)
      CALL MOVE(SK( 7,13),SKQSS( 7,19),6,6,18,30)
      CALL MOVE(SK(13,13),SKQSS(19,19),6,6,18,30)
      CALL MOVE(SK( 1,13),SKQSS(25,19),6,6,18,30)
      CALL MOVE(SK( 7, 1),SKQSS( 7,25),6,6,18,30)
      CALL MOVE(SK(13, 1),SKQSS(19,25),6,6,18,30)
      GO TO 300
C 3RD SUBDIVIDED PLATE OF QUADRILATERAL                                1729
  130 CALL MOVE(SK(13,13),SKQSS(13,13),6,6,18,30)
      CALL MOVE(SK( 7,13),SKQSS(19,13),6,6,18,30)
      CALL MOVE(SK( 1,13),SKQSS(25,13),6,6,18,30)
      CALL MOVE(SK(13, 7),SKQSS(13,19),6,6,18,30)
      CALL MOVE(SK( 7, 7),SKQSS(19,19),6,6,18,30)
      CALL MOVE(SK( 1, 7),SKQSS(25,19),6,6,18,30)
```

```
      CALL MOVE(SK(13, 1),SKQSS(13,25),6,6,18,30)
      CALL MOVE(SK( 7, 1),SKQSS(19,25),6,6,18,30)
      GO TO 300
C 4TH SUBDIVIDED PLATE OF QUADRILATERAL                                 1741
  150 CALL MOVE(SK(13,13),SKQSS( 1, 1),6,6,18,30)
      CALL MOVE(SK( 7,13),SKQSS(13, 1),6,6,18,30)
      CALL MOVE(SK( 1,13),SKQSS(25, 1),6,6,18,30)
      CALL MOVE(SK(13, 7),SKQSS( 1,13),6,6,18,30)
      CALL MOVE(SK( 7, 7),SKQSS(13,13),6,6,18,30)
      CALL MOVE(SK( 1, 7),SKQSS(25,13),6,6,18,30)
      CALL MOVE(SK(13, 1),SKQSS( 1,25),6,6,18,30)
      CALL MOVE(SK( 7, 1),SKQSS(13,25),6,6,18,30)
  300 IF(IBUC.EQ.0)GO TO 200
      SK( 6, 6)=SK( 6, 6)+SKPRE(1)
      SK(12, 6)=SK(12, 6)+SKPRE(2)
      SK(12,12)=SK(12,12)+SKPRE(3)
      SK(18, 6)=SK(18, 6)+SKPRE(4)
      SK(18,12)=SK(18,12)+SKPRE(5)
      SK(18,18)=SK(18,18)+SKPRE(6)
      SK(6,12) = SK(12,6)
      SK(6,18) = SK(18,6)
      SK(12,18) = SK(18,12)
      IF(N4.NE.0)GO TO 310
      CALL MOVE(SK(1,1),SKQ(1,1),18,18,18,30)                           1704
      GO TO 200                                                         1705
  310 CALL MOVE(SK(1,1),SKQ(25,25),6,6,18,30)                           1706
      GO TO(320,330,340,350),NQUA
  320 CALL MOVE(SK(7,7),SKQ(1,1),12,12,18,30)                           1712
      CALL MOVE(SK(1,7),SKQ(25,1),6,12,18,30)                           1713
      CALL MOVE(SK(7,1),SKQ(1,25),12,6,18,30)                           1714
      RETURN                                                            1715
  330 CALL MOVE(SK(7,7),SKQ(7,7),6,6,18,30)                             1719
      CALL MOVE(SK(13,7),SKQ(19,7),6,6,18,30)                           1720
      CALL MOVE(SK(1,7),SKQ(25,7),6,6,18,30)                            1721
      CALL MOVE(SK(7,13),SKQ(7,19),6,6,18,30)                           1722
      CALL MOVE(SK(13,13),SKQ(19,19),6,6,18,30)                         1723
      CALL MOVE(SK(1,13),SKQ(25,19),6,6,18,30)                          1724
      CALL MOVE(SK(7,1),SKQ(7,25),6,6,18,30)                            1725
      CALL MOVE(SK(13,1),SKQ(19,25),6,6,18,30)                          1726
      RETURN                                                            1727
  340 CALL MOVE(SK(13,13),SKQ(13,13),6,6,18,30)                         1731
      CALL MOVE(SK(7,13),SKQ(19,13),6,6,18,30)                          1732
      CALL MOVE(SK(1,13),SKQ(25,13),6,6,18,30)                          1733
      CALL MOVE(SK(13,7),SKQ(13,19),6,6,18,30)                          1734
      CALL MOVE(SK(7,7),SKQ(19,19),6,6,18,30)                           1735
      CALL MOVE(SK(1,7),SKQ(25,19),6,6,18,30)                           1736
      CALL MOVE(SK(13,1),SKQ(13,25),6,6,18,30)                          1737
      CALL MOVE(SK(7,1),SKQ(19,25),6,6,18,30)                           1738
      RETURN                                                            1739
  350 CALL MOVE(SK(13,13),SKQ(1,1),6,6,18,30)                           1743
      CALL MOVE(SK(7,13),SKQ(13,1),6,6,18,30)                           1744
      CALL MOVE(SK(1,13),SKQ(25,1),6,6,18,30)                           1745
      CALL MOVE(SK(13,7),SKQ(1,13),6,6,18,30)                           1746
      CALL MOVE(SK(7,7),SKQ(13,13),6,6,18,30)                           1747
      CALL MOVE(SK(1,7),SKQ(25,13),6,6,18,30)                           1748
      CALL MOVE(SK(13,1),SKQ(1,25),6,6,18,30)                           1749
      CALL MOVE(SK(7,1),SKQ(13,25),6,6,18,30)                           1750
  200 RETURN                                                            1751
      END                                                               1752
```

SUBROUTINE MOVE

```
$IBFTC MOVE*   DECK
       SUBROUTINE MOVE(A,B,N1,N2,N3,N4)                                 1755
C                                                                       1756
C ADD ELEMENTS OF A TO B AND STORE IN B                                 1757
C                                                                       1758
       DIMENSION A(N3,1),B(N4,1)                                        1759
C                                                                       1760
       DO 10 J=1,N2                                                     1761
       DO 10 I=1,N1                                                     1762
   10  B(I,J)=B(I,J)+A(I,J)                                             1763
       RETURN                                                           1764
       END                                                              1765
```

SUBROUTINE PMTR

```
$IBFTC PMTR*   DECK                                                          1834
      SUBROUTINE PMTR                                                        1835
C                                                                            1836
C COORDINATE TRANSFORMATION MATRIX FOR TRIANGULAR PLATE AND QUADRILATERA     1837
C                                                                            1838
C                                                                            1839
      COMMON/PMTR1/PT                                                        1840
      COMMON/PSTIF9/NP,N4                                                    1841
      COMMON/PSTIFG/XS1,YS1,ZS1,XS2,YS2,ZS2,XS3,YS3,ZS3,XS4,YS4,ZS4          1842
C                                                                            1843
C                                                                            1844
C                    **** DEFINITION OF ARGUMENTS ****                       1845
C                                                                            1846
C  XS1,YS1,ZS1 - STRUCTURAL COORDINATES AT NODE 1                            1847
C              FOR TRIANGULAR AND QUADRILATERAL PLATE                        1848
C  XS2,YS2,ZS2 - STRUCTURAL COORDINATES AT NODE 2                            1849
C              FOR TRIANGULAR AND QUADRILATERAL PLATE                        1850
C  XS3,YS3,ZS3 - STRUCTURAL COORDINATES AT NODE 3                            1851
C              FOR TRIANGULAR AND QUADRILATERAL PLATE                        1852
C  XS4,YS4,ZS4 - STRUCTURAL COORDINATES AT NODE 4 FOR QUADRILATERAL PLAT     1853
C  PT - TRANSFORMATION MATRIX IN PARTITION FORM FOR                          1854
C       TRIANGULAR OR QUADRILATERAL PLATE                                    1855
C                                                                            1866
      DIMENSION PT(3,3),XX(3),YY(3),ZZ(3),D(3)                               1868
C                                                                            1869
C THE FOLLOWING PROJECTIONS (LENGTHS) ARE USED TO COMPUTE                    1870
C DISTANCES BETWEEN NODES.                                                   1871
C                                                                            1872
      XX(1) = XS2-XS1                                                        1873
      YY(1) = YS2-YS1                                                        1874
      ZZ(1) = ZS2-ZS1                                                        1875
C                                                                            1876
      IF(N4.NE.0)GO TO 1                                                     1877
      XX(2) = XS3-XS1                                                        1878
      XX(3) = XS3-XS2                                                        1879
      YY(2) = YS3-YS1                                                        1880
      YY(3) = YS3-YS2                                                        1881
      ZZ(2) = ZS3-ZS1                                                        1882
      ZZ(3) = ZS3-ZS2                                                        1883
      GO TO 4                                                                1884
    1 XA = (XS4+XS3)/2.0                                                     1885
      YA = (YS4+YS3)/2.0                                                     1886
      ZA = (ZS4+ZS3)/2.0                                                     1887
      XX(2) = XA-XS1                                                         1888
      YY(2) = YA-YS1                                                         1889
      ZZ(2) = ZA-ZS1                                                         1890
C                                                                            1891
C D(1) IS L21,ETC. WHERE L21 IS DISTANCE BETWEEN NODES 1 AND 2,ETC.          1892
C                                                                            1893
    4 DO 10 I=1,3                                                            1894
      IF(N4.EQ.0)GO TO 8                                                     1895
      IF(I.EQ.3) GO TO 20                                                    1896
    8 D(I) = SQRT(XX(I)**2 + YY(I)**2 + ZZ(I)**2)                            1897
   10 CONTINUE                                                               1898
C                                                                            1899
C FORM UPPER LEFT PARTITION OF PT. (SHOWN ABOVE)                             1900
C
```

```
C  ***************************************************                1901
C                                                                     1902
C  PT(1,1) IS L1, PT(1,2) IS L2, PT(1,3) IS L3                        1903
C  PT(2,1) IS M1, PT(2,2) IS M2, PT(2,3) IS M3                        1904
C  PT(3,1) IS N1, PT(3,2) IS N2, PT(3,3) IS N3                        1905
C                                                                     1906
C  ***************************************************                1907
C                                                                     1908
   20 PT(1,1) = XX(1)/D(1)                                            1909
      PT(2,1) = YY(1)/D(1)                                            1910
      PT(3,1) = ZZ(1)/D(1)                                            1911
C                                                                     1912
C DD IS (L31 X L32)                                                   1913
C                                                                     1914
      IF(N4.NE.0)GO TO 40                                             1915
      DD = D(2)*D(3)                                                  1916
      DL3 =(YY(2)*ZZ(3) - YY(3)*ZZ(2))/DD                             1917
      DM3 =(XX(2)*ZZ(3) - XX(3)*ZZ(2))/DD                             1918
      DN3 =(XX(2)*YY(3) - XX(3)*YY(2))/DD                             1919
      GO TO 50                                                        1920
   40 DL3 =(PT(2,1)*ZZ(2) - PT(3,1)*YY(2))/D(2)                       1921
      DM3 =(PT(1,1)*ZZ(2) - PT(3,1)*XX(2))/D(2)                       1922
      DN3 =(PT(1,1)*YY(2) - PT(2,1)*XX(2))/D(2)                       1923
   50 SQD=SQRT(DL3**2+DM3**2+DN3**2)                                  1924
      PT(1,3) =   DL3/SQD                                             1925
      PT(2,3) = - DM3/SQD                                             1926
      PT(3,3) =   DN3/SQD                                             1927
C                                                                     1928
      PT(1,2) = PT(2,3)*PT(3,1) - PT(2,1)*PT(3,3)                     1929
      PT(2,2) = PT(1,1)*PT(3,3) - PT(3,1)*PT(1,3)                     1930
      PT(3,2) = PT(2,1)*PT(1,3) - PT(1,1)*PT(2,3)                     1931
C                                                                     1935
C  ***************************************************                1936
C                                                                     1937
      RETURN                                                          1952
      END                                                             1953
```

SUBROUTINE SMTR

```
$IBFTC SMTR*   DECK
      SUBROUTINE SMTR                                                    1957
      COMMON /PSTIF9/ NP,N4                                              1958
      COMMON /PSTIFH/ XL1,YL1,XL2,YL2,XL3,YL3,XL4,YL4                    1959
      COMMON/STRAN/GQUAD(8,24)                                           1960
      DIMENSION X(4),Y(4),XI(4),YI(4),NR(4),NU(4)                        1961
      DIMENSION Z(16)                                                    1962
      X(1)=XL1                                                           1963
      X(2)=XL2                                                           1964
      X(3)=XL3                                                           1965
      X(4)=XL4                                                           1966
      Y(1)=YL1                                                           1967
      Y(2)=YL2                                                           1968
      Y(3)=YL3                                                           1969
      Y(4)=YL4                                                           1970
      DO 3 I=1,8                                                         1971
      DO 1 J=1,24                                                        1972
    1 GQUAD(I,J)=0.0                                                     1973
    3 CONTINUE                                                           1974
      IF(N4.LE.0)GO TO 4                                                 1975
      XB=3.0*(X(4)*Y(3)-X(3)*Y(4)+X(2)*Y(4))                             1976
      XG=(X(4)+X(3))/3.0+(X(2)*Y(4)*(X(2)-X(3)))/XB                      1977
      YG=(Y(4)+Y(3))/3.0-X(2)*Y(4)*Y(3)/XB                               1978
      IF(X(2).EQ.X(4))GO TO 35                                           1979
      YI(2)=Y(4)*(XG-X(2))/(X(4)-X(2))                                   1980
      GO TO 36                                                           1981
   35 YI(2)=1.0E 20                                                      1982
   36 IF(X(4).EQ.X(3))GO TO 37                                           1983
      YI(3)=(Y(4)-Y(3))*(XG-X(3))/(X(4)-X(3))+Y(3)                       1984
      GO TO 38                                                           1985
   37 YI(3)=1.0E+20                                                      1986
   38 IF(X(3).EQ.0.)GO TO 39                                             1987
      YI(4)=Y(3)*XG/X(3)                                                 1988
      GO TO 40                                                           1989
   39 YI(4)=1.0E+20                                                      1990
   40 NT=4                                                               1991
      GO TO 5                                                            1992
    4 XG=(X(2)+X(3))/3.0                                                 1993
      YG=Y(3)/3.0                                                        1994
      IF(X(2).EQ.X(3))GO TO 45                                           1995
      YI(2)=Y(3)*(XG-X(2))/(X(3)-X(2))                                   1996
      GO TO 46                                                           1997
   45 YI(2)=1.0E 21                                                      1998
   46 IF(X(3).EQ.0.)GO TO 47                                             1999
      YI(3)=Y(3)*XG/X(3)                                                 2000
      GO TO 48                                                           2001
   47 YI(3)=1.0E+20                                                      2002
   48 NT=3                                                               2003
    5 CONTINUE                                                           2004
      YT=1.0E 20                                                         2005
      YB=0                                                               2006
      DO 8 I=2,NT                                                        2007
      IF(YI(I).GT.YG)GO TO 6                                             2008
      IF(YI(I).LE.YB)GO TO 8                                             2009
      YB=YI(I)                                                           2010
      GO TO 8                                                            2011
    6 IF(YI(I).GT.YT)GO TO 8                                             2012
```

```
      YT=YI(I)                                                          2013
    8 CONTINUE                                                          2014
      WY=YT-YB                                                          2015
      YC=(YT+YB)/2.0                                                    2016
      EIY=(WY**3)/12.0                                                  2017
      ELY=YC-YG                                                         2018
      JR=0                                                              2019
      DO 10 I=1,NT                                                      2020
      IF(X(I).LE.XG)GO TO 10                                            2021
      JR=JR+1                                                           2022
      NR(JR)=I                                                          2023
   10 CONTINUE                                                          2024
      IF(N4.GT.0)GO TO 11                                               2025
      XL=X(3)*YG/Y(3)                                                   2026
      XR=(X(3)-X(2))*YG/Y(3)+X(2)                                       2027
      GO TO 15                                                          2028
   11 XI(1)=(X(4)-X(2))*YG/Y(4)+X(2)                                    2029
      IF(Y(3).EQ.Y(4))GO TO 110                                         2030
      XI(2)=(YG-Y(3))*(X(4)-X(3))/(Y(4)-Y(3))+X(3)                      2031
      GO TO 111                                                         2032
  110 XI(2)=1.0E 21                                                     2033
  111 XI(3)=X(3)*YG/Y(3)                                                2034
      XR=1.0E 20                                                        2035
      XL=-1.0E 20                                                       2036
      DO 14 I=1,3                                                       2037
      IF(XI(I).GT.XG)GO TO 12                                           2038
      IF(XI(I).LE.XL)GO TO 14                                           2039
      XL=XI(I)                                                          2040
      GO TO 14                                                          2041
   12 IF(XI(I).GE.XR)GO TO 14                                           2042
      XR=XI(I)                                                          2043
   14 CONTINUE                                                          2044
   15 CONTINUE                                                          2045
      WX=XR-XL                                                          2046
      XC=(XR+XL)/2.0                                                    2047
      EIX=(WX**3)/12.0                                                  2048
      ELX=XG-XC                                                         2049
      JT=0                                                              2050
      DO 16 I=3,NT                                                      2051
      IF(Y(I).LE.YG)GO TO 16                                            2052
      JT=JT+1                                                           2053
      NU(JT)=I                                                          2054
   16 CONTINUE                                                          2055
      DO 17 J=1,JR                                                      2056
      K=(NR(J)-1)*6+3                                                   2057
      NJ=NR(J)                                                          2058
      GQUAD(1,K)=ELY/EIY                                                2059
      GQUAD(1,K+1)=-(Y(NJ)-YC)*ELY/EIY+1.0/WY                           2060
      GQUAD(1,K+2)=(X(NJ)-XG)*ELY/EIY                                   2061
      K=(NR(J)-1)*6+5                                                   2062
      GQUAD(3,K)=1.0/WY                                                 2063
      K=(NR(J)-1)*6+6                                                   2064
      GQUAD(4,K)=-1.0/WY                                                2065
      K=(NR(J)-1)*6+2                                                   2066
      GQUAD(7,K)=-1.0/WY                                                2067
      GQUAD(7,K+4)=(X(NJ)-XG)/WY                                        2068
   17 CONTINUE                                                          2069
      DO 18 J=1,JT                                                      2070
      K=(NU(J)-1)*6+3                                                   2071
      NJ=NU(J)                                                          2072
```

```
      GQUAD(2,K)=ELX/EIX                                                2073
      GQUAD(2,K+1)=-(Y(NJ)-YG)*ELX/EIX                                  2074
      GQUAD(2,K+2)=(X(NJ)-XC)*ELX/EIX+1.0/WX                            2075
      K=(NU(J)-1)*6+6                                                   2076
      GQUAD(5,K)=-1.0/WX                                                2077
      K=(NU(J)-1)*6+1                                                   2078
      GQUAD(6,K)=1.0/WX                                                 2079
      GQUAD(6,K+5)=(Y(NJ)-YG)/WX                                        2080
      K=(NU(J)-1)*6+2                                                   2081
      GQUAD(8,K)=1.0/WX                                                 2082
      GQUAD(8,K+4)=-(X(NJ)-XG)/WX                                       2083
   18 CONTINUE                                                          2084
   30 RETURN                                                            2085
      END                                                               2086
```

SUBROUTINE LOCAL

```
$IBFTC LOCAL*  DECK
      SUBROUTINE LOCAL                                                  2089
C                                                                       2090
C FORM LOCAL COORDINATES FROM STRUCTURAL COORDINATES                    2091
C FOR TRIANGULAR OR QUADRILATERAL PLATE                                 2092
C                                                                       2093
      COMMON/RENT/NRENT,KRENT                                           2094
      COMMON/PSTIF9/NP,N4                                               2095
      COMMON/PSTIFG/XS1,YS1,ZS1,XS2,YS2,ZS2,XS3,YS3,ZS3,XS4,YS4,ZS4     2096
      COMMON/PSTIFH/XL1,YL1,XL2,YL2,XL3,YL3,XL4,YL4                     2097
      COMMON/PSTIFL/ISWAP                                               2098
      COMMON/TAPES/MT1,MT2,MT3,MT4,MT5,MT6,MT7,MT8,MT9,MT10,MT11,MT12,
     * MT13,MT14,MT15,MT16,MT17
C                                                                       2102
      D(D1,D2,D3,D4,D5,D6)=(D1-D2)**2+(D3-D4)**2+(D5-D6)**2              2103
C                                                                       2104
C GENERATION OF LOCAL COORDINATES FOR FIRST THREE NODES IS THE          2105
C SAME FOR TRIANGULAR AND QUADRILATERAL PLATES                          2106
C                                                                       2107
C D21 IS THE DISTANCE SQUARED BETWEEN NODES 2 AND 1 FOUND FROM          2108
C STRUCTURAL COORDINATES, ETC.                                          2109
C                                                                       2110
      IOUT = MT6
      D21 = D(XS2,XS1,YS2,YS1,ZS2,ZS1)                                  2111
      D31 = D(XS3,XS1,YS3,YS1,ZS3,ZS1)                                  2112
      D32 = D(XS3,XS2,YS3,YS2,ZS3,ZS2)                                  2113
      ISWAP =0                                                          2114
      KRENT=0                                                           2115
C                                                                       2116
C NODE 1                                                                2117
C                                                                       2118
      XL1=0.0                                                           2119
      YL1=0.0                                                           2120
C                                                                       2121
C NODE 2                                                                2122
C                                                                       2123
      XL2=SQRT(D21)                                                     2124
      YL2=0.0                                                           2125
C                                                                       2126
C NODE 3                                                                2127
C                                                                       2128
      XL3=(D31-D32+D21)/(2.0*XL2)                                       2129
      YL3=SQRT(D31-XL3**2)                                              2130
C                                                                       2131
C NODE 4                                                                2132
C                                                                       2133
      IF(N4.EQ.0)GO TO 50                                               2134
   40 D41 = D(XS4,XS1,YS4,YS1,ZS4,ZS1)                                  2135
      D42 = D(XS4,XS2,YS4,YS2,ZS4,ZS2)                                  2136
      D43=D(XS4,XS3,YS4,YS3,ZS4,ZS3)                                    2137
      XL4=(D41-D42+D21)/(2.0*XL2)                                       2138
      YL4=SQRT(D41-XL4**2)                                              2139
      D43L=    ((XL4-XL3)**2+(YL4-YL3)**2)                              2140
      IF(ABS(D43-D43L).GE..10*D43)GO TO 44                              2141
C                                                                       2142
C CHECK SEQUENCE OF LOCAL COORDINATES OF QUADRILATERAL PLATE            2143
C                                                                       2144
```

```
      IF ((XL4*YL3)-(XL3*YL4)) 42,4,4
C                                                                       2150
C SWAP COORDINATES FOR NODES 3 AND 4                                    2151
C                                                                       2152
   42 ISWAP=1                                                           2153
      RETURN                                                            2154
C                                                                       2155
C CHECK FOR REENTRANT CORNER                                            2156
C                                                                       2157
    4 CONTINUE
      X32 = XL3 - XL2
      X42 = XL4 - XL2
      IF ((X42*YL3)-(X32*YL4)) 44,50,50
   44 WRITE(IOUT,1000)NP,XL1,XL2,XL3,XL4,YL1,YL2,YL3,YL4                2163
      NRENT =1                                                          2164
      KRENT=1                                                           2165
   50 RETURN                                                            2166
 1000 FORMAT(25H QUADRILATERAL PLATE NO. ,I4,                           2167
     *23H HAS A REENTRANT CORNER/23HOLOCAL COORDINATES ARE,/             2168
     *14HOX-COORDINATES,4(3X,E15.8)/14HOY-COORDINATES,4(3X,E15.8))      2169
      END                                                               2170
```

SUBROUTINE COPLAN

```
$IBFTC COPLA*  DECK
      SUBROUTINE COPLAN(NERR)                                                2173
C                                                                            2174
C DETERMINE THE DEGREE OF COPLANARTY                                         2175
C                                                                            2176
      COMMON/PSTIF9/NP,N4                                                    2177
      COMMON/PSTIFG/XS1,YS1,ZS1,XS2,YS2,ZS2,XS3,YS3,ZS3,XS4,YS4,ZS4          2178
      COMMON/PSTIFH/XL1,YL1,XL2,YL2,XL3,YL3,XL4,YL4                          2179
      COMMON/NOMERG/INER                                                     2180
      COMMON/CHECK/ACPT,GROSS                                                2181
      COMMON/TAPES/MT1,MT2,MT3,MT4,MT5,MT6,MT7,MT8,MT9,MT10,MT11,MT12,
     * MT13,MT14,MT15,MT16,MT17
C                                                                            2185
C ACPT - ACCEPTABLE PERCENT OF NONCOPLANARITY OF QUADRILATERAL PLATE          2186
C GROSS - NOT ACCEPTABLE PERCENT OF NONCOPLANARITY OF QUADRILATERAL PLAT      2187
C                                                                            2188
      IOUT = MT6
      NERR=0                                                                 2189
      RTD43=SQRT((XS4-XS3)**2+(YS4-YS3)**2+(ZS4-ZS3)**2)                     2190
      PERCT=(100.0*(RTD43-SQRT((XL4-XL3)**2+(YL4-YL3)**2)))/RTD43            2191
C                                                                            2192
C TEST FOR EXCEPTABLE ERROR WITH NO COMMENT                                  2193
C                                                                            2194
      IF(ABS(PERCT).LE.ACPT) GO TO 50                                        2195
C                                                                            2196
C TEST FOR MINOR OR GROSS ERROR WITH COMMENT                                 2197
C                                                                            2198
      IF(ABS(PERCT).GE.GROSS) GO TO 40                                       2199
C                                                                            2200
C MINOR ERROR - CONTINUE ANALYSIS                                            2201
C                                                                            2202
      WRITE(IOUT,1000)NP,PERCT                                               2203
      GO TO 50                                                               2204
C                                                                            2205
C GROSS ERROR - DELETE ANALYSIS                                              2206
C                                                                            2207
   40 NERR=1                                                                 2208
      INER=1                                                                 2209
      WRITE(IOUT,1001)NP,PERCT                                               2210
   50 RETURN                                                                 2211
 1000 FORMAT(25H0QUADRILATERAL PLATE NO. ,I4/                                2212
     *9X,25HDEGREE OF NONCOPLANARITY ,F6.3,8H PERCENT/                       2213
     *9X,31HACCEPTABLE - ANALYSIS CONTINUED)                                 2214
 1001 FORMAT(25H0QUADRILATERAL PLATE NO. ,I4/                                2215
     *9X,25HDEGREE OF NONCOPLANARITY ,F8.3,8H PERCENT/                       2216
     *9X,33HNOT ACCEPTABLE - ANALYSIS DELETED)                               2217
      END                                                                    2218
```

SUBROUTINE BEAM

```
$IBFTC BEAM*   DECK
      SUBROUTINE BEAM                                                    222
      COMMON/TITL/TITLE(13)
C                                                                        222
      DIMENSION XL(50),TEMP(50)
      DIMENSION PROP(600),FLX(6,6),SKAB(6,6),GAM1(6,6),GAM2(6,6),
     2AK(12,12),Q(12,12),BK(12,12),TLAM(12,12),GAMMA(12,12),QG(12,12),   222
     3OFSET(6),JFIX(6),ALAM(6,6),ALAMA(6,6),ALAMB(6,6),ALSL(6,6),        222
     4XKG(12,12)
C                                                                        223
      COMMON/CONT1/JPART(800)                                            223
      COMMON/CONT2/KPART(800)                                            223
      COMMON/LASTND/LN(200)                                              223
      COMMON/CORD/XN(2000),YN(2000),ZN(2000)                             223
      COMMON/TERMS/NBEAM,NPLATE,NNODE,NCOND,IPT,NPS ,IUM2                223
      COMMON/TAPES/MT1,MT2,MT3,MT4,MT5,MT6,MT7,MT8,MT9,MT10,MT11,MT12,
     * MT13,MT14,MT15,MT16,MT17
C                                                                        224
      COMMON/PROPT/YI(100),ZA(100),XA(100),ZI(100),YA(100),GJ(100)       224
      COMMON/ADPRO/EM,G,RC,ALFA,DARC,DL                                  224
      COMMON/SSTR/EMM,GG                                                 224
      COMMON/VAR/NPTS,FNPTS                                              224
      COMMON/N3N3/N3                                                     224
      COMMON/FLAG/NFLAG                                                  224
C                                                                        224
      EQUIVALENCE(PROP,YI)                                               224
C                                                                        225
      IN = MT5
      IOUT = MT6
      ISTRS = MT16
      ISTIF = MT2
      NPTS=100                                                           225
      FNPTS=100.0                                                        225
      K1=NPS+1                                                           225
C                                                                        225
C                                                                        226
C                                                                        226
 1999 CONTINUE                                                           226
      NB=0                                                               226
      LINE=0
      WRITE(IOUT,9011)(II,II=1,3)

C*****NB COUNTS BEAMS

 2001 NB=NB+1                                                            226
      ILOAD=0                                                            227
C                                                                        227

C****** READ BEAM DATA                                                   BEA

      RC=0.
      READ(IN,9030)KB,N1,N2,N3,ISP,NSP,IOC,IOFSET,
     1IFRC,(JFIX(K),K=1,6),IBUC,EM,G
      IF(IBUC.NE.0)READ(IN,9090)SK,SSK
      IF((IBUC.NE.0) .AND. (SK .NE. 0.0)) WRITE(IOUT,9351) SK
      IF((IBUC .NE. 0) .AND. (SSK .NE. 0.0)) WRITE(IOUT,9351) SSK
      IF(IFRC.NE.0)  READ(IN,9090) RC
```

```
      DO 16 K=1,6
   16 JFIX(K)=IABS(JFIX(K))
C -                                                                    227

C*****CHECK RANGE OF NODES

      IF(NNODE-N1     )25,20,20                                        227
   20 IF(NNODE-N2     )25,30,30                                        227
   25 WRITE (IOUT,9050)KB                                              227
      NFLAG=1                                                          227

C****** CHECK VALIDITY OF DATA

   30 IF(EM)35,40,45                                                   228
   35 WRITE(IOUT,9060)KB                                               228
      EM=ABS(EM)                                                       228
      GO TO 45                                                         228
   40 EM=EMM                                                           228
   45 IF(G)50,55,60                                                    228
   50 WRITE(IOUT,9070)KB                                               228
      G=ABS(G)                                                         228
      GO TO 60                                                         228
   55 G=GG                                                             228
   60 IF(KB     -NB)54,70,54                                           229
   54 WRITE(IOUT,9080)KB     ,NB                                       229
      CALL EXIT                                                        229
C                                                                      229
C     OFFSET NODES                                                     229
C                                                                      229
   70 IF(IOFSET)56,57,56                                               229
C                                                                      229
   56 READ  (IN,9090)(OFSET(I),I=1,6)                                  229
C                                                                      229
      GO TO 59                                                         230
   57 DO 58 I=1,6                                                      230
   58 OFSET(I)=0.0                                                     230
   59 M=N1                                                             230
      X1=XN(M)+OFSET(1)                                                230
      Y1=YN(M)+OFSET(2)                                                230
      Z1=ZN(M)+OFSET(3)                                                230
      XA1=ABS(XN(M))                                                   230
      YA1=ABS(YN(M))                                                   230
      ZA1=ABS(ZN(M))                                                   230
      M=N2                                                             231
      X2=XN(M)+OFSET(4)                                                231
      Y2=YN(M)+OFSET(5)                                                231
      Z2=ZN(M)+OFSET(6)                                                231
      XA2=ABS(XN(M))                                                   231
      YA2=ABS(YN(M))                                                   231
      ZA2=ABS(ZN(M))                                                   231
      M=N3                                                             231
      X3=XN(M)                                                         231
      Y3=YN(M)                                                         231
      Z3=ZN(M)                                                         232
C                                                                      232

C*****COMPUTE OFFSET TRANSFORMATION MATRIX

      IF(IOFSET.NE.0) CALL OFST(OFSET,GAMMA,GAM1,GAM2)                 232
C                                                                      232
```

```
C                                                                       232
C     CONNECTIVITY INFO FOR BEAM                                        232
C                                                                       232
C                                                                       232
      MPT=IPT-1                                                         232
      DO 65 I=1,MPT                                                     232
      IF((N1.LE.LN(I+1)).AND.(N1.GT.LN(I)))NI=I+1                       233
      IF((N2.LE.LN(I+1)).AND.(N2.GT.LN(I)))NJ=I+1                       233

C*****N1 IS IN THE NI-TH PARTITION
C*****N2 IS IN THE NJ-TH PARTITION

      IF(N1.LE.LN(1))NI=1                                               233
      IF(N2.LE.LN(1))NJ=1                                               233
   65 CONTINUE                                                          233
C                                                                       233
      IF(NI.GE.NJ)NINJ=NI*1000+NJ                                       233
      IF(NJ.GT.NI) NINJ=NJ*1000+NI                                      233
      NINI=1001*NI                                                      233
      NJNJ=1001*NJ                                                      234
C                                                                       234
C                                                                       234
      K11=K1-1                                                          234
      KC=0                                                              234
      KC1=0                                                             234
      KC2=0                                                             234
C

C*****K1 COUNTS PARTITIONS
C*****FIRST DIGIT OF KPART = FIRST BEAM IN THE PARTITION
C*****  LAST DIGIT OF KPART = LAST BEAM IN THE PARTITION

      DO 66 I=1,800                                                     234
      IF(JPART(I).EQ.NINJ)KC=I                                          234
      IF(JPART(I).EQ.NINI)KC1=I                                         234
      IF(JPART(I).EQ.NJNJ)KC2=I                                         235
   66 CONTINUE                                                          235
C                                                                       235
      IF(KC.EQ.0)GO TO 81                                               235

C*****UPDATE KPART

      IF(KPART(KC ).EQ.0) KPART(KC )=10001*KB                           235
      KPART(KC )=(KPART(KC )/10000)*10000+KB                            235
      GO TO 82                                                          235
   81 CONTINUE                                                          235

C*****NEW PARTITION - COMPUTE JPART, KPART

      JPART(K1)=NINJ                                                    235
      KPART(K1 )=10001*KB                                               235
      K1=K1+1                                                           236
   82 IF(KC1.EQ.0)GO TO 83                                              236

C*****UPDATE KPART

      IF(KPART(KC1).EQ.0) KPART(KC1)=10001*KB                           236
      KPART(KC1)=(KPART(KC1)/10000)*10000+KB                            236
      GO TO 84                                                          236
   83 CONTINUE                                                          236
```

```
C*****DON'T COUNT PARTITION AGAIN IF ALREADY COUNTED

      IF(NINJ.EQ.NINI)GO TO 67                                          236

C*****NEW PARTITION - COMPUTE JPART, KPART

      JPART(K1)=NINI                                                    236
      KPART(K1 )=10001*KB                                               236
      K1=K1+1                                                           236
   67 CONTINUE                                                          237
   84 IF(KC2.EQ.0)GO TO 85                                              237

C*****UPDATE KPART

      IF(KPART(KC2).EQ.0) KPART(KC2)=10001*KB                           237
      KPART(KC2)=(KPART(KC2)/10000)*10000+KB                            237
      GO TO 86                                                          237
   85 CONTINUE                                                          237

C*****DON'T COUNT PARTITION AGAIN IF ALREADY COUNTED

      IF((NJNJ.EQ.NINI).OR.(NJNJ.EQ.NINJ))GO TO 71                      237

C*****NEW PARTITION - COMPUTE JPART, KPART

      JPART(K1)=NJNJ                                                    237
      KPART(K1 )=10001*KB                                               237
      K1=K1+1                                                           237
   71 CONTINUE                                                          238
   86 CONTINUE                                                          238
C                                                                       238
C     SECTION PROPERTIES                                                238
C                                                                       238
      IF(ISP)75,810,75
C                                                                       238
   75 NCT=0                                                             238
      NI=0                                                              238
      GO TO(80,90,130),ISP                                              238
C                                                                       239

C****** UNIFORM PROPERTIES

   80 READ  (IN,9090)YI(1),ZA(1),XA(1),ZI(1),YA(1),GJ(1)                239
  810 WRITE(IOUT,9012)KB,N1,N2,N3,JFIX,YI(1),ZA(1),XA(1),ZI(1),YA(1),
     1   GJ(1),EM,G
      LINE=LINE+1
      IF(LINE.LT.50)GO TO 803
      LINE=0
      CALL PAGHED
      WRITE(IOUT,9011)(II,II=1,3)
  803 CONTINUE
      IF(IOFSET.EQ.0) GO TO 802
      WRITE(IOUT,9014)(OFSET(I),I=1,6)
      LINE=LINE+1
      IF(LINE.LT.50)GO TO 802
      LINE=0
      CALL PAGHED
      WRITE(IOUT,9011)(II,II=1,3)
  802 CONTINUE
```

```
      IF(ISP.EQ.0) GO TO 190                                          239
C                                                                     239
      DO 89 I=2,100                                                   239
      PROP(I)=YI(1)                                                   239
      PROP(I+100)=ZA(1)                                               239
      PROP(I+200)=XA(1)                                               239
      PROP(I+300)=ZI(1)                                               239
      PROP(I+400)=YA(1)                                               239
   89 PROP(I+500)=GJ(1)                                               240
      GO TO 190                                                       240
C
C****** PROPERTIES VARY ACCORDING TO FORMULA
   90 DO 125 I=1,6                                                    240
      LL=100*I                                                        240
      KK=LL-99                                                        240
      IF(NI-I)95,100,115                                              240
C                                                                     240
   95 READ  (IN,9150)NI,K,L,M,N,A,B,C
      WRITE(IOUT,9016)KB,N1,N2,N3,JFIX,NI,K,L,M,N,A,B,C
      LINE=LINE+1
      IF(LINE.LT.50)GO TO 804
      LINE=0
      CALL PAGHED
      WRITE(IOUT,9011)(II,II=1,3)
  804 CONTINUE                                                        240
C                                                                     240
      IF(NI-I)126,100,115                                             241
  100 AJ=-.005                                                        241
      NCT=NCT+1                                                       241
      DO 105 J=KK,LL                                                  241
      AJ=AJ+.01                                                       241
      PROP(J)=C*(1.0+A*(AJ)**K)**M*(1.0+B*(AJ)**L)**N                 241
  105 CONTINUE                                                        241
      IF(NCT-NSP)125,110,125                                          241
  110 NI=7                                                            241
      GO TO 125                                                       241
  115 DO 120 J=KK,LL                                                  241
      PROP(J)=0.0                                                     242
  120 CONTINUE                                                        242
  125 CONTINUE                                                        242
C                                                                     242
      GO TO 190                                                       242
  126 WRITE (IOUT,9160)NI                                             242
      CALL EXIT                                                       242
C
C****** PROPERTIES VARY ACCORDING TO TABLE
  130 DO 185 K=1,6                                                    242
      LL=100*K                                                        242
      KK=LL-99                                                        243
      IF(NI-K)135,140,175                                             243
C                                                                     243
  135 READ  (IN,9010)NI,LTBL,BL                                       243
C                                                                     243
      IF(NI-K)126,140,175                                             243
C                                                                     243
  140 READ  (IN,9090)(XL(J),TEMP(J),J=1,LTBL)                         243
```

```
      WRITE(IOUT,9017)KB,N1,N2,N3,JFIX,(XL(J),TEMP(J),J=1,LTBL)
      LINE=LINE+1
      IF(LINE.LT.50)GO TO 805
      LINE=0
      CALL PAGHED
      WRITE(IOUT,9011)(II,II=1,3)
  805 CONTINUE
C                                                                       243
      DO 141 I=1,LTBL                                                   243
  141 XL(I)=XL(I)/BL                                                    244
      DELT=-.005                                                        244
      I=2                                                               244
      DO 165 J=KK,LL                                                    244
      DELT=DELT+.01                                                     244
  145 IF(DELT-XL(I))150,155,160                                         244
  150 PROP( J)=TEMP(I)-((TEMP(I)-TEMP(I-1))*(XL(I)-DELT)/               244
     1(XL(I)-XL(I-1)))                                                  244
      GO TO 165                                                         244
  155 PROP(J)=TEMP(I)                                                   244
      GO TO 165                                                         245
  160 I=I+1                                                             245
      GO TO 145                                                         245
  165 CONTINUE                                                          245
      NCT=NCT+1                                                         245
      IF(NCT-NSP)185,170,185                                            245
  170 NI=7                                                              245
      GO TO 185                                                         245
  175 CONTINUE                                                          245
      DO 180 L=KK,LL                                                    245
  180 PROP(L)=0.0                                                       246
  185 CONTINUE                                                          246
C                                                                       246
  190 CONTINUE                                                          246
C                                                                       246

C****** COMPUTE TRANSFORMATION AND STIFFNESS MATRICES

C                                                                       246
      IF(RC)205,200,205                                                 246
C                                                                       246
C                                                                       246
  200 CALL SBMTR(X1,Y1,Z1,X2,Y2,Z2,X3,Y3,Z3,ALAM,TLAM)                  246
      CALL SSTIF(AK)
      GO TO 206                                                         247
C                                                                       247
  205 CALL CBMTR(X1,Y1,Z1,X2,Y2,Z2,X3,Y3,Z3,ALAMA,ALAMB,TLAM)           247
      CALL CSTIF(AK,FLX,SKAB)                                           247
C                                                                       247
  206 CONTINUE                                                          247
C                                                                       247
      IF(IBUC.NE.0) CALL SBGS(SK,SSK,DL,XKG)
      IF(IBUC.NE.0) CALL MAD(AK,XKG)
C                                                                       247
C                                                                       248
      DO 201 I=1,6
      DO 201 L=1,6
      SKAB(I,L)=AK(I,L+6)
  201 FLX(I,L)=AK(I+6,L+6)
C                                                                       248
  591 DO 208 I=1,6
```

```
      IF(JFIX(I),NE,1)GO TO 208                                          248
      M=I                                                                248
      IF(I,GT,3)M=I+3                                                    248
      CALL REDUCE(AK,12,M)
  208 CONTINUE                                                           248
  710 CONTINUE                                                           280
      IF(IOFSET)210,209,210                                              248
C                                                                        248
  209 CONTINUE                                                           249
      CALL MULT(AK,TLAM,Q,12,12,12,1,0)                                  249
      CALL MULT(TLAM,Q,BK,12,12,12,2,0)                                  249
      GO TO 211                                                          249
C                                                                        249
  210 CONTINUE                                                           249
      CALL MULT(TLAM,GAMMA,QG,12,12,12,1,0)                              2 9
      CALL MULT(AK,QG,Q,12,12,12,1,0)                                    249
      CALL MULT(QG,Q,BK,12,12,12,2,0)                                    249
C                                                                        249
  211 CONTINUE                                                           250
C                                                                        250
C     ENGINEERING SIGN CONVENTION                                        250
      DO 213 I=1,12                                                      250
      Q(1,I)=-Q(1,I)                                                     250
      Q(3,I)=-Q(3,I)                                                     250
      Q(4,I)=-Q(4,I)                                                     250
      Q(8,I)=-Q(8,I)                                                     250
      Q(11,I)=-Q(11,I)                                                   250
      Q(12,I)=-Q(12,I)                                                   250
  213 CONTINUE                                                           251
C                                                                        251
C                                                                        251
C                                                                        251
C     WRITING STRESS MATRIX                                              251
C                                                                        251
      WRITE(ISTRS)KB,N1,N2                                               251
      WRITE(ISTRS)((Q(I,J),I=1,12),J=1,12)                               251
C                                                                        251
C                                                                        251
C                                                                        252
C     WRITING STIFFNESS MATRIX                                           252
C                                                                        252
      WRITE(ISTIF)N1,N2                                                  252
C                                                                        252
      WRITE(ISTIF)((BK(J,I),J=1,6),I=1,6) ,                              252
     1((BK(J,I),J=1,6),I=7,12),((BK(J,I),J=7,12),I=1,6),                 252
     2((BK(J,I),J=7,12),I=7,12)                                          252
C                                                                        252
C                                                                        281
C                                                                        281
      IF(IOC,EQ,0)GO TO 1950                                             281
C                                                                        281
      WRITE(IOUT,9320)                                                   281
      CALL PRINT(TLAM,12,12,1,4HTLAM,1,12)                               281
      WRITE (IOUT,9270)                                                  281
      CALL PRINT(AK,12,12,1,4HAK  ,1,12)                                 281
      WRITE (IOUT,9310)                                                  281
      CALL PRINT(Q,12,12,1,4HQ   ,1,12)                                  281
      WRITE (IOUT,9300)                                                  282
      CALL PRINT(BK,12,12,1,4HBK  ,1,12)                                 282
 1950 CONTINUE                                                           282
```

```
C                                                                       282
 2000 IF(NBEAM-NB)2002,2002,2001                                        282
C                                                                       282
C                                                                       283
 2002 CONTINUE                                                          283
      NPS=K1-1                                                          283
C                                                                       284
C
 9010 FORMAT(2I4,4X,E12.4)
 9011 FORMAT(/56X,9HBEAM DATA//
     1 1X,   5H BEAM,3(5H NODE ),4X,6HFIXITY,4X,4HI(Y),8X,5HA(XY),9X,
     21HA,9X,4HI(Z),8X,5HA(XZ),9X,1HJ,9X,6HYG MOD,6X,6HSH MOD//4X,3I5//)
 9012 FORMAT(4I5,4X,6I1,8(1PE12.3))
 9013 FORMAT(9H RADIUS =E14.4)
 9014 FORMAT(9H OFFSETS 6E12.4)
 9016 FORMAT(4I5,4X,6I1,10H PROPERTY I1,16H   CONSTANTS  K=I5,3H L=I5,
     1  3H M=I5,3H N=I5,3H A=1PE11.3,3H B=1PE11.3,3H C=1PE11.3)
 9017 FORMAT(4I5,4X,6I1,10H PROPERTY I1,3(3H X=E12.4,5H VAL=E12.4))
 9030 FORMAT(8I4,4X,I4,6I1,I2,2E12.4)
 9050 FORMAT(33H1INCORRECT NODE NUMBER, BEAM NO. I3)                    284
 9060 FORMAT(42H1NEGATIVE MODULUS OF ELASTICITY, BEAM NO. I3)           284
 9070 FORMAT(40H1NEGATIVE MODULUS OF RIGIDITY, BEAM NO. I3)             284
 9080 FORMAT(40H1INPUT NOT IN PROPER SEQUENCE, BEAM NO. I3,             285
     111H SHOULD BE I3)                                                 285
 9090 FORMAT(6E12.4)                                                    285
 9150 FORMAT(5I4,4X,3E12.4)
 9160 FORMAT(1H117HSECTION PROPERTY I1,18HIS OUT OF SEQUEN3E)           285
 9170 FORMAT(3I4,5E12.4)                                                285
 9190 FORMAT(E12.4,4I4)                                                 285
 9270 FORMAT(23H1LOCAL STIFFNESS MATRIX)                                285
 9290 FORMAT(29H1OFFSET TRANSFORMATION MATRIX)                          286
 9300 FORMAT(28H1STRUCTURAL STIFFNESS MATRIX)                           286
 9310 FORMAT(14H1STRESS MATRIX)                                         286
 9320 FORMAT(22H1TRANSFORMATION MATRIX)                                 286
 9350 FORMAT(1H I4,6E12.4)                                              286
 9351 FORMAT(20H0INITIAL BEAM THRUST,F10.2,5H LBS.//)
C                                                                       286
      RETURN                                                            286
      END                                                               286
```

SUBROUTINE TINVR

```
$IBFTC TINVR*  DECK
      SUBROUTINE TINVR (ELEM,N,IND)                                    3700
      DIMENSION ELEM(6,6)                                              3701
      CALL OVERFL(K000FX)                                              3702
       GO TO(99,99),K000FX                                             3703
 99   CALL DVCHK (K000FX)                                              3704
       GO TO(100,100),K000FX                                           3705
C     COMPUTE EQUIVALENT TRIANGULAR MATRIX                             3706
 100  DO 111 I=1,N                                                     3707
      IF (ELEM(I,I)) 107,107,108                                       3708
 107  IND=-1                                                           3709
      GO TO 310                                                        3710
 108  ELEM(I,I)=SQRT(ELEM(I,I))                                        3711
      L=I+1                                                            3712
      IF (L-N) 103,103,116                                             3713
 103  DO 102 J=L,N                                                     3714
 102  ELEM(J,I)=ELEM(I,J)/ELEM(I,I)                                    3715
      CALL DVCHK (K000FX)                                              3716
       GO TO(107,106),K000FX                                           3717
 106  DO 111 K=L,N                                                     3718
      IF (ELEM(K,I))112,111,112                                        3719
 112  DO 110 J=K,N                                                     3720
 110  ELEM(K,J)=ELEM(K,J)-ELEM(K,I)*ELEM(J,I)                          3721
 111  CONTINUE                                                         3722
C     INVERT TRIANGULAR MATRIX                                         3723
 116  M=N-1                                                            3724
      DO 198 J=1,M                                                     3725
      L=J+1                                                            3726
      DO 198 K=L,N                                                     3727
 198  ELEM(J,K)=0.0                                                    3728
      DO 199 I=1,N                                                     3729
 199  ELEM(I,I)=1.0/ELEM(I,I)                                          3730
      DO 206 I=1,M                                                     3731
      L=I+1                                                            3732
      DO 206 J=L,N                                                     3733
      M=J-1                                                            3734
      DO 204 K=I,M                                                     3735
 204  ELEM(I,J)=ELEM(I,J)-ELEM(J,K)*ELEM(I,K)                          3736
 206  ELEM(I,J)=ELEM(J,J)*ELEM(I,J)                                    3737
      CALL OVERFL(K000FX)                                              3738
       GO TO(200,207),K000FX                                           3739
 200  IND=-2                                                           3740
      GO TO 310                                                        3741
C     EXPAND TRIANGULAR INVERSE                                        3742
 207  DO 299 I=2,N                                                     3743
      L=I-1                                                            3744
      DO 299 J=1,L                                                     3745
 299  ELEM(I,J)=0.0                                                    3746
      DO 306 I=1,N                                                     3747
      DO 306 J=I,N                                                     3748
      E=0.0                                                            3749
      DO 305 K=J,N                                                     3750
 305  E=E+ELEM(I,K)*ELEM(J,K)                                          3751
 306  ELEM(J,I)=E                                                      3752
      DO 307 I=2,N                                                     3753
      L=I-1                                                            3754
      DO 307 J=1,L                                                     3755
```

```
307   ELEM(J,I)=ELEM(I,J)                                          3756
      CALL OVERFL(K000FX)                                          3757
       GO TO(300,308),K000FX                                       3758
300   IND=-3                                                       3759
      GO TO 310                                                    3760
308   IND=+1                                                       3761
 310  RETURN                                                       3762
      END                                                          3763
```

SUBROUTINE SMULT

```
$IBFTC SMULT*  DECK
       SUBROUTINE SMULT(A,B,C,N1,N2,N3)                                 3766
C                                                                       3767
       DIMENSION A(N1,N2),B(N2,N3),C(N1,N3)                             3768
C                                                                       3769
C      C=C-A*B                                                          3770
C      WHERE C IS AN N1*N3                                              3771
C            A IS AN N1*N2                                              3772
C            B IS AN N2*N3                                              3773
C                                                                       3774
       DO 100 I=1,N1                                                    3775
       DO 100 J=1,N3                                                    3776
       C(I,J)=0.0                                                       3777
       DO 100 K=1,N2                                                    3778
       C(I,J)=C(I,J)-A(I,K)*B(K,J)                                      3779
  100  CONTINUE                                                         3780
C                                                                       3781
       RETURN                                                           3782
       END                                                              3783
```

SUBROUTINE MULT

```
$IBFTC MULT*   DECK
       SUBROUTINE MULT(A,B,C,N1,N2,N3,N4,IM )                          3785
C      N1,N2,AND N3 MUST BE IN THEIR ORDER AFTER TRANSPOSITION         3786
       DIMENSION A(N1,N2),B(N2,N3),C(N1,N3)                            3787
C      N4=1...NORMAL                                                   3788
C      N4=2...(A(TRANSPOSE))*B                                         3789
C                                                                      3790
C      C=C+A*B                                                         3791
C      WHERE C IS AN N1*N3                                             3792
C            A IS AN N1*N2                                             3793
C            B IS AN N2*N3                                             3794
C                                                                      3795
C      IM=0 NORMAL                                                     3796
C        =1 MOVE C TO B                                                3797
C                                                                      3798
       GO TO(1,2),N4                                                   3799
C                                                                      3800
     2 CONTINUE                                                        3801
       DO 10 I=1,N1                                                    3802
       DO 10 J=1,N3                                                    3803
       C(I,J)=0.0                                                      3804
       DO 10 K=1,N2                                                    3805
       C(I,J)=C(I,J)+A(K,I)*B(K,J)                                     3806
    10 CONTINUE                                                        3807
       GO TO 98                                                        3808
     1 CONTINUE                                                        3809
       DO 100 I=1,N1                                                   3810
       DO 100 J=1,N3                                                   3811
       C(I,J)=0.0                                                      3812
       DO 100 K=1,N2                                                   3813
       C(I,J)=C(I,J)+A(I,K)* B(K,J)                                    3814
   100 CONTINUE                                                        3815
    98 CONTINUE                                                        3816
       IF(IM)300,200,300                                               3817
   200 RETURN                                                          3818
   300 DO 320 I=1,N1                                                   3819
       DO 320 J=1,N3                                                   3820
       B(I,J)=C(I,J)                                                   3821
   320 CONTINUE                                                        3822
       GO TO 200                                                       3823
       END                                                             3824
```

SUBROUTINE SBMTR

```
$IBFTC SBMTR*  DECK
      SUBROUTINE SBMTR(XI,YI,ZI,XF,YF,ZF,XC,YC,ZC,ALAM,SLAM)                3312
                                                                            3313
C                                                                           3314
C     STRAIGHT BEAM TRANSFORMATION MATRIX                                   3315
C                                                                           3316
      DIMENSION ALAM(6,6),SLAM(12,12)                                       3317
C
      COMMON/TAPES/MT1,MT2,MT3,MT4,MT5,MT6,MT7,MT8,MT9,MT10,MT11,MT12,
     * MT13,MT14,MT15,MT16,MT17
      COMMON/VAR/NPTS,FNPTS                                                 3319
      COMMON/ADPRO/EM,G,RC,ALFA,DARC,DL                                     3320
      COMMON/N3N3/N3                                                        3321
C                                                                           3322
      XFI=XF-XI                                                             3323
      YFI=YF-YI                                                             3324
      ZFI=ZF-ZI                                                             3325
      DS=SQRT(XFI**2+YFI**2)                                                3326
      DL=SQRT(XFI**2+YFI**2+ZFI**2)                                         3327
      DARC=DL/FNPTS                                                         3328
C                                                                           3329
      IOUT = MT6
      IF(N3.EQ.0)GO TO 5                                                    3330
      XCI=XC-XI                                                             3331
      YCI=YC-YI                                                             3332
      ZCI=ZC-ZI                                                             3333
      F=(XCI*XFI+YCI*YFI)/DS                                                3334
      B=-(XCI*YFI-YCI*XFI)/DS                                               3335
      D=(-F*ZFI+ZCI*DS)/DL                                                  3336
      DIV=SQRT(D*D+B*B)                                                     3337
      DIVR=SQRT(XCI**2+YCI**2)                                              3338
    5 CONTINUE                                                              3339
C                                                                           3340
      DO 10 I=1,6                                                           3341
      DO 10 J=1,6                                                           3342
   10 ALAM(I,J)=0.0                                                         3343
C                                                                           3344
      DO 15 I=1,12                                                          3345
      DO 15 J=1,12                                                          3346
   15 SLAM(I,J)=0.0                                                         3347
C                                                                           3348
      IF(DS)20,40,60                                                        3349
C                                                                           3350
   20 WRITE(IOUT,9000)XI,YI,ZI,XF,YF,ZF                                     3351
 9000 FORMAT(19H1LENGTH IS NEGATIVE///(6E10.3))                             3352
C                                                                           3353
   40 CONTINUE                                                              3354
      IF(N3.NE.0)GO TO 41                                                   3355
      SINB=0.                                                               3356
      COSB=1.                                                               3357
      GO TO 42                                                              3358
   41 SINB=YCI/DIVR                                                         3359
      COSB=XCI/DIVR                                                         3360
   42 ALAM(1,3)=ZFI/DL                                                      3361
      ALAM(2,1)=SINB*(ZFI/DL)                                               3362
      ALAM(2,2)=-COSB*(ZFI/DL)                                              3363
      ALAM(3,1)=COSB                                                        3364
      ALAM(3,2)=SINB                                                        3365
```

```
      GO TO 65                                                          3366
C                                                                       3367
   60 CONTINUE                                                          3368
      IF(N3.NE.0) GO TO 61                                              3369
      SINB=0.                                                           3370
      COSB=1.                                                           3371
      GO TO 62                                                          3372
   61 COSB=D/DIV                                                        3373
      SINB=B/DIV                                                        3374
   62 ALAM(1,1)=XFI/DL                                                  3375
      ALAM(1,2)=YFI/DL                                                  3376
      ALAM(1,3)=ZFI/DL                                                  3377
      ALAM(2,1)=-COSB*(YFI/DS)+SINB*(ZFI/DL)*(XFI/DS)                   3378
      ALAM(2,2)=COSB*(XFI/DS)+SINB*(ZFI/DL)*(YFI/DS)                    3379
      ALAM(2,3)=-SINB*(DS/DL)                                           3380
      ALAM(3,1)=-SINB*(YFI/DS)-COSB*(ZFI/DL)*(XFI/DS)                   3381
      ALAM(3,2)=SINB*(XFI/DS)-COSB*(ZFI/DL)*(YFI/DS)                    3382
      ALAM(3,3)=COSB*(DS/DL)                                            3383
C                                                                       3384
   65 CONTINUE                                                          3385
C                                                                       3386
      DO 90 I=1,3                                                       3387
      DO 90 J=1,3                                                       3388
      ALAM(I+3,J+3)=ALAM(I,J)                                           3389
   90 CONTINUE                                                          3390
C                                                                       3391
      DO 110 I=1,6                                                      3392
      DO 110 J=1,6                                                      3393
      SLAM(I,J)=ALAM(I,J)                                               3394
      SLAM(I+6,J+6)=ALAM(I,J)                                           3395
  110 CONTINUE                                                          3396
C                                                                       3397
      RETURN                                                            3398
      END                                                               3399
```

SUBROUTINE SSTIF

```
$IBFTC SSTIF*  DECK
      SUBROUTINE SSTIF(AK)                                                  3067
C                                                                           3068
C     STRAIGHT BEAM STIFFNESS                                               3069
C
      DIMENSIONAK(12,12)                                                    3071
C
      COMMON/VAR/NPTS,FNPTS                                                 3075
      COMMON/ADPRO/EM,G,RC,ALFA,DARC,DL                                     3076
      COMMON/PROPT/YI(100),ZA(100),XA(100),ZI(100),YA(100),GJ(100)          3077
C                                                                           3078
C                                                                           3079
      YSUM1=0.0                                                             3080
      YSUM2=0.0                                                             3081
      YSUM3=0.0                                                             3082
      YSUM4=0.0                                                             3083
      ZSUM1=0.0                                                             3084
      ZSUM2=0.0                                                             3085
      ZSUM3=0.0                                                             3086
      ZSUM4=0.0                                                             3087
      SUM5=0.0                                                              3088
      SUM6=0.0                                                              3089
      A=0.0                                                                 3090
      B=201.0                                                               3091
      C=-1.0                                                                3092
      S=DL                                                                  3093
C                                                                           3094
      DO 100 I=1,12                                                         3095
      DO 100 J=1,I                                                          3096
  100 AK(I,J)=0.0                                                           3097
C                                                                           3098
      DO 200 I=1,NPTS                                                       3099
      A=A+1.0                                                               3100
      B=B-2.0                                                               3101
      C=C+2.0                                                               3102
      YSUM1=YSUM1+B**2/ZI(I)                                                3103
      YSUM2=YSUM2+1.0/YA(I)                                                 3104
      YSUM3=YSUM3+C**2/ZI(I)                                                3105
      YSUM4=YSUM4+C*B/ZI(I)                                                 3106
      ZSUM1=ZSUM1+B**2/YI(I)                                                3107
      ZSUM2=ZSUM2+1.0/ZA(I)                                                 3108
      ZSUM3=ZSUM3+C**2/YI(I)                                                3109
      ZSUM4=ZSUM4+C*B/YI(I)                                                 3110
      SUM5=SUM5+1.0/XA(I)                                                   3111
      SUM6=SUM6+1.0/GJ(I)                                                   3112
  200 CONTINUE                                                              3113
C                                                                           3114
      PARTC2=YSUM2/(G*S*FNPTS)                                              3115
      B=S/(EM*(4.0*FNPTS**3))                                               3116
      PARTC1=B*ZSUM1                                                        3117
      C1 = PARTC1 + PARTC2                                                  3118
      C2=B*ZSUM3+PARTC2                                                     3119
      C3=B*ZSUM4-PARTC2                                                     3120
      C4 = C1*C2 - C3**2                                                    3121
      AK(2,2)=C2/C4                                                         3122
      AK(8,2)=C3/C4                                                         3123
      AK(8,8)=C1/C4                                                         3124
```

```
      AK(6,6)=(AK(2,2)+2.0*AK(8,2)+AK(8,8))/S**2                        3125
      AK(4,4)=FNPTS*EM/(S*SUM5)                                         3126
      B=S/(EM*(4.0*FNPTS**3))                                           3127
      PARTC1=B*YSUM1                                                    3128
      PARTC2=ZSUM2/(G*S*FNPTS)                                          3129
      C1=PARTC1+PARTC2                                                  3130
      C2=B*YSUM3+PARTC2                                                 3131
      C3=B*YSUM4-PARTC2                                                 3132
      C4=C1*C2-C3**2                                                    3133
C                                                                       3134
      AK(3,3)=C2/C4                                                     3135
      AK(9,3)=C3/C4                                                     3136
      AK(9,9)=C1/C4                                                     3137
      AK(12,2)=(AK(2,2)+AK(8,2))/S                                      3138
      AK(9,5)=(AK(9,3)+AK(9,9))/S                                       3139
      AK(1,1)=FNPTS*G/(S*SUM6)                                          3140
      AK(5,5)=(AK(3,3)+2.0*AK(9,3)+AK(9,9))/S**2                        3141
      AK(8,6)=-(AK(8,2)+AK(8,8))/S                                      3142
      AK(11,9)=-(AK(9,3)+AK(9,9))/S                                     3143
      AK(12,8)=(AK(8,2)+AK(8,8))/S                                      3144
      AK(7,1)=-AK(1,1)                                                  3145
      AK(10,4)=-AK(4,4)                                                 3146
      AK(11,5)=-AK(5,5)                                                 3147
      AK(12,6)=-AK(6,6)                                                 3148
      AK(6,2)=-AK(12,2)                                                 3149
      AK(5,3)=(AK(3,3)+AK(9,3))/S                                       3150
      AK(11,3)=-AK(5,3)                                                 3151
      AK(7,7)=AK(1,1)                                                   3152
      AK(10,10)=AK(4,4)                                                 3153
      AK(11,11)=AK(5,5)                                                 3154
      AK(12,12)=AK(6,6)                                                 3155
C                                                                       3156
      DO 300 I=1,11                                                     3157
      IDIAG=I+1
      DO 300 J=IDIAG,12                                                 3158
  300 AK(I,J)=AK(J,I)                                                   3159
C                                                                       3160
C                                                                       3166
      RETURN                                                            3167
      END                                                               3168
```

SUBROUTINE MAD

```
$IBFTC MAD*     DECK
       SUBROUTINE MAD(A,B)
       DIMENSION A(144),B(144)
       DO 10 I=1,144
       A(I)=A(I)+B(I)
    10 CONTINUE
       RETURN
       END
```

SUBROUTINE SBGS

```
$IBFTC SBGS*   DECK
       SUBROUTINE SBGS(SK,SSK,S,XKG)
       DIMENSION XKG(12,12)
C
C *** COMPUTES STRAIGHT BEAM GEOMETRIC STIFFNESS MATRIX-XKG
C     **CLEAR ARRAY**
       DO 10 I=1,12
       DO 10 J=1,I
       XKG(I,J)=0.0
   10 CONTINUE
C
C     **COMPUTE ELEMENTS OF LOWER TRIANGLE**
       SK= -SK
C
       XKG(6,6)=-1.2/S*SK
       XKG(12,12)=XKG(6,6)
       XKG(12,6)=-XKG(6,6)
       XKG(12,2)=-.1*SK
       XKG(12,8)=XKG(12,2)
       XKG(6,2)=-XKG(12,2)
       XKG(8,6)=XKG(6,2)
       XKG(8,2) =S/30.*SK
       XKG(2,2)=-4.*XKG(8,2)
       XKG(8,8)=XKG(2,2)
C
       SSK=-SSK
       XKG(5,5)=-1.2/S*SSK
       XKG(11,11)=XKG(5,5)
       XKG(11,5)=-XKG(5,5)
       XKG(11,3)=-.1*SSK
       XKG(11,9)=XKG(11,3)
       XKG(5,3)=-XKG(11,3)
       XKG(9,5)=XKG(5,3)
       XKG(9,3)=S/30.*SSK
       XKG(3,3)=-4.*XKG(9,3)
       XKG(9,9)=XKG(3,3)
C     **FILL UPPER TRIANGLE**
C
       DO 20 I=1,11
       IDIAG=I+1
       DO 20 J=IDIAG,12
       XKG(I,J)=XKG(J,I)
   20 CONTINUE
C
       RETURN
       END
```

SUBROUTINE OFST

```
$IBFTC OFST*   DECK
      SUBROUTINE OFST(OFSET,GAMMA,GAM1,GAM2)                              3586
C                                                                         3587
C     OFFSET BEAM TRANSFORMATION                                          3588
C                                                                         3589
      DIMENSION OFSET(6),GAMMA(12,12),GAM1(6,6),GAM2(6,6)                 3590
C                                                                         3591
      DO 100 I=1,12                                                       3592
      DO 100 J=1,12                                                       3593
      GAMMA(I,J)=0.0                                                      3594
  100 GAMMA(I,I)=1.0                                                      3595
C                                                                         3596
      GAMMA(5,1)=-OFSET(3)                                                3597
      GAMMA(6,1)=OFSET(2)                                                 3598
      GAMMA(4,2)=OFSET(3)                                                 3599
      GAMMA(6,2)=-OFSET(1)                                                3600
      GAMMA(4,3)=-OFSET(2)                                                3601
      GAMMA(5,3)=OFSET(1)                                                 3602
C                                                                         3603
      GAMMA(11,7)=-OFSET(6)                                               3604
      GAMMA(12,7)=OFSET(5)                                                3605
      GAMMA(10,8)=OFSET(6)                                                3606
      GAMMA(12,8)=-OFSET(4)                                               3607
      GAMMA(10,9)=-OFSET(5)                                               3608
      GAMMA(11,9)=OFSET(4)                                                3609
C                                                                         3610
      DO 500 I=1,6                                                        3611
      DO 500 J=1,6                                                        3612
      GAM1(I,J)=GAMMA(I,J)                                                3613
  500 GAM2(I,J)=GAMMA(I+6,J+6)                                            3614
C                                                                         3615
      RETURN                                                              3616
      END                                                                 3617
```

SUBROUTINE CSTIF

```
$IBFTC CSTIF*  DECK
      SUBROUTINE CSTIF (AK,FLX,SKAB)                                    3171
C                                                                       3172
C     CURVED BEAM STIFFNESS                                             3173
C                                                                       3174
      DIMENSION AK(12,12),FLX(6,6),SKAA(6,6),SKAB(6,6),S(22),           3175
     1TMAB(6,6),TMABT(6,6)                                              3176
C                                                                       3177
      COMMON/TAPES/MT1,MT2,MT3,MT4,MT5,MT6,MT7,MT8,MT9,MT10,MT11,MT12,
     * MT13,MT14,MT15,MT16,MT17
      COMMON/VAR/NPTS,FNPTS                                             3181
      COMMON/ADPRO/EM,G,RC,ALFA,DARC,DL                                 3182
      COMMON/PROPT/YI(100),ZA(100),XA(100),ZI(100),YA(100),GJ(100)      3183
C                                                                       3184
      IOUT = MT6
      DO 10 I=1,22                                                      3185
   10 S(I)=0.0                                                          3186
C                                                                       3187
      REM=DARC/EM                                                       3188
      RG=DARC/G                                                         3189
      RDE=RC*DARC/EM                                                    3190
      DSIN=SIN(ALFA/FNPTS)                                              3191
      DCOS=COS(ALFA/FNPTS)                                              3192
C     FLEXIBILITY MATRIX AT B                                           3193
C                                                                       3194
      DO 40 I=1,NPTS                                                    3195
C                                                                       3196
      IF(I.NE.1)GO TO 20                                                3197
C                                                                       3198
      SSIN=SIN(ALFA/(2.0*FNPTS))                                        3199
      SCOS=COS(ALFA/(2.0*FNPTS))                                        3200
      GO TO 30                                                          3201
C                                                                       3202
   20 SSIN=ESIN*DCOS+ECOS*DSIN                                          3203
      SCOS=ECOS*DCOS-ESIN*DSIN                                          3204
C                                                                       3205
   30 CONTINUE                                                          3206
      SINSQ=SSIN**2                                                     3207
      COSSQ=SCOS**2                                                     3208
      SICO=SSIN*SCOS                                                    3209
      SCOS1=(1.0-SCOS)                                                  3210
C                                                                       3211
      S(1)=S(1)+COSSQ/XA(I)                                             3212
      S(2)=S(2)+SINSQ/YA(I)                                             3213
      S(3)=S(3)+SCOS1**2/ZI(I)                                          3214
      S(4)=S(4)+SICO/XA(I)                                              3215
      S(5)=S(5)+SICO/YA(I)                                              3216
      S(6)=S(6)+(SCOS1*SSIN)/ZI(I)                                      3217
      S(7)=S(7)+SCOS1/ZI(I)                                             3218
      S(8)=S(8)+SINSQ/XA(I)                                             3219
      S(9)=S(9)+COSSQ/YA(I)                                             3220
      S(10)=S(10)+SINSQ/ZI(I)                                           3221
      S(11)=S(11)+SSIN/ZI(I)                                            3222
      S(12)=S(12)+1.0/ZA(I)                                             3223
      S(13)=S(13)+SCOS1**2/GJ(I)                                        3224
      S(14)=S(14)+SINSQ/YI(I)                                           3225
      S(15)=S(15)+(SCOS1*SCOS)/GJ(I)                                    3226
```

```
      S(16)=S(16)+(SCOS1*SSIN)/GJ(I)                                    3227
      S(17)=S(17)+SICO/YI(I)                                            3228
      S(18)=S(18)+COSSQ/GJ(I)                                           3229
      S(19)=S(19)+SICO/GJ(I)                                            3230
      S(20)=S(20)+SINSQ/GJ(I)                                           3231
      S(21)=S(21)+COSSQ/YI(I)                                           3232
      S(22)=S(22)+1.0/ZI(I)                                             3233
C                                                                       3234
      ESIN=SSIN                                                         3235
      ECOS=SCOS                                                         3236
C                                                                       3237
   40 CONTINUE                                                          3238
C                                                                       3239
      DO 50 I=1,6                                                       3240
      DO 50 J=1,6                                                       3241
      FLX(J,I)=0.0                                                      3242
      SKAA(J,I)=0.0                                                     3243
      SKAB(J,I)=0.0                                                     3244
      TMAB(J,I)=0.0                                                     3245
      TMABT(J,I)=0.0                                                    3246
   50 CONTINUE                                                          3247
C                                                                       3248
      FLX(1,1)=S(18)*RG+S(14)*REM                                       3249
      FLX(1,2)=S(19)*RG-S(17)*REM                                       3250
      FLX(1,6)=-S(15)*RC*RG+S(14)*RDE                                   3251
      FLX(2,2)=S(20)*RG+S(21)*REM                                       3252
      FLX(2,6)=-S(16)*RC*RG-S(17)*RDE                                   3253
      FLX(3,3)=S(22)*REM                                                3254
      FLX(3,4)=-S(7)*RDE                                                3255
      FLX(3,5)=S(11)*RDE                                                3256
      FLX(4,4)=S(1)*REM+S(2)*RG+S(3)*RC*RDE                             3257
      FLX(4,5)=S(4)*REM-S(5)*RG-S(6)*RC*RDE                             3258
      FLX(5,5)=S(8)*REM+S(9)*RG+S(10)*RC*RDE                            3259
      FLX(6,6)=S(12)*RG+S(13)*RC**2*RG+S(14)*RC*RDE                     3260
C                                                                       3261
      DO 60 I=1,6                                                       3262
      DO 60 J=I,6                                                       3263
   60 FLX(J,I)=FLX(I,J)                                                 3264
C                                                                       3265
C     INVERT FLX TO GET STIFFNESS AT B                                  3266
C                                                                       3267
      CALL TINVR(FLX,6,IND)                                             3268
C                                                                       3269
      IF(IGD.NE.0)GO TO 80                                              3271
      WRITE(IOUT,9000)                                                  3272
 9000 FORMAT(31H1FLEXIBILITY MATRIX IS SINGULAR)                        3273
   80 CONTINUE                                                          3274
C                                                                       3275
      SSIN=SIN(ALFA)                                                    3276
      SCOS=COS(ALFA)                                                    3277
      TMAB(1,1)=SCOS                                                    3278
      TMAB(1,2)=SSIN                                                    3279
      TMAB(1,6)=-RC*(1.0-SCOS)                                          3280
      TMAB(2,1)=-SSIN                                                   3281
      TMAB(2,2)=SCOS                                                    3282
      TMAB(2,6)=-RC*SSIN                                                3283
      TMAB(3,3)=1.0                                                     3284
      TMAB(3,4)=TMAB(1,6)                                               3285
      TMAB(3,5)=-TMAB(2,6)                                              3286
      TMAB(4,4)=SCOS
```

```
      TMAB(4,5)=SSIN                                                    3287
      TMAB(5,4)=-SSIN                                                   3288
      TMAB(5,5)=SCOS                                                    3289
      TMAB(6,6)=1.0                                                     3290
C                                                                       3291
      CALL SMULT(TMAB,FLX,SKAB,6,6,6)                                   3292
C                                                                       3293
      DO 90 I=1,6                                                       3294
      DO 90 J=1,6                                                       3295
   90 TMABT(J,I)=TMAB(I,J)                                              3296
C                                                                       3297
      CALL SMULT(SKAB,TMABT,SKAA,6,6,6)                                 3298
C                                                                       3299
      DO 100 I=1,6                                                      3300
      DO 100 J=1,6                                                      3301
      AK(I,J)=SKAA(I,J)                                                 3302
      AK(I,J+6)=SKAB(I,J)                                               3303
      AK(I+6,J)=SKAB(J,I)                                               3304
      AK(I+6,J+6)=FLX(I,J)                                              3305
  100 CONTINUE                                                          3306
C                                                                       3307
      RETURN                                                            3308
      END                                                               3309
```

SUBROUTINE CBMTR

```
$IBFTC CBMTR*  DECK
      SUBROUTINE CBMTR(XI,YI,ZI,XF,YF,ZF,XC,YC,ZC,ALAMA,ALAMB,CLAM)          3402
C                                                                            3403
C     CURVE BEAM TRANSFORMATION MATRIX                                       3404
C                                                                            3405
C                                                                            3406
      DIMENSION ALAM(3,3),ALAMA(6,6),ALAMB(6,6),CLAM(12,12),                 3407
     1RCP(3,3),RCN(3,3), RCF(3,3)                                            3408
C                                                                            3409
      COMMON/TAPES/MT1,MT2,MT3,MT4,MT5,MT6,MT7,MT8,MT9,MT10,MT11,MT12,
     * MT13,MT14,MT15,MT16,MT17
      COMMON/VAR/NPTS,FNPTS                                                  3413
      COMMON/ADPRO/EM,G,RC,ALFA,DARC,DL                                      3414
      COMMON/LTRAN/L1,M1,N1,L2,M2,N2,L3,M3,N3                                3415
      REAL L1,M1,N1,L2,M2,N2,L3,M3,N3                                        3416
C                                                                            3417
      IOUT = MT6
      XFI=XF-XI                                                              3418
      YFI=YF-YI                                                              3419
      ZFI=ZF-ZI                                                              3420
      XCI=XC-XI                                                              3421
      YCI=YC-YI                                                              3422
      ZCI=ZC-ZI                                                              3423
      DS=SQRT(XFI**2+YFI**2)                                                 3424
      DL=SQRT(XFI**2+YFI**2+ZFI**2)                                          3425
      F=(XCI*XFI+YCI*YFI)/DS                                                 3426
      B=-(XCI*YFI-YCI*XFI)/DS                                                3427
      D=(-F*ZFI+ZCI*DS)/DL                                                   3428
      DIV=SQRT(D*D+B*B)                                                      3429
      DIVR=SQRT(XCI**2+YCI**2)                                               3430
C                                                                            3431
      DO 5 I=1,3                                                             3432
      DO 5 J=1,3                                                             3433
   10 ALAM(I,J)=0.0                                                          3434
      RCN(I,J)=0.0                                                           3435
    5 RCP(I,J)=0.0                                                           3436
C                                                                            3437
      RCN(1,1)=1.0                                                           3438
      RCN(2,3)=1.0                                                           3439
      RCN(3,2)=-1.0                                                          3440
C                                                                            3441
      RCP(1,1)=1.0                                                           3442
      RCP(2,3)=-1.0                                                          3443
      RCP(3,2)=1.0                                                           3444
C                                                                            3445
C                                                                            3446
      IF(DS)20,40,60                                                         3447
C                                                                            3448
   20 WRITE(IOUT,9000)XI,YI,ZI,XF,YF,ZF                                      3449
 9000 FORMAT(19H1LENGTH IS NEGATIVE///(6E10.3))                              3450
C                                                                            3451
   40 CONTINUE                                                               3452
      SINB=YCI/DIVR                                                          3453
      COSB=XCI/DIVR                                                          3454
      ALAM(1,3)=ZFI/DL                                                       3455
      ALAM(2,1)=SINB*(ZFI/DL)                                                3456
      ALAM(2,2)=COSB*(ZFI/DL)                                                3457
```

```
      ALAM(3,1)=COSB                                                    3458
      ALAM(3,2)=SINB                                                    3459
      GO TO 65                                                          3460
C                                                                       3461
   60 CONTINUE                                                          3462
      COSB=D/DIV                                                        3463
      SINB=B/DIV                                                        3464
      ALAM(1,1)=XFI/DL                                                  3465
      ALAM(1,2)=YFI/DL                                                  3466
      ALAM(1,3)=ZFI/DL                                                  3467
      ALAM(2,1)=-COSB*(YFI/DS)+SINB*(ZFI/DL)*(XFI/DS)                   3468
      ALAM(2,2)=COSB*(XFI/DS)+SINB*(ZFI/DL)*(YFI/DS)                    3469
      ALAM(2,3)=-SINB*(DS/DL)                                           3470
      ALAM(3,1)=-SINB*(YFI/DS)-COSB*(ZFI/DL)*(XFI/DS                    3471
      ALAM(3,2)=SINB*(XFI/DS)-COSB*(ZFI/DL)*(YFI/DS)                    3472
      ALAM(3,3)=COSB*(DS/DL)                                            3473
C                                                                       3474
   65 CONTINUE                                                          3475
C                                                                       3476
      IF(RC.LT.0.0)GO TO 80                                             3477
      CALL MULT (RCP,ALAM,RCF,3,3,3,1,0)                                3478
      GO TO 85                                                          3479
   80 CALL MULT(RCN,ALAM,RCF,3,3,3,1,0)                                 3480
   85 CONTINUE                                                          3481
C                                                                       3482
      L1=RCF(1,1)                                                       3483
      L2=RCF(2,1)                                                       3484
      L3=RCF(3,1)                                                       3485
      M1=RCF(1,2)                                                       3486
      M2=RCF(2,2)                                                       3487
      M3=RCF(3,2)                                                       3488
      N1=RCF(1,3)                                                       3489
      N2=RCF(2,3)                                                       3490
      N3=RCF(3,3)                                                       3491
C                                                                       3492
      DO 110 I=1,6                                                      3493
      DO 110 J=1,6                                                      3494
      ALAMA(I,J)=0.0                                                    3495
  110 ALAMB(I,J)=0.0                                                    3496
C                                                                       3497
      DO 115 I=1,12                                                     3498
      DO 115 J=1,12                                                     3499
  115 CLAM(I,J)=0.0                                                     3500
C                                                                       3501
      RC=ABS(RC)                                                        3502
      ALFA = DL/(2.0*RC)                                                3503
      ALFA=ASIN(ALFA)                                                   3504
      CALFA=COS(ALFA)                                                   3505
      SALFA=SIN(ALFA)                                                   3506
      DARC=2.0*(RC*ALFA/FNPTS)                                          3507
      DL=DARC*FNPTS                                                     3508
      ALFA=2.0*ALFA                                                     3509
C                                                                       3510
C                                                                       3511
      ALAMA(1,1)=L1*CALFA+L2*SALFA                                      3512
      ALAMA(1,2)=M1*CALFA+M2*SALFA                                      3513
      ALAMA(1,3)=N1*CALFA+N2*SALFA                                      3514
      ALAMA(2,1)=-L1*SALFA+L2*CALFA                                     3515
      ALAMA(2,2)=-M1*SALFA+M2*CALFA                                     3516
      ALAMA(2,3)=-N1*SALFA+N2*CALFA                                     3517
```

```
      ALAMA(3,1)=L3                                                     3518
      ALAMA(3,2)=M3                                                     3519
      ALAMA(3,3)=N3                                                     3520
C                                                                       3521
      ALAMB(1,1)=L1*CALFA-L2*SALFA                                      3522
      ALAMB(1,2)=M1*CALFA-M2*SALFA                                      3523
      ALAMB(1,3)=N1*CALFA-N2*SALFA                                      3524
      ALAMB(2,1)=L1*SALFA+L2*CALFA                                      3525
      ALAMB(2,2)=M1*SALFA+M2*CALFA                                      3526
      ALAMB(2,3)=N1*SALFA+N2*CALFA                                      3527
      ALAMB(3,1)=L3                                                     3528
      ALAMB(3,2)=M3                                                     3529
      ALAMB(3,3)=N3                                                     3530
C                                                                       3531
      DO 120 I=1,3                                                      3532
      DO 120 J=1,3                                                      3533
      ALAMA(I+3,J+3)=ALAMA(I,J)                                         3534
      ALAMB(I+3,J+3)=ALAMB(I,J)                                         3535
  120 CONTINUE                                                          3536
C                                                                       3537
      DO 130 I=1,6                                                      3538
      DO 130 J=1,6                                                      3539
      CLAM(I,J)=ALAMA(I,J)                                              3540
      CLAM(I+6,J+6)=ALAMB(I,J)                                          3541
  130 CONTINUE                                                          3542
C                                                                       3543
      RETURN                                                            3544
      END                                                               3545
```

SUBROUTINE MERGE

```
$IBFTC MERGE*  DECK
      SUBROUTINE MERGE                                                  3828
C     CONTROL SECTION FOR MERGE AND BOUNDARY                            3829
      COMMON/CONT1/JPART(800)                                           3830
      COMMON/CONT2/KPART(800)                                           3831
      COMMON/CONT3/LPART(800)                                           3832
      COMMON/LASTND/LN(200)                                             3833
      COMMON/CONTRL/NDEFL,NKSP,NREX,NNF,NPSTR,NBSTR,NVIB                3834
      COMMON/TERMS/NBEAM,NPLATE,NNODE,NCOND,NPS,NTOL,NP,NOPT(4)
      COMMON/COMS/NSIZE(200)                                            3836
      COMMON/SKIP/NBSP,NBSB,NBSPI,NBSBI                                 3837
      COMMON/TAPES/MT1,MT2,MT3,MT4,MT5,MT6,MT7,MT8,MT9,MT10,MT11,MT12,
     * MT13,MT14,MT15,MT16,MT17
      COMMON/REDUC/NTEST,NTEST2
C                                                                       3841
C                                                                       3842
      IOUT = MT6
      ISTIF = MT2
      IKBC = MT3
      IKDF = MT4
      KFF = MT11
      KSTRES = MT8
C                                                                       3843
C     SPACING FOR TAPES LOGIC 2 AND LOGIC 15                            3844
C     NBSP=NUMBER OF LOGICAL RECORDS FOR PLATE STRESS                   3845
      NBSP=NPLATE*8                                                     3846
C     NBSPI=NUMBER OF LOGICAL RECORDS FOR PLATE INTERNAL LOADS          3847
      NBSPI=NPLATE*2                                                    3848
C     NBSB=NUMBER OF LOGICAL RECORDS FOR BEAM STRESS                    3849
      NBSB=NBEAM*4                                                      3850
C     NBSBI=NUMBER OF LOGICAL RECORDS FOR BEAM INTERNAL LOADS           3851
      NBSBI=NBEAM*3                                                     3852
C                                                                       3853
      REWIND ISTIF                                                      3854
      REWIND IKBC                                                       3855
      REWIND IKDF                                                       3856
      REWIND KFF                                                        3857
      REWIND KSTRES                                                     3860
C                                                                       3861
      CALL SOR (JPART,KPART,LPART,800)                                  3863
C                                                                       3865
C                                                                       3866
      CALL MERGBC                                                       3867
      IF (NOPT(3) .NE. NTEST2) CALL STRESS
C                                                                       3870
      REWIND ISTIF                                                      3871
      REWIND IKBC                                                       3872
      REWIND IKDF                                                       3873
      REWIND KFF                                                        3874
      REWIND KSTRES                                                     3877
C                                                                       3878
 1000 RETURN                                                            3881
      END                                                               3882
```

SUBROUTINE SOR

```
$IBFTC SOR*     DECK
       SUBROUTINE SOR (JPART,KPART,LPART,NPS)                          3885
C                                                                      3886
C      SORTS THE CONNECTIVITY DATA SO THAT JPART                       3887
C      ARRAY STARTS WITH THE FIRST PARTITION                           3888
C      AND GOES THRU EVERY ROW OF PARTITIONS                           3889
C      CONSECUTIVELY***1001,2001,2002,3001,ETC.***                     3890
C                                                                      3891
       DIMENSION JPART(1),KPART(1),LPART(1)                            3892
       NPS1=NPS-1                                                      3893
       DO 40 I=1,NPS1                                                  3894
       IF(JPART(I).EQ.0)GO TO 40                                       3895
       MIN=JPART(I)                                                    3896
       K=I                                                             3897
       DO 20 M=I,NPS1                                                  3898
       IF(JPART(M+1).EQ.0)GO TO 20                                     3899
       IF(MIN.LT.JPART(M+1))GO TO 20                                   3900
       MIN=JPART(M+1)                                                  3901
       K=M+1                                                           3902
    20 CONTINUE                                                        3903
       JPART(K)=JPART(I)                                               3904
       JPART(I)=MIN                                                    3905
       MIN1=KPART(K)                                                   3906
       KPART(K)=KPART(I)                                               3907
       KPART(I)=MIN1                                                   3908
       MIN2=LPART(K)                                                   3909
       LPART(K)= LPART(I)                                              3910
       LPART(I)=MIN2                                                   3911
       K=I +1                                                          3912
    40 CONTINUE                                                        3913
       RETURN                                                          3914
       END                                                             3915
```

SUBROUTINE MERGBC

```
$IBFTC MERGB*  DECK
       SUBROUTINE MERGBC                                                3919
                                                                        3920
C                                                                       3921
C      MERGE AND BOUNDARY OF STIFFNESS                                  3922
C                                                                       3923
       COMMON/CONT1/JPART(800)                                          3924
       COMMON/CONT2/KPART(800)                                          3925
       COMMON/CONT3/LPART(800)                                          3926
       COMMON/COMS/NSIZE(200)                                           3927
       COMMON/LASTND/LN(200)                                            3928
       COMMON/TERMS/NBEAM,NPLATE,NNODE,NCOND,NPS,NTOL,NP                3929
       COMMON/CONTRL/NDEFL,NKSP,NREX,NNF,NPSTR,NBSTR,NVIB,IF88
       COMMON/TAPES/MT1,MT2,MT3,MT4,MT5,MT6,MT7,MT8,MT9,MT10,MT11,MT12,
      * MT13,MT14,MT15,MT16,MT17
                                                                        3933
C                                                                       3934
       DIMENSION TEMP(  60),ID(60),SPRING(60),                          3935
      1NNI1(3),NNJ1(3),NDIAG(3)                                         3936
      2,FKK(6),JKK(6),IDD(60),MNN(4)                                    3937
       DIMENSION NT(3),NB(3),NR(3),NL(3),ELEM(60,60,3),A(6,6,4),        3938
      1B(6,6,16),JPAR(3),KPAR(3),LPAR(3),ICODE(200,10)                  3939
       DIMENSION ARRAYB(144),ARRAYP(576),IDRRY(12)                      3940
       EQUIVALENCE(ARRAYB,A),(ARRAYP,B)                                 3941
C                                                                       3942
C      0  FREE                                                          3943
C      1  ZERO DEFLECTION                                               3944
C      2  SPECIFIED DEFLECTION                                          3945
C      3  SPRUNG                                                        3946
C                                                                       3947
C      NVIB IS A PRINT OPTION                                           3948
       INTEGER QZ,QP                                                    3949
       LOGICAL NFIND,NSPR                                               3950
C
       IOUT = MT6
       QP = MT8
       QZ = MT2
       KFF = MT11
       IKBC = MT3
       REWIND QP                                                        3951
       REWIND QZ                                                        3952
       REWIND KFF                                                       3953
       REWIND IKBC                                                      3956
       PRINT 6969                                                       3957
C                                                                       3958
       ZERO=0.0                                                         3959
       DO 2 I=1,200                                                     3962
     2 NSIZE(I)=0                                                       3963
C                                                                       3964
       IDI=0                                                            3965
       NPM=0                                                            3966
       IMK=0                                                            3967
       L1=1                                                             3968
       L2=2                                                             3969
       L3=3                                                             3970
C                                                                       3971
C                                                                       3972
     4 CONTINUE                                                         3973
C                                                                       3974
```

```
C                                                                       3975
C     WORK WITH THREE PARTITIONS                                        3976
C****** DETERMINE FIRST AND LAST BEAMS AND PLATES IN THIS SET.
      JPAR(1)=JPART(L1)                                                 3977
      JPAR(2)=JPART(L2)                                                 3978
      JPAR(3)=JPART(L3)                                                 3979
C****** KPART(K)=I00J MEANS THE I-TH BEAM IS THE FIRST
C****** BEAM CONNECTED TO A NODE IN THE K-TH PARTITION,
C****** AND THE J-TH BEAM IS THE LAST BEAM IN THE K-TH
C****** PARTITION. LPART IS A SIMILAR ARRAY FOR PLATES.
      KMIN=KPART(L1)/10000                                              3980
      LMIN=LPART(L1)/10000                                              3981
      KMAX=KPART(L1)-10000*KMIN                                         3982
      LMAX=LPART(L1)-10000*LMIN                                         3983
      DO 20 I=L2,L3                                                     3984
      KF=KPART(I)/10000                                                 3985
      KL=KPART(I)-10000*KF                                              3986
      LF=LPART(I)/10000                                                 3987
      LL=LPART(I)-10000*LF                                              3988
      IF(KL.GT.KMAX)KMAX=KL                                             3989
      IF(LL.GT.LMAX)LMAX=LL                                             3990
      IF((KF.NE.0).AND.(KF.LT.KMIN))KMIN=KF                             3991
      IF((LF.NE.0).AND.(LF.LT.LMIN))LMIN=LF                             3992
      IF(KMIN.EQ.0)KMIN=KF                                              3993
      IF(LMIN.EQ.0)LMIN=LF                                              3994
   20 CONTINUE                                                          3995
C                                                                       3996
C****** ZERO OUT AND SIZE THE THREE PARTITIONS                         MERGBC
      DO 8 K=1,3                                                        3998
C                                                                       3999
      NI=JPAR (K)/1000                                                  4000
      NJ=JPAR (K)-NI*1000                                               4001
      NNI1(K)=NI                                                        4002
      NNJ1(K)=NJ                                                        4003
      IF(NI.NE.0)GO TO 7                                                4004
      NB(K)=0                                                           4005
      NT(K)=0                                                           4006
      NR(K)=0                                                           4007
      NL(K)=0                                                           4008
      GO TO 9                                                           4009
    7 CONTINUE                                                          4010
      IF(NI.NE.1)NT(K)=LN(NI-1)+1                                       4011
      IF(NI.EQ.1)NT(K)=1                                                4012
      NB(K)=LN(NI)                                                      4013
      IF(NJ.NE.1)NL(K)=LN(NJ-1)+1                                       4014
      IF(NJ.EQ.1)NL(K)=1                                                4015
      NR(K)=LN(NJ)                                                      4016
      NRL=(NR(K)-NL(K)+1)*6                                             4017
      NBT=(NB(K)-NT(K)+1)*6                                             4018
      DO 8 J=1,NRL                                                      4019
      DO 8 M=1,NBT                                                      4020
      ELEM(M,J,K)=0.                                                    4021
    8 CONTINUE                                                          4022
    9 CONTINUE                                                          4023
C                                                                       4025
      IF(NVIB.NE.1)GO TO 8100                                           4026
      WRITE(IOUT,9600)(NT(K),NB(K),NR(K),NL(K),K=1,3)                   4027
 8100 CONTINUE                                                          4028
C                                                                       4029
C                                                                       4030
```

```
      IF(NBEAM.EQ.0)GO TO 1004                                          4031
C                                                                       4032
C                                                                       4033
C****** SPACE TO FIRST BEAM IN THIS SET OF PARTITIONS
      KB=0                                                              4034
      IF(KMIN-1)1004,22,23                                              4035
   23 KMIN1=(KMIN-1)*2                                                  4036
      DO 21 ISPA=1,KMIN1                                                4037
      KB=KB+1                                                           4038
   21 READ(QZ)DUMMY                                                     4039
      KB=KB/2                                                           4040
   22 CONTINUE                                                          4041
      PRINT 6971,(JPAR(IMT),IMT=1,3),KMIN,KMAX                          4042
C     READING BEAM STIFFNESS                                            4043
   15 READ(QZ)N1,N2                                                     4044
      NFIND=.FALSE.                                                     4045
      KB=KB+1                                                           4046
C                                                                       4047
C****** TEST ALL COMBINATIONS OF NODES
      DO 1000 I=1,4                                                     4048
      GO TO(100,200,300,400),I                                          4049
  100 M1=N1                                                             4050
      M2=N1                                                             4051
      GO TO 500                                                         4052
  200 M1=N1                                                             4053
      M2=N2                                                             4054
      GO TO 500                                                         4055
  300 M1=N2                                                             4056
      M2=N1                                                             4057
      GO TO 500                                                         4058
  400 M1=N2                                                             4059
      M2=N2                                                             4060
  500 CONTINUE                                                          4061
C                                                                       4062
      DO 800 L=1,3                                                      4063
      IF((M1.GT.NB(L)).OR.(M1.LT.NT(L))) GO TO 800                      4064
      IF((M2.GT.NR(L)).OR.(M2.LT.NL(L))) GO TO 800                      4065
C                                                                       4066
C****** READ STIFFNESS, IF NECESSARY
      IF(NFIND) GO TO 501                                               4067
      NFIND=.TRUE.                                                      4068
      READ(QZ)(ARRAYB(MJ),MJ=1,144)                                     4069
  501 CONTINUE                                                          4070
C     ADDING IN BEAM STIFFNESSES                                        4071
      DO 700 N=1,6                                                      4072
      KROW=(M1-NT(L))*6+N                                               4073
      DO 700 M=1,6                                                      4074
      KCOL=(M2-NL(L))*6+M                                               4075
  700 ELEM(KROW,KCOL,L)=ELEM(KROW,KCOL,L)+A(N,M,I)                      4076
      GO TO 1000                                                        4077
  800 CONTINUE                                                          4078
C                                                                       4079
 1000 CONTINUE                                                          4080
      IF(.NOT.NFIND) READ(QZ)DUMMY                                      4081
C                                                                       4082
C****** LOOP FOR ALL BEAMS IN THIS SET
      IF(KB.LT.KMAX)GO TO 15                                            4085
C                                                                       4086
      REWIND QZ                                                         4087
C                                                                       4088
```

```
 1004 CONTINUE                                                          4089
      IF(NPLATE.EQ.0)GO TO 4000                                         4090
C                                                                       4091
C                                                                       4092
C****** SPACE DOWN TAPE TO FIRST PLATE IN THIS SET.
      NQ=0                                                              4093
      IF(LMIN-1)4000,25,24                                              4094
   24 LMIN1=(LMIN-1)*2                                                  4095
      DO 26 ISPA=1,LMIN1                                                4096
      NQ=NQ+1                                                           4097
   26 READ(QP)DUMMY                                                     4098
      NQ=NQ/2                                                           4099
   25 CONTINUE                                                          4100
      PRINT 6972,LMIN,LMAX                                              4101
C     READING PLATE STIFFNESS                                           4102
 1005 READ(QP)(MNN(I),I=1,4)                                            4103
C****** TEST ALL COMBINATIONS OF NODES.
      NFIND=.FALSE.                                                     4104
      NQ=NQ+1                                                           4105
C                                                                       4106
      MLT=4                                                             4107
      IF(MNN(4).EQ.0)MLT=3                                              4108
C                                                                       4109
      DO 2900 I=1,MLT                                                   4110
      M1=MNN(I)                                                         4111
      DO 2900 J=1,MLT                                                   4112
      M2=MNN(J)                                                         4113
C                                                                       4114
      DO 2800 L=1,3                                                     4115
      IF((M1.GT.NB(L)).OR.(M1.LT.NT(L)))GO TO 2800                      4116
      IF((M2.GT.NR(L)).OR.(M2.LT.NL(L))) GO TO 2800                     4117
C****** READ PLATE STIFFNESS, IF NECESSARY.
      IF(NFIND) GO TO 2901                                              4118
      NFIND=.TRUE.                                                      4119
      READ(QP) (ARRAYP(MJ),MJ=1,576)                                    4120
 2901 CONTINUE                                                          4121
C                                                                       4122
C     ADDING IN PLATE STIFFNESSES                                       4123
      DO 2700 N=1,6                                                     4124
      KROW=(M1-NT(L))*6+N                                               4125
      DO 2700 M=1,6                                                     4126
      KCOL=(M2-NL(L))*6+M                                               4127
      NII=4*(I-1)+J                                                     4128
 2700 ELEM(KROW,KCOL,L)=ELEM(KROW,KCOL,L)+B(N,M,NII)                    4129
      GO TO 2900                                                        4130
 2800 CONTINUE                                                          4131
C                                                                       4132
 2900 CONTINUE                                                          4133
      IF(.NOT.NFIND) READ (QP) DUMMY                                    4134
C                                                                       4135
C****** LOOP FOR ALL PLATES IN THIS SET.
      IF(NQ.LT.LMAX) GO TO 1005                                         4138
C                                                                       4139
      REWIND QP                                                         4140
C                                                                       4141
 4000 CONTINUE                                                          4142
C                                                                       4143
C                                                                       4145
C                                                                       4146
      PRINT 6973                                                        4147
```

```
C      BOUNDARY CONDITIONS                                                    4148
C                                                                             4149
C                                                                             4150
C                                                                             4151
C****** LOOP FOR 3 PARTITIONS.
       DO 7300 N=1,3                                                          4152
C                                                                             4153
       IF(JPAR(N).EQ.0) GO TO 7300                                            4154
       NDIAG(N)=0                                                             4155
       NUNC=0                                                                 4156
       NUNR=0                                                                 4157
       NBOT=NB(N)-NT(N)+1                                                     4158
       NBOTF=NBOT*6                                                           4159
       NBOTF1=NBOTF-1                                                         4160
       NRL=NR(N)-NL(N)+1                                                      4161
       NRLF=NRL*6                                                             4162
       NRLF1=NRLF-1                                                           4163
       IF(IMK.EQ.NNI1(N))GO TO 7008                                           4164
       K=1                                                                    4165
       NI=NNI1(N)                                                             4166
C                                                                             4167
       DO 7007 I=1,NBOT                                                       4168
       READ(IKBC)M,IJKLMN                                                     4169
       CALL UNPACK(IJKLMN,JKK(1),JKK(2),JKK(3),JKK(4),JKK(5),JKK(6))          4170
       ICODE(NI,I)=IJKLMN                                                     4171
       NSPR = .FALSE.                                                         4172
       DO 7005 L=1,6                                                          4174
       IF(JKK(L).NE.3) GO TO 7005                                             4175
       NSPR = .TRUE.                                                          4176
       READ(IKBC)(FKK(JJ),JJ=1,6)                                             4177
       GO TO 7006                                                             4178
 7005 CONTINUE                                                                4179
 7006 CONTINUE                                                                4180
       DO 107 L=1,6                                                           4181
       M=L+K-1                                                                4182
       IF (.NOT.NSPR) GO TO 107                                               4183
       SPRING(M)=FKK(L)                                                       4184
  107  ID(M)=JKK(L)                                                           4185
       K=K+6                                                                  4186
 7007 CONTINUE                                                                4187
C                                                                             4188
C                                                                             4190
 7008 CONTINUE                                                                4191
C                                                                             4192
       DO 7009 L=1,60                                                         4193
 7009 IDD(L)=ID(L)                                                            4194
C****** NDIAG(N)=1, IF THE N-TH PARTITION IN THIS SET
C****** IS A DIAGONAL ONE.
C                                                                             4195
       IF(NNI1(N).EQ.NNJ1(N))NDIAG (N)=1                                      4196
       IK=0                                                                   4197
       IFIX=0                                                                 4198
       DO 7010 I=1,NBOTF                                                      4199
       IF((ID(I).EQ.1).OR.(ID(I).EQ.2))IK=IK+1                                4200
 7010 CONTINUE                                                                4201
       IF(IK.EQ.NBOTF)IFIX=1                                                  4202
       IF(NDIAG(N).NE.1)GO TO 7030                                            4203
       IDI=IDI+1                                                              4204
       NSIZE(IDI)=NBOTF-IK                                                    4205
       IF(.NOT. NSPR) GO TO 7030
```

```
C      ADD SPRING CONSTANTS TO DIAGONAL TERMS                                  4207
       DO 7025 I=1,NBOTF                                                       4208
       IF(ID(I).NE.3)GO TO 7025                                                4209
       ELEM(I,I,N)=ELEM(I,I,N)+SPRING(I)                                       4210
 7025 CONTINUE                                                                 4211
 7030 CONTINUE                                                                 4212
C                                                                              4213
C****** SORT CONSTRAINED ELEMENTS TO BOTTOM.
       DO 7035 I=1,NBOTF                                                       4214
       II=NBOTF-I+1                                                            4215
       IF((ID(II).EQ.1).OR.(ID(II).EQ.2))GO TO 7038                            4216
 7035 CONTINUE                                                                 4217
 7038 ILAST=II                                                                 4218
       JJJ=0                                                                   4219
C                                                                              4220
C                                                                              4221
       DO 7150 I=1,ILAST                                                       4222
C                                                                              4223
       IJK=0                                                                   4224
       IF((ID(I).NE.1).AND.(ID(I).NE.2))GO TO 7150                             4225
       IJK=1                                                                   4226
       IF(JJJ.EQ.0)IJK=0                                                       4227
       JJJ=JJJ+1                                                               4228
       KK=I-JJJ+1                                                              4229
       NUNR=NUNR+1                                                             4230
       IF(NDIAG(N).NE.1)GO TO 7044                                             4231
       NUNC=NUNR                                                               4232
       GO TO 7050                                                              4233
 7044 CONTINUE                                                                 4234
       ITR=NNJ1(N)                                                             4235
       NUN1=NSIZE(ITR)                                                         4236
       NUNC=NRLF-NUN1                                                          4237
 7050 CONTINUE                                                                 4238
C                                                                              4239
       DO 7060 L=1,NRLF                                                        4240
       TEMP(L)=ELEM(KK,L,N)                                                    4241
 7060 CONTINUE                                                                 4242
C                                                                              4243
       DO 7080 K=KK,NBOTF1                                                     4244
       DO 7080 M=1,NRLF                                                        4245
       ELEM(K,M,N)=ELEM(K+1,M,N)                                               4246
 7080 CONTINUE                                                                 4247
C                                                                              4248
C****** SORT CONSTRAINED ELEMENTS TO RIGHT.
       DO 7090 II=1,NRLF                                                       4249
       ELEM(NBOTF,II,N)=TEMP(II)                                               4250
 7090 CONTINUE                                                                 4251
 7150 CONTINUE                                                                 4252
C                                                                              4253
       NJ=NNJ1(N)                                                              4254
       K=1                                                                     4255
       DO 108 I=1,NRL                                                          4256
       CALL UNPACK(ICODE(NJ,I),ID(K),ID(K+1),ID(K+2),ID(K+3),ID(K+4),          4257
      1 ID(K+5))                                                               4258
  108 K=K+6                                                                    4259
       DO 101 I=1,NRLF                                                         4260
       ILAST=NRLF-I+1                                                          4261
       IF((ID(ILAST).EQ.1).OR.(ID(ILAST).EQ.2)) GO TO 102                      4262
  101 CONTINUE                                                                 4263
  102 JJJ=0                                                                    4264
```

```
      DO 103 I=1,ILAST                                                  4265
      IF((ID(I).NE.1).AND.(ID(I).NE.2)) GO TO 103                       4266
      JJJ=JJJ+1                                                         4267
      KK=I-JJJ+1                                                        4268
      DO 104 L=1,NBOTF                                                  4269
  104 TEMP(L)=ELEM(L,KK,N)                                              4270
      DO 105 K=KK,NRLF1                                                 4271
      DO 105 M=1,NBOTF                                                  4272
  105 ELEM(M,K,N)=ELEM(M,K+1,N)                                         4273
      DO 106 L=1,NBOTF                                                  4274
  106 ELEM(L,NRLF,N)=TEMP(L)                                            4275
  103 CONTINUE                                                          4276
C                                                                       4277
      IF((IK.NE.0).OR.(NDIAG(N).EQ.1))GO TO 7155                        4278
      ITR=NNJ1(N)                                                       4279
      NUN1=NSIZE(ITR)                                                   4280
      NUNC=NRLF-NUN1                                                    4281
 7155 CONTINUE                                                          4282
      IF((IFIX.NE.1).OR.(NDIAG(N).EQ.1))GO TO 7165                      4283
      ITR=NNJ1(N)                                                       4284
      NUN1=NSIZE(ITR)                                                   4285
      NUNC=NRLF-NUN1                                                    4286
 7165 CONTINUE                                                          4287
C                                                                       4288
C                                                                       4289
      NROW=NBOTF-NUNR                                                   4290
      NCOL=NRLF-NUNC                                                    4291
      NBR=NROW+1                                                        4292
      NBC=NCOL+1                                                        4293
C                                                                       4294
C                                                                       4298
C****** WRITE STIFFNESS PARTITIONS ON TAPE.
      IF((IK.EQ.0).AND.(NDIAG(N).EQ.1))GO TO 7280                       4299
      IF((IK.EQ.0).AND.(NDIAG(N).NE.1))GO TO 7168                       4300
      IF((IFIX.NE.1).OR.(NDIAG(N).NE.1))GO TO 7162                      4301
C                                                                       4302
      K8=JPAR(N)+1000000
      K8COL=NRLF                                                        4307
      K8ROW=NBOTF                                                       4308
 8200 CONTINUE                                                          4321
      GO TO 7296                                                        4322
C                                                                       4323
 7162 CONTINUE                                                          4324
      IF((IFIX.NE.1).OR.(NDIAG(N).EQ.1))GO TO 7164                      4325
      ITR=NNJ1(N)                                                       4326
      NUN1=NSIZE(ITR)                                                   4327
      IF(NUN1 .NE. NRLF) GO TO 7163                                     4328
      K8=JPAR(N)+2000000
      K8COL=NRLF                                                        4333
      K8ROW=NBOTF                                                       4334
 8300 CONTINUE                                                          4347
      GO TO 7296                                                        4348
 7163 CONTINUE                                                          4349
 7164 CONTINUE                                                          4350
C                                                                       4351
      IF(NBC.GT.NRLF) GO TO 8400                                        4352
      K88=K88+1                                                         4355
      K8=JPAR(N)+1000000
      K8ROW=NBOTF-NBR+1                                                 4358
      K8COL=NRLF-NBC+1
```

```
 8400 CONTINUE                                                          4371
      IF(NCOL.EQ.0)GO TO 8500
      K8=JPAR(N)+2000000
      K8COL=NCOL                                                        4376
      K8ROW=NBOTF-NBR+1                                                 4377
 8500 CONTINUE                                                          4390
      IF((IFIX.EQ.1).AND.(NDIAG(N).NE.1))GO TO 7296                     4391
C                                                                       4392
 7168 CONTINUE                                                          4393
      IF(NNI1(N).EQ.NNJ1(N))GO TO 7200                                  4394
      IF((IK.NE.0).OR.(NDIAG(N).EQ.1))GO TO 7178                        4395
      ITR=NNJ1(N)                                                       4396
      NUN1=NSIZE(ITR)                                                   4397
      IF(NUN1.NE.NRLF)GO TO 7176                                        4398
      CALL WRTETP(ELEM(1,1,N),60,JPAR(N),NBOTF,NRLF,IDRRY,0,0,KFF,IRROR)
      K8=JPAR(N)+4000000                                                4403
      K8COL=NRLF                                                        4404
      K8ROW=NBOTF                                                       4415
      IF(NVIB.NE.1)GO TO 8600                                           4416
      CALL PRINT(ELEM(1,1,N),K8ROW,K8COL,1,5HSTIFF,0,60)                4417
 8600 CONTINUE                                                          4418
      GO TO 7296                                                        4419
 7176 CONTINUE                                                          4420
C                                                                       4421
 7178 CONTINUE                                                          4422
      IF(NBC.GT.NRLF) GO TO 8700                                        4423
      IFPAR=NNJ1(N)*1000+NNI1(N)                                        4424
C     KCF TERM FOR OFF DIAGONAL PARTITION
      K8=JPAR(N)+3000000                                                4429
      K8COL=NRLF-NBC+1                                                  4430
      K8ROW=NROW                                                        4443
 8700 CONTINUE                                                          4444
 7200 CONTINUE                                                          4445
C                                                                       4446
 7280 CONTINUE                                                          4447
      IF(NCOL.EQ.0)GO TO 8800
      CALL WRTETP(ELEM(1,1,N),60,JPAR(N),NROW,NCOL,IDARRY,0,0,KFF,IRROR)
      K8=JPAR(N)+4000000                                                4452
      K8COL=NCOL                                                        4453
      K8ROW=NROW                                                        4464
      IF(NVIB.NE.1)GO TO 8800                                           4465
      CALL PRINT(ELEM(1,1,N),K8ROW,K8COL,1,5HSTIFF,0,60)                4466
 8800 CONTINUE                                                          4467
C                                                                       4468
 7296 CONTINUE                                                          4469
C                                                                       4470
      DO 7298 L=1,60                                                    4471
 7298 ID(L)=IDD(L)                                                      4472
      IMK=NNI1(N)                                                       4473
C                                                                       4474
C                                                                       4475
C                                                                       4476
 7300 CONTINUE                                                          4477
C                                                                       4478
C                                                                       4479
C                                                                       4480
      NPM=NPM+3                                                         4481
      L1=L1+3                                                           4482
      L2=L2+3                                                           4483
      L3=L3+3
```

```
C                                                                       4484
C       TEST FOR TOTAL NUMBER OF PARITIONS                              4485
        PRINT 6974                                                      4486
        IF(NPM.LT.NTOL)GO TO 4                                          4487
        REWIND KFF                                                      4489
        REWIND IKBC                                                     4492
C****** CONVERT KPART AND LPART ARRAYS TO FORM COMPATIBLE WITH
C****** SUBROUTINE MFORCE.
        DO 30 I=1,NTOL                                                  4496
        KPART(I)=KPART(I)-(KPART(I)/10000)*10000                        4497
   30   LPART(I)=LPART(I)-(LPART(I)/10000)*10000                        4498
        PRINT 6970                                                      4499
C                                                                       4500
 9001 FORMAT(16H JR88 IDENT NO. I6,10X,I2,4H BY I2)
 9600 FORMAT(24H1NT      NB      NR      NL///(4I8))                    4507
 6969 FORMAT(1H ,11HBEGIN MERGE)                                        4508
 6970 FORMAT(1H ,09HEND MERGE)                                          4509
 6971 FORMAT(1H ,7HJPAR = ,3(I6,1H,),7HKMIN = ,I6,7HKMAX = ,I6)         4510
 6972 FORMAT(1H ,7HLMIN = ,I6,7HLMAX = ,I6)                             4511
 6973 FORMAT(1H ,8HSTART BC)                                            4512
 6974 FORMAT(1H ,6HEND BC)                                              4513
C                                                                       4514
 7400 RETURN                                                            4515
        END                                                             4516
```

SUBROUTINE STRESS

```
$IBFTC STRES*  DECK                                                    4849
      SUBROUTINE STRESS                                                4850
C                                                                      4851
C     CONTROL SECTION FOR STRESS CALCULATIONS                          4852
C                                                                      4853
      COMMON/TERMS/NBEAM,NPLATE,NNODE,NCOND,NPS,NTOL,NP                4854
      COMMON/CONTRL/NDEFL,NKSP,NREX,NNF,NPSTR,NBSTR,NVIB               4855
      COMMON/SKIP/NBSP,NBSB,NBSPI,NBSBI
      COMMON/PBSIZE/IPB,IPBL,IPBN,NELEM,NOD
      COMMON/TAPES/MT1,MT2,MT3,MT4,MT5,MT6,MT7,MT8,MT9,MT10,MT11,MT12,
     * MT13,MT14,MT15,MT16,MT17
      COMMON/LOADS/BIG1(96,60)                                         4860
C                                                                      4861
C     LIST OF ARGUMENTS FOR PLATES AND BEAM MERGER                     4862
C                                                                      4866
C     NELEM=   NPLATE       NBEAM                                      4867
C
      IOUT = MT6
      ISTRS = MT16
      REWIND ISTRS
      IF(NPLATE .EQ. 0) GO TO 10                                       4869
C
      IPB = 12
      IPBL = 96                                                        4872
      IPBN=8                                                           4873
      NELEM=NPLATE
      NOD=4
      PRINT 91
      PRINT 92
      CALL MSTRES                                                      4876
   10 CONTINUE                                                         4877
C
      IF(NBEAM .EQ. 0) GO TO 20                                        4879
C
      IPB = 8
      IPBL = 96                                                        4882
      IPBN=12                                                          4883
      NELEM=NBEAM
      NOD=2                                                            4884
C
      PRINT 93
      PRINT 94
      CALL MSTRES                                                      4887
   20 CONTINUE                                                         4888
C                                                                      4889
      RETURN
   91 FORMAT(12H CALL IPLATE)
   92 FORMAT(12H CALL PSTRES)
   93 FORMAT(12H CALL  IBEAM)
   94 FORMAT(12H CALL BSTRES)                                          4890
      END
```

SUBROUTINE MSTRES

```
$IBFTC MSTRE*  DECK
       SUBROUTINE MSTRES
       COMMON/CONTRL/NDEFL,NKSP,NREX,NNF,NPSTR,NBSTR,NVIB                    5195
       COMMON/LASTND/LN(200)                                                 5196
       COMMON/TERMS/NBEAM,NPLATE,NNODE,NCOND,NPS,NTOL,NP                     5197
       COMMON/SKIP/NBSP,NBSB,NBSPI,NBSBI                                     5198
       COMMON/PBSIZE/IPB,IPBL,IPBN,NELEM,NOD
       COMMON/TAPES/MT1,MT2,MT3,MT4,MT5,MT6,MT7,MT8,MT9,MT10,MT11,MT12,
      * MT13,MT14,MT15,MT16,MT17
       COMMON/MAPSTR/IPTOT,IBTOT
C      NOD=2 FOR BEAMS,   4 FOR PLATES.
       DIMENSION BIG(96,60),A(68,8,6),NN(68)
       EQUIVALENCE(A,B)
       DIMENSION B(22,12,6)
       COMMON/LOADS/BIG
       LOGICAL LAST


       IOUT = MT6
       ISTRS = MT16
       KSTRES = MT8
C      INITIALIZE
       NTOT = 0
       IF((NPLATE.EQ.0).OR.(NPSTR.NE.0)) GO TO 11
       DO 10 I=1,NBSPI
   10  READ(ISTRS) NDUM
   11  CONTINUE
       NTIM=NELEM/IPB
       NTIML=NELEM-NTIM*IPB
       NLIM=NTIM
       IF(NTIML.NE.0)NLIM=NLIM+1
       LAST=.FALSE.
C       LOOP FOR SETS OF ELEMENTS
       DO 2000 LPL=1,NLIM
       IF(LPL.GT.NTIM) LAST=.TRUE.
 2001  MLIM=IPB
       IF(LAST)MLIM=NTIML
C       READ IN STRESSES
       DO 3100 K=1,MLIM
       KN1=K+MLIM
       KN2=KN1+MLIM
       KN3=KN2+MLIM
       IF(NOD.EQ.2) GO TO 2002
       READ(ISTRS) NB    ,NN(K),NN(KN1),NN(KN2),NN(KN3)
       READ(ISTRS)((A(K,I,J),I=1,IPBN),J=1,6),((A(KN1,I,J),I=1,IPBN),J=1
      1,6),((A(KN2,I,J),I=1,IPBN),J=1,6),((A(KN3,I,J),I=1,IPBN),J=1,6)
       GO TO 3100
 2002  READ(ISTRS) NB    ,NN(K),NN(KN1)
       READ(ISTRS) ((B(K,I,J),I=1,IPBN),J=1,6),((B(KN1,I,J),I=1,IPBN),J=1
      1 ,6)
 3100  CONTINUE
       LLIM=KN3
       IF(NOD.EQ.2) LLIM=KN1
C        LOOP FOR PARTITIONS
       DO 4100 K=1,NPS                                                       5306
       IRITE=0                                                               5307
       DO 300 I=1,IPBL
```

```
      DO 300 J=1,60
  300 BIG(I,J)=0.0
      DO 3800 L=1,LLIM
C       CHECK TO SEE IF NODE IS IN THIS PARTITION
      IF(K.NE.1)GO TO 3450                                              5312
      IF(NN(L).EQ.0) GO TO 3800
      IF(NN(L).GT.LN(1))GO TO 3800                                      5313
      GO TO 3400
 3450 CONTINUE                                                          5314
      IF((NN(L).GT.LN(K)).OR. (NN(L).LE.LN(K-1)))GO TO 3800             5316
 3400 CONTINUE                                                          5317
      IRITE=1                                                           5318
C                                                                       5319
C       ADD IN STRESSES
      DO 3700 N=1,IPBN
      L1=L-1
      KROW=(MOD(L1,MLIM))*IPBN+N
      DO 3700 M=1,6                                                     5322
      IF(K.NE.1)GO TO 3460                                              5323
      KCOL=(NN(L)-1)*6+M                                                5324
      GO TO 3480                                                        5325
 3460 CONTINUE                                                          5326
      KCOL=(NN(L)-LN(K-1)-1)*6+M                                        5327
 3480 CONTINUE                                                          5328
      IF(NOD.EQ.4) GO TO 3701
      BIG(KROW,KCOL)=B(L,N,M)
      GO TO 3700
 3701 BIG(KROW,KCOL)=A(L,N,M)
 3700 CONTINUE
      NN(L)=0                                                           5330
 3800 CONTINUE                                                          5331
C                                                                       5332
      IF(IRITE.NE.1)GO TO 4000                                          5333
      NTOT=NTOT+1
C       WRITE STRESS PARTITION ON TAPE
      MAP=1000*LPL+K
      WRITE(KSTRES)MAP
      NSIZE=LN(1)*6                                                     5336
      IF(K.GT.1)NSIZE=(LN(K)-LN(K-1))*6                                 5337
      KNN=IPBL
      IF(LAST)KNN=IPBN*NTIML
      WRITE(KSTRES)((BIG(I,J),I=1,KNN),J=1,NSIZE)
      IF(NVIB.NE.1)GO TO 3900                                           5340
      IF(NOD.EQ.2) WRITE(IOUT,9001) MAP
      IF(NOD.EQ.4)WRITE(IOUT,9002)  MAP
      CALL PRINT(BIG,KNN,NSIZE,1,4H BIG,1,48)
 3900 CONTINUE                                                          5343
 4000 CONTINUE                                                          5347
C                                                                       5348
 4100 CONTINUE                                                          5349
C                                                                       5350
 2000 CONTINUE
      IF(NOD.EQ.2) GO TO 5000
      IPTOT=NTOT
      GO TO 5001
 5000 IBTOT=NTOT
 9001 FORMAT(22H MERGED BEAM STRESSES,I8,15HWRITTEN ON TAPE)
 9002 FORMAT(23H MERGED PLATE STRESSES,I8,15HWRITTEN ON TAPE)
 5001 RETURN
      END
```

SUBROUTINE SOLN

```
$IBFTC SORCON  DECK
       SUBROUTINE SOLN
       COMMON/SORT/NSAVE(7000),NZ(200),NPART(200),NPART2(200)
       COMMON/TERMS/NBEAMS,NPLATE,NNODE,NCOND,NPS,NTOL,NP,NOPT(4)
       COMMON/TAPES/MT1,MT2,MT3,MT4,MT5,MT6,MT7,MT8,MT9,MT10,MT11,MT12,
      *MT13,MT14,MT15,MT16,MT17
       COMMON/COMS/NSIZE(200)
       COMMON/CONT1/JPART(800)
       COMMON/RSIZE/ISIZE(500),NROW,JSIZE(200)
       COMMON/MAPSTR/IPTOT,IBTOT
       COMMON/CONTEM/NPRR,NPR
       COMMON/REDUC/NTEST,NTEST2
       CALL FKSORT
       CALL KFFSRT
       CALL CONECT
       IF (NOPT(3) .EQ. NTEST2) GO TO 20
       NELEM = 12
       NSTRS = 8
       IF (NPLATE .EQ. 0) GO TO 10
C SORT PLATE STRESS MATRIX
       REWIND MT8
       CALL DELETE (0,NELM,NSTRS)
       CALL SSORT (0,NELEM,NSTRS)
   10  CONTINUE
       IF (NBEAM .EQ. 0) GO TO 20
       NELEM = 8
       NSTRS = 12
C SORT BEAM STRESS MATRIX
       CALL DELETE (1,NELEM,NSTRS)
       CALL SSORT(1,NELEM,NSTRS)
   20  CONTINUE
       RETURN
       END
```

SUBROUTINE TEST

```
$IBFTC TEST*   DECK
       SUBROUTINE TEST(M,N,MAT,NPR,ITEST)                              5493
C***   TESTS MATRIX MAP MAT(I) TO SEE IF PARTION 1000*M+N
C        APPEARS IN STIFFNESS MATRIX
       DIMENSION MAT(1)                                                5494
       NUM=1000*M+N                                                    5498
       DO 1 I=1,NPR                                                    5499
       IF(NUM.EQ.MAT(I))GO TO 2                                        5500
     1 CONTINUE                                                        5501
       NUM=1000*N+M                                                    5502
       DO 10 I=1,NPR                                                   5503
       IF(NUM.EQ.MAT(I))GO TO 2                                        5504
    10 CONTINUE                                                        5505
       ITEST=0                                                         5506
       GO TO 3                                                         5507
     2 ITEST=1                                                         5508
     3 CONTINUE                                                        5509
       RETURN                                                          5510
       END                                                             5511
```

SUBROUTINE FKSORT

```
$IBFTC FKSRT   DECK
      SUBROUTINE FKSORT
C
C THIS SUBROUTINE CREATES THE NSAVE ARRAY WHICH CONTAINS THE LIST OF ELEMENT
C NUMBERS THAT ARE TO BE RETAINED AND THE NZ ARRAY WHICH CONTAINS THE NUMBER
C OF RETAINED ELEMENTS IN EACH PARTITION.
C
      COMMON/LASTND/LN(200)
      COMMON/TERMS/NBEAM,NPLATE,NNODE,NCOND,NPS,NMAX,NP
      COMMON/TAPES/MT1,MT2,MT3,MT4,MT5,MT6,MT7,MT8,MT9,MT10,MT11,MT12,
     * MT13,MT14,MT15,MT16,MT17
      COMMON/SORT/NSAVE( 7000),NZ(200)
      COMMON/CONT1/JPART(800)
      COMMON/CONTEM/NPRR,NPR
      COMMON/COMS/NSIZE(200)
      DIMENSION IR(60),NTEMP(800),IC(60)
C *** NOMENCLATURE ***   ***   ***   ***   ***   ***   ***   ***   ***   ***
C     NTOL = TOTAL NUMBER OF RETAINED FREEDOMS
C     NNZ COUNTS RETAINED FREEDOMS FOR THIS PARTITION
C     NZ = ARRAY CONTAINING THE NUMBER OF RETAINED FREEDOMS FOR EACH PARTITION
C     NSAVE = ARRAY OF ELEMENT NUMBERS OF THE STIFFNESS MATRIX WHICH ARE TO BE
C     RETAINED
C     IR = ARRAY CONTAINING REDUCTION INFORMATION FOR THIS PARTITION
C     IR(N) = 1 IF THE NTH FREEDOM IS TO BE RETAINED, AND ZERO IF IT IS TO BE
C     REDUCED.
C     LN = ARRAY CONTAINING LAST NODES FOR EACH PARTITION.
C     NMAX = TOTAL NUMBER OF PARTITIONS (ELEMENTS IN JPART ARRAY)
C     NPR = NUMBER OF KFF PARTITIONS (ELEMENTS IN NTEMP)
C
C THE NSAVE ARRAY IS CONSTUCTED AS SHOWN BELOW
C     NSAVE(1) = FIRST RETAINED ELEMENT NUMBER****
C     NSAVE(2) = 2ND   RETAINED ELEMENT NUMBER    *
C        *                 *                       *
C        *                 *                        *FIRST PARTITION
C        *                ETC                      *
C        *                                        *
C     NSAVE(NZ(1)) = LAST RETAINED ELEM. NUMBER***
C     NSAVE(NZ(1)+1) = 1ST RET. ELEM. NUMBER     *
C     NSAVE(NZ(1)+2)   2ND RET. ELEM. NUMBER      *
C        *                 *                       *SECOND PARTTITION
C        *                 *                      *
C        *                ETC.                    *
C     NSAVE(NZ(1)+NZ(2))*************************
C        *                                       *
C        *                                        *
C        *                                         *LAST PARTITION
C        *                                        *
C        *                                       *
C     NSAVE(NTOL) = LAST RET. ELEM. NUMBER********


C INITIALIZE ALL ARRAYS AND COUNTERS.
      IOUT = MT6
      N18 = MT1
      ISTIF = MT2
      IKBC = MT3
```

```
C
      REWIND N18
      REWIND ISTIF
      REWIND IKBC

      NPR = 0
      DO 4 K=1,NPS
      IF (NSIZE(K) .EQ. 0) GO TO 4
      NPR = NPR + 1
      NTEMP(NPR) = K*1000 + K
      IF (K .EQ. 1) GO TO 4
      NEND = K-1
      DO 3 I=1,NEND
      IF (NSIZE(I) .EQ. 0) GO TO 3
      CALL TEST (K,I,JPART,NMAX,ITEST)
      IF (ITEST .EQ. 0) GO TO 3
      NPR = NPR + 1
      NTEMP(NPR) = K*1000 + I
    3 CONTINUE
    4 CONTINUE

      DO 6 I=1,NPR
      JPART(I) = NTEMP(I)
    6 CONTINUE

      NTOL = 0
      DO 5 J=1,200
    5 NZ(J) = 0

C LOOP FOR EACH PARTITION
C
      DO 100 I=1,NP

C CALCULATE FIRST AND LAST NODE NUMBERS FOR THIS PARTITION
C
      IF(I .EQ. 1) GO TO 7
      N1 = LN(I-1) + 1
      N2 = LN(I)
      GO TO 8
    7 CONTINUE
      N1 = 1
      N2 = LN(1)
    8 CONTINUE
      NROW = 6*(N2-N1+1)
      M = 1

C READ RETAINED FREEDOMS FROM TAPE FOR ONE NODE AND ADD TO LIST FOR THIS PART.
C
      DO 10 N=N1,N2
      READ (N18) NCODE
      IF(NCODE .NE. N) GO TO 9990
      READ (N18) IR(M),IR(M+1),IR(M+2),IR(M+3),IR(M+4),IR(M+5)

C READ BOUNDARY CONDITIONS
      READ(IKBC) NODE,IJKLMN
      IF(NODE .NE. N) GO TO 9991
      CALL UNPACK(IJKLMN,IC(M),IC(M+1),IC(M+2),IC(M+3),IC(M+4),IC(M+5))
      ISP = 0
      M6 = M+5
      DO 9 IT=M,M6
```

```
      IF(IC(IT) .EQ. 3) ISP = 1
    9 CONTINUE
      IF(ISP .EQ. 1) READ(IKBC) D1,D2,D3,D4,D5,D6
      M = M+6
   10 CONTINUE
   20 CONTINUE
      NNZ = 0

C START SORTING LOOP
C
      J = 0
      DO 50 JJ=1,NROW
      IF((IC(JJ) .EQ. 0) .OR. (IC(JJ) .EQ. 3)) J = J + 1

C TEST FOR RETAINED FREEDOM
C
      IF(IR(JJ) .NE. 1) GO TO 50

C UPDATE COUNT OF RETAINED ELEMENTS AND ADD J TO ARRAY OF RETAINED ELEMENT NO.S
C
      NTOL = NTOL + 1
      NNZ = NNZ + 1
      NSAVE(NTOL) = J
   50 CONTINUE

C SAVE THE TOTAL NUMBER OF RETAINED FREEDOMS FOR THIS PARTITION
C
      NZ(I) = NNZ

  100 CONTINUE
      REWIND IKBC
      RETURN


 9990 WRITE (IOUT,9000) NCODE,N
      STOP
 9991 WRITE (IOUT,9001) NODE,N
      STOP


 9000 FORMAT(103H1ERROR IN SUBROUTINE FKSORT. THE PARTITION NUMBER READ
     *FROM TAPE DOES NOT AGREE WITH WHAT WAS EXPECTED.//27H PARTITION NU
     *MBER READ WAS I3,14H IT SHOULD BE I3)
 9001 FORMAT(//52H NODE NUMBER READ FROM IKBC IN FKSORT WAS INCORRECT./
     *16H NUMBER READ WAS,I8,5X,19HIT SHOULD HAVE BEEN,I8)
      END
```

SUBROUTINE KFFSRT

```
$IBFTC KFSRT   DECK
      SUBROUTINE KFFSRT
      COMMON/SORT/NSAVE( 7000),NZ(200),NPART(200),NPART2(200)
      COMMON/TAPES/MT1,MT2,MT3,MT4,MT5,MT6,MT7,MT8,MT9,MT10,MT11,MT12,
     * MT13,MT14,MT15,MT16,MT17
      COMMON/TERMS/NBEAM,NPLATE,NNODE,NCOND,NPS,NTOL,NP
      COMMON/COMS/NSIZE(200)
      COMMON/CONT1/MAT(800)
      COMMON/CONTEM/NPRR,NPR
      DIMENSION STIFF(60,60),SORTED(60,60),B(12)
C
C THIS SUBROUTINE SORTS THE KFF MATRIX INTO RETAINED (K11), DELETED (K22), AND
C DELETED-RETAINED (K21) PARTITIONS. IT THEN WRITES THEM ON TAPE IN TL01 FORMAT.
C
C*** NOMENCLATURE ***   ***   ***   ***   ***   ***   ***   ***   ***   ***   **
C     NP = TOTAL NUMBER OF PARTITIONS
C     NZ = ARRAY CONTAINING TOTAL NUMBER OF RETAINED FREEDOMS FOR EACH PARTITION
C     NSAVE = ARRAY OF RETAINED ELEMENT NUMBERS
C     ISIZE = SIZE OF THE PARTITION BEING PROCESSED.
C     STIFF = UNSORTED STIFFNESS MATRIX READ IN FROM TAPE
C     SORTED = SORTED STIFFNESS MATRIX WRITTEN OUT ON TAPE IN TL01 FORMAT
C
      IOUT = MT6
      KFF11 = MT16
      KFF12 = MT12
      KFF21 = MT2
      KFF22 = MT1
      KFF = MT11
C
      REWIND KFF11
      REWIND KFF12
      REWIND KFF21
      REWIND KFF22
      NFILE = 0
      NMAT =0
      REWIND KFF
      DO 205 K=1,NP
      NCNT = 0
      NCOUNT = 0
      NCNT1 = 0
      NCNT2 = 0
      DO 200 L=1,K
      DO 5 I=1,60
      DO 5 J=1,60
      SORTED(I,J) = 0.0
      STIFF(I,J) = 0.0
    5 CONTINUE
      N12 = 0
      N11 = 0
      ID = 1000*K + L
      CALL TEST (K,L,MAT,NPR,ITEST)
C
C IF THERE IS NO KOOL PARTITION, NULL MATRICES WILL BE WRITTEN ON TAPE
      ISIZEN = NSIZE(K)
      ISIZEM = NSIZE(L)
      IF (ITEST .NE. 1) GO TO 160
      N22 = NZ(K)
      N21 = NZ(K)
      IF(NSIZE(K) .EQ. 0) GO TO 201
      IF(NSIZE(L) .EQ. 0) GO TO 200
```

```
C
C CALCULATE LOCATION OF RETAINED FREEDOMS FOR THIS PARTITION
      IF (K .EQ. 1) GO TO 10
      NFIRST = 0
      DO 7 N1=2,K
      NFIRST = NFIRST + NZ(N1-1)
    7 CONTINUE
      NFIRST = NFIRST + 1
      GO TO 13
   10 NFIRST = 1
   13 NLAST = NFIRST + NZ(K) - 1
      IF(L .EQ. 1) GO TO 16
      MFIRST = 0

      DO 15 N2=2,L
      MFIRST = MFIRST + NZ(N2-1)
   15 CONTINUE
      MFIRST = MFIRST + 1
      GO TO 17
   16 MFIRST = 1
   17 MLAST = MFIRST + NZ(L) - 1
C
C READ STIFFNESS PARTITION (KFF) FROM TAPE.
      NAME = K*1000 + L
      CALL READTP(STIFF,60,NAME,NSIZE(K),NSIZE(L),B,0,0,KFF,IERR)
      IF (IERR .EQ. 1) GO TO 900
C
C IF NUMBER OF RETAINED FREEDOMS = 0, PLACE ALL ELEMENTS IN K22
      IF((NZ(K) .EQ. 0).AND.(NZ(L) .EQ. 0)) GO TO 150
C IF ALL FREEDOMS ARE RETAINED, PLACE ALL ELEMENTS IN K11
      IF((NZ(K) .EQ. ISIZEN) .AND. (NZ(L) .EQ. ISIZEM)) GO TO 140


      DO 100 N=1,ISIZEN
      M12 = NZ(L)
      M22 = NZ(L)
      M21 = 0
      M11 = 0
      IF (NZ(K) .EQ. 0) GO TO 21

      DO 20 NTEST=NFIRST,NLAST
      IF (NSAVE(NTEST) .EQ. N) GO TO 50
   20 CONTINUE

   21 CONTINUE
      N21 = N21+1
      N22 = N22+1

      DO 40 M=1,ISIZEM
      IF (NZ(L).EQ. 0) GO TO 31

      DO 30 MTEST=MFIRST,MLAST
      IF (NSAVE(MTEST) .EQ. M) GO TO 35
   30 CONTINUE

   31 CONTINUE
C PLACE THE N,M ELEMENT OF KFF INTO THE K22 PARTITION
      M22 = M22+1
      SORTED(N22,M22) = STIFF(N,M)
```

```
      GO TO 40
   35 CONTINUE
C PLACE THE N,M ELEMENT OF KFF INTO THE K21 PARTITION
      M21 = M21+1
      SORTED(N21,M21) = STIFF(N,M)
   40 CONTINUE
      GO TO 100

   50 CONTINUE
      N11 = N11+1
      N12 = N12+1

      DO 90 M=1,ISIZEM
      IF (NZ(L) .EQ. 0) GO TO 71

      DO 70 MTEST=MFIRST,MLAST
      IF (NSAVE(MTEST) .EQ. M) GO TO 75
   70 CONTINUE

   71 CONTINUE
C PLACE THE N,M ELEMENT OF KFF INTO THE K12 PARTITION
      M12 = M12+1
      SORTED(N12,M12) = STIFF(N,M)
      GO TO 90
   75 CONTINUE
C PLACE THE N,M ELEMENT OF KFF INTO THE K11 PARTITION
      M11 = M11+1
      SORTED(N11,M11) = STIFF(N,M)
   90 CONTINUE

  100 CONTINUE
      GO TO 160

C ALL FREEDOMS RETAINED. ALL ELEMENTS OF KFF GO INTO K11.
  140 CONTINUE
      ISIZEN = NSIZE(K)
      ISIZEM = NSIZE(L)
      NCNT1 = NCNT1 + 1
      CALL WRTETP(STIFF,60,ID,ISIZEN,ISIZEM,B,0,0,KFF11,IERROR)
      GO TO 200

C NO RETAINED FREEDOMS. ALL ELEMENTS OF KFF GO INTO K22.
  150 CONTINUE
      ISIZEN = NSIZE(K)
      ISIZEM = NSIZE(L)
      NCNT2 = NCNT2 + 1
      CALL WRTETP (STIFF,60,  ID,ISIZEN,ISIZEM ,  B,NFILE,NMAT,KFF22,
     *IERROR)
      GO TO 200
  160 CONTINUE

C WRITE OUT K22 PARTITION ONTO TAPE
      N = NZ(K) + 1
      M = NZ(L) + 1
      IROW = ISIZEN - NZ(K)
      ICOL = ISIZEM - NZ(L)
      IF ((IROW .EQ. 0) .OR. (ICOL .EQ. 0)) GO TO 170
      WRITE (IOUT,6500) ID,IROW,ICOL,KFF22
      NCNT2 = NCNT2 + 1
      CALL WRTETP (SORTED(N,M),60,    ID     ,IROW,ICOL,B,NFILE,NMAT,KFF22
```

```
     *,IERROR)
  170 CONTINUE

C WRITE OUT K21 PARTITION ONTO TAPE
      ICOL = NZ(L)
      IF ((IROW .EQ. 0) .OR. (ICOL .EQ. 0)) GO TO 180
      NCOUNT = NCOUNT+1
      CALL WRTETP (SORTED(N,1),60,     ID    ,IROW,ICOL,B,NFILE,NMAT,KFF21
     *,IERROR)
  180 CONTINUE

C WRITE OUT K12 PARTITION ONTO TAPE
      IROW = NZ(K)
      ICOL = ISIZEM - NZ(L)
      IF ((IROW .EQ. 0) .OR. (ICOL .EQ. 0)) GO TO 190
      NCNT = NCNT + 1
      CALL WRTETP(SORTED(1,M),60,ID,IROW,ICOL,B,0,0,KFF12,IERROR)
  190 CONTINUE

C WRITE OUT K11 PARTITION ONTO TAPE
      ICOL = NZ(L)
      IF ((IROW .EQ. 0) .OR. (ICOL .EQ. 0)) GO TO 200
      NCNT1 = NCNT1 + 1
      CALL WRTETP (SORTED(1,1),60,     ID    ,IROW,ICOL,B,NFILE,NMAT,KFF11
     *,IERROR)
  200 CONTINUE
  201 CONTINUE
      NPART2(K) = NCNT
      NPART(K) = NCOUNT
C
C WRITE END OF FILES ON ALL TAPES INDICATING THE END OF A ROW
      IF (NCOUNT .NE. 0) END FILE KFF21
      IF (NCNT .NE. 0) END FILE KFF12
      IF (NCNT1 .NE. 0) END FILE KFF11
      IF (NCNT2 .NE. 0) END FILE KFF22
  205 CONTINUE
      REWIND KFF22
      REWIND KFF21
      REWIND KFF12
      REWIND KFF11
      RETURN
  900 WRITE (IOUT,9000)
 9000 FORMAT(///48H ERROR IN READ THE KFF TAPE IN SUBROUTINE KFSORT)
      STOP
      END
```

SUBROUTINE CONECT

```
$IBFTC CNCT*   DECK
       SUBROUTINE CONECT
       COMMON/SORT/ NSAVE( 7000),NZ(200),NPART(200),NPART2(200)
       COMMON/TERMS/DUMMY(6),NP
       COMMON/TAPES/MT1,MT2,MT3,MT4,MT5,MT6,MT7,MT8,MT9,MT10,MT11,MT12,
      * MT13,MT14,MT15,MT16,MT17
       COMMON/COMS/NSIZE(200)
       DIMENSION ISIZE(200),JSIZE(200),NPARA(50),B(12),PARA(50)
       EQUIVALENCE (PARA,NPARA)

       IOUT = MT6
       NTAPE4 = MT4
       NTP11 = MT11
       KFF11 = MT16
       KFF22 = MT1
       K21 = MT16
       KFF21 = MT2
       KFF12 = MT12

       N = 0
       DO 10 I=1,NP
       IF(NZ(I) .EQ. 0) GO TO 10
       N = N+1
       ISIZE(N) = NZ(I)
    10 CONTINUE
       NUMBER = N
       N = 0

C SET JSIZE ARRAY
       DO 20 I=1,NP
       IF(NZ(I) .EQ. NSIZE(I)) GO TO 20
       N = N+1
       JSIZE(N) = NSIZE(I) - NZ(I)
    20 CONTINUE
       NUMB = N

C WRITE PARAMETERS IN FILE 1 OF NTAPE4
       REWIND NTAPE4
       NPARA(1) = NUMB
       NPARA(2) = 2*NUMB
       NPARA(3) = NUMBER
       PARA(4) = 1.0
       NPARA(5) = NUMBER
       DO 30 I=6,50
       NPARA(I) = 0.0
    30 CONTINUE
       NAME = 0
       CALL WRTETP(NPARA,1,NAME,50,1,B,0,0,NTAPE4,IERR)
       END FILE NTAPE4
C
C EXPAND KFF11 TAPE ONTO TAPE NTAPE4 (2ND FILE)
       CALL EXPAND(ISIZE,NUMBER,KFF11,NTAPE4)
       END FILE NTAPE4

C
C EXPAND KFF22 TAPE ONTO TAPE NTP11
       REWIND NTP11
```

```
      CALL EXPAND(JSIZE,NUMB,KFF22,NTP11)
      END FILE NTP11
      REWIND NTP11
      N = 0
C SET UP ARGUMENTS FOR EXTRAN

C ELIMINATE ZEROS FROM THE NPART ARRAY ONLY IF THE FIRST NON-ZERO
C  ELEMENT IS NOT EQUAL TO THE CORRESPONDING ELEMENT IN NPART2.
      INC = 0
      NZERO = 0
   50 INC = INC + 1
      IF(NPART(INC) .NE. 0) GO TO 60
      NZERO = NZERO + 1
      GO TO 50
   60 CONTINUE
      IF ((NZERO .NE. 0) .AND. (NPART(INC) .NE. NPART2(INC))) GO TO 110
      DO 70 I=1,NUMB
      NSUB = I + NZERO
      NPART(I) = NPART(NSUB)
   70 CONTINUE

C ELIMINATE ZEROS FROM THE NPART2 ARRAY
      INC = 0
      NZERO = 0
   80 INC = INC + 1
      IF(NPART2(INC) .NE. 0) GO TO 90
      NZERO = NZERO + 1
      GO TO 80
   90 CONTINUE
      DO 100 I=1,NUMBER
      NSUB = I + NZERO
      NPART2(I) = NPART2(NSUB)
  100 CONTINUE
  110 CONTINUE

      CALL EXTRAN(NPART,NPART2,NUMBER,NUMB,NTAPE4,K21,KFF21,KFF12)
      REWIND NTAPE4
      RETURN
      END
```

SUBROUTINE EXPAND

```
$IBFTC EXPND*  DECK
      SUBROUTINE EXPAND(ISIZE,NUMBER,KTAPE,ITAPE)
      DIMENSION TRANSP(60,60),TEMP(60,60),ISIZE(1),B(12)
C
C THIS SUBROUTINE EXPANDS THE K11 AND K22 PARTITIONS INTO FULL MATRIX FORM.
C **** TAPE USEAGE ****   ****   ****   ****   ****   ****   ****   ****   ****
C KTAPE CONTAINS INPUT IN LOWER TRIANGULAR FORM
C ITAPE CONTAINS OUTPUT IN FULL FORM
C ITAPE MUST BE POSITIONED PROPERLY BY CALLING ROUTINE
C *** NOMENCLATURE ***   ***   ***   ***   ***   ***   ***   ***   ***   ***
C     NROW = ROW OF PARTITIONS BEING FORMED
C     ISIZE = ARRAY CONTAING SIZES OF THE PARTITIONS
C     NROWR = ROW OF PARTITIONS IN WHICH WE READ THE PARTITION THAT WILL BE
C     TRANSPOSED AND WRITTEN IN THE PARTITION ROW BEING FORMED
C     ID = THE IDENTIFYING WORD THAT IS ASSCOIATED WITH A GIVEN PARTITION
C     N = INDICATES THE NUMBER OF ROWS THAT HAVE ALREADY BEEN FORMED
C     NUMBER = NUMBER OF ROWS (AND COLUMNS) OF PARTITIONS


      REWIND KTAPE
      N = 0
      L = 0
      NFILE = 1
      NZERO = 0
    5 CONTINUE
      IF(N .NE. 0) CALL FSF(N,KTAPE,IERRA)
      NROW = N+1
      NROWR = NROW
      IF(ISIZE(NROW).EQ. 0) GO TO 40
C
C READ ROW OF PARTITIONS AND ADD THEM TO THE NEW TAPE
      DO 10 I=1,NROW
      NCOL = I
      ID = 0
      CALL READTP(TEMP,60,ID,IROW,ICOL,B,0,0,KTAPE,IERROR)
      IF(IERROR .NE. 0) GO TO 990
      CALL WRTETP(TEMP,60,ID,IROW,ICOL,B,0,0,ITAPE,IERR)
      IF (IERR .NE. 0) GO TO 991
   10 CONTINUE
      IF (NROW .GE. NUMBER) GO TO 50

C FORWARD SPACE TO READ TRANSPOSE OF NEXT PARTITION TO BE WRITEN
   15 CONTINUE
      NROWR = NROWR + 1
      ID = 0
      CALL READTP(TEMP,60,ID,IROW,ICOL,B,NFILE,N,KTAPE,IERROR)
      IF(IERROR .NE. 0) GO TO 990
C
C FORM TRANSPOSE
      DO 20 I=1,IROW
      DO 20 J=1,ICOL
      TRANSP(J,I) = TEMP(I,J)
   20 CONTINUE
      IDT = (ID - (ID/1000)*1000)*1000 + ID/1000
      CALL WRTETP(TRANSP,60,IDT,ICOL,IROW,B,0,0,ITAPE,IERR)
      IF (IERR .NE. 0) GO TO 991
      IF(NROWR .GE. NUMBER) GO TO 30
```

```
      GO TO 15
   30 CONTINUE
      IF(NROW .GE. NUMBER) GO TO 50
      REWIND KTAPE
   40 N = N + 1
      GO TO 5
   50 CONTINUE
      REWIND KTAPE
      RETURN
  990 WRITE(6,9000) IERROR
 9000 FORMAT(1H1,23HREADTP ERROR IN EXPAND.,13H ERROR CODE =,I5)
      STOP
  991 WRITE(6,9001) IERR
 9001 FORMAT(1H1,23HWRTETP ERROR IN EXPAND.,13H ERROR CODE =,I5)
      STOP
      END
```

SUBROUTINE EXTRAN

```
$IBFTC EXTRN*  DECK
      SUBROUTINE EXTRAN(NP21,NP12,NP11,NP22,NTAPE4,K21,KFF21,KFF12)
C THIS SUBROUTINE CREATES A FULL K21 MATRIX USING BOTH K21 AND K12 TAPES OUTPUT
C FROM THE KFSORT AND ALSO A FULL K12 MATRIX (BY TRANSPOSING K21).
C**** TAPE USEAGE ****    ****    ****    ****    ****    ****    ****    ****
C NTAPE4 IS THE OUTPUT FOR K21 AND K12 IN FULL FORM. IT MUST BE POSITIONED AT
C THE BEGINING OF FILE 3 AT THE START OF THIS ROUTINE AND K12 WILL BE WRITTEN
C AS FILE 3 WITH K21 FOLLOWING IN FILE 4.
C**** NOMENCLATURE ****    ****    ****    ****    ****    ****    ****    ****
C     NP21 = ARRAY GIVING THE NUMBER OF PARTITIONS IN EACH ROW OF K21
C     NP12 = ARRAY GIVING THE NUMBER OF PARTITIONS IN EACH ROW OF K12
C     NP11 = NUMBER OF PARTITIONS IN K11
C     NP22 = NUMBER OF PARTITIONS IN K22
C     N = NUMBER OF ROWS OF PARTITIONS WRITTEN ON K21
C     M = NUMBER OF ROWS OF PARTITIONS WRITTEN FOR K12 (ON NTAPE4)
C     NMOD = NUMBER OF ROWS OF PARTITIONS READ FROM KFF21

      DIMENSION TEMP(60,60),TRANSP(60,60),B(12),NP21(1),NP12(1)

      IERR = 0
      IOUT = 6
      REWIND K21
      REWIND KFF21
      REWIND KFF12
      N = 0
      NMOD = 0

    5 CONTINUE
      AREA = 0.0
      IF (NMOD .GT. 0) CALL FSF(NMOD,KFF21,IERR)
      IF ((NMOD .GT. 0) .AND. (IERR .NE. 0)) GO TO 992
      NPART = NP21(N+1)
      IF (NPART .EQ. 0) GO TO 12

C COPY NPART PARTITIONS FROM KFF21 TAPE TO K21 TAPE
      DO 10 I=1,NPART
      ID = 0
      AREA  =  1.0
      CALL READTP(TEMP,60,ID,IROW,ICOL,B,0,0,KFF21,IERR)
      IF (IERR .NE. 0) GO TO 990
      AREA  =  2.0
      CALL WRTETP(TEMP,60,ID,IROW,ICOL,B,0,0,K21,IERR)
      IF (IERR .NE. 0) GO TO 990
   10 CONTINUE
      NMOD = NMOD + 1
   12 CONTINUE
      IF (NPART .GE. NP11) GO TO 25

C SKIP NSKIP ROWS OF PARTITIONS ON KFF12 TAPE

      NSKIP = NPART - 1
      IF (NSKIP .GT. 0) CALL FSF(NSKIP,KFF12,IERR)
      AREA  =  3.0
      IF ((NSKIP .GT. 0) .AND. (IERR .NE. 0)) GO TO 992

      NFILE = 1
      IF (NSKIP .LT. 0) NFILE = 0
```

```
      NSTART = NPART + 1
      DO 20 I=NSTART,NP11
      ID = 0
      AREA  =  4.0
      CALL READTP(TEMP,60,ID,IROW,ICOL,B,NFILE,N,KFF12,IERR)
      IF (IERR .NE. 0) GO TO 990

C FORM TRANSPOSE
      DO 15 J=1,IROW
      DO 15 K=1,ICOL
      TRANSP(K,J) = TEMP(J,K)
   15 CONTINUE

C WRITE OUT ON TAPE K21
      IDT = (ID - (ID/1000)*1000)*1000 + ID/1000
      AREA  =  5.0
      CALL WRTETP(TRANSP,60,IDT,ICOL,IROW,B,0,0,K21,IERR)
      IF (IERR .NE. 0) GO TO 990
      NFILE = 1
   20 CONTINUE


C UPDATE COUNT OF ROWS ALREADY FORMED
   25 CONTINUE
      N=N+1
      REWIND KFF21
      REWIND KFF12
      END FILE K21
      IF (N .GE. NP22) GO TO 30
      GO TO 5

C K21 COMPLETED
   30 CONTINUE
      REWIND K21

C FORM K12 ON TAPE 4
      M = 0
   32 CONTINUE
      NFILE = 0

      DO 40 I=1,NP22
      ID = 0
      AREA  =  6.0
      CALL READTP(TEMP,60,ID,IROW,ICOL,B,NFILE,M,K21,IERR)
      IF (IERR .NE. 0) GO TO 990

C FORM TRANSPOSE
      DO 35 J=1,IROW
      DO 35 K=1,ICOL
      TRANSP(K,J) = TEMP(J,K)
   35 CONTINUE
      IDT = (ID - (ID/1000)*1000)*1000 + ID/1000
      AREA  =  7.0
      CALL WRTETP(TRANSP,60,IDT,ICOL,IROW,B,0,0,NTAPE4,IERR)
      IF (IERR .NE. 0) GO TO 990
      NFILE =1
   40 CONTINUE

      M = M+1
      IF (M .GE. NP11) GO TO 50
```

```
      REWIND K21
      GO TO 32

C K12 COMPLETED
   50 CONTINUE
      REWIND K21
      END FILE NTAPE4


C WRITE K21 ONTO NTAPE4
      DO 60 I=1,NP22
      DO 55 J=1,NP11
      ID = 0
      AREA  =  8.0
      CALL READTP(TEMP,60,ID,IROW,ICOL,B,0,0,K21,IERR)
      IF (IERR .NE. 0) GO TO 990
      AREA  =  9.0
      CALL WRTETP(TEMP,60,ID,IROW,ICOL,B,0,0,NTAPE4,IERR)
      IF (IERR .NE. 0) GO TO 990
   55 CONTINUE
      IF (I .LT. NP22) CALL FSF(1,K21,IERR)
      IF ((I .LT. NP22) .AND. (IERR .NE. 0)) GO TO 992
   60 CONTINUE

C NTAPE4 COMPLETED
      END FILE NTAPE4
      REWIND K21
      RETURN

C ERROR COMMENTS
  990 WRITE (6,9000) AREA,IERR
      STOP
  992 WRITE (6,9002) AREA,IERR
      STOP
 9000 FORMAT(// 7H AREA =,F5.0,10X,13HERRROR CODE = I5)
 9002 FORMAT(//18H FSF ERROR. AREA =,F5.0,13H ERROR CODE =,I3)
      END
```

SUBROUTINE DELETE

```
$IBFTC DELET*  DECK
       SUBROUTINE DELETE(ITYPE,NELEM,NSTRS)
       COMMON/TAPES/MT1,MT2,MT3,MT4,MT5,MT6,MT7,MT8,MT9,MT10,MT11,MT12,
      * MT13,MT14,MT15,MT16,MT17
       COMMON/TERMS/NBEAM,NPLATE,NNODE,NCOND,NPS,NTOL,NP
       COMMON/RSIZE/ISIZE(500),NROW,JSIZE(200)
       COMMON/LASTND/LN(200)
       COMMON/MAPSTR/IPTOT,IBTOT
       COMMON/TEMPO/STRESS(96,60)
       DIMENSION SAVE(96),IC(60)

C THIS SUBROUTINE DELETES THE COLUMNS OF THE STRESS MATRIX THAT CORRESPOND TO
C CONSTRAINED DEGREES OF FREEDOM

C **** NOMENCLATURE ****    ****     ****     ****     ****     ****     ****     ****
C      ITYPE = TYPE OF ELEMENT IN STRESS MATRIX. 1 FOR BEAMS, 0 FOR PLATES
C      NELEM = THE MAXIMUM NUMBER OF ELEMENTS PER PARTITION
C      NSTRS = THE NUMBER OF STRESSES PER ELEMENT
C      ISIZE = THE ARRAY OF ROW DIMENSIONS FOR THE STRESS PARTITIONS
C      NROW = NUMBER OF ROWS OF PARTITIONS IN STRESS MATRIX BEING PROCESSED
C      JSIZE = COLUMN DIMENSIONS FOR STRESS PARTITIONS (SORTED)
C ****     ****     ****     ****     ****     ****     ****     ****     ****     ****


       NSTRES = MT12
       KSTRES = MT8
       IKBC = MT3
       IOUT = MT6
       REWIND NSTRES

C CALCULATE PARTITION COLUMN SIZE ARRAY
       DO 10 I=1,NP
       IF(I .EQ. 1) GO TO 5
       JSIZE(I) =(LN(I) - LN(I-1))*6
       GO TO 10
     5 CONTINUE
       JSIZE(I) = LN(I) *6
    10 CONTINUE

       IF (ITYPE .EQ. 1) NUMBER = NBEAM
       IF (ITYPE .EQ. 0) NUMBER = NPLATE
       IF(ITYPE .EQ. 1) NTOT = IBTOT
       IF(ITYPE .EQ. 0) NTOT = IPTOT
       NTOT2 = NTOT
       IREM = 0

C CALCULATE NUMBER OF ROWS OF PARTITIONS AND ROW DIMENSIONS
       NROW = NUMBER/NELEM
       IF (NUMBER - NROW*NELEM .EQ. 0) GO TO 15
       NROW = NROW + 1
       IREM = 1
    15 DO 20 I=1,NROW
       ISIZE(I) = NELEM*NSTRS
    20 CONTINUE
       IF (IREM .EQ. 1) ISIZE(NROW) =(NUMBER - (NROW - 1)*NELEM)*NSTRS
```

```
C BEGIN SORTING PROCESS

      NCOUNT = 0
C LOOP ON PARTITION ROWS
      DO 200 I=1,NROW
      REWIND IKBC

C LOOP ON PARTITION COLUMNS
      DO 190 J=1,NP
      IMAX = ISIZE(I)
      DO 25 I2=1,IMAX
      DO 25 J2=1,60
      STRESS(I2,J2) = 0.0
   25 CONTINUE

      ID = 1000*I + J
      IF(NCOUNT .GE. NTOT) GO TO 210
      READ (KSTRES) NCODE

C READ IN BOUNDARY CONDITIONS FROM IKBC
      IF (J .EQ. 1) GO TO 32
      N1 = LN(J-1) + 1
      N2 = LN(J)
      GO TO 33
   32 CONTINUE
      N1 = 1
      N2 = LN(J)
   33 CONTINUE
      M = 1

      DO 35  NN=N1,N2
      READ (IKBC) NODE,IJKLMN
      IF (NODE .NE. NN) GO TO 990
      CALL UNPACK(IJKLMN,IC(M),IC(M+1),IC(M+2),IC(M+3),IC(M+4),IC(M+5))
      ISP = 0
      M6 = M+5
      DO 34 IT=M,M6
      IF(IC(IT) .EQ. 3) ISP = 1
   34 CONTINUE
      IF(ISP .EQ. 1) READ(IKBC) D1,D2,D3,D4,D5,D6
      M = M + 6
   35 CONTINUE

      IF (NCODE .NE. ID) GO TO 180

C SORT THIS PARTITION (NCODE = ID)
      NCOUNT = NCOUNT + 1

C READ IN STRESS PARTITION FROM KSTRES
      IROW = ISIZE(I)
      ICOL = JSIZE(J)
      READ (KSTRES) ((STRESS(M,N),M=1,IROW),N=1,ICOL)

      ICON = 0

C COUNT FREEDOMS TO BE DELETED
      DO 40 NN=1,ICOL
      IF((IC(NN) .EQ. 1) .OR. (IC(NN) .EQ. 2)) ICON = ICON + 1
   40 CONTINUE
```

```
C TEST FOR NO CONSTRAINED FREEDOMS OR FOR ALL CONSTRAINED FREEDOMS
      IF (ICON .EQ. 0) GO TO 120
      IF(ICON .EQ. ICOL) NTOT2 = NTOT2 - 1
      IF (ICON .EQ. ICOL) GO TO 190


C BEGIN SORTING
      JS = 0
      JS1 = 0
      JS2 = 0
   50 JS = JS + 1

      IF((IC(JS) .NE. 1) .AND. (IC(JS) .NE. 2)) GO TO 100

C DELETE THIS COLUMN
      N1 = JS1 + 2
      N2 = ICOL
      DO 66 JNC1=N1,N2
      DO 65 INC1=1,IMAX
      STRESS(INC1,JNC1-1) = STRESS(INC1,JNC1)
   65 CONTINUE

   66 CONTINUE
      IF (JS .LT. ICOL) GO TO 50

  100 CONTINUE

C UPDATE COUNT OF RETAINED COLUMNS
      JS1 = JS1 + 1
      IF (JS .LT. ICOL) GO TO 50

C WRITE OUT THE NEW PARTITION ON TAPE NSTRES
      WRITE(NSTRES) NCODE
      WRITE(NSTRES)((STRESS(M,N),M=1,IROW),N=1,N2)
      GO TO 190

  120 CONTINUE


C ALL FREEDOMS RETAINED. NO SORTING NEEDED
      WRITE(NSTRES) NCODE
      WRITE(NSTRES) ((STRESS(M,N),M=1,IROW),N=1,ICOL)
      GO TO 190
  180 CONTINUE
      BACKSPACE KSTRES
  190 CONTINUE

  200 CONTINUE

  210 CONTINUE
      IF(ITYPE .EQ. 0) IPTOT = NTOT2
      IF(ITYPE .EQ. 1) IBTOT = NTOT2
      REWIND IKBC
      REWIND NSTRES
      RETURN


C ERROR COMMENT
  990 WRITE (IOUT,9000) NODE,NN
      STOP
```

```
9000 FORMAT(//87H NODE NUMBER READ FROM TAPE IN SUBROUTINE DELETE DOES
     *NOT AGREE WITH WHAT WAS EXPECTED./ 6H READ ,I8,5X,10H EXPECTED I8)
      END
```

SUBROUTINE SSORT

```
$IBFTC SSORT*  DECK
       SUBROUTINE SSORT(ITYPE,NELEM,NSTRS)
       COMMON/SORT/NSAVE(7000),NZ(200),NPART(200)
       COMMON/TAPES/MT1,MT2,MT3,MT4,MT5,MT6,MT7,MT8,MT9,MT10,MT11,MT12,
      * MT13,MT14,MT15,MT16,MT17
       COMMON/TERMS/NBEAM,NPLATE,NNODE,NCOND,NPS,NTOL,NP
       COMMON/RSIZE/ISIZE(500),NROW,JSIZE(200)
       COMMON/COMS/NSIZE(200)
       COMMON/MAPSTR/IPTOT,IBTOT
       COMMON/TEMPO/STRESS(96,60)
       DIMENSION SAVE(96),NPARA(50),PARA(50),SORTED(96,60),B(12)
       EQUIVALENCE(STRESS,SORTED),(NPARA,PARA)
C THIS SUBROUTINE SORTS THE STRESS MATRIX FOR EITHER BEAMS OR PLATES INTO TWO
C PARTS - S1 (RETAINED) AND S2 (REDUCED)
C*****NOMENCLATURE*****
C      ITYPE = TYPE OF STRESS MATRIX. 1 FOR BEAMS, 0 FOR PLATES
C      NELEM = THE MAXIMUM NUMBER OF ELEMENTS PER PARTITION
C      NSTRS = THE NUMBER OF STRESSES PER ELEMENT
C      NSIZE = ARRAY OF COLUMN DIMENSION FOR THE STRESS PARTITIONS
       IOUT = MT6
       KSTRES = MT12
       KS2 = MT16


       IF(ITYPE .EQ. 1) NTOT = IBTOT
       IF(ITYPE .EQ. 0) NTOT = IPTOT
       IF(ITYPE .EQ. 1) NUMBER = NBEAM
       IF(ITYPE .EQ. 0) NUMBER = NPLATE
C CALCULATE NUMBER OF ROWS OF PARTITIONS AND ROW DIMENSIONS

       NSTAPE = MT3
       IF (ITYPE .EQ. 0) NSTAPE = MT1
       REWIND NSTAPE
       REWIND KS2

C COUNT NUMBER OF S1 AND S2 PARTITIONS
       NS1 = 0
       NS2 = 0
       DO 4 I=1,NP
       IF (NZ(I) .NE. 0) NS1 = NS1 + 1
       IF (NZ(I) .NE. NSIZE(I)) NS2 = NS2 + 1
     4 CONTINUE

C SET PARAMETER MATRIX AND WRITE OUT AS FIRST FILE OF TAPE NSB OR NSP
       NPARA(1) = NS2
       NPARA(2) = 2*NS2
       NPARA(3) = NS1
       PARA(4) = 1.0
       NPARA(5) = NROW
       DO 6 I =6,50
     6 NPARA(I) = 0.0
       NAME = 0
       CALL WRTETP(NPARA,1,NAME,50,1,B,0,0,NSTAPE,IERROR)
       END FILE NSTAPE
       NCOUNT = 0

C LOOP ON PARTITION ROWS
```

```
      DO 200 I=1,NROW
C LOOP ON PARTITION COLUMNS
      DO 190 J=1,NP
      IMAX = ISIZE(I)
      DO 5 INC=1,IMAX
      DO 5 JNC=1,60
      STRESS(INC,JNC) = 0.0
    5 CONTINUE
      ID = 1000*I + J
      IF(NCOUNT .GE. NTOT) GO TO 150
      READ (KSTRES) NCODE
      IF (NCODE .NE. ID) GO TO 150
C THIS PARTITION IS TO BE SORTED BY COLUMNS.
      NCOUNT = NCOUNT + 1
      JS1 = 0
      JS2 = NZ(J)
C CALCULATE LOCATION OF RETAINED FREEDOMS FOR THIS PARTITION
      IF (J .EQ. 1) GO TO 20
      NFIRST = 0
      DO 15 N1=2,J
      NFIRST = NFIRST + NZ(N1-1)
   15 CONTINUE
      NFIRST = NFIRST + 1
      GO TO 23
   20 CONTINUE
      NFIRST = 1
   23 CONTINUE
      NLAST = NFIRST + NZ(J) - 1
C READ STRESS PARTITION FROM TAPE
      IROW = ISIZE(I)
      ICOL = NSIZE(J)
      READ (KSTRES) ((STRESS(IS,JS),IS=1,IROW),JS=1,ICOL)

C IF NO FREEDOMS ARE RETAINED, PLACE ELEMENTS IN S2.
      IF (NZ(J) .EQ. 0) GO TO 100
C IF ALL FREEDOMS ARE RETAINED, PLACE ELEMENTS IN S1
      IF (NZ(J) .EQ. NSIZE(J)) GO TO 90

C BEGIN SORTING LOOP
      JS = 0
      JS1 = 0
      JS2 = 0
   30 JS = JS + 1
      DO 40 NTEST=NFIRST,NLAST
      IF (NSAVE(NTEST) .EQ. JS) GO TO 60
   40 CONTINUE
C PLACE COLUMN IN S2
      JS2 = JS2 + 1
      N1 = JS1 + 1
      DO 50 IS2=1,IROW
      SAVE(IS2) = STRESS(IS2,N1)
   50 CONTINUE
      N2 = NZ(J)
      N3 = NZ(J) + JS2
      N4 = JS1 + 2

      DO 56 JNC1=N4,N2
      DO 55 INC1=1,IMAX
      STRESS(INC1,JNC1-1) = STRESS(INC1,JNC1)
   55 CONTINUE
```

```
   56 CONTINUE

      DO 57 INC1=1,IMAX
      STRESS(INC1,N2) = STRESS(INC1,N3)
   57 CONTINUE

      DO 58 INC2=1,IMAX
      STRESS(INC2,N3 ) = SAVE(INC2)
   58 CONTINUE

      GO TO 70

   60 CONTINUE
C PLACE COLUMN IN S1
      JS1 = JS1 + 1

   70 CONTINUE
      IF (JS .LT. NSAVE(NLAST)) GO TO 30
      GO TO 120

C ALL FREEDOMS RETAINED. PLACE ALL COLUMNS IN THE S1 PART
   90 CONTINUE
      CALL WRTETP(STRESS,96,ID,IROW,ICOL,B,0,0,NSTAPE,IERROR)
     *INTO S1)
      GO TO 190

C NO RETAINED FREEDOMS. PLACE ALL COLUMNS IN THE S2 PART.
  100 CONTINUE
      CALL WRTETP(STRESS,96,ID,IROW,ICOL,B,0,0,KS2,IERROR)
      GO TO 190

C WRITE OUT S1 AND S2 PARTS
  120 CONTINUE
      IROW = ISIZE(I)
      ICOL = NZ(J)
      J1 = NZ(J) + 1
      CALL WRTETP(STRESS(1,1),96,ID,IROW,ICOL,B,0,0,NSTAPE,IERROR)
      ICOL = NSIZE(J) - NZ(J)
      CALL WRTETP(STRESS(1,J1),96,ID,IROW,ICOL,B,0,0,KS2,IERROR)
      GO TO 190
C WRITE NULL PARTITIONS FOR S1 AND S2 ONTO TAPE
  150 CONTINUE
      IROW = ISIZE(I)
      ICOL = NZ(J)
      IF (NZ(J) .EQ. 0) GO TO 160
      CALL WRTETP(STRESS,96,ID,IROW,ICOL,B,0,0,NSTAPE,IERROR)
  160 CONTINUE
      IF (NZ(J) .EQ. NSIZE(J)) GO TO 170
      ICOL = NSIZE(J) - NZ(J)
      CALL WRTETP(STRESS,96,ID,IROW,ICOL,B,0,0,KS2,IERROR)
  170 CONTINUE
      BACKSPACE KSTRES
  190 CONTINUE
  200 CONTINUE

      REWIND KS2
      END FILE NSTAPE

C COPY S2 FROM KS2 TO FILE 3 OF NSTAPE
      DO 300 I=1,NROW
```

```
      DO 300 J=1,NS2
      ID = 0
      CALL READTP(STRESS,96,ID,IROW,ICOL,B,0,0,KS2,IERR)
      CALL WRTETP(STRESS,96,ID,IROW,ICOL,B,0,0,NSTAPE,IERROR)
  300 CONTINUE
      END FILE NSTAPE
      REWIND NSTAPE
      REWIND KS2
      RETURN

      END
```

SUBROUTINE FREMOD

```
$IBFTC FREMO*  DECK
      SUBROUTINE FREMOD
C
C     FORMULA NUMBERS 1 TO 5999 ARE NORMAL PROGRAM
C     FORMULA NUMBERS 6000 TO 7499 ARE DIAGNOSTICS
C     FORMULA NUMBER 7500 IS THE CALL EXIT STATEMENT
C     FORMULA NUMBERS 8000 TO 8999 ARE INPUT FORMATS
C     FORMULA NUMBERS 9000 TO 9999 ARE OUTPUT FORMATS
C
      DIMENSION DYNMAT(100,100), AMASS(100), CURNTD(100,100),
     1CMAT(100,3), FLEXIB(100,100), GUESS(100), TEMP1(100), VECMAT(100,
     225), TEMP(100), TEMPRY(100), TMPRY(100,3), DIAG(100), B(12),
     3NORMEL(25), FREQ(25), ITER(25), PCTBIG(25), CTMC(3,3), NTAPE(10),
     4BMASS(100),EVAL(100)
C
      EQUIVALENCE  (DYNMAT,FLEXIB,CURNTD,BMASS),(AMASS,VECMAT),
     1(DIFF,IDIFF), (IERNOW,ERRNOW), (ERROR,IERR), (ERLGST,IERLG),
     2              (DIFSML,IDFSML),                (TMPRY(1,1),TEMP),
     3(TMPRY(1,2),TEMPRY),(TMPRY(1,3),TEMP1)
C
      INTEGER T6,T5
      T5 = 5
      T6 =6
      ITP=0
      PCTLMT=10.0
      MAXITR=1500
      NTAPED=3
      SF=1.0
C
   10 READ (T5,8000) N, MODES
C
  140 DO  145  I=1,N
      AMASS(I) = 0.0
      DO  145  J=1,N
      FLEXIB (I,J) = 0.0
  145 CONTINUE
C
C     READ FLEXIBILITY MATRIX
C
  150 NSTART=0
      NEND=0
      NSTART=NEND+1
      ITEMP = 10
      IPART=1
      NAME=0
      NFILE=0
      NMAT = 1
  175 REWIND ITEMP
      CALL READTP(FLEXIB(NSTART,NSTART),100,NAME,K,K,B,NFILE,NMAT,ITEMP
     1,IRR)
      IF(IRR.NE.0)GO TO 6015
  225 NEND=NEND+K
C
C     READ DIAGONAL MASS MATRIX
C
      READ(T5,8005) (AMASS(I),I=NSTART,NEND)
C
```

```
  450 WRITE(T6,9000)SF
      DO 475 I=1,N
  475 WRITE(T6,9005)I,(FLEXIB(I,J),J=1,N)
C
C     SCALE FLEXIBILITY MATRIX
C
  495 DO 500 I=1,N
      DO 500 J=1,N
  500 FLEXIB(I,J)=FLEXIB(I,J)*SF
C
C     FORM DYNAMIC MATRIX FROM FLEXIB AND AMASS
C
      DO 550 I=1,N
      DO 550 J = 1,N
  550 DYNMAT(I,J) = FLEXIB(I,J)*AMASS(J)
C     PUT ORIGINAL DYNAMIC MATRIX ON SCRATCH TAPE
C
  551 REWIND NTAPED
      CALL WRTETP(DYNMAT,100,0,N,N,B,0,0,NTAPED,IRR)
      IF(IRR.NE.0) GO TO 6055
  575 WRITE(T6,9010)
      DO 600 I=1,N
  600 WRITE(T6,9005)I,(DYNMAT(I,J),J=1,N)
C
C     WRITE OUT MASS MATRIX
C
  605 WRITE(6,9015) (I,AMASS(I),I=1,N)
      CALL WRTETP(AMASS,1,0,1,N,B,0,0,NTAPED,IRR)
      IF(IRR.NE.0)GO TO 6040
      ENDFILE NTAPED
      REWIND NTAPED
C
C     GET VALUES AND VECTORS.
C
      CALL VALVCT(DYNMAT,N,MODES,EVAL,VECMAT)
C
C     LOOP TO EXAMINE ROOTS 1 BY 1
C
      REWIND NTAPED
      CALL READTP(DYNMAT,100,0,N,N,B,0,0,NTAPED,IRR)
      IF (IRR.NE.0) GO TO 6060
      DO 1350 MODE=1,MODES
C
C     FORM  D V  AND  EVAL V TO SEE IF THIS MODE IS A
C     GOOD ONE.
C
      DIFSML=1.0E38
      DO 1250 I=1,N
      IF (ABS(VECMAT(I,MODE) - 0.01))1250,1175,1175
 1175 TEMPRY(I) = EVAL(MODE)*VECMAT(I,MODE)
      CALL INRPRD( DYNMAT(I,1),100,VECMAT(1,MODE),1,TEMP(I),N )
      DIFF = ABS(TEMP(I)-TEMPRY(I))*2.0/(ABS(TEMP(I)+TEMPRY(I)))
 1200 IF (IDFSML-IDIFF) 1250,1250,1225
 1225 DIFSML=DIFF
 1250 CONTINUE
      PCTBIG(MODE)=(1.0-DIFSML)*100.0
 1260 IF(PCTBIG(MODE).GE.0.0)GO TO 1270
 1265 PCTBIG(MODE)= 0.0
C
C     IS THE BEST PERCENTAGE GOOD ENOUGH
```

```
C
 1270 IF (PCTBIG(MODE)-PCTLMT) 1275,1325,1325
C
C      THIS MODE SHAPE IS NO GOOD, PRINT OUT ERROR
C
 1275 WRITE(T6,9055)MODE,PCTBIG(MODE),PCTLMT,(TEMPRY(I),I,TEMP(I),I=1,N)
C
C      MODE IS FOUND CORRECTLY -ARE MORE REQUESTED
C
 1325 IF (EVAL(MODE).LT.0.0) GO TO 6500
      IF(EVAL(MODE).EQ.0.0)GO TO 1330
      EVAL(MODE)=SQRT(1.0/EVAL(MODE))
 1330 FREQ(MODE)=EVAL(MODE)/6.28318530
 1350 CONTINUE
C
C      ALL DONE WITH FINDING VECTORS,FORM ORTHOGONALITY
C      MATRIX VT M V
C
      TMINDG=1.0E38
      TMAXDG=-TMINDG
      TMAXOD=TMAXDG
      CALL READTP(BMASS,1,0,NR,NC,B,0,0,NTAPED,IRR)
      IF (IRR .NE. 0) GO TO 6066
      WRITE(T6,9001)
      DO 1400 I=1,MODES
      DO 1360 J = 1,N
 1360 TEMP1(J) = VECMAT(J,I)*BMASS(J)
      DO 1375 J=1,MODES
 1375 CALL INRPRD (TEMP1  , 1 ,VECMAT(1,J),1,TEMP(J),N)
C
C      PRINT VT M V - ONE ROW
C
      WRITE(T6,9065)I,(TEMP(J),J=1,MODES)
      TEMPRY(I)=TEMP(I)
      DO 1400 J=1,MODES
      IF (I-J) 1380,1385,1380
 1380 TMAXOD=AMAX1(ABS(TEMP(J)),TMAXOD)
      GO TO 1400
 1385 TMAXDG=AMAX1(TEMP(J),TMAXDG)
      TMINDG=AMIN1(TEMP(J),TMINDG)
 1400 CONTINUE
      WRITE(T6,9060)TMAXDG,TMINDG,TMAXOD
C
C      PRINT INERTIA MATRIX
C
      WRITE(T6,9070)(I,TEMPRY(I),I=1,MODES)
C
C      PRINT MODE SHAPES
C
      K=MODES
      K1=1
      K2=6
      IF (K-6) 1430,1435,1435
 1430 K2=MODES
 1435 WRITE(T6,9075)K1,K2
      DO 1440 I=1,N
 1440 WRITE(T6,9076)(I,VECMAT(I,J),J=K1,K2)
      K=K-6
      IF (K) 1550,1550,1450
 1450 K1=K2+1
```

```
      IF (K-6) 1430,1430,1525
 1525 K2=K1+5
      GO TO 1435
C
C     WRITE OUT TABLE OF RESULTS
C
 1550 WRITE(T6,9080)(I,EVAL(I),FREQ(I),PCTBIG(I),I,I=1,MODES)
C
C
C     WRITE RESULTS ON TAPE IF DESIRED
C
 1575 NTAPOT = 10
      NAME=0
      NFILE = 2
      NMAT=0
C
C     WRITE FREQUENCIES, MODE SHAPES, INERTIA MATRIX, AND MASS MATRIX.
C
      CALL WRTETP (FREQ,1,NAME,MODES,1,B,NFILE,NMAT,NTAPOT,IRR)
      IF(IRR.NE.0)GO TO 6030
 1600 CALL WRTETP(VECMAT,100,NAME+1,N,MODES,B,0,0,NTAPOT,IRR)
      IF(IRR.NE.0)GO TO 6030
C
      CALL WRTETP(TEMPRY,1,NAME+2,1,MODES,B,0,0,NTAPOT,IRR)
      IF(IRR.NE.0)GO TO 6030
      END FILE NTAPOT
C
      CALL WRTETP(BMASS,1,NAME+3,1,N,B,0,0,NTAPOT,IRR)
      IF(IRR.NE.0)GO TO 6030
      END FILE NTAPOT
C
 1640 GO TO 7500
C
C     THE FOLLOWING ARE ALL DIAGNOSTIC PRINT EXITS
C
 6015 WRITE(T6,9100)
 6016 NTAPDG=ITEMP
      GO TO 7000
 6030 WRITE(T6,9115)NFILE,NMAT
      NTAPDG=NTAPOT
      GO TO 7000
 6040 WRITE(T6,9210)
      NTAPDG=NTAPED
      GO TO 7000
 6055 WRITE  (T6,9200)
 6056 NTAPDG = NTAPED
      GO TO 7000
 6060 WRITE  (T6,9205)
      GO TO 6056
 6066 WRITE  (T6,9215)
      GO TO 6056
 7000 WRITE(T6,9125)NTAPDG,IRR
      GO TO 7500
 6500 WRITE (T6,9155) EVAL,MODE
 7500 CALL UNLOAD (ITEMP)
      RETURN
 8000 FORMAT(5I10)
 8001 FORMAT(5I10/4I10,E20.0)
 8005 FORMAT (7E10.0)
 8010 FORMAT (I10,E10.0)
```

```
8015 FORMAT (I10/(7E10.0))
9000 FORMAT (1H1,5X,70HBELOW IS THE COMPLETE FLEXIBILITY MATRIX UNSCALE
    1D. THE SCALE FACTOR = 1PE16.6/1H0)
9001 FORMAT(1H1)
9005 FORMAT (1H0,I5,1P7E16.6/(E22.6,6E16.6))
9010 FORMAT (1H1,5X,28HBELOW IS THE DYNAMIC MATRIX./1H0)
9015 FORMAT(1H1,5X,25HBELOW IS THE MASS MATRIX./1H0/(10X,I5,F20.7) )
9055 FORMAT (1H1,5X,5HMODE I5,9HHAD ONLY F7.2,53H PERCENT ACCURACY, WHI
    1CH IS UNDER THE GIVEN LIMIT OF F7.2/1H0,5X,58HTHE TWO VECTORS WHIC
    2H SHOULD BE IDENTICAL ARE GIVEN BELOW./1H0/(20X,1PE16.7,3X,I5,2X,E
    316.7))
9060 FORMAT (1H0,5X,77HABOVE IS THE MATRIX V TRANSPOSE  M  V  ,WHERE V
    1IS THE MATRIX OF MODE SHAPES./1H0,5X,44HTHE MAXIMUM AND MINIMUM DI
    2AGONAL VALUES ARE F15.3,5H AND F15.3/1H0,5X,37HTHE MAXIMUM OFF DIA
    3GONAL MAGNITUDE = F15.8)
9065 FORMAT(1H0,I5,3X,1P9E12.2/(9X,1P9E12.2))
9070 FORMAT (1H1,5X,28HBELOW IS THE INERTIA MATRIX./1H0/(10X,I5,3X,F20.
    17))
9075 FORMAT (1H1,5X,31HBELOW ARE THE SHAPES FOR MODES I5,8H THROUGHI5/1
    1H0)
9076 FORMAT (6(3X,I3,F14.7))
9080 FORMAT (1H1,50X,22HTABLE OF FINAL RESULTS
    1/1H0,74X,10HPERCENTAGE
    2/33X,4HMODE,6X,7HRADIANS,10X,9HFREQUENCY,6X,8HACCURACY,7X,4HMODE
    3/33X,6HNUMBER,4X,10HPER SECOND,7X,6HIN CPS,9X,12HOF FREQUENCY,3X
    4,6HNUMBER/(1H0,32X,I3,5X,F14.7,4X,F14.7,4X,F8.3,4X,I3))
9100 FORMAT(1H1,38HFLEXIBILITY MATRIX COULD NOT BE FOUND.)
9115 FORMAT (1H1,65HSPACING ERROR OCCURED WHILE TRYING TO WRITE BINARY
    1TAPE. NFILE = I5,7HNMAT = I5)
9125 FORMAT (1H0,22HCURRENT TAPE IN USE = I5,14H ERROR CODE = I5)
9155 FORMAT(1H1,14H EIGENVALUE = E13.6,11H  FOR MODE I3,71H HAS A NEGAT
    1IVE VALUE, WHICH DOES NOT DESCRIBE A PROPER PHYSICAL SYSTEM)
9200 FORMAT(1H0,43HERROR WHILE WRITING DYNAMIC MATRIX ON TAPE.)
9205 FORMAT(1H0,45HERROR WHILE READING DYNAMIC MATRIX FROM TAPE.)
9210 FORMAT(1H0,40HERROR WHILE WRITING MASS MATRIX ON TAPE.)
9215 FORMAT(1H0,42HERROR WHILE READING MASS MATRIX FROM TAPE.)
     END
```

SUBROUTINE VALVCT

```
$IBFTC WDVALV  DECK
      SUBROUTINE VALVCT(D,N,MODES,EVALS,VECTRS)
      DIMENSION CN(100),SN(100),D(100,100),IR(100),ORDER(100),
     1 EVALS(100),VECTRS(100,25),X(100)
      INTEGER ORDER
      KR=100
C
C     SUBROUTINE HESSEN TRANSFORMS THE DYNAMIC MATRIX TO UPPER HESSENBERG
C     SUBROUTINE HESSEN USES TAPE 4
      CALL HESSEN(D,KR,N,MODES,CN,SN,VECTRS,EVALS,JL,IR,ORDER,X)
C
C     SUBROUTINE QRITER TRANSFORMS THE HESSENBERG MATRIX TO TRIANGULAR,
C     THE EIGENVALUES ARE THE DIAGONAL ELEMENTS.
C     SUBROUTINE QRITER USES TAPE 2
      CALL QRITER(D,KR,N,MODES,CN,SN,VECTRS,EVALS,JL,IR,ORDER,X)
C
C     SUBROUTINE SORTRT ORDERS THE ROOTS ACCORDING TO ABSOLUTE VALUE, LGST FIRST
      CALL SORTRT(D,KR,L,N,MODES,CN,SN,VECTRS,EVALS,JL,IR,ORDER,X)
      ITMO=L-JL+1
      IF(ITMO.LT.MODES)MODES=ITMO
C
C     SUBROUTINE VECTOR COMPUTES THE VECTORS FOR THE TRIANGULAR MATRIX.
      CALL VECTOR(D,KR,L,N,MODES,CN,SN,VECTRS,EVALS,JL,IR,ORDER,X)
C
C     SUBROUTINE TRANS1 TRANSFORMS THE VECTORS TO CORRESPOND TO THE
C     HESSENBERG MATRIX.
C     SUBROUTINE TRANS1 USES TAPE 2
      CALL TRANS1(D,KR,L,N,MODES,CN,SN,VECTRS,EVALS,JL,IR,X)
C
C     SUBROUTINE TRANS2 TRANSFORMS THE VECTORS TO CORRESPOND TO THE
C     ORIGINAL MATRIX.
C     SUBROUTINE TRANS2 USES TAPE 4
      CALL TRANS2(D,KR,L,N,MODES,CN,SN,VECTRS,EVALS,JL,IR,X)
 7500 RETURN
      END
```

SUBROUTINE HESSEN

```
$IBFTC WDHESS  DECK
      SUBROUTINE HESSEN(D,KR,N,MODES,CN,SN,VECTRS,EVALS,JL,IR,ORDER,X)
      DIMENSION CN(KR),SN(KR),D(KR,KR),IR(KR),ORDER(KR),
     1 EVALS(KR),VECTRS(KR,1),X(KR)
      INTEGER ORDER,R,R2,R3
C
C     TRANSFORM ORIGINAL MATRIX TO UPPER HESSENBERG FORM.
      IH = 4
      REWIND IH
      N2 =N-2
      DO 100 R=1,N2
      R2= R+1
      XR = 0.0
      DO 50 I=R2,N
      XR2 =  ABS(D(I,R))
      IF(XR2.LE.XR)GO TO 50
      XR = XR2
      K = I
   50 CONTINUE
      IR(R) = 0
      IF(XR.EQ.0.0 )GO TO 100
      IF(K.EQ.R2)GO TO 65
      IR(R)=K
      DO 55 J=R,N
      XR = D(R2,J)
      D(R2,J) = D(K,J)
   55 D(K,J) =XR
      DO 60 I=1,N
      XR = D(I,R2)
      D(I,R2) = D(I,K)
   60 D(I,K) =XR
   65 R3 =R+2
      DO 80 I=R3,N
      IF(D(I,R).EQ.0.0 )GO TO 80
      XR =-D(I,R)/D(R2,R)
      DO 70 J=R2,N
   70 D(I,J) = D(I,J)+XR*D(R2,J)
      DO 75 J=1,N
   75 D(J,R2 )=D(J,R2 )-XR*D(J,I)
      D(I,R) = -XR
   80 CONTINUE
  100 CONTINUE
C
C     STORE UPPER HESSENBERG MATRIX ON TAPE,
C     USED TO TRANSFORM VECTORS TO CORRESPOND TO ORIGINAL MATRIX.
      WRITE(IH)((D(I,J),J=1,I),I=1,N)
      ENDFILE IH
      REWIND IH
      RETURN
      END
```

SUBROUTINE QRITER

```
$IBFTC WDQRIT  DECK
      SUBROUTINE QRITER(D,KR,N,MODES,CN,SN,VECTRS,EVALS,JL,IR,ORDER,X)
      DIMENSION CN(KR),SN(KR),D(KR,KR),IR(KR),ORDER(KR),
     1 EVALS(KR),VECTRS(KR,1),X(KR)
      INTEGER ORDER
C
C     TRANSFORM UPPER HESSENBERG MATRIX TO TRIANGULAR FORM,
C     EIGENVALUES WILL BE THE DIAGONAL ELEMENTS.
      TOL = 1.E-14
      JS1 = 0
      JS2 = 0
      JCTR = 0
      IQ = 2
      LIMIT = 10
      REWIND IQ
      IDUM = 1
      IDUM2= 0
      N1=N-1
      J1=1
      ITER = 0
   40 ITER = ITER+1
      DO 50 I=1,N1
      IF( ABS(D(I+1,I)).GT. ABS(D(I+1,I+1))*TOL)GO TO 55
   50 CONTINUE
C
C     WRITE ZEROS ON TAPE TO INDICATE END OF DATA.
   52 WRITE(IQ)IDUM,IDUM,(IDUM2,I=1,4)
      REWIND IQ
      GO TO 7500
   55 I1=I .
      J1P=J1+1
      DO 60 I=J1P,N1
      IF( ABS(D(I+1,I)).LE. ABS(D(I+1,I+1))*TOL) GO TO 65
   60 CONTINUE
      J2 = N
      GO TO 70
   65 J2 = I
   70 DO 75 I=1,N
      IF(D(I,I).NE.0.0 )GO TO 80
      WRITE(6,1100)I
   75 CONTINUE
      PRINT 1
      WRITE(6,1)
      CALL EXIT
   80 JL=I
   85 M = J2-1
      B = -D(M,M)-D(J2,J2)
      C = D(M,M)*D(J2,J2)-D(J2,M)*D(M,J2)
      RAD = B**2-4.0 *C
      IF(RAD.LT.0.0 )GO TO 90
      RAD =  SQRT(RAD)
      SHIFT = .50 *(-B+RAD)
      T = .50 *(-B-RAD)
      IF( ABS(SHIFT-D(J2,J2)).GT. ABS(T-D(J2,J2)))SHIFT = T
      GO TO 95
   90 SHIFT =-.50 *B
   95 DO 100 I=J1,J2
```

```
  100 D(I,I) = D(I,I)-SHIFT
C
C     DO QR PRE-MULT.
      DO 120 J= J1,M
  115 RAD =  SQRT(D(J+1,J)**2+D(J,J)**2)
      CN(J) = D(J,J)/RAD
      SN(J) = D(J+1,J)/RAD
      DO 120 I=J,N
      B = CN(J)*D(J,I)+SN(J)*D(J+1,I)
      D(J+1,I) = -SN(J)*D(J,I)+CN(J)*D(J+1,I)
  120 D(J,I) = B
C
C     DO QR POST-MULT.
      DO 140 J=J1,M
      JP = J+1
  135 DO 140 I=1,JP
      B = CN(J)*D(I,J)+SN(J)*D(I,J+1)
      D(I,J+1) = -SN(J)*D(I,J)+CN(J)*D(I,J+1)
  140 D(I,J) = B
      DO 150 I= J1,J2
  150 D(I,I) = D(I,I)+SHIFT
C
C     STORE ELEMENTS USED TO TRANSFORM VECTORS TO CORRESPOND TO HESSENBERG MTX.
      WRITE(IQ)J1,M,(SN(I),CN(I),I=J1,M)
      IF( JS1 .EQ. J1.AND. JS2 .EQ. J2 ) GO TO 200
      JCTR = 1
      JS1 = J1
      JS2 = J2
      IF( ITER .LT. 500 ) GO TO 40
      WRITE(6,1150)
      GO TO 52
 200  JCTR = JCTR + 1
      IF( JCTR .GE. LIMIT ) GO TO 250
      GO TO 40
 250  D(J2,J2-1)=0.
      GO TO 40
 7500 RETURN
 1    FORMAT(1H0,39HDIAGONAL OF TRANSFORMED MATRIX IS ZERO.)
 1100 FORMAT(1H0,16HDIAGONAL ELEMENT, I5, 9H IS ZERO.)
 1150 FORMAT(1H0,32HITERATION LIMIT OF 500 EXCEEDED.)
 1200 FORMAT(1H0,10I10)
 1300 FORMAT(1H0,8E16.8)
 1400 FORMAT(1H0,I5,(7E16.8))
      END
```

SUBROUTINE SORTRT

```
$IBFTC WDSORT  DECK
      SUBROUTINE SORTRT(D,KR,L,N,MODES,CN,SN,VECTRS,EVALS,JL,IR,ORDER,X)
      DIMENSION CN(KR),SN(KR),D(KR,KR),IR(KR),ORDER(KR),
     1 EVALS(KR),VECTRS(KR,1),X(KR)
      INTEGER ORDER
C
      DO 20 I= 1,N
   20 X(I) = D(I,I)
C
C     ORDER THE ROOTS ACCORDING TO ABS.VALUE.
      DO 80 I= JL,N
      A = 0.0
      DO 60 J=JL,N
      IF( ABS(X(J)).LE.A)GO TO 60
      A =  ABS(X(J))
      ORDER(I) = J
   60 CONTINUE
      IF(A.EQ.0.0 )GO TO 85
      J = ORDER(I)
   80 X(J) = 0.0
      L = N
      GO TO 86
   85 L = I-1
C
C     MOVE THE ROOTS BACK INTO ARRAY X IN CORRECT ORDER.
   86 DO 90 I = JL,L
      M = ORDER(I)
   90 X(I) = D(M,M)
      M = JL+MODES-1
      IF(M.GT.L)M=L
C
C     CHECK FOR NEGATIVE ROOTS AMONG THE FREQUENCIES.
      DO 100 I=JL,M
      IF(X(I).GE.0.0)GO TO 100
      WRITE(6,1300)I,X(I)
  100 CONTINUE
      M2=0
      DO 130 I=JL,L
      M2=M2+1
  130 EVALS(M2) = X(I)
 7500 RETURN
 1300 FORMAT(1H0,I5,21H NEGATIVE EIGENVALUE= ,E16.8)
      END
```

SUBROUTINE VECTOR

```
$IBFTC WDVECT  DECK
      SUBROUTINE VECTOR(D,KR,L,N,MODES,CN,SN,VECTRS,EVALS,JL,IR,ORDER,X)
      DIMENSION CN(KR),SN(KR),D(KR,KR),IR(KR),ORDER(KR),
     1 EVALS(KR),VECTRS(KR,1),X(KR)
      INTEGER ORDER
C
C     COMPUTE VECTORS CORRESPONDING TO THE TRIANGULAR MATRIX.
      M = JL+MODES-1
      IF(M.GT.L)M=L
      DO 120 L5 = JL,M
      K = ORDER(L5).
      DO 50 I=JL,N
   50 X(I) = 0.0
      CURRT = D(K,K)
      DO 60 I = JL,K
   60 D(I,I) = D(I,I)-CURRT
      X(K) = 1.0
      J = K
   65 IF(J.EQ.JL) GO TO 80
      IF(D(J-1,J-1).EQ.0.0 ) GO TO 130
      SUM = 0.0
      DO 70 I = J,K
   70 SUM = SUM+D(J-1,I)*X(I)
      X(J-1) = -SUM/D(J-1,J-1)
      J = J-1
      GO TO 65
   80 DO 90 I = JL,K
   90 D(I,I) = D(I,I)+CURRT
      SUM = 0.0
      DO 100 I = JL,K
      CURRT =  ABS(X(I))
      IF(CURRT.GT.SUM)SUM = CURRT
  100 CONTINUE
      DO 110 I = JL,K
110   VECTRS(I,L5)=X(I)/SUM
      K1=K+1
      DO 115 I=K1,N
115   VECTRS(I,L5)=X(I)
120   CONTINUE
      GO TO 7500
  130 PRINT 1
      MODES=L5-1
7500  RETURN
    1 FORMAT(38HOVECTOR SOL. FAILS DUE TO ZERO ON DIAG)
      END
```

SUBROUTINE TRANS1

```
$IBFTC WDTRN1  DECK
      SUBROUTINE TRANS1(D,KR,L,N,MODES,CN,SN,VECTRS,EVALS,JL,IR,X)
      DIMENSION CN(KR),SN(KR),D(KR,KR),IR(KR),
     1 EVALS(KR),VECTRS(KR,1),X(KR)
C
C     TRANSFORM VECTORS TO CORRESPOND TO THE HESSENBERG MATRIX.
      IQ = 2
      DO 50 I = 1,N
      DO 50 J = 1,N
   50 D(I,J) = 0.0
      DO 60 I = 1,N
   60 D(I,I) = 1.00
   70 READ(IQ)J1,M,(SN(I),CN(I),I = J1,M)
      IF(SN(J1).EQ.0.0 .AND.CN(J1).EQ.0.0 ) GO TO 150
      DO 140 J = J1,M
      DO 140 I = 1,N
      SUM = CN(J)*D(I,J)+SN(J)*D(I,J+1)
      D(I,J+1) = -SN(J)*D(I,J)+CN(J)*D(I,J+1)
  140 D(I,J) = SUM
      GO TO 70
  150 REWIND IQ
      DO 170 I = 1,N
  170 X(I) = 0.0
      M = JL+MODES-1
      IF(M.GT.L)M=L
      DO 250 K = JL,M
      DO 200 I=JL,N
  200 X(I)=VECTRS(I,K)
      DO 240 I = 1,N
      VECTRS(I,K)=0.0
  240 CALL INRPRD(D(I,1),KR,X,1,VECTRS(I,K),N)
  250 CONTINUE
      RETURN
      END
```

SUBROUTINE TRANS2

```
$IBFTC WDTRN2  DECK
      SUBROUTINE TRANS2(D,KR,LIN,N,MODES,CN,SN,VECTRS,EVALS,JL,IR,X)
      DIMENSION CN(KR),SN(KR),D(KR,KR),IR(KR),
     1 EVALS(KR),VECTRS(KR,1),X(KR)
      INTEGER R,R2
C
C     TRANSFORM VECTORS TO CORRESPOND TO THE ORIGINAL MATRIX.
      IH = 4
      READ(IH)((D(I,J),J = 1,I),I = 1,N)
      N2 = N-2
      DO 300 K = 1,MODES
      JLK=JL+K-1
      IF(JLK.GT.LIN)GO TO 500
      DO 100 I=1,N
  100 X(I)=VECTRS(I,JLK)
      R=N2
      DO 175 I1=JL,N2
  160 R2 = R+2
      J=N
      DO 170 I2=R2,N
      X(J)=X(J)+D(J,R)*X(R+1)
  170 J=J-1
      L = IR(R)
      IF(L.EQ.0)GO TO 175
      SUM = X(L)
      X(L) = X(R+1)
      X(R+1) = SUM
  175 R=R-1
      SUM = 0.0
      DO 180 I = 1,N
      IF( ABS(X(I)).GT. ABS(SUM)) SUM=X(I)
  180 CONTINUE
      DO 190 I = 1,N
  190 X(I) = X(I)/SUM
  200 DO 250 I=1,N
  250 VECTRS(I,K)=X(I)
  300 CONTINUE
  500 RETURN
      END
```

SUBROUTINE AMERGE

```
$IBFTC AMERG*  DECK
      SUBROUTINE AMERGE( ITAPE, NTAPE, NF1 )
C**** SUBROUTINE TO FORM FULL STIFFNESS.AND FLEXIBILITY MATRIX FROM
C                 SUB-MATRICES
C
C                   *                                          *
C                  *S U B R O U T I N E    M E R G E          *
C                 *                                            *
C                 *  MERGES THE STIFFNESS/FLEXIBILITY MATRIX   *
C                  **                                         *
C                                                            *
C
C        NT          FINAL STIFFNESS/FLEXIBILITY MATRIX SIZE
C        NROW        NO OF ROW PARTITIONS TO MERGE
C        NCOL        NO OF COL PARTITIONS TO MERGE
C
C        SMAT(I,J)   FINAL MERGED NT X NT MATRIX
C
C        NPRNTK = 0 NO PRINTOUT OF STIFFNESS MATRIX
C               = NOT 0  PRINTOUT THE STIFFNESS MATRIX
C
C      ITAPE = A PARAMETR MATRIX, STIFFNESS MATRIX AND THE FLEXIBILITY
C              MATRIX IS STORED ON THIS TAPE.
C      NTAPE = THE MERGED MATRIX IS STORED ON THIS TAPE
C      NF1=1  IF THE STIFFNESS MATRIX IS DESIRED
C      NF1=2  IF THE FLEXIBILITY MATRIX IS DESIRED
C      THE FLEX MATRIX IS USED AS INPUT TO EIGENVALUE-EIGENVECTOR ROUTINE TV-105W
C      THE MAX SIZE OF THE STIFFNESS AND FLEX MATRIX IS (100X100)
C
C
C
       COMMON /PRNT/ NPRNTK
      DIMENSION SMAT(100,100), SCRAT(60,60),B(16), PARAM(50), IPARAM(50)
      EQUIVALENCE (PARAM,IPARAM)
      REWIND ITAPE
      ISUM=0
      JK=0
      IJ=0
      NT=0
      NST = NF1
      NAME=0
      NFILE = 0
      NMAT = 0
      CALL READTP(PARAM,1,NAME,M,N,B,NFILE,NMAT,ITAPE,IRR)
      IF(IRR .NE. 0) GO TO 1010
      NROW=IPARAM(5)
      NCOL=IPARAM(3)
      NMAT = 0
      NFILE = 0
      M=0
C
C
      DO 500 II=1,NROW
      JSUM=0
      ISUM=ISUM+M
      N=0
      NT=NT+M
```

```
      DO 500 I=1,NCOL
      JSUM=JSUM+N
      NAME=0
      CALL READTP(SCRAT,60,NAME,M,N,B,NF1,NMAT,ITAPE,IRR)
      IF(IRR .NE. 0)GO TO 1010
      NF1 = 0
      DO 200 J=1,M
      IJ=ISUM+J
      DO 100 K=1,N
      JK=JSUM+K
  100 SMAT(IJ,JK)=SCRAT(J,K)
  200 CONTINUE
  500 CONTINUE
C
C
      NT=NT+M
      NAME = 0
      NFILE = 0
      NMAT = 0
      CALL WRTETP(SMAT,100,NAME,NT,NT,B,NFILE,NMAT,NTAPE,IRR)
      IF(IRR .NE. 0)GO TO 1020
      IF( NST .NE. 1 ) GO TO 7500
      IF( NPRNTK .EQ. 0 ) RETURN
      WRITE(6,9600) NT,NT
C
      DO 9800 I = 1,NT
      WRITE(6,9720)I,(SMAT(I,J),J=1,NT )
 9800 CONTINUE
C
      GO TO 7500
 1010 WRITE (6,6001) IRR
      GO TO 7500
 1020 WRITE (6,6002) IRR
 6001 FORMAT(22H1ERROR CODE IN READTP=,I3)
 6002 FORMAT(22H1ERROR CODE IN WRTETP=,I3)
 9600 FORMAT(1H1,40X,17H STIFFNESS MATRIX ,8X,I3,4H BY I3/1H0 )
 9720 FORMAT(1H0,I5,1P7E16.6/(E22.6,6E16.6) )
 7500 RETURN
      END
```

SUBROUTINE SMERGE

```
$IBFTC SMERG*  DECK
      SUBROUTINE SMERGE ( ITAPE,NTAPE,ITEST )
C***  SUBROUTINE TO MERGE AND REPARTITION THE STRESS MATRIX
C
C                                                                    *
C           *                                                       *
C          * M E R G E   R O U T I N E   T O   M E R G E           *
C         *                                                         *
C         *  A N D   R E P A R T I T I O N   S T R E S S E S       *
C         *                                                         *
C          * F O R   P L A T E S   A N D   B E A M S               *
C           *                                                      *
C
C
C        ITEST = 8   PLATES
C              = 6   BEAMS
C
C
C        NSTRSP = 0       DONT MERGE THE STRESS FOR PLATES
C               NOT 0     MERGE THE STRESS FOR PLATES
C
C        NSTRSB = 0       DONT MERGE THE STRESS FOR BEAMS
C               NOT 0     MERGE THE STRESS FOR BEAMS
C
C
C
C     ITAPE = THE PARAMETER AND STRESS MATRIX IS STORED ON THIS TAPE
C     NTAPE = THE MERGED MATRIX IS STORED ON THIS TAPE
C
C***ICNT1       COUNT OF END 1 OR END 2
C***ICNT2       COUNT OF NUMBER OF BEAM ELEMENTS
C
      COMMON/PRNT/NPRNTK,NSTRSP,NSTRSB
      DIMENSION SMAT(96,100),SCRAT(96,60),B(16),IPARAM(50),PARAM(50),
     1 DMAT(8,100)
      EQUIVALENCE ( PARAM,IPARAM )
      REWIND ITAPE
      JK = 0
      NT = 0
      NAME = 0
      NFILE = 0
      NMAT  = 0
      CALL READTP( PARAM,1,NAME,M,N,B,NFILE,NMAT,ITAPE,IRR )
      IF( IRR .NE. 0 ) GO TO 1010
C***  THE NO. OF COL. PARTITIONS IS NCOL AND ROW PARTITIONS IS NROW
C
C
      NROW = IPARAM(5)
      NCOL = IPARAM(3)
      NMAT = 0
      NFILE = 0
      M = 0
C
C
      NF1 = 1
      IF( ITEST .EQ. 8 ) GO TO 50
      IF ( NSTRSB .EQ. 0 ) GO TO 60
      WRITE(6,9000)
      GO TO 60
```

```
 50   IF( NSTRSP .EQ. 0 ) GO TO 60
      WRITE(6,9001)
 60   CONTINUE
C***            INITIALIZE ICNT1 AND ICNT2
      ICNT1 = 1
      ICNT2 = 0
      DO 600 II = 1,NROW
C
C
      JSUM = 0
      N = 0
      NC= 0
C
      DO 500 I = 1,NCOL
      JSUM = JSUM + N
      NAME = 0
      CALL READTP(SCRAT,96,NAME,M,N,B,NF1,NMAT,ITAPE,IRR )
      IF( IRR .NE. 0 ) GO TO 1010
      NF1 = 0
      DO 200 J = 1,M
      DO 100 K = 1,N
      JK = JSUM + K
      NK = JK
 100  SMAT(J,JK) = SCRAT(J,K)
 200  CONTINUE
      NC = NC + N
C
 500  CONTINUE
C
C***  RE-PARTITION FOR EACH ELEMENT OF BEAM OR PLATE
      JJ = 0
C***  SET ITEST = 8  FOR PLATES
C***  SET ITEST = 6 FOR BEAMS
      NT = NT + M
      IJ = 0
      DO 550 I = 1,M
      JJ = JJ + 1
      IJ = IJ + 1
      DO 505 J = 1,NC
      DMAT(JJ,J) = SMAT(IJ,J)
 505  CONTINUE
      NMAT = 0
      NAME = 0
      IF( JJ .LT. ITEST ) GO TO 550
      JJ = 0
      CALL WRTETP( DMAT,8,NAME,ITEST,NC,B,NFILE,NMAT,NTAPE,IRR )
      IF( IRR .NE. 0 ) GO TO 1020
C
C
      IF( ITEST .EQ. 8 ) GO TO 510
      IF( NSTRSB .EQ. 0 ) GO TO 550
C***  P R I N T   O U T   B E A M   S T R E S S E S ( 6 X N )
      IF( ICNT2 .NE. 0 ) GO TO 506
      ICNT2 = ICNT2 + 1
      WRITE(6,9002) ICNT2
      GO TO 508
 506  IF( ICNT1 .EQ. 2 ) GO TO 508
      ICNT1 = 1
      ICNT2 = ICNT2 + 1
      WRITE(6,9002) ICNT2
```

```
  508  WRITE(6,9006) ICNT1
       DO 509 LL = 1,ITEST
       WRITE(6,9003) LL,( DMAT(LL,MM),MM=1,NC )
  509  CONTINUE
       ICNT1 = ICNT1 + 1
       GO TO 550
  510  IF( NSTRSP .EQ. 0) GO TO 550
C***   P R I N T    O U T    P L A T E    S T R E S S E S ( 8 X N )
       ICNT2 = ICNT2 + 1
       WRITE(6,9004) ICNT2
       DO 520 LL = 1,ITEST
  520  WRITE(6,9005) LL, ( DMAT(LL,MM),MM = 1,NC )
C
C
  550  CONTINUE
C
C
  600  CONTINUE
C
C
C
       GO TO 7500
 1010  WRITE(6,6001)IRR
       GO TO 7500
 1020  WRITE(6,6002)IRR
       GO TO 7500
 6001  FORMAT(22H ERROR CODE IN READTP= I3 )
 6002  FORMAT(22H ERROR CODE IN WRTETP= I3 )
 9000  FORMAT(1H1,40X,21H BEAM STRESS MATRICES /1H0 )
 9001  FORMAT(1H1,40X,22H PLATE STRESS MATRICES /1H0 )
 9002  FORMAT(1H0,5HBEAM I3 )
 9003  FORMAT(1H0,13X,I3,1P7E16.6/(17X,7E16.6) )
 9004  FORMAT(1H0,5HPLATEI3)
 9005  FORMAT(1H0,8X,I3,1P7E16.6/(12X,7E16.6) )
 9006  FORMAT(1H0,8X,3HENDI2)
 7500  RETURN
       END
```

PHASE I TLO1 DATA LISTING

```
8000B                                   200B        29B         CLEAR
8200B                                               0B          LOAD AND EXEC.
          9000B                                     0B
          -40000000B                                0B
          -80000000B                                0B
          -110000000B                               0B
          -120000000B                               0B
40000000B              8000B                        0B
8003B                  8004B        8005B          27B
          40030000B                                 0B
1008B                               8003B          0B          IS K22 UNPARTITIONED
110000000B             1B                           0B
1B                     1B                           18B
40000000B              2B                           0B
1B        2B           3B                           6B
3B                     80000000B                    0B
-3B                                 8005B          0B
1079B                               999B           0B
110000000B             1B                           0B
1B                     80000000B                    0B
-2B                                 8003B          0B
40000000B              1B                           0B
1B                     80000000B                    0B
-2B                                 8005B          0B
-6B                                 8003B          0B
          -40000000B                                0B
          -80000000B                                0B
          -110000000B                               0B
8005B                               4B             29B
8003B                  8006B        0B             27B         SAVE N FOR LOOPING
8003B                               -1B            27B         DECREMENT N
8004B                               -1B            27B         DECREMENT K
80000000B              1B                           0B          PIVOT PARTITION      VT1
1B                     1B           0B              18B         INVERT
-1014B    -1B                       1000B          27B
80000000B              2B                           0B          REST OF PIVOT ROW    VT2
1B        2B           3B           0B              6B          NEW PIVOT ROW
-1014B                              100B           27B         CHANGE NAME
3B                     110000000B                   0B          ON TAPE 2
-4B                                 8004B          0B          ALL OF PIVOT ROW
          110000000B                                0B
          -110000000B                               0B
80000000B              1B                           0B          PART. TO BE ZEROED   VT3
110000000B             2B                           0B          PIVOT ROW
1B        2B           3B           22B             6B          MULTIPLY
3B                     2B           23B             0B          COPY
80000000B              3B                           0B          ROW OF ZERO          VT4
3B        2B           3B           0B              2B          NEW TERM OF ROW
-1014B                              10B            27B         CHANGE NAME
3B                     12000000B                    0B          TO TAPE 1 OR 3       VT5
-7B                                 8004B          0B          ALL TERMS OF ROW
-10B                                8003B          0B          FOR N-1 ROWS
          -80000000B                                0B                               VT6
          -110000000B                               0B
110000000B             1B                           0B          PIVOT ROW
1B                     12000000B                    0B          TO TAPE 3 OR 1       VT7
-2B                                 8004B          0B          ALL OF THE ROW
          12000000B                                 0B                               VT8
          -12000000B                                0B                               VT9
          -110000000B                               0B
```

```
10138                                   80058       0B          EVERY OTHER TIME
-12918      -1B                         40000008    27B         CHANGE VT1 TO 3
-12718      -1B                         40000008    27B         CHANGE VT2 TO 3
-12118      -1B                         40000008    27B         CHANGE VT3 TO 3
-11818      -1B                         40000008    27B         CHANGE VT4 TO 3
-11638      -1B                         -40000008   27B         CHANGE VT5 TO 1
-11428      -1B                         -40000008   27B         CHANGE VT6 TO 3
-11238      -1B                         -40000008   27B         CHANGE VT7 TO 1
-11128      -1B                         -40000008   27B         CHANGE VT8 TO 1
-11128      -1B                         40000008    27B         CHANGE VT9 TO 1
80058                                   1B          29B         CLEAR
80058                                   2B          27B         SET UP FOR NEXT
10118                                   2558        0B          TO END POINT
-14118      -1B                         -40000008   27B         CHANGE VT1 TO 1
-13918      -1B                         -40000008   27B         CHANGE VT2 TO 1
-13318      -1B                         -40000008   27B         CHANGE VT3 TO 1
-13018      -1B                         -40000008   27B         CHANGE VT4 TO 1
-12838      -1B                         40000008    27B         CHANGE VT5 TO 3
-12628      -1B                         40000008    27B         CHANGE VT6 TO 1
-12438      -1B                         40000008    27B         CHANGE VT7 TO 3
-12328      -1B                         40000008    27B         CHANGE VT8 TO 3
-12328      -1B                         -40000008   27B         CHANGE VT9 TO 3
80058                                   1B          29B         CLEAR
-528                                    8006B       0B
10078                                   8004B       0B
10028                                   8005B       0B          IF 8005=0 SOLUTION ON 8
10048                                   999B        0B          LEAVE ON TP 8
120000008               1B                          0B
1B                      8000000B                    0B          COPY ONTO TP 8
-2B                                     8006B       0B
10068                                   999B        0B
10058                                   8005B       0B          IF 8005=0 SOLUTION IS ON 8
80000008                1B                          0B
1B                      120000008                               COPY ONTO 12
-2B                                     8004B       0B
-3B                                     8006B       0B
          -40000008B                                0B
          -80000008B                                0B
          -110000008B                               0B
          -120000008B                               0B
8175B     1B                            9B          20B
                                                                                    +
      PHASE 1.1  IS COMPLETE
8000B                                   2008        29B         CLEAR 200 CELLS
8200B                                               0B          LOAD+EXECUTE
          -40000008B                                0B          REW 4
          -20000008B                                0B
          -80000008B                                0B          REW 8
          -110000008B                               0B          REW 11
          -120000008B                               0B
          -160000008B                               0B
          9000B                                     0B          PARTITION
40000008B                 8000B                     0B          READ PARAMETERS
8000B                     20000008B                 0B          COPY PARAMETERS
          20000008B                                 0B          EOF T2
1082B                                   8005B       27B
1002B                                   8003B       0B          ONE ROW PARTITION
1013B                                   999B        0B
1004B                                   8005B       0B          ONE COL PARTITION
1011B                                   999B        0B
```

```
10128                                   1B          27B
          12000000B                                 0B        FS T12/F0/M(I)
10028                                   999B        0B
          11999999B                                 0B
120000008           1B                              0B
1B                  80000008B                       0B        X=(K22)INV*K21
-3B                                     8003B       0B        FORM A COLUMN
          -120000008                                0B
-8B                                     8005B       0B
          8000008B                                  0B        EOF
          -8000008B                                           INITIAL REWINDS
          40020008                                  0B
40000008            1B                              0B        READ K12
1B                  110000008B                      0B        STORE K12
-2B                                     8003B       0B        CYCLE N TIMES
          110000008B                                0B        EOF
-4B                                     8005B       0B        CYCLE M TIMES
          -110000008B                               0B        REW 11
          -4000008B                                 0B
110000008B          1B                              0B        READ K12
80000008B           2B                              0B        READ K22-1*K21
1B        2B        3B                              6B        MULTIPLY
3B                  1B                              0B        COPY
1004B                                   999B        0B
110000008B          2B                              0B        READ K12
80000008B           3B                              0B        K22-1 * K21
2B        3B        1B                              30B       MULTIPLY
-3B                                     8003B       0B        CYCLE N-1
1001B                                   999B        0B
          40010008                                  0B
-1021B                                  2B          27B
40000008B           2B                              0B        READ K11
2B        1B        2B                              2B        SUBTRACT
2B                  20000008B                       0B
          -110010008B                               0B        BS T11/F1/M0
-16B                                    8005B       0B
          -8000008B                                 0B        REW 8
          110010008B                                0B        FS T11/F1/M0
-19B                                    8005B       0B
          2000008B                                  0B
          -110000008B                               0B        REW 11
          -2000008B                                 0B
80000008B           1B                              0B
1B                  160000008B                      0B
-2B                                     8003B       0B
-3B                                     8005B       0B
          160000008B                                0B
          -160000008B                               0B
          -8000008B                                 0B        REW 8
          -4000008B                                 0B        REW 4
8160B     1B                            9B          20B       PRINT COMMENT
                                                              EXIT                    +
      PHASE 1.2  IS COMPLETE


8000B                                   100B        29B
8100B                                               0B        LOAD
          -4000008B                                 0B
          -2000008B                                 0B
          9000B                                     0B        PARTITION
```

```
4000000B                8000B                   0B      READ PARAMETERS
          -4000000B                             0B      REW 4
          20010008                              0B
8075B                                   9B      19B     PRINT TITLE
2000000B                1B                      0B      READ K INV
1014B                                   1B      27B     PREPARE NAME
1B                      1B              1000B   0B
1B                                      8075B   20B     PRINT K INV
-4B                                     8005B   0B      CYCLE M
-10348                                  -1B     27B     DECREMENT NAME
-1B                                     8005B   0B
-10548      -1B                         1000B   27B     INCREMENT NAME
-8B                                     8005B   0B      CYCLE M
          -20000008                             0B      REW 7-B1
8055B       1B                          9B      20B     COMMENT
                                                        EXIT                    +

          K RED -- MATRIX NUMBER = I*1000 + J
          PRINTING OF K RED IS COMPLETE


8000B                                   200B    29B     CLEAR
8200B                                           0B      LOAD+EXECUTE
          -4000000B                             0B      REW 4
          -8000000B                             0B      REW 8
          -11000000B                            0B      REW 11
          -2000000B                             0B      REW 2
          9000B                                 0B      PARTITION
4000000B                8000B                   0B      READ PARAMETERS
          -4000000B                             0B
          20010008                              0B      FS T2/F1/M0
1009B                                   999B    0B
2000000B                1B                      0B      READ K
1B          1B          2B                      2B      SUBTRACT
2B          8006B       2B                      17B     ADD 1.0 TO DIAGONAL
2B                      11000000B               0B      STORE IDENTITY MAT
2000000B                1B                      0B      READ K
1B          1B          2B                      2B      SUBTRACT
2B                      11000000B               0B
-3B                                     8005B   0B      CYCLE M
-8B                                     8005B   0B
2000000B                1B                      0B      READ K
1B          1B          2B                      2B      NULL MATRIX
2B          8006B       2B                      17B     ADD 1.0 TO DIA
2B                      11000000B               0B
          11000000B                             0B
          -11000000B                            0B      REW 11
          -20010008                             0B      BS T2/F1/M0
2000000B                1B                      0B      READ K L.HS
1B                      80000008                0B
-2B                                     8005B   0B
11000000B               1B                      0B      READ R.H.S
1B                      80000008                0B
-2B                                     8005B   0B      CYCLE M
-6B                                     8005B   0B      CYCLE M
          80000008                              0B      EOF
          11000000B                             0B      REW 11
          -80000008                             0B      REW 8
          20010008                              0B      FS T2/F1/M0
8003B                                   2B      29B
8003B                                   8005B   27B     8003=M
```

```
8003B                   8004B       8005B       27B     8004=2M
8160B       1B                      9B          20B     PRINT
                                                        EXIT                    +
            PHASE 3 -- PART 1 IS COMPLETE
8200B                                           0B      LOAD+EXECUTE
1014B                               8003B               SKIP IF N NOT 1
            9000B                                   0B  PARTITION
80000000B               1B                      0B      D11
1B                      1B                      18B     INVERT
1004B                               999B
80000000B               2B                      0B      D10
1B          2B          3B                      6B      MULTIPLY
3B                      20000000B               0B      COPY K INV
-3B                                 8004B
            20000000B                           0B      END OF FILE
            -80000000B                          0B
            -20000000B                          0B      REW 2
1073B                               999B        0B
8200B                               5B          20B
            9000B                               0B      PARTITION
            -80000000B                                  INITIAL REWINDS
            -11000000B                          0B
            -12000000B                          0B
            -11000000B
            -12000000B
8005B                               4B          29B
8003B                   8006B       0B          27B     SAVE N FOR LOOPING
8003B                               -1B         27B     DECREMENT N
8004B                               -1B         27B     DECREMENT K
80000000B               1B                      0B      PIVOT PARTITION         VT1
1B                      1B                      18B     INVERT
-1014B      -1B                     1000B       27B
80000000B               2B                      0B      REST OF PIVOT ROW       VT2
1B          2B          3B          0B          6B      NEW PIVOT ROW
-1014B                              100B        27B     CHANGE NAME
3B                      11000000B               0B      ON TAPE 11
-4B                                 8004B       0B      ALL OF PIVOT ROW
            11000000B                           0B
            -11000000B                          0B
80000000B               1B                      0B      PART. TO BE ZEROED      VT3
11000000B               2B                      0B      PIVOT ROW
1B          2B          3B          22B         6B      MULTIPLY
3B                      2B          23B         0B      COPY
80000000B               3B                      0B      ROW OF ZERO             VT4
3B          2B          3B          0B          2B      NEW TERM OF ROW
-1014B                              10B         27B     CHANGE NAME
3B                      12000000B               0B      TO TAPE 1 OR 3          VT5
-7B                                 8004B       0B      ALL TERMS OF ROW
-10B                                8003B       0B      FOR N-1 ROWS
            -80000000B                          0B                              VT6
            -11000000B                          0B
11000000B               1B                      0B      PIVOT ROW
1B                      12000000B               0B      TO TAPE 3 OR 1          VT7
-2B                                 8004B       0B
            12000000B                           0B                              VT8
            -12000000B                          0B                              VT9
            -11000000B                          0B
1013B                               8005B       0B      EVERY OTHER TIME
-1291B      -1B                     40000000B   27B     CHANGE VT1 TO 3
-1271B      -1B                     40000000B   27B     CHANGE VT2 TO 3
```

```
-1211B     -1B                  4000000B   27B       CHANGE VT3 TO 3
-1181B     -1B                  4000000B   27B       CHANGE VT4 TO 3
-1163B     -1B                  -4000000B  27B       CHANGE VT5 TO 1
-1142B     -1B                  -4000000B  27B       CHANGE VT6 TO 3
-1123B     -1B                  -4000000B  27B       CHANGE VT7 TO 1
-1112B     -1B                  -4000000B  27B       CHANGE VT8 TO 1
-1112B     -1B                  4000000B   27B       CHANGE VT9 TO 1
8005B                           1B         29B       CLEAR
8005B                           2B         27B       SET UP FOR NEXT
10111B                          255B       0B        TO END POINT
-1411B     -1B                  -4000000B  27B       CHANGE VT1 TO 1
-1391B     -1B                  -4000000B  27B       CHANGE VT2 TO 1
-1331B     -1B                  -4000000B  27B       CHANGE VT3 TO 1
-1301B     -1B                  -4000000B  27B       CHANGE VT4 TO 1
-1283B     -1B                  4000000B   27B       CHANGE VT5 TO 3
-1262B     -1B                  4000000B   27B       CHANGE VT6 TO 1
-1243B     -1B                  4000000B   27B       CHANGE VT7 TO 3
-1232B     -1B                  4000000B   27B       CHANGE VT8 TO 3
-1232B     -1B                  -4000000B  27B       CHANGE VT9 TO 3
8005B                           1B         29B       CLEAR
-52B                            8006B      0B
10028                           8005B      0B        IF 8005=0.OUT
10111B     -1B                  -4000000B  27B
120000008                1B                 0B        READ
1B                       2000000B           0B
-2B                             8004B      0B        NO OF COL PARTS
-3B                             8006B      0B        NO OF ROW PARTS
           2000000B                        0B        END OF FILE
           -2000000B                       0B        REW 2
           -8000000B                       0B
8160B      1B                   9B         20B       PRINT COMMENT                    +

           PHASE 3 IS COMPLETE


8000B                           100B       29B
8100B                                      0B        LOAD
           -4000000B                       0B
           -2000000B                       0B
           9000B                           0B        PARTITION
40000008                 8000B              0B        READ PARAMETERS
           -4000000B                       0B        REW 4
           2002000B                        0B        FS T7/F2/M0
8075B                           9B         19B       PRINT TITLE
2000000B                 1B                 0B        READ K INV
1014B                           1B         27B       PREPARE NAME
1B                       1B     1000B      0B
1B                              8075B      20B       PRINT K INV
-4B                             8005B      0B        CYCLE M
-1034B                          -1B        27B       DECREMENT NAME
-1B                             8005B      0B
-1054B     -1B                  1000B      27B       INCREMENT NAME
-8B                             8005B      0B        CYCLE M
           -2000000B                       0B        REW 7-B1
8055B      1B                   9B         20B       COMMENT
                                           EXIT                               +

           K INV -- MATRIX NUMBER = I*1000 + J
           PRINTING OF K INV IS COMPLETE
```

```
8000B                                   200B        29B       CLEAR 200 CELLS
8200B                                               0B        LOAD+EXECUTE
          -10000000B                                0B        REW 1
          -8000000B                                 0B        REW 8
          -16000000B                                0B        REW 16
          -11000000B                                0B        REW 11
          9000B                                     0B        PARTITION
10000000B              8000B                        0B        READ PARAMETERS
8000B                  80000000B                    0B        COPY PARAMETERS
          80000000B                                 0B        EOF
          10020000B                                 0B        FS/F2/M0 TO READ S2
10000000B              1B                           0B        READ S2
1B                     110000000B                   0B        STORE S2
-2B                                     8003B       0B        CYCLE N TIMES
          110000000B                                0B        EOF
-4B                                     8007B       0B        CYCLE J TIMES
          -110000000B                               0B        REW 11
          -10020000B                                0B        BS/F2/M0 TO READ S1
110000000B             1B                           0B        READ S2
160000000B             2B                           0B        READ K22-1*K21
1B        2B           3B                           6B        MULTIPLY
3B                     1B                           0B        COPY
1004B                                   999B        0B
110000000B             2B                           0B        READ S2
160000000B             3B                           0B        READ K22-1*K21
2B        3B           1B                           30B       MULTIPLY
-3B                                     8003B       0B        CYCLE N-1 TIMES
10000000B              2B                           0B        READ S1
2B        1B           2B                           2B        SUBTRACT
2B                     80000000B                    0B        STORE SR
          -110010000B                               0B        BS/F1/M0
-13B                                    8005B       0B        CYCLE M TIMES
          -160000000B                               0B        REW 16
          110010000B                                0B        FS/F1/M0
-16B                                    8007B       0B        CYCLE J TIMES
          80000000B                                 0B        EOF
          -110000000B                               0B        REW 11
          -80000000B                                0B        REW 8
          -10000000B                                0B        REW 1
8160B     1B                            9B          20B       PRINT COMMENT
                                                                EXIT                    +
          PHASE P IS COMPLETE


8000B                                   100B        29B
8100B                                               0B        LOAD
          -80000000B                                0B
          9000B                                     0B        PARTITION
80000000B              8000B                        0B        READ PARAMETERS
          80010000B                                 0B
8075B                                   9B          19B       PRINT TITLE
80000000B              1B                           0B        READ S REDUCED
1014B                                   1B          27B       PREPARE NAME
1B                     1B               1000B       0B
1B                                      8075B       20B       PRINT S REDUCED/
-4B                                     8005B       0B        CYCLE M
-1034B                                  -1B         27B       DECREMENT NAME
-1B                                     8005B       0B
-1054B    -1B                           1000B       27B       INCREMENT NAME
-8B                                     8007B       0B        CYCLE J
```

```
          -8000000B                          0B        REW 8
8055B     1B                      9B         20B       COMMENT
                                                       EXIT                    +
          S RED -- MATRIX NUMBER = I*1000 + J
          PRINTING OF S RED IS COMPLETE


8000B                                 200B    29B       CLEAR 200 CELLS
8200B                                         0B        LOAD+EXECUTE
          -30000000B                          0B        REW 3
          -120000000B                         0B        REW 12
          -160000000B                         0B        REW 16
          -110000000B                         0B        REW 11
          9000B                               0B        PARTITION
30000000B           8000B                     0B        READ PARAMETERS
8000B               120000000B                0B        COPY PARAMETERS
          120000000B                          0B        EOF
          30020008                            0B        FS/F2/M0 TO READ S2
30000008            1B                        0B        READ S2
1B                  110000008                 0B        STORE S2
-2B                                   8003B   0B        CYCLE N TIMES
          110000008                           0B        EOF
-4B                                   8007B   0B        CYCLE J TIMES
          -110000008                          0B        REW 11
          -30020008                           0B        BS/F2/M0 TO READ S1
110000008           1B                        0B        READ S2
160000008           2B                        0B        READ K22-1*K21
1B        2B        3B                        6B        MULTIPLY
3B                  1B                        0B        COPY
1004B                                 999B    0B
110000008           2B                        0B        READ S2
160000008           3B                        0B        READ K22-1*K21
2B        3B        1B                        30B       MULTIPLY
-3B                                   8003B   0B        CYCLE N-1 TIMES
30000008            2B                        0B        READ S1
2B        1B        2B                        2B        SUBTRACT
2B                  120000008                 0B        STORE SR
          -110010008                          0B        BS/F1/M0
-13B                                  8005B   0B        CYCLE M TIMES
          -160000008                          0B        REW 16
          110010008                           0B        FS/F1/M0
-16B                                  8007B   0B        CYCLE J TIMES
          120000008                           0B        EOF
          -110000008                          0B        REW 11
          -120000008                          0B        REW 12
          -30000008                           0B        REW 3
8160B     1B                      9B         20B       PRINT COMMENT
                                                       EXIT                    +
          PHASE B IS COMPLETE


8000B                                 100B    29B
8100B                                         0B        LOAD
          -120000008                          0B
          9000B                               0B        PARTITION
120000008           8000B                     0B        READ PARAMETERS
8075B                                 9B      19B       PRINT TITLE
120000008           1B                        0B        READ S REDUCED
1014B                                 1B      27B       PREPARE NAME
1B                  1B                1000B   0B
```

```
1B                              8075B       20B         PRINT S REDUCED
-4B                             8005B       0B          CYCLE M
-10348                          -1B         27B         DECREMENT NAME
-1B                             8005B       0B
-10548      -1B                 1000B       27B         INCREMENT NAME
-8B                             8007B       0B          CYCLE J
            -12000000B                      0B          REW 12
8055B       1B                  9B          20B         COMMENT
                                                        EXIT                        +
            S RED -- MATRIX NUMBER = I*1000 + J
            PRINTING OF S RED IS COMPLETE


$EOF
```

APPENDIX IV

PHASE II PROGRAM LISTING

This appendix contains the following listings:

PHASE II PROGRAM LISTING

PHASE II – MAIN PROGRAM

```
$IBFTC PHASE2  DECK
C
C                    *
C                  *     ( P H A S E     2 )   *
C                *                                *
C              *   R A N D O M   R E S P O N S E     *
C              *                                     *
C              *  S O L U T I O N   P R O G R A M    *
C                *                                 *
C                  *                             *
C                    *        R A N V I B       *
C
C
C      RANVIB - RANDOM VIBRATION ANALYSIS SYSTEM FOR COMPLEX STRUCTURES
C***           CALCULATES THE CROS-POWER SPECTRAL DENSITY, JOINT MOMENTS
C                         AND SPECTRAL MOMENTS.
C
C
C      *** A N A L Y S I S    O P T I O N S ***
C
C    OPTION 1          GENERAL ANALYSIS-ARBITRARY PRESSURE SPECTRA WITH
C                                      VISCOUS DAMPING.
C
C     OPTION 2          SIMPLIFIED ANALYSIS WITH CROSS TERMS-SLOWLY VARYING
C                                      PRESSURE SPECTRA, WITH DAMPING
C                                      PROPORTIONAL TO A LINEAR
C                                      COMBINATION OF INERTIA AND
C                                      STIFFNESS OR  STRUCTURAL DAMPING.
C
C     OPTION 3          SIMPLIFIED ANALYSIS WITHOUT CROSS TERMS-SLOWLY
C                                      VARYING PRESSURE SPECTRA, CROSS
C                                      TERMS DELETED. DAMPING ASSUMPTIONS
C                                      ARE SAME AS OPTION 2.
C
C
C                       FLG1=1  CALCULATE OPTION 1
C                            =2  CALCULATE OPTION 2
C                            =3  CALCULATE OPTION 3
C
C
C
C                       FLG2=1  CALCULATE JOINT DEFLECTION ( JD )
C                            =2  CALCULATE CROSS POWER SPECTRAL DENSITY -
C                                 ( CPSD )
C
C
C                       FLG3=0  NO STRESSES ARE CALCULATED
C                            =1  STRESS ARE CALCULATED
C
C
C                       FLG4=0  NO SECOND MOMENTS ARE CALCULATED
C                            =1  SECOND MOMENTS ARE CALCULATED
C
C
C
      COMMON/BLK1/FREQ,AMASS,OMEGA
      COMMON /BLK2/M,N,G,ALAM,CMU,K,NF,NPLATE,NBEAMS,NLOOP
```

```
      COMMON/BLK3/FLG1,FLG2,FLG3,FLG4,MF,NRI
      DIMENSION IRROR(5),B(16),FREQ(25),AMASS(25),OMEGA(100)
      INTEGER FLG1,FLG2,FLG3,FLG4
      READ(5,8888) FLG1,FLG2,FLG3,FLG4,NPLATE,NBEAMS
      WRITE(6,8999)
      WRITE(6,7000)FLG1,FLG2,FLG3,FLG4,NPLATE,NBEAMS
C
C***INITIALIZE TAPES
C***NTAP1 - PHASE 1 DATA OUTPUT TAPE
C***ITAPE - PHASE 2 MASTER TAPE
C***NTAP2, NTAP3, NTAP8, NTAP11, NTAP12, NTAP15 - SCRATCH TAPES
      NTAP1 = 10
      NTAP3 = 3
      NTAP2 = 4
      NTAP8 = 8
      NTAP11 = 11
      ITAPE = 9
      NTAP12 = 12
      NTAP15 = 15
C
C***REWIND TAPES
      REWIND NTAP1
      REWIND NTAP2
      REWIND NTAP3
      REWIND NTAP8
      REWIND NTAP11
      IRELOC = 0
C
C                          ***READ NATURAL FREQUENCIES***
C
      NAME = 0
      NMAT = 0
      NFILE = 2
      CALL READTP(FREQ,1,NAME,M,1,B,NFILE,NMAT,NTAP1,IRR)
      IF ( IRR .NE. 0 ) GO TO 9985
C
C***FREQUENCIES(FREQ)          CONVERT CYCLES/SEC TO RADIANS BY MULT THE
C                              FREQ BY 2*PI,WHERE PI=3.14159265
      DO 10 I = 1,M
 10   FREQ(I) = FREQ(I)*6.2831853
C
C                          ***READ GENERALIZED MASSES***
C
      NAME = 0
      NMAT = 1
      NFILE = 0
      CALL READTP(AMASS,1,NAME,1,M,B,NFILE,NMAT,NTAP1,IRR)
      IF ( IRR .NE. 0 ) GO TO 9985
      WRITE(6,7001)
      WRITE(6,7002)(I,FREQ(I),AMASS(I),I=1,M)
C
C***                       CARD INPUT NUMBER OF DEGREES OF FREEDOM(N)
C                                  DAMPING CONTANTS(ALAM,AMU,G)
C                                  AND OFF DIAGONAL CONSTANT K
C
      READ(5,9000) ALAM,CMU,G,N,K,NF
      WRITE(6,7003)ALAM,CMU,G,N,K,NF
C
C***  CALL SUBROUTINE TO FORWARD SPACE ON MASTER TAPE UNIT 9 TO START
C     READING THE CORRECT TL01 ROUTINES .
```

```
      CALL TAPOS
C
C
C
C***                          TEST TO SEE IF FREQUENCIES FOR THE CPSD
C                                  CALCULATIONS ARE DESIRED
      IF ( FLG2 .EQ. 2 ) GO TO 40
      IF ( FLG1 .EQ. 1 ) GO TO 40
      GO TO 50
C
C***                          READ IN THE FREQUENCIES(OMEGA)
C
 40   READ(5,9001) ( OMEGA(I),I=1,NF )
C
      WRITE(6,7004)
      WRITE(6,7005) ( I,OMEGA(I),I=1,NF )
C
C
C***                          TEST TO SEE IF OPTIONS 1,2 OR 3 IS DESIRED
C
 50   CONTINUE
C
      GO TO ( 100,378,300), FLG1
 100  CONTINUE
C
C
C
C                    *                    *
C                   *                      *
C                  * O  P  T  I  O  N     1  *
C                   *                      *
C                    *                    *
C
C
C
C                                FLG1 = 1   CALCULATE OPTION 1
C
C***                          FORM THE EXCITATIONS CO- AND QUAD-POWER
C                                  SPECTRAL DENSITY
C
      CALL RANLOD
C
C                             FORM THE REAL AND IMAGINARY PART OF THE
C                                  MATRIX TO BE INVERTED.
C
C
C*SUBROUTINE PEDAN IS CALLED
      CALL PEDAN
C
C***                          FORM THE ADMITTANCE RESPONSE MATRIX
C                              (COMPLEX MATRIX INVERSION AT ALL OMEGAS)
C
C*SUBROUTINE COMINV IS CALLED
      CALL TLO1( ITAPE,IRELOC,IRROR )
      IF( IRROR .NE. 0 ) GO TO 9997
C
C***                          FORM THE DEFLECTION RESPONSE CPSD
C
C*SUBROUTINE CPSD1 IS CALLED
      CALL TLO1( ITAPE,IRELOC,IRROR )
```

```
      IF( IRROR .NE. 0 ) GO TO 9997
      CALL PRINTA
C
      IF ( FLG3 .EQ. 0 ) GO TO 110
      IF ( FLG2 .EQ. 1 ) GO TO 110
C*SUBROUTINE SRESP1 IS CALLED
      CALL TLO1( ITAPE,IRELOC,IRROR )
      IF( IRROR .NE. 0 ) GO TO 9997
      WRITE(6,9020)
      CALL PRINTB
 110  IF ( FLG2 .NE. 1 ) GO TO 1000
      NFILE = 0
      CALL FSF(NFILE,ITAPE,LERR)
C
C
C
C***                          FORM THE INTEGRATION CONSTANTS FOR THE
C                                  TRAPEZOIDAL METHOD ROUTINE
C
      CALL CONS
C
C***                          FORM THE JOINT DEFLECTION BY INTEGRATING
C                                  OVER NF FREQUENCIES-COMPLEX MATRIX
C                                  TRAPEZOIDAL INTEGRATION ROUTINE.
C
C*SUBROUTINE TRAPM IS CALLED
      CALL TLO1( ITAPE,IRELOC,IRROR )
      IF( IRROR .NE. 0 ) GO TO 9997
      CALL PRINTC
      IF ( FLG3 .EQ. 0 ) GO TO 1000
C
C***                          TEST TO SEE IF THE JOINT STRESS CPSD IS
C                                  CALCULATED
C*SUBROUTINE SJNT1 IS CALLED
      CALL TLO1( ITAPE,IRELOC,IRROR )
C
      IF( IRROR .NE. 0 ) GO TO 9997
      WRITE(6,9010)
      NL = 2
      NFF = 1
      CALL PRINTD(NL,NTAP3,NFF)
C***                          TEST TO SEE IF THE DEFLECTION SECOND
C                                  MOMENT IS DESIRED.
      IF ( FLG4 .EQ. 0 ) GO TO 1000
      CALL DSECM1
C*SUBROUTINE SSECM IS CALLED
      CALL TLO1( ITAPE,IRELOC,IRROR )
      WRITE(6,9011)
      NL = 2
      NFF = 1
C
C***THE STRESS SECOND SPECTRAL MOMENT MATRICES ARE PRINTED IN PRINTD
      CALL PRINTD(NL,NTAP3,NFF)
      GO TO 1000
C
C
C
C
C                    *                    *
C                   *                      *
```

```
C                    * O  P  T  I  O  N     3  *
C                     *                       *
C                      *                     *
C
C
C
C
 300  CONTINUE
      NRI = 1
      GO TO ( 310,365 ), FLG2
C***                                CALCULATE JOINT DEFLECTION-OPTION 3
C
C***                                CALC THE CO-PSD ONLY
C
 310  CALL RANLOD
C***                                CALC THE ADMITTANCE INTEGRAL SCALARS
      CALL ADMIN3
C***                                CALC THE JOINT DEFLECTIONS
C*SUBROUTINE DJNT3 IS CALLED
      CALL TLO1( ITAPE,IRELOC,IRROR )
      IF( IRROR .NE. 0 ) GO TO 9997
C***                                SUM THE JOINT DEFLECTIONS AT LIKE MODES
      NO = 1
      MF = M
      CALL ADDMAT(NTAP8,NTAP3,NO)
C***                                TEST TO SEE IF THE STRESS IS DESIRED
      IF ( FLG3 .EQ. 0 ) GO TO 1000
C***                                     CALCULATE THE STRESSES
C*SUBROUTINE SJNT3 IS CALLED
      CALL TLO1( ITAPE,IRELOC,IRROR )
      IF( IRROR .NE. 0 ) GO TO 9997
      WRITE(6,9012)
      NL = 1
      NFF = 1
      CALL PRINTD(NL,NTAP15,NFF)
C
C***                                TEST TO SEE IF THE DEFLECTION SECOND
C                                        MOMENTS ARE DESIRED
      IF ( FLG4 .EQ. 0 ) GO TO 1000
C***                                CALCULATE THE DEFLECTION SECOND MOMENTS
C
C***                                CALC THE SECOND MOM ADMITTANCE INTEGRAL
C                                        SCALARS
C
C***                                CALC THE DEFLECTION SECOND MOMENTS
C
      CALL DSECM3
C***                                CALCULATE THE DEFLECTION JOINT STRESS
C* SUBROUTINE DSJNT3 IS CALLED
      CALL TLO1( ITAPE,IRELOC,IRROR )
      IF ( IRROR .NE. 0 ) GO TO 9997
      WRITE(6,9013)
      NL = 1
      NFF = 1
      CALL PRINTD(NL,NTAP15,NFF)
      GO TO 1000
C***                           *** OPTION 3-DEFLECTION CPSD ***
 365  CONTINUE
C***                                FORM THE CO- AND QUAD-PSD AT NF FREQ.
      CALL RANLOD
C***                                FORM THE ADMITTANCE SCALARS-OPTION 3
```

```
      CALL ADMIT3
C***                            FORM THE DEFLECTION RESPONSE PSD
C*SUBROUTINE CPSD3 IS CALLED
      CALL TLO1( ITAPE,IRELOC,IRROR )
      IF( IRROR .NE. 0 ) GO TO 9997
C***                            SUM THE DEFLECTION RESPONSE CPSD FOR
C                                   ALL LIKE MODES.
      MF = M
      CALL ADDMAT(NTAP8,NTAP3,NF)
      IF ( FLG3 .EQ. 0 ) GO TO 1000
C***                            CALCULATE THE STRESS RESPONSE PSD
C*SUBROUTINE SRESP3 IS CALLED
      CALL TLO1( ITAPE,IRELOC,IRROR )
      IF( IRROR .NE. 0 ) GO TO 9997
      WRITE(6,9014)
      NL = 1
      NFF = NF
      CALL PRINTD(NL,NTAP15,NFF)
      GO TO 1000
  378 CONTINUE
C
C
C
C                      *                       *
C                     *                         *
C                    * O  P  T  I  O  N     2   *
C                     *                         *
C                      *                       *
C
C
C
C                           *** OPTION 2-JOINT DEFLECTIONS ***
C
      NLOOP = (M-1)*K - K*(K-1)/2
C                               FORM THE CO- AND QUAD-PSD
      GO TO ( 380,400 ), FLG2
  380 CALL RANLOD
      CALL ADMIN3
      CALL TLO1(ITAPE,IRELOC,IRROR)
      IF ( IRROR .NE. 0 ) GO TO 9997
      NRI = 1
      NO = 1
      MF = M
      CALL ADDMAT(NTAP8,NTAP3,NO)
C***                            FORM THE ADMITTANCE INTEGRAL SCALARS
      CALL ADMIN2
C***                            CALCULATE THE EXCITATIONS
      CALL CQJD
C***                            FORM THE JOINT DEFLECTION CPSD
C*SUBROUTINE DJNT2 IS CALLED
      CALL TLO1( ITAPE,IRELOC,IRROR )
      IF( IRROR .NE. 0 ) GO TO 9997
C***                            SUM THE JOINT DEFLECTIONS WITH ALL CROSS
C                                   MODES INCLUDED WITH LIKE MODE EFFECTS.
      NRI = 2
      NO = 1
      MF = M-1
      CALL ADDMAT(NTAP8,NTAP2,NO)
      NO = 1
      CALL SUMT(NO,NTAP2)
```

```
C***                                SUM OPTION 2 EFFECT TO OPTION 3
      NO = 2
      NCN = 1
      CALL SUM2(NTAP3,NTAP2,NO,NCN)
C***                               TEST TO SEE IF THE SECOND MOMENT SCALARS
C                                       IN THE FATIGUE LIFE CALC ARE TO BE
C                                       COMPUTED.
      IF ( FLG3 .EQ. 0 ) GO TO 1000
C*SUBROUTINE SJNT2 IS CALLED
      CALL TLO1( ITAPE,IRELOC,IRROR )
      IF ( IRROR .NE. 0 ) GO TO 9997
      WRITE(6,9015)
      NL = 2
      NFF = 1
      CALL PRINTD(NL,NTAP15,NFF)
      IF ( FLG4 .EQ. 0 ) GO TO 1000
      CALL SECM3
      CALL TLO1(ITAPE,IRELOC,IRROR)
      IF ( IRROR .NE. 0 ) GO TO 9997
      NRI = 1
      NO = 1
      MF = M
      CALL ADDMAT(NTAP8,NTAP3,NO)
C
C***                               FORM THE SEC MOM ADM INT SCALARS
      CALL SECM2
C***                               CALCULATE THE EXCITATIONS
      CALL CQJD
C***                               FORM THE DEFLECTION SECOND MOMENTS
C*SUBROUTINE DSECM2 IS CALLED
      CALL TLO1( ITAPE,IRELOC,IRROR )
      IF( IRROR .NE. 0 ) GO TO 9997
      NRI = 2
      NO = 1
      MF = M-1
      CALL ADDMAT(NTAP8,NTAP2,NO)
      NO = 1
      CALL SUMT(NO,NTAP2)
      NO = 2
      NCN = 2
      CALL SUM2(NTAP3,NTAP2,NO,NCN)
C
C*SUBROUTINE SECM2 IS CALLED
      CALL TLO1( ITAPE,IRELOC,IRROR )
      IF ( IRROR .NE. 0 ) GO TO 9997
      WRITE(6,9016)
      NL = 2
      NFF = 1
      CALL PRINTD(NL,NTAP15,NFF)
      GO TO 1000
 400  CONTINUE
C                                *** OPTION 2-CPSD ***
C***                                CALC THE CO- AND QUAD-(PSD OPTION 2)
      CALL RANLOD
      CALL ADMIT3
      CALL TLO1(ITAPE,IRELOC,IRROR)
      IF ( IRROR .NE. 0 ) GO TO 9997
      NRI = 1
      MF = M
      CALL ADDMAT(NTAP8,NTAP3,NF)
```

```
C***                            FORM THE ADMITTANCE SCALARS
      CALL ADMIT2
C***                            CALC THE EXCITATIONS FOR OPTION 2-CPSD
      CALL CQCPSD
C***                            CALC THE DEFLECTION RESPONSE CPSD
C*SUBROUTINE DRESP2 IS CALLED
      CALL TLO1( ITAPE,IREL-C,IRROR )
      IF( IRROR .NE. 0 ) GO TO 9997
C***                            SUM OVER ALL CROSS-MODE EFFECTS
      NRI = 1
      NO = 2*NF
      MF = M-1
      CALL ADDMAT(NTAP8,NTAP2,NO)
      NO = NF
      CALL SUMT(NO,NTAP2)
C***                            SUM THE CROSS-MODE EFFECTS TO OPTION 3
C                                   LIKE MODES TO GET THE TOTAL DEFLECTION
C                                   RESPONSE CPSD
      WRITE(6,9018)
      CALL SUM3(NTAP3,NTAP2,NTAP8)
C
C
C***                            TEST TO SEE IF THE STRESSES ARE DESIRED
      IF ( FLG3 .EQ. 0 ) GO TO 1000
C***                            CALCULATE THE STRESS RESPONSE CPSD
C*SUBROUTINE SRESP2 IS CALLED
      CALL TLO1( ITAPE,IRELOC,IRROR )
      IF( IRROR .NE. 0 ) GO TO 9997
      WRITE(6,9017)
      CALL PRINTE
C
      GO TO 1000
 9985 WRITE(6,9990) IRR
      RETURN
 9997 WRITE(6,9998)
      WRITE(6,9999) IRROR
 1000 CONTINUE
      WRITE(6,9019)
      CALL UNLOAD(NTAP1)
      RETURN
 7000 FORMAT(1H0,16HOPTION CONTROLS 7HFLAG 1=I2,5X,7HFLAG 2=I2,5X,
     17HFLAG 3=I2,5X,7HFLAG 4=I2,5X,7HNPLATE=I4,5X,7HNBEAMS=I4///)
 7001 FORMAT(10X,32HNATURAL FREQUENCIES(RADIANS/SEC),10X,18HGENERALIZED
     1MASSES//)
 7002 FORMAT(20X,I5,5X,E14.7,10X,E14.7)
 7003 FORMAT(1H0,7HLAMBDA=F12.6,5X,3HMU=F12.6,5X,2HG=F12.6,5X,2HN=I5,
     15X,2HK=I5,5X,3HNF=I5////)
 7004 FORMAT(1H1,31HCROSS-PSD FREQUENCIES(RAD/SEC) ///)
 7005 FORMAT(1X,I5,5X,E14.7)
 8888 FORMAT(6I10)
 8999 FORMAT(1H1,25X,74HRANDOM VIBRATION ANALYSIS SYSTEM FOR COMPLEX STR
     1UCTURES (  R A N V I B  )  //// )
 9000 FORMAT( 3F10.0,3I10 )
 9001 FORMAT(7F10.0)
 9010 FORMAT(1H1,40X,26HSTRESS COVARIANCE MATRICES ////)
 9011 FORMAT(1H1,40X,38HSTRESS SECOND SPECTRAL MOMENT MATRICES ////)
 9012 FORMAT(1H1,40X,38HSTRESS COVARIANCE MATRICES (REAL PART)  ///)
 9013 FORMAT(1H1,40X,49HSTRESS SECOND SPECTRAL MOMENT MATRIX (REAL PART)
     1   ////)
 9014 FORMAT(1H1,40X,25HSTRESS CROSS PSD MATRICES   ///)
```

```
9015 FORMAT(1H1,40X,26HSTRESS COVARIANCE MATRICES ////)
9016 FORMAT(1H1,40X,39HSTRESS SECOND SPECTRAL MOMENT MATRICES ////)
9017 FORMAT(1H1,40X,25HSTRESS CROSS-PSD MATRICES ////)
9018 FORMAT(1H1,40X,30HDEFLECTION CROSS PSD MATRICES  ////)
9019 FORMAT(1H0,30X,58H R A N V I B    P R O G R A M    I S    C O M P
    1L E T E D  )
9020 FORMAT(1H1,40X,25HSTRESS CROSS-PSD MATRICES ////)
9990 FORMAT(28H ERROR IN READTP-ERROR CODE=I5)
9998 FORMAT(104H 5 INSTRUCTION FIELDS OF THE CARD THAT TL01 WAS TRYING
    1TO INTERPRET AT THE TIME AN ERROR WAS ENCOUNTERED )
9999 FORMAT( 5(5X,I10) )
     END
```

SUBROUTINE SCALE

```
$IBFTC SCALE*  DECK
      SUBROUTINE SCALE(FREQ,NFREQ,SCAL)
C***  SUBROUTINE TO CALCULATE THE ADMITTANCE INTEGRAL SCALE FACTOR
C     AND SCALE THE FREQUENCIES
C***  FREQ-THE SCALED FREQUENCIES
C***  NFREQ-NO OF FREQUENCIES
C***  SCALE-THE SCALE FACTOR
C  SCAL         SCALING FACTOR, DETERMINED BY EXAMINING THE MAGNITUDE
C               OF THE FIRST NATURAL FREQUENCY.  THIS ROUTINE WILL ONLY
C               SCALE FREQUENCIES IN THE RANGE OF ( 0 TO 100,000 ). AFTER
C               THE SCALING FACTOR IS FOUND ALL FREQ. WILL BE SCALED.
C          ****(FREQUENCY RANGE)* * * * * * * * * * * * * * * *SCALE FACTOR
C                 0 TO 100 INCLUSIVE                                    10
C           GREATER THAN 100 AND LESS THAN OR EQUAL TO 1,000            100
C           GREATER THAN 1,000 AND LESS THAN OR EQ TO 10,000           1,000
C           GREATER THAN 10,000 AND LESS THAN OR EQ TO 100,000        10,000
C
C
C               AFTER THE QUOTIENT HAS BEEN FOUND THE RESULT IS DIVIDED
C               BY THE SCALE FACTOR CUBED TO OBTAIN THE CORRECT RESULTS.
C               THIS ROUTINE WILL SCALE THE FREQUENCIES.  THERE WILL BE
C               AN OVERFLOW IN THE CALCULATIONS OF THE ADMITTANCE
C               INTEGRALS IF THE FREQUENCIES ARE NOT SCALED.
C
C
C
      DIMENSION FREQ(1)
      DO 100 I=2,5
      P=10**I
      IF(FREQ(1)-P)20,20,100
 20   SCAL = P/10.
      GO TO 200
 100  CONTINUE
 200  DO 300 I=1,NFREQ
 300  FREQ(I) = FREQ(I)/SCAL
      RETURN
      END
```

SUBROUTINE TAPOS

```
$IBFTC TAPOS*  DECK
       SUBROUTINE TAPOS
       COMMON /BLK3/FLG1,FLG2,FLG3,FLG4
       INTEGER FLG1, FLG2, FLG3, FLG4
C
C
C          SUBROUTINE TO POSITION TO THE START OF THE TL01 ROUTINES
C        ON THE MASTER TAPE UNIT 9 FOR THE JOINT DEFLECTIONS AND
C        CROSS POWER SPECTRAL DENSITY CALCULATIONS FOR OPTIONS 1, 2,
C        AND 3 .
C
C
C                   FLG1 = 1   OPTION 1
C                        = 2   OPTION 2
C                        = 3   OPTION 3
C
C
C                   FLG2 = 1   JOINT DEFLECTIONS
C                        = 2   CROSS PSD
C
C
C                   FLG3 = 0   NO STRESSES
C                        = 1   STRESSES
C
C
C                   FLG4 = 0   NO DEFLECTION SECOND MOMENTS
C                        = 1   DEFLECTION SECOND MOMENTS
C
C
C        ROUTINE FSF IS USED TO DO THE FORWARD SPACING OF THE NUMBER
C        OF FILE MARKS .
C
C
C
      ITAPE = 9
      GO TO ( 100, 200, 300 ), FLG1
C
C***  O P T I O N   1
C
 100  GO TO ( 10, 20 ), FLG2
 10   NFILE = 5
      GO TO 500
 20   NFILE = 5
      GO TO 500
C
C***  O P T I O N   2
C
 200  GO TO ( 210, 220 ), FLG2
 210  NFILE = 3
      GO TO 500
 220  NFILE = 4
      GO TO 500
C
C
C***  O P T I O N   3
C
 300  GO TO ( 310, 320 ), FLG2
```

```
310   NFILE = 1
      GO TO 500
320   NFILE = 2
500   CALL FSF ( NFILE, ITAPE, LERR )
      RETURN
      END
```

SUBROUTINE RANLOD

```
$IBFTC RANLO*  DECK
      SUBROUTINE  RANLOD
C               FORCE CROSS POWER SPECTRAL DENSITY GENERATION PROGRAM
C
C          DEFINITION OF OPTION FLAGS IN COMMON BLK3
C               FLG1 = 1  SOLUTION OPTION 1
C                    = 2  SOLUTION OPTION 2
C                    = 3  SOLUTION OPTION 3
C               FLG2 = 1  JOINT DEFLECTIONS
C                    = 2  CPSD
C               FLG3 AND FLG4 ARE NOT USED IN THIS PROGRAM
C          MISCELLANEOUS
C               NFREQ =  NO. OF NATURAL FREQUENCIES
C               NRTNDS=  NO. OF D.O.F. (RETAINED NODES)
C               KDIG = NO. OFF DIAGONAL TERMS DESIRED
C               NF =  NO. OF SELECTED CPSD FREQUENCIES
      COMMON/BLK1/FREQ,AMASS,OMEGA
      COMMON/BLK2/NFREQ,NSIZE ,G,ALAM,CM,KDIG,NF
      COMMON/BLK3/FLG1,FLG2,FLG3,FLG4
      COMMON/BLK4/IT5,IT6,ITAPE,JTAPE,IFILE,IMAT,IRR,NAME,NDIM,C
      COMMON/BLK5/N1,N2,CRD1,CRD2,AREA,IPT
      COMMON/BLK6/PHI,CF,QF
      INTEGER FLG1,FLG2,FLG3,FLG4
      DIMENSION FREQ(25),AMASS(25),OMEGA(100)
C
      DIMENSION IPT(4),CRD1(100),CRD2(100),AREA(100),PHI(100)
      DIMENSION CF(85,85),QF(85,85),C(12)
      DIMENSION TITLE(14)
C************************ TAPE AND I/O INITIALIZATION SECTION **********
      I3FLG = 0
      IT5 = 5
      IT6 = 6
      ITAPE = 17
      JTAPE = 14
      IFG = FLG2
      IF ( FLG1 .EQ. 1 ) FLG2 = 2
C
      IFILE = 0
      IMAT = 0
      IRR = 0
      NAME = 0
      NDIM = 85
C******   SPECIAL HANDLING IS REQUIRED FOR OPT. 2
C          ASK QUESTION - IS THIS OPTION 2
      IF (FLG1.NE.2) GO TO 19
C
      I3FLG = 2
      FLG1 =  3
C*****     OPTION 3 IS DONE FIRST, THEN OPTION 2 ,DIFFERENT TAPE
C          SETUP REQUIRED FOR J.D. AND CPSD
 19   IF (FLG1-1) 40,40,20
 20   IF (FLG1-2)  30,30,40
 30   REWIND JTAPE
      IF (FLG2-1)  40,40,50
 40   REWIND ITAPE
 50   IF (I3FLG-2) 55,55,53
 53   IF (FLG2-2) 80,210,210
```

```
 55   READ(IT5,1000) TITLE
C
C************************ INPUT DATA SECTION ***************************
C
      READ(IT5,1001) (IPT(I),I=1,4)
      READ(IT5,1002) D,CX,CY
      READ(IT5,1001) N1,N2
      READ(IT5,1002) (CRD1(I),I=1,N1)
      READ(IT5,1002) (CRD2(I),I=1,N2)
      IT= IPT(1)
      GO TO (60,70), IT
C
C     READ AREAS
C
 60   READ(IT5,1002) (AREA(I),I=1,NSIZE)
      GO TO 80
C
C     CALCULATE  AREAS
C
 70   CALL ARIA
C
C     COMPUTE NO. OF FREQS FOR LIMIT ON READING PHI,S
C
 80   GO TO (100,90), FLG2
C
C     CPSD SOLUTION-NO. OF FREQUENCIES EQUAL  NF
C
 90   ILIM  = NF
      GO TO 130
C
C     JOINT DEFLECTION SOLUTION
C
 100  GO TO (110,110,120),FLG1
C
 110  ILIM =  (NFREQ-1)*KDIG -  (KDIG*(KDIG-1))/2
      GO TO 130
 120  ILIM = NFREQ
C
 130  READ(IT5,1002) (PHI(I),I=1,ILIM)
      IF (I3FLG - 2)  131,131,135
C
C
C*****M******************* PRINT INPUT SECTION ***************************
  131 WRITE(IT6,2000)
      WRITE(IT6,2001)
      WRITE(IT6,2002) TITLE
      WRITE(IT6,2003)
      WRITE(IT6,2004)(IPT(I),I=1,4)
      WRITE(IT6,2005)D,CX,CY
      WRITE(IT6,2006) N1,N2
      WRITE(IT6,2007)
      WRITE(IT6,2008)(CRD1(I),I=1,N1)
      WRITE(IT6,2009)
      WRITE(IT6,2008) (CRD2(I),I=1,N2)
      WRITE(IT6,2010)
      WRITE(IT6,2008)(AREA(I),I=1,NSIZE)
 135  WRITE(IT6,2011)
      WRITE(IT6,2008) (PHI(I),I=1,ILIM)
C********************** CALCULATE CT AND THETA  **********************
C                        FOR PROGRESSIVE WAVE
```

```
C
      IF(I3FLG-2) 136,136,134
  134 WRITE(IT6,2014)
      GO TO 210
  136 IF (IPT(2)-1) 160,160,140
  140 CALL CONST(CX,CY,CT,THETA)
      WRITE(IT6,2012)
      WRITE(IT6,2013) CT,THETA
C     REARRANGE COORDINATES FOR SEPERATION CALCULATION IN NOISOR
C
  160 DO 170 I=1,N1
      CRD1(I)=CRD1(I+1)
  170 CONTINUE
      DO 180 I=1,N2
      CRD2(I)=CRD2(I+1)
  180 CONTINUE
      N1 = N1 - 2
      N2 = N2 - 2
C
C     ASSURE THE FREQUENCIES ARE STORED IN OMEGA
C
      IF(FLG2-1)  190,190,210
  190 DO 200 I=1,NFREQ
      OMEGA(I)= FREQ(I)
  200 CONTINUE
C
C     SET UP TO CALL GENERATION ROUTINE - NOISOR
C
C     IS THIS A JOINT DEFLECTION OPTION 2 SOLUTION
  210 GO TO(220,240), FLG2
C     YES - JOINT DEFLECTION
  220 GO TO(240,230,250), FLG1
C
C     YES - OPTION 2
  230 ILIM = NFREQ - 1
      KLIM = 0
      GO TO 260
  240 KLIM = NF
      ILIM = 1
      GO TO 260
  250 ILIM = 1
      KLIM = NFREQ
C
  260 WRITE (IT6,2016)
      CALL NOISOR(ILIM,KLIM,D,CT,THETA,CX,CY)
C
C*************************** CLOSE OFF TAPES ************************
C
      GO TO (280, 270, 280),FLG1
  270 END FILE JTAPE
      REWIND JTAPE
      GO TO (280,290), FLG2
  280 END FILE ITAPE
      REWIND ITAPE
  290 IF (I3FLG-2)320,310,320
C
C******   IF THIS IS AN OPT 2 SOLUTION, THE PROGRAM MUST BE RECYCLED
  310 FLG1 = 2
      ITAPE  = 2
      I3FLG = 3
```

```
      GO TO 19
  320 IF ( FLG1 .EQ. 1 ) FLG2 = IFG
      WRITE(IT6,2015)
      RETURN
C
C*********************** I N P U T  F O R M A T S **********************
C
 1000 FORMAT(14A6)
 1001 FORMAT(7I10)
 1002 FORMAT(7F10.0)
C*********************** O U T P U T  F O R M A T S ********************
C
 2000 FORMAT(1H1,50X,28HRANDOM LOADING MODULE OUTPUT///)
 2001 FORMAT(1H0,86HFORCE CROSS POWER SPECTRAL DENSITY FOR DECAYED PROGR
     1ESSIVE WAVES FOR USE WITH PANEL - )
 2002 FORMAT(1H ,14A6///)
 2003 FORMAT(1H0,55X,18HI N P U T  D A T A//)
 2004 FORMAT(1H0,104HTHE FOLLOWING FOUR OPTIONS HAVE BEEN SELECTED (THEY
     1 APPEAR IN THE ORDER IN WHICH THEY WERE CARD INPUT) -//1X,16HOPTIO
     2N(S) (1) = ,I2, 8H  (2) = ,I2,8H  (3) = ,I2,8H  (4) = ,I2/)
 2005 FORMAT(1H0,28HPARAMETER VALUES ARE - D  = ,E15.8/24X, 5HCX = ,E15.
     18/24X, 5HCY = ,E15.8/)
 2006 FORMAT(1H0, 66HNUMBER OF COORDINATES IN THE DIRECTION OF CYCLIC NO
     1DE NUMBERING - ,I3/1X, 94HNUMBER OF COORDINATES IN THE DIRECTION -
     2ERPENDICULAR.TO THE CYCLIC NODE NUMBERING DIRECTION - ,I3/)
 2007 FORMAT(1H0, 60HORIGIN-TO-NODE LINE DISTANCES IN THE CYCLIC DIRECTI
     1ON ARE - )
 2008 FORMAT(1H0,6E16.7/(E17.7,5E16.7))
 2009 FORMAT(1H0, 74HORIGIN-TO-NODE LINE DISTANCES PERPENDICULAR TO THE
     1CYCLIC DIRECTION ARE - )
 2010 FORMAT(1H0,42HAREA ASSOCIATED WITH EACH RETAINED NODE - )
 2011 FORMAT(1H0,40HPRESSURE POWER SPECTRAL DENSITIES ARE - )
 2012 FORMAT(//1H0,28H*****COMPUTED CONSTANTS*****)
 2013 FORMAT(1H0,38HTRACE VELOCITY OF PRESSURE WAVE, CT = ,E15.8/1X,66HA
     1NGLE BETWEEN TRACE WAVE PROPAGATION DIRECTION AND X-AXIS, THETA =
     2,E15.8)
 2014 FORMAT(///1H0,102H***** THIS IS AN OPTION 2 SOLUTION - RESULTS OF
     1OPTION 3 APPEAR ABOVE FOLLOWED BY OPTION 2 BELOW ***** )
 2015 FORMAT(1H0, 29H**** GENERATION COMPLETE ****)
 2016 FORMAT(1H0,75HTHE FOLLOWING FORCE CROSS POWER SPECTRAL DENSITY MAT
     1RICES WERE GENERATED - )
      END
```

SUBROUTINE ARIA

```
$IBFTC ARIA**  DECK
       SUBROUTINE ARIA
C          THIS ROUTINE CALCULATES THE NODAL AREAS FOR THE RETAINED
C          NODES IN A RECTANGULAR PANEL CONFIGURATION. THE GRID CAN
C          HAVE UNEQUALLY SPACED COORDINATES IN EITHER OR BOTH THE
C          X AND Y DIRECTION. THE COORDINATE ORIGIN CAN BE (0,0) OR
C          (X0,Y0).
C
C                     DESCRIPTION OF CALLING SEQUENCE
C          1.    X-ARRAY  THE SET OF X-COORDINATES FOR THE RETAINED
C                CRD1     NODES, WHERE X(1) AND X(LAST) ARE THE
C                         BOUNDARY VALUES -INPUT
C          2.    Y-ARRAY  THE SET OF Y-COORDINATES CORRESPONDING -INPUT
C                CRD2     WITH X - INPUT
C          3.    NX-N1    THE NUMBER OF RETAINED NODES IN THE X-DIRECTION
C          4.    NY-N2    THE NUMBER OF RETAINED NODES IN THE Y-DIRECTION
C          5.    AREA     THE SET OF AREAS FOR RETAINED NODES - OUTPUT
C
C      L IS A COUNTER   L=1,NO. OF RETAINED NODES = NX*NY
       COMMON/BLK1/FREQ,AMASS,OMEGA
       COMMON/BLK2/NFREQ,NSIZE ,G,ALAM,CM,KDIG,NF
       COMMON/BLK3/FLG1,FLG2,FLG3,FLG4
       COMMON/BLK4/IT5,IT6,ITAPE,JTAPE,IFILE,IMAT,IRR,NAME,NDIM,C
       COMMON/BLK5/N1,N2,CRD1,CRD2,AREA,IPT
       COMMON/BLK6/PHI,CF,QF
       INTEGER FLG1,FLG2,FLG3,FLG4
       DIMENSION FREQ(25),AMASS(25),OMEGA(100)
C
       DIMENSION IPT(4),CRD1(100),CRD2(100),AREA(100),PHI(100)
       DIMENSION CF(85,85),QF(85,85),C(12)
       L = 0
       M = N1 - 1
       N = N2 - 1
C
       DO 100 I=2,N
C
       DELY = (CRD2(I+1)- CRD2(I-1))/4.0
       DO 100 J= 2,M
C
       L = L + 1
C
       AREA(L) =(CRD1(J+1)- CRD1(J-1))*DELY
  100  CONTINUE
       RETURN
       E=D
```

SUBROUTINE CONST

```
$IBFTC CONST*  DECK
       SUBROUTINE CONST(CX,CY,CT,THETA)
C          THIS ROUTINE CALCULATES THE TRACE VELOCITY (CT) AND
C          THETA - THE ANGLE BETWEEN DIRECTION OF SOUND PROPAGATION
C          AND X AND Y AXIS OF PANEL IN THE CASE OF A PROGRESSIVE
C          WAVE.CX IS THE PHASE VELOCITY ALONG PANEL IN X-DIRECTION
C          CY IN Y-DIRECTION.
C          CALCULATE CT.
C
       IF (CX) 20,10,20
C
  10   CT = CY
       THETA = 1.57079
       GO TO 100
  20   IF (CY) 40,30,40
C
  30   CT = CX
       THETA = 0.0
       GO TO 100
  40   THETA = ASIN(  1.0/(SQRT(1.0 + (CY/CX)**2)))
       IF (THETA - 0.2)  50,50,60
C
  50   CT = CX*COS(THETA)
       GO TO 100
C
  60   CT = CY*SIN(THETA)
  100  RETURN
       END
```

SUBROUTINE NOISOR

```
$IBFTC NOISO*  DECK
      SUBROUTINE NOISOR(ILIM,KLIM,D,CT,THETA,CX,CY)
C         SUBROUTINE NOISOR SIMULATES TWO DIFFERENT NOISE SOURCE
C         CONDITIONS. (1) NORMAL INCIDENCE WAVES - OCCURS WHEN THE
C         TRAIN OF WAVE FRONTS ARE PARLLEL TO THE PANEL.
C                     (2) PROGRESSIVE WAVE - OCCURS WHEN THE TRAIN
C         OF INCIDENT PLANE WAVE FRONTS ARE NOT PARALLEL TO THE
C         PANEL FACE.
      COMMON/BLK1/FREQ,AMASS,OMEGA
      COMMON/BLK2/NFREQ,NSIZE ,G,ALAM,CM,KDIG,NF
      COMMON/BLK3/FLG1,FLG2,FLG3,FLG4
      COMMON/BLK4/IT5,IT6,ITAPE,JTAPE,IFILE,IMAT,IRR,NAME,NDIM,C
      COMMON/BLK5/N1,N2,CRD1,CRD2,AREA,IPT
      COMMON/BLK6/PHI,CF,QF
      INTEGER FLG1,FLG2,FLG3,FLG4
      DIMENSION FREQ(25),AMASS(25),OMEGA(100)
C
      DIMENSION IPT(4),CRD1(100),CRD2(100),AREA(100),PHI(100)
      DIMENSION CF(85,85),QF(85,85),C(12)
      CSTHET = COS(THETA)
      SNTHET = SIN(THETA)
      IF (CX) 1,2,2
 1    CSTHET = -COS(THETA)
 2    IF (CY) 3,4,4
 3    SNTHET = -SIN(THETA)
 4    IF (IPT(4)-1) 5,5,6
 5    DUM = SNTHET
      SNTHET = CSTHET
      CSTHET = DUM
 6    NPHI = 0
      DO 600 L=1,ILIM
      IF (ILIM-1) 20,20,10
C     IF ILIM = 1, THEN THIS IS NOT  JOINT DEFLECTION OPT 2 .
C
C     JOINT DEFL. OPT. 2
 10   LL = L + 1
      KLIM =  L + KDIG
      IF(KLIM.GT.NFREQ) KLIM=NFREQ
      GO TO 30
C     OUTER  LOOP = 1
 20   LL = 1
 30   DO 500 K = LL,KLIM
      NPHI = NPHI + 1
      OMEGL = OMEGA(L)
      IF (ILIM-1)  50,50,40
 40   OMCT = (OMEGL + OMEGA(K))/2.0
      GO TO 60
 50   OMCT = OMEGA(K)
 60   PHIO = PHI(NPHI)
      OMG = OMCT
      OMCT = OMCT/CT
C     BEGIN  ELEMENT BY ELEMENT MATRIX GENERATION - FORMED IN THE
C     UPPER TRIANGULAR AND TLACE IN THE LOWER
      DO 400 += 1,NSIZE
      ARI = AREA(I)
      DO 300 J=I,NSIZE
      AR = ARI*AREA(J)
```

```
C*****M************************SEPARATION CALCULATION*********************
      N1J  = MOD(J,N1)
      N1I  = MOD(I,N1)
      IF (N1J)  80,70,80
 70   N2J  = J/N1
      N1J = N1
      GO TO 90
 80   N2J =J/N1 + 1
 90   IF (N1I)  110,100,110
 100  N2I  = I/N1
      N1I  = N1
      GO TO 120
 110  N2I  = I/N1 + 1
 120  SEP2 = CRD2(N2J) - CRD2(N2I)
      SEP1 = CRD1(N1J) - CRD1(N1I)
C*************************************************************************
C     CHECK IF A DIAGONAL ELEMENT
      IF (I - J) 150,130,150
C
 130  CF(I,J)= AR*PHIO
C     CHECK  IF  QF(I,J) IS NEEDED
      IF (FLG1-3) 140,300,140
 140  QF(I,J)= 0.0
      GO TO 300
 150  EXYSQ = SQRT(SEP1**2 + SEP2**2)
      EXPON = EXP(-D*EXYSQ)
C     CHECK IF NORMAL INCIDENCE OR PROGRESSIVE WAVE
 160  IF (IPT(2)-1)  180,170,180
C     NORMAL INCIDENCE
 170  CP2= 1.0
      GO TO 190
C     PROGRESSIVE WAVE
 180  P2 = OMCT*(SEP1*CSTHET + SEP2*SNTHET)
      CP2 = COS(P2)
 190  CF(I,J) = AR*PHIO*EXPON*CP2
      CF(J,I) = CF(I,J)
C     CHECK IF QF(I,J)  IS NEEDED
      IF (FLG1 - 3)  200,300,200
C     CHECK IF NORMAL INCIDENCE  OR PROGRESSIVE WAVE
 200  IF (IPT(2)-1)  220,210,220
C     NORMAL INCIDENCE
 210  QF(I,J)= 0.0
      QF(J,I)= QF(I,J)
      GO TO 300
C     PROGRESSIVE WAVE
 220  QP2 = SIN(P2)
      QF(I,J) = AR*PHIO*EXPON*QP2
      QF(J,I) = -QF(I,J)
C
 300  CONTINUE
 400  CONTINUE
      CALL OUTPUT(NPHI,OMG)
 500  CONTINUE
 600  CONTINUE
      RETURN
      END
```

SUBROUTINE OUTPT

```
$IBFTC OUTPUT  DECK
      SUBROUTINE OUTPT(NPHI,OMG)
C         THIS SUBROUTINE HANDLES THE OUTPUT FOR EACH FREQUENY,
C         BOTH THE PRINTED OUTPUT AND THE BINARY TAPES.
C         NPHI IS THE CURRENT FREQUENCY NUMBER
      COMMON/BLK1/FREQ,AMASS,OMEGA
      COMMON/BLK2/NFREQ,NSIZE ,G,ALAM,CM,KDIG,NF
      COMMON/BLK3/FLG1,FLG2,FLG3,FLG4
      COMMON/BLK4/IT5,IT6,ITAPE,JTAPE,IFILE,IMAT,IRR,NAME,NDIM,C
      COMMON/BLK5/N1,N2,CRD1,CRD2,AREA,IPT
      COMMON/BLK6/PHI,CF,QF
      INTEGER FLG1,FLG2,FLG3,FLG4
      DIMENSION FREQ(25),AMASS(25),OMEGA(100)
C
      DIMENSION IPT(4),CRD1(100),CRD2(100),AREA(100),PHI(100)
      DIMENSION CF(85,85),QF(85,85),C(12)
C************************PRINT SECTION**********************************
C
      IF (IPT(3)-NPHI)  200,10,10
C
 10   WRITE(IT6,2000) NPHI, OMG
C
C     PRINT CF(OMEGA(NPHI))
C
      DO 20 IROW=1,NSIZE
      WRITE(IT6,2001) IROW,(CF(IROW,J),J=1,NSIZE)
 20   CONTINUE
C
C     TEST IF QF(OMEGA) IS NEEDED-IE YES IF EITHER OPTION 1 OR 2
      IF(FLG1-2)  30,30,200
C
 30   WRITE(IT6,2002) NPHI, OMG
C
C     PRINT QF(OMEGA(NPHI))
C
      DO 40 IROW =1,NSIZE
      WRITE(IT6,2001) IROW,(QF(IROW,J),J=1,NSIZE)
 40   CONTINUE
C************************TAPE GENERATION SECTION************************
C
C     FIRST DETERMINE WHETHER JOINT DEFLECTIONS OR CPSD
C
 200  IF(FLG2-1)  210,210, 300
C
C******** JOINT DEFLECTIONS
C
C     TEST FOR OPTION  2 OR 3
C
 210  IF (FLG1-2)  220,220,250
C
C***** OPTION 2
C
 220  IFILE = 0
      IMAT = 0
      CALL WRTETP(CF,NDIM,NAME,NSIZE,NSIZE,C,IFILE,IMAT,ITAPE,IRR)
      CALL WRTETP(QF,NDIM,NAME,NSIZE,NSIZE,C,IFILE,IMAT,JTAPE,IRR)
      GO TO 500
```

```
C***** OPTION 3
 250  CALL WRTETP(CF,NDIM,NAME,NSIZE,NSIZE,C,IFILE,IMAT,ITAPE,IRR)
      GO TO 500
C
C******** CPSD
C
C     TEST FOR OPTION 1,2,OR 3
C
 300  GO TO (310,330,350), FLG1
C
C***** OPTION 1
C
 310  END FILE ITAPE
      CALL WRTETP(QF,NDIM,NAME,NSIZE,NSIZE,C,IFILE,IMAT,ITAPE,IRR)
      CALL WRTETP(CF,NDIM,NAME,NSIZE,NSIZE,C,IFILE,IMAT,ITAPE,IRR)
      GO TO 500
C
C***** OPTION 2
C
 330  END FILE JTAPE
      CALL WRTETP(CF,NDIM,NAME,NSIZE,NSIZE,C,IFILE,IMAT,JTAPE,IRR)
      CALL WRTETP(QF,NDIM,NAME,NSIZE,NSIZE,C,IFILE,IMAT,JTAPE,IRR)
      GO TO 500
C
C***** OPTION 3
C
 350  END FILE ITAPE
      CALL WRTETP(CF,NDIM,NAME,NSIZE,NSIZE,C,IFILE,IMAT,ITAPE,IRR)
C
C     TAPE WRITING COMPLETE - TAPE CLOSEOFF DONE IN MAIN PROGRAM
C
C     TEST IF ERROR FLAG IRR HAS BEEN SET
C
 500  IF (IRR)  510,520,510
 510  WRITE(IT6,2003) IRR, FLG1,FLG2,NPHI
      CALL EXIT
 520  RETURN
C
C*****************************F O R M A T S ****************************
C
 2000 FORMAT(1H1,2X,14HFREQUENCY NO. ,I3,3X,14HCF(I,J) MATRIX,3X,
     1 8HOMEGA = ,E15.8//)
 2001 FORMAT(1H0,I5,1P6E16.7/(E22.7,5E16.7))
 2002 FORMAT(1H1,2X,14HFREQUENCY NO. ,I3,3X,14HQF(I,J) MATRIX,3X,
     1 8HOMEGA = ,E15.8//)
 2003 FORMAT(1H0,2X,33HERROR RETURN FROM WRTETP , IRR = ,I3,2X,
     1 7HFLG1 = ,I3,2X, 7HFLG2 = ,I3,2X, 7HNPHI = ,I3)
      END
```

SUBROUTINE PEDAN

```
$IBFTC PEDAN*  DECK
      SUBROUTINE PEDAN
C
C
C
C***PEDAN***  IMPEDANCE MATRIX(PEDAN)
C                    SUBROUTINE TO FORM THE REAL AND IMAGINARY PART
C                                OF THE ADMITTANCE MATRIX BEFORE INVERTING
C
C
C
      COMMON /BLK1/ FREQ,AMASS,OMEGA
      COMMON /BLK2/ M,N,G,ALAM,CMU,K,NF,NPLATE,NBEAMS
         DIMENSION AMASS(25),C(90,90),STIFF(90,90),OMEGA(100),B(16),
     1   AMAT(90,90),FREQ(25),IPARAM(5)
      DIMENSION BMASS(25)
      EQUIVALENCE ( C,STIFF )
      NTAP1 = 10
      NTAP14 = 14
      NTAP15 = 15
      NTAP16 = 16
      REWIND NTAP1
      REWIND NTAP14
      REWIND NTAP15
      REWIND NTAP16
C***                 FORM THE REAL PART OF THE IMPEDANCE MATRIX
C                         *OMEGA**2AMASS(I) + STIFF(I,J)
C
      NAME = 0
      NMAT = 0
      NFILE = 0
C***                 READ IN THE STIFFNESS MATRIX( STIFF )
C
      CALL READTP(STIFF,90,NAME,N,N,B,NFILE,NMAT,NTAP1,IRR)
      IF ( IRR .NE. 0 ) GO TO 9985
C
      NAME = 0
      NMAT = 0
      NFILE = 3
      CALL READTP(BMASS,1,NAME,1,N,B,NFILE,NMAT,NTAP1,IRR)
      IF ( IRR .NE. 0 ) GO TO 9985
C
      NAME = 0
      NMAT = 0
      NFILE = 0
C***
C    START LOOP ( NF TIMES )
C
      DO 200 II = 1,NF
C
C
      DO 150 I = 1,N
      DO 150 J = 1,N
      IF ( I .EQ. J ) GO TO 100
      AMAT(I,J) = STIFF(I,J)
      GO TO 150
 100  AMAT(I,J) = -OMEGA(II)**2*BMASS(I) + STIFF(I,J)
```

```
 150  CONTINUE
C
      CALL WRTETP( AMAT,90,NAME,N,N,B,NFILE,NMAT,NTAP16,IRR )
      IF ( IRR .NE. 0 ) GO TO 9986
C
 200  CONTINUE
C
C
C***READ IN THE DAMPING MATRIX C(NXN)
      READ(5,9002) ((C(I,J),J=1,N),I=1,N)
      WRITE(6,9003)
      DO 250 I = 1,N
      WRITE(6,9004)I,( C(I,J),J=1,N )
 250  CONTINUE
C
C***FORM THE IMAGINARY PART OF THE IMPEDANCE MATRIX
C                 OMEGA*C(I,J)
C
      NAME = 0
      NMAT = 0
      NFILE = 0
C***
C    START LOOP
C
      DO 400 II = 1,NF
C
C
      DO 300 I = 1,N
      DO 300 J = 1,N
 300  AMAT(I,J) = OMEGA(II)*C(I,J)
C
      CALL WRTETP( AMAT,90,NAME,N,N,B,NFILE,NMAT,NTAP15,IRR)
      CALL WRTETP( AMAT,90,NAME,N,N,B,NFILE,NMAT,NTAP15,IRR)
      IF ( IRR .NE. 0 ) GO TO 9986
C
 400  CONTINUE
C
C
      END FILE NTAP15
      END FILE NTAP16
      REWIND NTAP15
      REWIND NTAP16
      IF ( N .LE. 50 ) GO TO 500
      IPARAM(1) = 4
      GO TO 510
 500  IPARAM(1) = N
 510  IPARAM(2) = N*N + 3
      IPARAM(3) = NF
      IPARAM(4) = M
      IPARAM(5) = NPLATE + 2*NBEAMS
      NAME = 0
      NMAT = 0
      NFILE = 0
      CALL WRTETP( IPARAM,1,NAME,1,5,B,NFILE,NMAT,NTAP14,IRR )
      IF ( IRR .NE. 0 ) GO TO 9986
      END FILE NTAP14
      RETURN
 9985 WRITE(6,9990) IRR
      CALL EXIT
 9986 WRITE(6,9991) IRR
```

```
      CALL EXIT
      RETURN
 9002 FORMAT(7F10.0)
 9003 FORMAT(1H1,40X,14HDAMPING MATRIX //// )
 9004 FORMAT(1H0,I5,1P7E16.6/(E22.6,6E16.6) )
 9990 FORMAT(28H ERROR IN READTP-ERROR CODE=I5)
 9991 FORMAT(28H ERROR IN WRTETP-ERROR CODE=I5)
      END
```

SUBROUTINE PRINTA

```
$IBFTC PRNTA*  DECK
      SUBROUTINE PRINTA
C
C
C***                THE DEFLECTION CROSS-PSD MATRICES FOR OPTION 1
C                     ARE PRINTED.
C
C
C
      COMMON/BLK1/FREQ,AMASS,OMEGA
      COMMON/BLK2/M,N,G,ALAM,CMU,K,NF,NPLATE,NBEAMS
      DIMENSION A(90,90),FREQ(25),AMASS(25),OMEGA(100),B(16)
      NTAPE = 15
      ITAPE = 16
      REWIND NTAPE
      REWIND ITAPE
      NAME = 0
      NMAT = 0
      NFILE = 0
      WRITE(6,9000)
C
C
C
C***CYCLE ON NUMBER OF FREQUENCIES
      DO 1000 II = 1,NF
      WRITE(6,9005)
      WRITE(6,9001) OMEGA(II)
      CALL READTP(A,90,NAME,N,N,B,NFILE,NMAT,NTAPE,IRR)
      IF ( IRR .NE. 0 ) GO TO 9985
C
      DO 100 I = 1,N
      WRITE(6,9002)I,(A(I,J),J=1,N)
  100 CONTINUE
      WRITE(6,9005)
      WRITE(6,9003) OMEGA(II)
      CALL READTP(A,90,NAME,N,N,B,NFILE,NMAT,ITAPE,IRR)
      IF ( IRR .NE. 0 ) GO TO 9985
C
      DO 200 I = 1,N
      WRITE(6,9002)I,(A(I,J),J=1,N)
  200 CONTINUE
 1000 CONTINUE
C
C
      REWIND NTAPE
      REWIND ITAPE
      RETURN
 9985 WRITE(6,9990) IRR
      CALL EXIT
      RETURN
 9000 FORMAT(1H1,50X,29HDEFLECTION CROSS PSD MATRICES ////)
 9001 FORMAT(1H0,10X,26HDEFLECTION CO-POWER, FREQ=E14.7,12H  (RAD/SEC) /
     1// )
 9002 FORMAT(1H0,I5,1P7E16.6/(E22.6,6E16.6) )
 9003 FORMAT(1H0,10X,28HDEFLECTION QUAD-POWER, FREQ=E14.7,12H  (RAD/SEC)
     1 /// )
 9005 FORMAT(1H0)
```

```
9990 FORMAT(28H ERROR IN READTP-ERROR CODE=I5)
     END
```

SUBROUTINE PRINTB

```
$IBFTC PRNTB*  DECK
       SUBROUTINE PRINTB
C
C
C***                  THE STRESS CROSS-PSD MATRICES ARE PRINTED FOR
C                       PLATES AND BEAMS FOR OPTION 1.
C
C
C
       COMMON/BLK1/FREQ,AMASS,OMEGA
       COMMON/BLK2/M,N,G,ALAM,CMU,K,NF,NPLATE,NBEAMS
       COMMON/BLK3/FLG1,FLG2,FLG3,FLG4
       INTEGER FLG1,FLG2,FLG3,FLG4
       DIMENSION FREQ(25),AMASS(25),OMEGA(100),S(8,8),B(16),SB(6,6)
       NTAPE = 3
       REWIND NTAPE
       NAME = 0
       NMAT = 0
       NFILE = 0
C
C***IF NPLATE EQUALS 0 - SKIP PLATE PRINTOUT
       IF( NPLATE .EQ. 0 ) GO TO 200
C
C
C
       DO 100 II = 1,NPLATE
       WRITE(6,9000) II
C
       DO 50 IJ = 1,NF
       WRITE(6,9001) OMEGA(IJ)
       WRITE(6,9002)
       CALL READTP(S,8,NAME,NR,NC,B,NFILE,NMAT,NTAPE,IRR)
       IF ( IRR .NE. 0 ) GO TO 9985
C
       DO 10 I = 1,8
       WRITE(6,9003)I,(S(I,J),J=1,8)
 10    CONTINUE
C
       WRITE(6,9004)
       CALL READTP(S,8,NAME,NR,NC,B,NFILE,NMAT,NTAPE,IRR)
       IF ( IRR .NE. 0 ) GO TO 9985
       DO 20 I = 1,8
       WRITE(6,9003)I,(S(I,J),J=1,8)
 20    CONTINUE
C
 50    CONTINUE
C
C
 100   CONTINUE
C
C
C
C***IF NBEAMS EQUALS 0 - SKIP BEAM PRINTOUT
 200   IF(NBEAMS .EQ. 0 ) RETURN
C
C
C
```

```
      DO 1000 II = 1,NBEAMS
      WRITE(6,9005) II
C
      DO 150 IJ = 1,NF
      WRITE(6,9001) OMEGA(IJ)
      DO 140 I1 = 1,2
      CALL READTP(SB,6,NAME,NR,NC,B,NFILE,NMAT,NTAPE,IRR)
      IF ( IRR .NE. 0 ) GO TO 9985
      WRITE(6,9010) I1
      WRITE(6,9002)
C
      DO 80 I = 1,6
      WRITE(6,9011)I,(SB(I,J),J=1,6)
 80   CONTINUE
C
      CALL READTP(SB,6,NAME,NR,NC,B,NFILE,NMAT,NTAPE,IRR)
      IF ( IRR .NE. 0 ) GO TO 9985
      WRITE(6,9004)
C
      DO 90 I=1,6
      WRITE(6,9012)I,(SB(I,J),J=1,6)
 90   CONTINUE
C
 140  CONTINUE
C
 150  CONTINUE
C
C
C
 1000 CONTINUE
C
C
C
      REWIND NTAPE
      RETURN
 9985 WRITE(6,9990) IRR
      CALL EXIT
      RETURN
 9000 FORMAT(1H0,6HPLATE I3 )
 9001 FORMAT(5X,10HFREQUENCY=E14.7)
 9002 FORMAT(1H0,15X,9HREAL PART///)
 9003 FORMAT(1H0,14X,I5,8E14.5/(20X,8E14.5) )
 9004 FORMAT(1H0,15X,9HIMAG PART///)
 9005 FORMAT(1H0,6H BEAM I3//)
 9010 FORMAT(10X,4HEND I3///)
 9011 FORMAT(1H0,20X,I5,6E14.5/(26X,6E14.5) )
 9012 FORMAT(1H0,20X,I5,6E14.5/(26X,6E14.5) )
 9990 FORMAT(28H ERROR IN READTP-ERROR CODE=I5)
      END
```

SUBROUTINE CONS

```
$IBFTC CONS*   DECK
       SUBROUTINE CONS
C
C
C***                ROUTINE TO CALCULATE THE CONSTANTS USED IN THE
C                     TRAPEZOIDAL INTEGRATION FORMULA.
C
C
C
       COMMON /BLK1/ FREQ,AMASS,OMEGA
       COMMON /BLK2/M,N,G,ALAM,CMU,K,NF,NPLATE,NBEAMS
       DIMENSION FREQ(25),AMASS(25),OMEGA(100),X(100),H(100),C(100)
      1      ,B(16)
       EQUIVALENCE ( OMEGA,X )
C     THE CONSTANTS ARE C0, C1.AND C2 CALCULATED OVER THE TOTAL INTERVAL.
       NTAPE = 4
       REWIND NTAPE
       N2 = NF - 2
       NN = NF - 1
C
       DO 10 I=1,NN
  10   H(I) = X(I+1)-X(I)
C
       C(1) = H(1)/2.0
       C(NF) = H(NN)/2.
C
       DO 150 I = 2,NN
       C(I)=(H(I-1)+H(I))/2.
  150  CONTINUE
C
       NAME = 0
       NMAT = 0
       NFILE = 0
       CALL WRTETP(C,1,NAME,NF,1,B,NFILE,NMAT,NTAPE,IRR)
       IF ( IRR .NE. 0 ) GO TO 9986
       END FILE NTAPE
       RETURN
 9986  WRITE(6,9991) IRR
       CALL EXIT
       RETURN
 9991  FORMAT(28H ERROR IN WRTETP-ERROR CODE=I5)
       END
```

SUBROUTINE PRINTC

```
$IBFTC PRNTC*  DECK
       SUBROUTINE PRINTC
C
C
C***                  THE DEFLECTION CO-VARIANCE MATRICES ARE PRINTED FOR
C                       OPTION 1.
C
C
C
       COMMON/BLK2/M,N,G,ALAM,CMU,K,NF,NPLATE,NBEAMS
       DIMENSION A(90,90),B(16)
       NTAPE = 12
       REWIND NTAPE
       NAME = 0
       NMAT = 0
       NFILE = 0
       WRITE(6,9000)
       WRITE(6,9001)
       CALL READTP(A,90,NAME,N,N,B,NFILE,NMAT,NTAPE,IRR)
       IF ( IRR .NE. 0 ) GO TO 9985
C
       DO 100 I = 1,N
       WRITE(6,9002) I,(A(I,J),J=1,N )
 100   CONTINUE
C
       WRITE(6,9005)
       WRITE(6,9003)
       CALL READTP(A,90,NAME,N,N,B,NFILE,NMAT,NTAPE,IRR)
       IF ( IRR .NE. 0 ) GO TO 9985
C
       DO 200 I = 1,N
       WRITE(6,9002)9,(A(I,J),J=1,N)
 200   CONTINUE
C
       REWIND NTAPE
       RETURN
 9985  WRITE(6,9990) IRR
       CALL EXIT
       RETURN
 9000  FORMAT(1H1,30X,30HDEFLECTION CO-VARIANCE MATRIX  ///)
 9001  FORMAT(1H0,10X,9HREAL PART ///)
 9002  FORMAT(1H0,I5,1P7E16.6/(E22.6,6E16.6) )
 9003  FORMAT(1H0,10X,14HIMAGINARY PART  ///)
 9005  FORMAT(1H0)
 9990  FORMAT(28H ERROR IN READTP-ERROR CODE=I5)
       END
```

SUBROUTINE PRINTD

```
$IBFTC PRNTD*  DECK
      SUBROUTINE PRINTD(IR,NTAPE,NFF)
C
C
C***                THE STRESS MATRICES ARE PRINTED FOR
C                     OPTIONS  1, 2 AND 3
C
C
C
      COMMON /BLK1/FREQ,AMASS,OMEGA
      COMMON/BLK2/M,N,G,ALAM,CMU,K,NF,NPLATE,NBEAMS
      COMMON/BLK3/FLG1,FLG2,FLG3,FLG4
      DIMENSION S(8,8),B(16),SB(6,6)
      DIMENSION FREQ(25),AMASS(25),OMEGA(100)
      INTEGER FLG1,FLG2,FLG3,FLG4
      REWIND NTAPE
      NAME = 0
      NMAT = 0
      NFILE = 0
C
C
C
      DO 800 III = 1,IR
      IF ( III .EQ. 2 ) GO TO 80
      WRITE(6,9001)
      GO TO 90
 80   WRITE(6,9010)
 90   IF ( NPLATE .EQ. 0 ) GO TO 200
C
      DO 500 INF = 1,NFF
      IF ( NFF .EQ. 1 ) GO TO 99
      WRITE(6,9020) OMEGA(INF)
 99   CONTINUE
C
C
      DO 100 II = 1,NPLATE
      CALL READTP(S,8,NAME,NR,NC,B,NFILE,NMAT,NTAPE,IRR)
      IF ( IRR .NE. 0 ) GO TO 9985
      WRITE(6,9002)II
      DO 110 I = 1,8
      WRITE(6,9003)I,(S(I,J),J=1,8)
 110  CONTINUE
 100  CONTINUE
 200  IF ( NBEAMS .EQ. 0 ) GO TO 500
C
      DO 400 II = 1,NBEAMS
      WRITE(6,9004) II
      DO 380 IJ = 1,2
      WRITE(6,9005)IJ
      CALL READTP(SB,6,NAME,NR,NC,B,NFILE,NMAT,NTAPE,IRR)
      IF ( IRR .NE. 0 ) GO TO 9985
      DO 360 I = 1,6
      WRITE(6,9006)I,(SB(I,J),J=1,6 )
C
 360  CONTINUE
C
 380  CONTINUE
```

```
C
  400  CONTINUE
C
  500  CONTINUE
C
C
C
  800  CONTINUE
C
C
C
       REWIND NTAPE
       RETURN
 9985  WRITE(6,9990) IRR
       CALL EXIT
       RETURN
 9001  FORMAT(1H0,10X,9HREAL PART ///)
 9002  FORMAT(1H0,6HPLATE I3)
 9003  FORMAT(1H0,5X,I5,8E14.5/(11X,8E14.5) )
 9004  FORMAT(1H0,6H BEAM I3 )
 9005  FORMAT(11X,4HEND I3 )
 9006  FORMAT(20X,I5,6E14.5/(25X,6E14.5) )
 9010  FORMAT(1H0,10X,14HIMAGINARY PART ///)
 9020  FORMAT(1H0,27HSTRESS CO-POWER, FREQUENCY=E14.7,9H(RAD/SEC) )
 9990  FORMAT(28H ERROR IN READTP-ERROR CODE=I5)
       END
```

SUBROUTINE DSECM1

```
$IBFTC DSEC1*  DECK
       SUBROUTINE DSECM1
C
C
C***                  THE DEFLECTION SECOND SPECTRAL MOMENTS ARE
C                       CALCULATED FOR OPTION 1.
C                     THE JOINT DEFLECTION MATRICES FORMED IN SUBROUTINE
C                       DJNT1 ARE MULTIPLIED BY ITS APPROPRIATE FREQ**2
C                       AND CONSTANTS AND SUMMED TOGETHER TO FORM THE
C                       DEFLECTION SECOND SPECTRAL MOMENTS MAYRIES
C                       DEFLECTION SECOND SPECTRAL MOMENT MATRICES
C                       ( REAL PART AND IMAGINARY PART ).
C
C
C
       COMMON/BLK1/FREQ,AMASS,OMEGA
       COMMON/BLK2/M,N,G,ALAM,CMU,K,NF,NPLATE,NBEAMS
       DIMENSION FREQ(25),AMASS(25),OMEGA(100),AMAT(90,90),SMAT(90,90)
      1    ,B(16)
       DIMENSION C(100)
       NTAP12 = 12
       NTAP4 = 4
       NTAP15 = 15
       NTAP16 = 16
       REWIND NTAP12
       REWIND NTAP4
       REWIND NTAP15
       REWIND NTAP16
C
C***PRINT TITLE HEADINGS
       WRITE(6,9000)
       NAME = 0
       NMAT = 0
       NFILE = 0
       CALL READTP(C,1,NAME,NF,1,B,NFILE,NMAT,NTAP4 ,IRR)
       IF ( IRR .NE. 0 ) GO TO 9985
C
       DO 5 I = 1,N
       DO 5 J = 1,N
 5     SMAT(I,J) = 0.
C
       NTAPE = NTAP15
C
C***LOOP ON REAL AND IMAGINARY
       DO 50 III = 1,2
       IF ( III .EQ. 2 ) NTAPE = NTAP16
       DO 40 II = 1,NF
       NAME = 0
       NMAT = 0
       NFILE = 0
       CALL READTP(AMAT,90,NAME,N,N,B,NFILE,NMAT,NTAPE,IRR)
       IF ( IRR .NE. 0 ) GO TO 9985
       DO 20 I = 1,N
       DO 20 J = 1,N
 20    SMAT(I,J) = SMAT(I,J) + AMAT(I,J)*OMEGA(II)**2*C(II)
 40    CONTINUE
       NAME = 0
```

```
      NMAT = 0
      NFILE = 0
      CALL WRTETP(SMAT,90,NAME,N,N,B,NFILE,NMAT,NTAP12,IRR)
      IF ( IRR .NE. 0 ) GO TO 9986
      IF ( III .EQ. 1 ) GO TO 45
      WRITE(6,9005)
      WRITE(6,9001)
      DO 42 I = 1,N
      WRITE(6,9002)I,(SMAT(I,J),J=1,N)
   42 CONTINUE
      GO TO 50
   45 WRITE(6,9003)
      DO 48 I = 1,N
      WRITE(6,9002)I,(SMAT(I,J),J=1,N)
   48 CONTINUE
   50 CONTINUE
C
C
      END FILE NTAP12
      REWIND NTAP12
      RETURN
 9985 WRITE(6,9990) IRR
      CALL EXIT
 9986 WRITE(6,9991) IRR
      CALL EXIT
      RETURN
 9000 FORMAT(1H1,30X,41HDEFLECTION SECOND SPECTRAL MOMENT MATRIX   ////)
 9001 FORMAT(1H0,10X,14HIMAGINARY PART  ///)
 9002 FORMAT(1H0,I5,1P7E16.6/(E22.6,6E16.6) )
 9003 FORMAT(1H0,10X,9HREAL PART  ///)
 9005 FORMAT(1H0)
 9990 FORMAT(28H ERROR IN READTP-ERROR CODE=I5)
 9991 FORMAT(28H ERROR IN WRTETP-ERROR CODE=I5)
      END
```

SUBROUTINE ADMIN3

```
$IBFTC ADMN3*  DECK
      SUBROUTINE ADMIN3
C
C
C
C
C                *                                          *
C               * A D M I T T A N C E   I N T E G R A L S *
C              *                                            *
C               * U S E D   I N   F O R M I N G   T H E *
C                *                                          *
C                 * J O I N T   D E F L E C T I O N S   *
C                  *                                        *
C                          ( OPTION 3 )
C
C
C  OPTION 3     BROAD BAND EXCITATIONS, DAMPING COEFFICIENTS PROPORTIONAL
C              TO A LINEAR COMBINATION OF THE MASS INERTIA AND STIFFNESS
C              COEFFICIENTS. ( NO CROSS MODAL COUPLING IS INCLUDED )
C
C  M           NUMBER OF FREQUENCIES, MASS AND MODE SHAPES
C  N           NUMBER OF RETAINED DEGREES OF FREEDOM
C  G           STRUCTURAL DAMPING
C  ALAM        A DAMPING PROPORTIONALITY FACTOR PROP. TO STIFFNESS
C  CMU         A DAMPING PROPORTIONALITY FACTOR FOR DAMPING PROPORTIONAL
C                TO MASS.
      DIMENSION FREQ(25), AMASS(25), AMU(25), DIAG(25), PHIM(25),
     1          PHI(100,25), B(16),NSTO(1)
      COMMON /BLK1/ FREQ, AMASS
      COMMON /BLK2/ M,N,G,ALAM,CMU
      COMMON/BLK3/FLG1,FLG2,FLG3,FLG4
      INTEGER FLG1,FLG2,FLG3,FLG4
      NTAP1 = 10
      NTAP11 = 11
      REWIND NTAP1
      REWIND NTAP11
C
C***FORM THE PARAMETER MATRIX AND STORE ON TAPE
      NSTO(1) = M
      NAME = 0
      NMAT = 0
      NFILE = 0
      CALL WRTETP(NSTO,1,NAME,1,1,B,NFILE,NMAT,NTAP11,IRR)
      IF ( IRR .NE. 0 ) GO TO 9986
C***                                        DAMPING FACTOR
C
      DO 10 I = 1,M
 10   AMU(I) = CMU + ALAM*FREQ(I)**2 + G*FREQ(I)
C
C
C***                                        DIAGONAL SCALARS ARE COMPUTED
      IF ( FLG1 .EQ. 2 ) GO TO 15
      WRITE(6,9000)
      GO TO 18
 15   WRITE(6,9002)
 18   DO 20 I = 1,M
 20   DIAG(I) = 3.14159265/(2.0*AMU(I)*FREQ(I)**2*AMASS(I)**2)
```

```
      WRITE(6,9001) ( I,DIAG(I),I=1,M )
      NAME = 0
      NMAT = 0
      NFILE = 0
      CALL WRTETP(DIAG,1,NAME,1,M,B,NFILE,NMAT,NTAP11,IRR)
      IF ( IRR .NE. 0 ) GO TO 9986
      NAME = 0
      NMAT = 1
      NFILE = 2
C
C***                                          THE MODE SHAPES ARE READ IN.
C
      CALL READTP(PHI,100,NAME,N,M,B,NFILE,NMAT,NTAP1,IRR)
      IF ( IRR .NE. 0 ) GO TO 9985
      NAME = 0
      NFILE =0
      NMAT = 0
C
      DO 50 J=1,M
      DO 40 I=1,N
 40   PHIM(I) = PHI(I,J)
      CALL WRTETP(PHIM,1,NAME,N,1,B,NFILE,NMAT,NTAP11,IRR)
 50   CONTINUE
C
      END FILE NTAP11
      REWIND NTAP1
      REWIND NTAP11
      RETURN
 9985 WRITE(6,9990) IRR
      CALL EXIT
 9986 WRITE(6,9991) IRR
      CALL EXIT
      RETURN
 9000 FORMAT(1H1,40X,20HADMITTANCE INTEGRALS //// )
 9001 FORMAT(10X,I5,5X,E14.7 )
 9002 FORMAT(1H1,40X,38HADMITTANCE INTEGRALS (NO CROSS TERMS) /// )
 9990 FORMAT(28H ERROR IN READTP-ERROR CODE=I5)
 9991 FORMAT(28H ERROR IN WRTETP-ERROR CODE=I5)
      END
```

SUBROUTINE ADDMAT

```
$IBFTC ADDMA*  DECK
       SUBROUTINE ADDMAT( INTAPE, OUTAPE, NO )
C
C                  * *                                  * *
C                 *          M  A  T  R  I  X             *
C                *                                        *
C                * A D D I T I O N   R O U T I N E         *
C                *                                        *
C                 * * *                                * * *
C
C                      (MATRIX SIZE IS NXN)
C
C***  INTAPE -          THERE ARE M NO. OF MATRICES STORED ON THIS TAPE.
C***  OUTAPE -          THE SUM OF M MATRICES ARE STORED ON THIS TAPE.
C
       COMMON/BLK1/FREQ,AMASS,OMEGA
       COMMON/BLK2/M,N,G,ALAM,CMU,K,NF,NPLATE,NBEAMS
       COMMON/BLK3/FLG1,FLG2,FLG3,FLG4,MF,NRI
       DIMENSION SUM(90,90),AMAT(90,90),B(16),IPARAM(2)
       DIMENSION FREQ(25),AMASS(25),OMEGA(100)
       INTEGER OUTAPE
       INTEGER FLG1,FLG2,FLG3,FLG4
       REWIND INTAPE
       REWIND OUTAPE
       NAME = 0
       NMAT = 0
       NFILE = 0
C
C***FORM PARAMETER  MATRIX IPARAM
       IPARAM(1) = NPLATE + 2*NBEAMS
       IPARAM(2) = NO
       CALL WRTETP(IPARAM,1,NAME,1,2,B,NFILE,NMAT,OUTAPE,IRR)
       IF ( IRR .NE. 0 ) GO TO 9986
       IF ( FLG1 .EQ. 3 ) GO TO 5
       GO TO 8
 5     IF ( FLG2 .EQ. 1 ) GO TO 6
       GO TO 7
 6     WRITE(6,9001)
       GO TO 8
 7     WRITE(6,9002)
 8     CONTINUE
C
C
       DO 500 III = 1,NO
       IF ( FLG1 .EQ. 2 ) GO TO 11
       IF ( FLG2 .EQ. 1 ) GO TO 9
       WRITE(6,9003) OMEGA(III)
 9     CONTINUE
C
C***LOOP ON REAL AND IMAGINARY
 11    DO 400 IR = 1,NRI
C
       DO 10 I=1,N
       DO 10 J=1,N
 10    SUM(I,J) = 0.
C
C***BEGINNING OF LOOP TO SUM M MATRICES.
```

```
C
C
C
      DO 300 II = 1,MF
      NAME = 0
      NFILE = 0
      NMAT = 0
      CALL READTP(AMAT,90,NAME,N,N,B,NFILE,NMAT,INTAPE,IRR)
      IF ( IRR .NE. 0 ) GO TO 9985
C***READ IN A MATRIX FROM INTAPE
C
      DO 100 I = 1,N
      DO 100 J = 1,N
  100 SUM(I,J) = SUM(I,J) + AMAT(I,J)
  300 CONTINUE
C
C***END OF LOOP TO SUM M MATRICES.
C
C
      IF ( FLG1 .EQ. 2 ) GO TO 360
      DO 350 I = 1,N
      WRITE(6,9000)I,(SUM(I,J),J=1,N)
  350 CONTINUE
C
  360 NAME = 0
      NFILE = 0
      NMAT = 0
C***WRITE THE SUMMATION MATRIX ON OUTAPE.
      CALL WRTETP(SUM,90,NAME,N,N,B,NFILE ,NMAT,OUTAPE ,IRR)
      IF ( IRR .NE. 0 ) GO TO 9986
  400 CONTINUE
  500 CONTINUE
C
C
      END FILE OUTAPE
      REWIND INTAPE
      REWIND OUTAPE
      RETURN
 9985 WRITE(6,9990) IRR
      CALL EXIT
 9986 WRITE(6,9991) IRR
      CALL EXIT
      RETURN
 9000 FORMAT(1H0,I5,1P7E16.6/(E22.6,6E16.6) )
 9001 FORMAT(1H1,30X,37HDEFLECTION CO-VARIANCE MATRIX (REAL) ///)
 9002 FORMAT(1H1,30X,39HBELOW ARE DEFLECTION CROSS-PSD MATRICES /// )
 9003 FORMAT(1H0,31HDEFLECTION CO-POWER, FREQUENCY=F12.6,11H (RAD/SEC) ,
 9990 FORMAT(28H ERROR IN READTP-ERROR CODE=I5)
 9991 FORMAT(28H ERROR IN WRTETP-ERROR CODE=I5)
      END
```

SUBROUTINE ADMIN2

```
$IBFTC ADMN2*  DECK
       SUBROUTINE ADMIN2
C                *                                          *
C               * A D M I T T A N C E   I N T E G R A L S *
C              *                                            *
C              * U S E D   I N   F O R M I N G   T H E     *
C               *                                          *
C                * J O I N T   D E F L E C T I O N S   *
C
C                *                                        *
C   OPTION 2     BROAD BAND EXCITATION, DAMPING COEFFICIENTS PROPORTIONAL
C                TO A LINEAR COMBINATION OF THE MASS INERTIA AND STIFFNESS
C                COEFFICIENTS. ( CROSS MODAL COUPLING IS INCLUDED )
C
C
C   M            NUMBER OF FREQUENCIES, MASS AND MODE SHAPES
C
C
C                THE FREQUENCIES(FREQ), GENERALIZED MASS(AMASS) AND
C                MODE SHAPES(PHI) COME FROM THE EIGENVECTOR-EIGENVALUE
C                ROUTINE TV-105W.
C
C   N            NUMBER OF RETAINED DEGREES OF FREEDOM
C   NF           NUMBER OF FREQUEUNCIES TO CALCULATE THE CPSD
C   G            STRUCTURAL DAMPING
C   ALAM         A DAMPING PROPORTIONALITY FACTOR PROP. TO STIFFNESS
C   CMU          A DAMPING PROPORTIONALITY FACTOR FOR DAMPING PROPORTIONAL
C                  TO MASS.
C
C***TAPE OUTPUT STORAGE - TAPE 16
C                    *****
C      MATRIX 1      *M-1* PARAMETER MATRIX(3X1)
C                    *K  *
C                    *N  *
C                    *****
C
C      MATRIX 2       0     PARAMETER K MATRIX USED FOR OFF-DIAG TEST
C                   OR -1       ( M-1 BY 1 )
C
C
C      MATRIX 3       PHI(I,J)     I=1,M-1
C                                  J=I,I+K
C
      DIMENSION DE(25,25), ED(25,25), DDEE(25,25), AMU(25), AMASS(25),
     1FREQ(25),OM4(25),X7(25),FREQ4(25),X8(25),SC1(25,25),
     2SC2(25,25),SMAT(3),B(16),PHI(100,25),PHIM(100),NSTO(3),KPARAM(24)
      COMMON /BLK1/ FREQ,AMASS
      COMMON/BLK2/M,N,G,ALAM,CMU,K,NF,NPLATE,NBEAMS,NLOOP
      PI = 3.14159265
      NTAP1 = 10
      NTAP12 = 12
      NTAP16=16
      REWIND NTAP1
      REWIND NTAP12
      REWIND NTAP16
C
      DO 20 I=1,M
```

```
  20    AMU(I)=CMU+ALAM*FREQ(I)**2+G*FREQ(I)
C
C       THE FREQUENCIES AND MU ARE SCALED
        CALL SCALE( FREQ, M, SCAL )
C
        DO 25 I = 1,M
  25    AMU(I) = AMU(I)/SCAL
C
        MM=M-1
C
C***FORM PARAMETER MATRIX NSTO
        NSTO(1) = MM
        NSTO(2) = K
        NSTO(3) = N
        NAME = 0
        NMAT = 0
        NFILE = 0
        CALL WRTETP(NSTO,1,NAME,3,1,B,NFILE,NMAT,NTAP16,IRR)
        IF ( IRR .NE. 0 ) GO TO 9999
C***FORM PARAMETER K MATRIX TO TEST FOR NUMBER OF OFF-DIAGONAL TERMS
C          ARE DESIRED
C
        MK = M-K
        DO 30 I=1,MK
  30    KPARAM(I) = 0
        MK = MK+1
        IF ( MK .GT. MM ) GO TO 50
        DO 40 I = MK,MM
  40    KPARAM(I) = -1
  50    NAME = 0
        NMAT = 0
        NFILE = 0
        IF ( K .EQ. 1 ) GO TO 55
        GO TO 58
  55    KPARAM(M) = -1
        MM = M
  58    CONTINUE
        CALL WRTETP(KPARAM,1,NAME,MM,1,B,NFILE,NMAT,NTAP16,IRR)
        IF ( IRR .NE. 0 ) GO TO 9999
        IF ( K .EQ. 1 ) MM = M-1
        NAME = 0
        NMAT=1
        NFILE = 2
        CALL READTP(PHI,100,NAME,N,M,B,NFILE,NMAT,NTAP1,IRR)
        IF(IRR .NE. 0)GO TO 9998
        NAME = 0
        NMAT=0
        NFILE = 0
C       THE MODE SHAPES PHI ARE STORED ON TAPE
C****
        DO 90 II=1,MM
        J1 = II
        ML = II+K
        IF( ML .GT. M ) ML = M
        DO 90 J = J1,ML
        DO 80 I=1,N
  80    PHIM(I)=PHI(I,J)
        CALL WRTETP(PHIM,1,NAME,N,1,B,NFILE,NMAT,NTAP16,IRR)
        IF ( IRR .NE. 0 ) GO TO 9999
  90    CONTINUE
```

```
C****
C****
C            THE ADMITTANCE INTEGRALS FOR OPTION 2 ARE FORMED AT M
C             NUMBER OF FREQUENCIES.
C****
      DO 100 I=1,M
      FREQ4(I)=FREQ(I)**4
      X7(I)=AMU(I)**2-2.*FREQ(I)**2
  100 X8(I)=SQRT(4.*FREQ(I)**2-AMU(I)**2)*AMU(I)
C
C
      DO 130 I=1,M
      DO 120 J=1,M
      IF(I .NE. J)GO TO 110
      DE(I,J) = 0.
      ED(J,I) = 0.
      GO TO 120
  110 X2=FREQ(I)**2-FREQ(J)**2
      X3 = FREQ4(I) - FREQ4(J)
      X4 = AMU(J)*FREQ(I)**2
      X5 = AMU(I)**2 - AMU(J)**2
      X6 = FREQ(J)**4/FREQ(I)**4
      BMAT = ( X3*(-X4*X7(I)-AMU(J)*FREQ4(I) )+X4*(FREQ4(I)*X7(J)-FREQ4
     1(J)*X7(I)) )/( FREQ4(I)*(X5-2.*X2)*(FREQ4(I)*X7(J)-FREQ4(J)*X7(I))
     2-FREQ4(I)*X3**2 )
      AMAT = ( BMAT*FREQ4(I)*( X5-2.*X2 )-X4 )/X3
      CMAT = X4/FREQ4(I) - X6*AMAT
      X=BMAT*ALOG(X6)/2.+(AMAT-BMAT*X7(I)/2.)*PI/X8(I)
      X = X + ( CMAT + BMAT*X7(J)/2. )*PI/X8(J)
      DE(I,J) = X/( 2.*AMASS(I)*AMASS(J)  )
      ED(J,I) = DE(I,J)
  120 CONTINUE
  130 CONTINUE
C
C
      DO 3000 I =1,M
      DO 3000 J = 1,M
      DDEE(I,J) = ( PI*( AMU(I)+AMU(J)) )/( AMASS(I)*AMASS(J)*( FREQ4(J)
     1 + AMU(I)**2 *FREQ(J)**2 + AMU(I)*AMU(J)*FREQ(J)**2 + FREQ4(I) +
     2 AMU(I)*AMU(J)*FREQ(I)**2 + AMU(J)**2*FREQ(I)**2 - 2.*FREQ(I)**2*
     3 FREQ(J)**2 ) )
 3000 CONTINUE
      DO 4000 I=1,M
      DO 4000 J=1,M
      DE(I,J) = DE(I,J)/SCAL**3
      ED(J,I)=DE(I,J)
C     THE SCALARS ARE RESCALED BY DIVIDING BY SCALE FACTOR CUBED.
      DDEE(I,J) = DDEE(I,J)/SCAL**3
 4000 CONTINUE
      WRITE(6,9000)
      WRITE(6,9001)
      DO 4100 I = 1,M
      WRITE(6,9002)I
      WRITE(6,9003) ( J,DE(I,J),DDEE(I,J),J=1,M)
      WRITE(6,9004)
 4100 CONTINUE
C
      MM=M-1
C
      DO 5000 I=1,MM
```

```
      M1=I+1
      DO 5000 J=M1,M
      SC1(I,J)=DE(I,J)-DE(J,I)
      SC2(I,J)=-SC1(I,J)
 5000 CONTINUE
C
      NAME = 0
      NMAT=0
      NFILE=0
      MM=M-1
C
C****
C   SMAT(1)   =   D(I)*D(J) + E(I)*E(J)     (INTEGRATE)
C   SMAT(2)   =   D(I)*E(J) - D(J)*E(I)     (INTEGRATE)
C   SMAT(3)   =   D(J)*E(I) - D(I)*E(J)     (INTEGRATE)
C
C
      NAME = 0
      NMAT = 0
      NFILE = 0
C
      DO 6000 I=1,MM
      M1=I+1
      ML = I + K
      IF( ML .GT. M ) ML = M
      DO 6000 J = M1,ML
      SMAT(1)=DDEE(I,J)
      SMAT(2)=SC1(I,J)
      SMAT(3)=SC2(I,J)
      CALL WRTETP(SMAT,1,NAME,3,1,B,NFILE,NMAT,NTAP12,IRR)
      IF ( IRR .NE. 0 ) GO TO 9999
 6000 CONTINUE
C
      DO 6050 I = 1,M
 6050 FREQ(I) = FREQ(I)*SCAL
C
      END FILE NTAP12
      END FILE NTAP16
      RETURN
 9998 WRITE(6,9010)IRR
      CALL EXIT
 9999 WRITE(6,9020)IRR
      CALL EXIT
      RETURN
 9000 FORMAT(1H1,50X,21HADMITTANCE INTEGRALS ////)
 9001 FORMAT(1H0,10X,9HD(I)*E(J),20X,22HD(I)*D(J) + E(I)*E(J)  ///)
 9002 FORMAT(1H0,3H I=I3)
 9003 FORMAT(7X,I3,1X,E14.7,10X,E14.7)
 9004 FORMAT(1H0)
 9010 FORMAT(28H ERROR IN READTP ERROR CODE=I5)
 9020 FORMAT(28H ERROR IN WRTETP ERROR CODE=I5)
      END
```

SUBROUTINE CQJD

```
$IBFTC CQJD*   DECK
       SUBROUTINE CQJD
       COMMON/BLK2/M,N,G,ALAM,CMU,K
          DIMENSION SMAT(3),B(16),CFW(90,90),QFW(90,90)
       NTAP12 = 12
       NTAP15 = 15
       NTAP17 = 2
       NTAP18 = 14
       REWIND NTAP12
       REWIND NTAP15
       REWIND NTAP17
       REWIND NTAP18
       IQ = 0
       MM = M-1
C
C
C***CALCULATE THE EXCITATIONS
C            LOOP II=1,2          WHEN II=1 THE EXCITATIONS FOR REAL PART
C                                          ARE STORED ON TAPE 15
C                                 WHEN II=2 THE EXCITATIONS FOR IMAGINARY
C                                          PART ARE STORED ON TAPE 15
       DO 500 II = 1,2
C
       DO 200 I = 1,MM
       M1 = I + 1
       ML = I+K
       IF( ML .GT. M ) ML=M
       DO 200 J = M1,ML
       NAME = 0
       NMAT = 0
       NFILE = 0
C***READ IN THE ADMITTANCE INTEGRAL SCALARS
       CALL READTP(SMAT,1,NAME,NR,NC,B,NFILE,NMAT,NTAP12,IRR)
       IF( IRR .NE. 0 ) GO TO 9998
       NAME = 0
       NMAT = 0
       NFILE = 0
C***READ IN THE CO-POWER SPECTRAL DENSITY(CFW)
       CALL READTP(CFW,90,NAME,N,N,B,NFILE,NMAT,NTAP17,IRR)
       IF( IRR .NE. 0 ) GO TO 9998
C***READ IN THE QUAD-POWER SPECTRAL DENSITY(QFW)
       NAME = 0
       NMAT = 0
       NFILE = 0
       CALL READTP(QFW,90,NAME,N,N,B,NFILE,NMAT,NTAP18,IRR)
       IF( IRR .NE. 0 ) GO TO 9998
C***FORM THE EXCITATION MATRIX TO BE STORED ON TAPE 15
       IF( IQ  .NE. 0 ) GO TO 54
       DO 50 K1=1,N
       DO 50 L1 = 1,N
 50    CFW(K1,L1) = CFW(K1,L1)*SMAT(1) + QFW(K1,L1)*SMAT(2)
       GO TO 80
 54    DO 55 K1 = 1,N
       DO 55 L1 = 1,N
 55    CFW(K1,L1) = CFW(K1,L1)*SMAT(3) + QFW(K1,L1)*SMAT(1)
 80    NAME = 0
       NMAT = 0
```

```
      NFILE = 0
      CALL WRTETP(CFW,90,NAME,N,N,B,NFILE,NMAT,NTAP15,IRR)
      IF( IRR .NE. 0 ) GO TO 9999
  200 CONTINUE
C***REWIND THE TAPES USED IN CALCULATING THE IMAGINARY PART
      REWIND NTAP12
      REWIND NTAP17
      REWIND NTAP18
      IQ = 1
  500 CONTINUE
C
C
      END FILE NTAP15
      RETURN
 9998 WRITE(6,9010) IRR
      CALL EXIT
 9999 WRITE(6,9020) IRR
      CALL EXIT
      RETURN
 9010 FORMAT(28H ERROR IN READTP-ERROR CODE=I5)
 9020 FORMAT(28H ERROR IN WRTETP-ERROR CODE=I5)
      END
```

SUBROUTINE SUMT

```
$IBFTC SUMT*   DECK
      SUBROUTINE SUMT ( NO, INTAPE)
C
C
C                  SUBROUTINE SUMT - A MATRIX WILL BE ADDED TO ITS
C                    TRANSPOSE .
C
C
C
      COMMON /BLK2/M,N,G,ALAM,CMU,K,NF,NPLATE,NBEAMS
      DIMENSION SUM(90,90),AMAT(90,90),B(16),IPARAM(2)
      NTAP8 = 8
      REWIND NTAP8
      REWIND INTAPE
C
C
C
C***CYCLE ON REAL AND IMAGINARY
      DO 600 III = 1,2
      IF ( III .EQ. 1 ) GO TO 50
      NMAT = 0
C
C***SET SIGN = -1  FOR THE IMAGINARY PART
      SIGN = -1.0
      GO TO 60
 50   NMAT = 1
C
C***SET SIGN = 1  FOR THE REAL PART
      SIGN = 1.0
 60   CONTINUE
C
      DO 500 II = 1,NO
      NAME = 0
      NFILE = 0
      CALL READTP(AMAT,90,NAME,N,N,B,NFILE,NMAT,INTAPE,IRR)
      IF ( IRR .NE. 0 ) GO TO 9985
      DO 100 I = 1,N
      DO 100 J = 1,N
 100  SUM(I,J) = AMAT(I,J) + AMAT(J,I)*SIGN
      NAME = 0
      NMAT = 0
      NFILE = 0
      CALL WRTETP(SUM,90,NAME,N,N,B,NFILE,NMAT,NTAP8,IRR)
      IF ( IRR .NE. 0 ) GO TO 9986
 500  CONTINUE
 600  CONTINUE
C
C
C
      END FILE NTAP8
      REWIND NTAP8
      REWIND INTAPE
      NAME = 0
      NMAT = 0
      NFILE = 0
C
C***A PARAMETER MATRIX IPARAM IS FORMED
```

```
      IPARAM(1) = NPLATE + 2*NBEAMS
      IPARAM(2) = NO
      CALL WRTETP(IPARAM,1,NAME,1,2,B,NFILE,NMAT,INTAPE,IRR)
      IF ( IRR .NE. 0 ) GO TO 9986
C
C
C
C***CYCLE ON REAL AND IMAGINARY
      DO 900 III = 1,2
      DO 800 II = 1,NO
      NAME = 0
      NMAT = 0
      NFILE = 0
      CALL READTP(AMAT,90,NAME,N,N,B,NFILE,NMAT,NTAP8,IRR)
      IF ( IRR .NE. 0 ) GO TO 9985
      NAME = 0
      NMAT = 0
      NFILE = 0
      CALL WRTETP(AMAT,90,NAME,N,N,B,NFILE,NMAT,INTAPE,IRR)
      IF ( IRR .NE. 0 ) GO TO 9986
 800  CONTINUE
 900  CONTINUE
C
C
C
      END FILE INTAPE
      REWIND INTAPE
      RETURN
 9985 WRITE(6,9990) IRR
      CALL EXIT
 9986 WRITE(6,9991) IRR
      CALL EXIT
      RETURN
 9990 FORMAT(28H ERROR IN READTP-ERROR CODE=I5)
 9991 FORMAT(28H ERROR IN WRTETP-ERROR CODE=I5)
      END
```

SUBROUTINE SUM2

```
$IBFTC SUM2*   DECK
      SUBROUTINE SUM2( ITP1,ITP2,NO,NCN )
C
C
C                  SUBROUTINE SUM2 - SUMS THE OPTION 3 LIKE
C                    MODE EFFECTS TO UNLIKE MODE EFFECTS TO FORM
C                    TOTAL OPTION 2 RESULTS.
C
C
C
      COMMON /BLK2/M,N,G,ALAM,CMU,K,NF,NPLATE,NBEAMS
      DIMENSION AMAT(90,90),BMAT(90,90),B(16),IPARAM(2)
      REWIND ITP1
      REWIND ITP2
      NAME = 0
      NMAT = 1
      NFILE = 0
      CALL READTP(AMAT,90,NAME,N,N,B,NFILE,NMAT,ITP1,IRR)
      IF ( IRR .NE. 0 ) GO TO 9985
      REWIND ITP1
      NAME = 0
      NMAT = 1
      NFILE = 0
      CALL READTP(BMAT,90,NAME,N,N,B,NFILE,NMAT,ITP2,IRR)
      IF ( IRR .NE. 0 ) GO TO 9985
C
      DO 100 I = 1,N
      DO 100 J = 1,N
 100  AMAT(I,J) = AMAT(I,J) + BMAT(I,J)
C
C*** ADD OPTION 3 LIKE MODES TO OPTION 2 UNLIKE MODE EFFECTS TO
C     GET THE TOTAL JOINT DEFLECTIONS
      IF ( NCN .EQ. 1 ) GO TO 200
      WRITE(6,9000)
      GO TO 250
 200  WRITE(6,9001)
C
 250  DO 500 I = 1,N
      WRITE(6,9002)I,(AMAT(I,J),J=1,N)
 500  CONTINUE
C
C
C***A PARAMETER MATRIX NSTO IS FORMED
      IPARAM(1) = NPLATE + 2*NBEAMS
      IPARAM(2) = NO
      NAME = 0
      NMAT = 0
      NFILE = 0
      CALL WRTETP(IPARAM,1,NAME,1,2,B,NFILE,NMAT,ITP1,IRR)
      IF ( IRR .NE. 0 ) GO TO 9986
      CALL WRTETP(AMAT,90,NAME,N,N,B,NFILE,NMAT,ITP1,IRR)
      IF ( IRR .NE. 0 ) GO TO 9986
      CALL READTP(AMAT,90,NAME,N,N,B,NFILE,NMAT,ITP2,IRR)
      IF ( IRR .NE. 0 ) GO TO 9985
      CALL WRTETP(AMAT,90,NAME,N,N,B,NFILE,NMAT,ITP1,IRR)
      IF ( IRR .NE. 0 ) GO TO 9986
      IF ( NCN .EQ. 1 ) GO TO 800
```

SUBROUTINE SECM3

```
$IBFTC SECM3*  DECK
      SUBROUTINE SECM3
C
C
C
C
C
C                 *
C                  * A D M I T T A N C E   I N T E G R A L S *
C                 *                                          *
C               *   U S E D   I N   F O R M I N G   T H E    *
C              *                                              *
C              *   D E F L E C T I O N   S E C O N D          *
C               *                                            *
C                * M O M E N T S .                           *
C                 *                                          *
C                  *                                        *
C
C
C                          ( OPTION 3 )
C
C
C  OPTION 3     BROAD BAND EXCITATIONS, DAMPING COEFFICIENTS PROPORTIONAL
C               TO A LINEAR COMBINATION OF THE MASS INERTIA AND STIFFNESS
C               COEFFICIENTS. ( NO CROSS MODAL COUPLING IS INCLUDED )
C
C  M            NUMBER OF FREQUENCIES, MASS AND MODE SHAPES
C  N            NUMBER OF RETAINED DEGREES OF FREEDOM
C  G            STRUCTURAL 4AMPING
C  ALAM         A DAMPING PROPORTIONALITY FACTOR PROP. TO STIFFNESS
C  CMU          A DAMPING PROPORTIONALITY FACTOR FOR DAMPING PROPORTIONAL
C                 TO MASS.
      D+MENSION FREQ(25), AMASS(25), AMU(25), DIAG(25), PHIM(25),
     1          PHI(100,25),B(16),NSTO(1)
      COMMON /BLK1/ FREQ, AMASS
      COMMON /BLK2/ M,N,G,ALAM,CMU
      NTAP1 = 10
      NTAP11 = 11
      REWIND NTAP1
      REWIND NTAP11
      NSTO(1) = M
      NAME = 0
      NMAT = 0
      NFILE = 0
      CALL WRTETP(NSTO,1,NAME,1,1,B,NFILE,NMAT,NTAP11,IRR)
      IF ( IRR .NE. 0 ) GO TO 9986
C***                                        DAMPING FACTOR
C
      DO 10 I = 1,M
 10   AMU(I) = CMU + ALAM*FREQ(I)**2 + G*FREQ(I)
C
C
C***                                        DIAGONAL SCALARS ARE COMPUTED
      DO 20 I = 1,M
      PI = 3.14159265
 20   DIAG(I) = PI/( 2.0*AMU(I)*AMASS(I)**2 )
      NAME = 0
```

```
      WRITE(6,9003)
      GO TO 900
  800 WRITE(6,9004)
C
  900 DO 950 I = 1,N
      WRITE(6,9002)I,(AMAT(I,J),J=1,N)
  950 CONTINUE
C
      END FILE ITP1
      RETURN
 9985 WRITE(6,9990) IRR
      CALL EXIT
 9986 WRITE(6,9991) IRR
      CALL EXIT
      RETURN
 9000 FORMAT(1H1,30X,55HDEFLECTION SECOND SPECTRAL MOMENT MATRIX ( REAL
     1PART )   ///)
 9001 FORMAT(1H1,30X,43HDEFLECTION COVARIANCE MATRIX ( REAL PART )  ///)
 9002 FORMAT(1H0,I5,1P7E16.6/(E22.6,6E16.6) )
 9003 FORMAT(1H1,30X,59HDEFLECTION SECOND SPECTRAL MOMENT MATRIX ( IMAGI
     1NARY PART ) //)
 9004 FORMAT(1H1,30X,48HDEFLECTION COVARIANCE MATRIX ( IMAGINARY PART )
     1  //)
 9990 FORMAT(28H ERROR IN READTP-ERROR CODE=I5)
 9991 FORMAT(28H ERROR IN WRTETP-ERROR CODE=I5)
      END
```

```
      NMAT = 0
      NFILE = 0
      CALL WRTETP(DIAG,1,NAME,1,M,B,NFILE,NMAT,NTAP11,IRR)
      NAME = 0
      NMAT = 1
      NFILE = 2
C
C***                                          THE MODE SHAPES ARE READ IN.
C
      CALL READTP(PHI,100,NAME,N,M,B,NFILE,NMAT,NTAP1,IRR)
      IF ( IRR .NE. 0 ) GO TO 9985
      NAME = 0
      NMAT = 0
      NFILE =0
C
      DO 50 J=1,M
      DO 40 I=1,N
 40   PHIM(I) = PHI(I,J)
      CALL WRTETP(PHIM,1,NAME,N,1,B,NFILE,NMAT,NTAP11,IRR)
      IF ( IRR .NE. 0 ) GO TO 9986
 50   CONTINUE
C
      END FILE NTAP11
      RETURN
 9985 WRITE(6,9990) IRR
      CALL EXIT
 9986 WRITE(6,9991) IRR
      CALL EXIT
      RETURN
 9990 FORMAT(28H ERROR IN READTP-ERROR CODE=I5)
 9991 FORMAT(28H ERROR IN WRTETP-ERROR CODE=I5)
      END
```

SUBROUTINE SECM2

```
$IBFTC SECM2*  DECK
       SUBROUTINE SECM2
C
C                *
C               * ADMITTANCE   INTEGRALS *
C              *                         *
C             *   USED   IN   FORMING   THE   *
C            *                               *
C             *   DEFLECTION   SECOND        *
C              *                             *
C               * MOMENTS.                   *
C                *                           *
C                 *                         *
C
C
C
C
C   OPTION 2     BROAD BAND EXCITATION, DAMPING COEFFICIENTS PROPORTIONAL
C                TO A LINEAR COMBINATION OF THE MASS INERTIA AND STIFFNESS
C                COEFFICIENTS. ( CROSS MODAL COUPLING IS INCLUDED )
C
C
C   M            NUMBER OF FREQUENCIES, MASS AND MODE SHAPES
C
C
C                THE FREQUENCIES(FREQ), GENERALIZED MASS(AMASS) AND
C                MODE SHAPES(PHI) COME FROM THE EIGENVECTOR-EIGENVALUE
C                ROUTINE TV-105W.
C
C   N            NUMBER OF RETAINED DEGREES OF FREEDOM
C   NF           NUMBER OF FREQUEUNCIES TO CALCULATE THE CPSD
C   G            STRUCTURAL DAMPING
C   ALAM         A DAMPING PROPORTIONALITY FACTOR PROP. TO STIFFNESS
C   CMU          A DAMPING PROPORTIONALITY FACTOR FOR DAMPING PROPORTIONAL
C                  TO MASS.
C
C***TAPE OUTPUT STORAGE - TAPE 16
C                    *****
C      MATRIX 1      *M-1* PARAMETER MATRIX(3X1)
C                    *K  *
C                    *N  *
C                    *****
C
C      MATRIX 2        0     PARAMETER K MATRIX USED FOR OFF-DIAG TEST
C                   OR -1        ( M-1 BY 1 )
C
C
C      MATRIX 3        PHI(I,J)     I=1,M-1
C                                   J=I,I+K
C
C
C
C
      DIMENSION DE(25,25), ED(25,25), DDEE(25,25), AMU(25), AMASS(25),
     1FREQ(25),OM4(25),X7(25),FREQ4(25),X8(25),SC1(25,25),
     2SC2(25,25),SMAT(3),B(16),PHI(100,25),PHIM(100),NSTO(3),KPARAM(24)
      COMMON /BLK1/ FREQ,AMASS
```

```
      COMMON /BLK2/ M,N,G,ALAM,CMU,K
      PI = 3.14159265
      NTAP1 = 10
      NTAP12 = 12
      NTAP16=16
      REWIND NTAP1
      REWIND NTAP12
      REWIND NTAP16
C
      DO 20 I=1,M
 20   AMU(I)=CMU+ALAM*FREQ(I)**2+G*FREQ(I)
C
C     THE FREQUENCIES AND MU ARE SCALED
      CALL SCALE( FREQ, M, SCAL )
      DO 25 I = 1,M
 25   AMU(I) = AMU(I)/SCAL
      MM=M-1
      NSTO(1) = MM
      NSTO(2) = K
      NSTO(3) = N
      NAME = 0
      NMAT = 0
      NFILE = 0
      CALL WRTETP(NSTO,1,NAME,3,1,B,NFILE,NMAT,NTAP16,IRR)
      IF ( IRR .NE. 0 ) GO TO 9999
C***FORM PARAMETER K MATRIX TO TEST FOR NUMBER OF OFF-DIAGONAL TERMS
C        ARE DESIRED
C
      MK = M-K
      DO 30 I=1,MK
 30   KPARAM(I) = 0
      MK = MK+1
      IF ( MK .GT. MM ) GO TO 50
      DO 40 I = MK,MM
 40   KPARAM(I) = -1
 50   NAME = 0
      NMAT = 0
      NFILE = 0
      IF ( K .EQ. 1 ) GO TO 55
      GO TO 58
 55   KPARAM(M) = -1
      MM = M
 58   CONTINUE
      CALL WRTETP(KPARAM,1,NAME,MM,1,B,NFILE,NMAT,NTAP16,IRR)
      IF ( IRR .NE. 0 ) GO TO 9999
      IF ( K .EQ. 1 ) MM = M-1
      NAME = 0
      NMAT=1
      NFILE = 2
      CALL READTP(PHI,100,NAME,N,M,B,NFILE,NMAT,NTAP1,IRR)
      IF(IRR .NE. 0)GO TO 9998
      NAME = 0
      NMAT=0
      NFILE = 0
C     THE MODE SHAPES PHI ARE STORED ON TAPE
C****
      DO 90 II=1,MM
      J1 = II
      ML = II+K
      IF( ML .GT. M ) ML = M
```

```
      DO 90 J = J1,ML
      DO 80 I=1,N
   80 PHIM(I)=PHI(I,J)
      CALL WRTETP(PHIM,1,NAME,N,1,B,NFILE,NMAT,NTAP16,IRR)
      IF ( IRR .NE. 0 ) GO TO 9999
   90 CONTINUE
C****
C****
C           THE ADMITTANCE INTEGRALS FOR OPTION 2 ARE FORMED AT M
C            NUMBER OF FREQUENCIEV.
C****
      DO 100 I=1,M
      FREQ4(I)=FREQ(I)**4
      X7(I)=AMU(I)**2-2.*FREQ(I)**2
  100 X8(I)=SQRT(4.*FREQ(I)**2-AMU(I)**2)*AMU(I)
C
C
      DO 130 I=1,M
      DO 120 J=1,M
      IF(I .NE. J)GO TO 110
      DE(I,J) = 0.
      ED(J,I) = 0.
      GO TO 120
  110 X2=FREQ(I)**2-FREQ(J)**2
      X3 = FREQ4(I) - FREQ4(J)
      X4 = AMU(J)*FREQ(I)**4
      X5 = AMU(I)**2 - AMU(J)**2
      X6 = FREQ(J)**4/FREQ(I)**4
      BMAT = (  AMU(J)*FREQ4(I)*FREQ(I)**2*X3 + X4*( FREQ4(I)*X7(J)-FREQ
     1       4(J)*X7(I) ) )/( FREQ4(I)*( X5-2.0*X2 )*( FREQ4(I)*X7(J)-
     2       FREQ4(J)*X7(I) ) - FREQ4(I)*X3**2  )
      AMAT = ( BMAT*FREQ4(I)*( X5-2.*X2 )-X4 )/X3
      CMAT = -AMAT*X6
      X = BMAT*ALOG(X6)/2.0 + ( AMAT-BMAT*X7(I)/2.0 )*PI/X8(I)
      X = X + ( CMAT + BMAT*X7(J)/2. )*PI/X8(J)
      DE(I,J) = X/( 2.*AMASS(I)*AMASS(J)  )
      ED(J,I) = DE(I,J)
  120 CONTINUE
  130 CONTINUE
C
C
      DO 3000 I =1,M
      DO 3000 J = 1,M
      IF( I .EQ. J ) GO TO 3000
 3000 DDEE(I,J) = (  PI*( AMU(I)*FREQ(J)**2+ AMU(J)*FREQ(I)**2 ) )/(
     1             AMASS(I)*AMASS(J)*( FREQ4(J) + AMU(I)**2*FREQ(J)**2 +
     2             AMU(I)*AMU(J)*FREQ(J)**2 + FREQ4(I) + AMU(I)*AMU(J)
     3             *FREQ(I)**2 + AMU(J)**2*FREQ(I)**2 - 2.0*FREQ(I)**2*
     4             FREQ(J)**2 ) )
      DO 4000 I=1,M
      DO 4000 J=1,M
      DE(I,J) = DE(I,J)/SCAL
      ED(J,I)=DE(I,J)
      IF( I .EQ. J ) GO TO 4000
C     THE SCALARS ARE RESCALED BY DIVIDING BY SCALE FACTOR CUBED.
      DDEE(I,J) = DDEE(I,J)/SCAL
 4000 CONTINUE
      MM=M-1
      DO 5000 I=1,MM
      M1=I+1
```

```
      DO 5000 J=M1,M
      SC1(I,J)=DE(I,J)-DE(J,I)
      SC2(I,J)=-SC1(I,J)
 5000 CONTINUE
      NAME = 0
      NMAT=0
      NFILE=0
      MM=M-1
C
C****
C  SMAT(1)  =  D(I)*D(J) + E(I)*E(J)     (INTEGRATE)
C  SMAT(2)  =  D(I)*E(J) - D(J)*E(I)     (INTEGRATE)
C  SMAT(3)  =  D(J)*E(I) - D(I)*E(J)     (INTEGRATE)
C
C
      NAME = 0
      NMAT = 0
      NFILE = 0
      DO 6000 I=1,MM
      M1=I+1
      ML = I + K
      IF( ML .GT. M ) ML = M
      DO 6000 J = M1,ML
      SMAT(1)=DDEE(I,J)
      SMAT(2)=SC1(I,J)
      SMAT(3)=SC2(I,J)
      CALL WRTETP(SMAT,1,NAME,3,1,B,NFILE,NMAT,NTAP12,IRR)
      IF ( IRR .NE. 0 ) GO TO 9999
 6000 CONTINUE
 54   CONTINUE
C
      DO 6050 I = 1,M
 6050 FREQ(I) = FREQ(I)*SCAL
      END FILE NTAP12
      END FILE NTAP16
      RETURN
 9998 WRITE(6,9010)IRR
      CALL EXIT
 9999 WRITE(6,9020)IRR
      CALL EXIT
      RETURN
 9010 FORMAT(28H ERROR IN READTP ERROR CODE=I5)
 9020 FORMAT(28H ERROR IN WRTETP ERROR CODE=I5)
      END
```

SUBROUTINE ADMIT3

```
$IBFTC ADMT3*  DECK
      SUBROUTINE ADMIT3
C
C                 *                                   *
C                  * A D M I T T A N C E    S C A L A R S *
C                *                                       *
C               *   USED IN CALCULATING THE DEFLECTION     *
C                *  RESPONSE CROSS POWER SPECTRAL DENSITY  *
C                 * ( CPSD )                             *
C                  *                                   *
C                   *       O P T I O N   3           *
C
C
C  OPTION 3           BROAD BAND EXCITATION, DAMPING COEFFICIENTS
C                     PROPORTIONAL TO A LINEAR COMBINATION OF THE
C                     MASS INERTIA AND STIFFNESS COEFFICIENTS.
C                      ( NO CROSS MODAL COUPLING )
C
C  D(I)               REAL FACTOR IN THE DIAGONAL ADMITTANCE SCALAR
C
C  E(I)               IMAGINARY FACTOR IN THE DIAGONAL ADMITTANCE SCALAR
C
C  N                  NUMBER OF RETAINED DEGREES OF FREEDOM
C  M                  NUMBER OF FREQUENCIES(FREQ).GENERALIZED MASS(AMASS)
C                     AND MODE SHAPES(PHI)
C  NF                 NUMBER OF FREQUEUNCIES TO CALCULATE THE CPSD
C  G                  STRUCTURAL DAMPING
C  ALAM               A DAMPING PROPORTIONALITY FACTOR
C  CMU                A DAMPING PROPORTIONALITY FACTOR
C
C
      DIMENSION FREQ(25), AMASS(25), OMEG(100), PHI(100,25),PHIM(100),
     1          AMU(25), D(25), E(25), DD(25), B(16), NSTO(3)
      COMMON /BLK1/ FREQ,AMASS,OMEG
      COMMON /BLK2/ M,N,G,ALAM,CMU,K,NF
      COMMON/BLK3/ FLG1,FLG2,FLG3,FLG4
      INTEGER FLG1,FLG2,FLG3,FLG4
      NTAP1 = 10
      NTAP2 = 2
      NTAP11 = 11
      REWIND NTAP1
      REWIND NTAP2
      REWIND NTAP11
      NAME = 0
      NMAT = 1
      NFILE = 2
C
C***                THE MODE SHAPES ARE READ IN.
C
      CALL READTP(PHI,100,NAME,N,M,B,NFILE,NMAT,NTAP1,IRR)
      IF ( IRR .NE. 0 ) GO TO 9985
      NSTO(1) = M
      NSTO(2) = M
      NSTO(3) = NF
      NAME = 0
      NMAT = 0
      NFILE = 0
```

```
      CALL WRTETP( NSTO, 1, NAME,3,1,B,NFILE,NMAT,NTAP2,IRR)
      IF ( IRR .NE. 0 ) GO TO 9986
      NAME = 0
      NMAT = 0
      NFILE = 0
      DO 20 J = 1,M
      DO 10 I = 1,N
 10   PHIM(I) = PHI(I,J)
      CALL WRTETP(PHIM,1,NAME,N,1,B,NFILE,NMAT,NTAP11,IRR)
      IF ( IRR .NE. 0 ) GO TO 9986
 20   CONTINUE
C
C
C
C
C****
C****
      WRITE(6,9000)
      NAME = 0
      NMAT = 0
      NFILE = 0
C               THE ADMITTANCE SCALARS FOR OPTION 3 ARE FORMED AT NF
C               NUMBER OF FREQUENCIES.
C****
C****
C
      DO 40 II=1,NF
      WRITE(6,9001) OMEG(II)
      DO 25 J=1,M
 25   AMU(J) = CMU + ALAM*FREQ(J)**2 + G*FREQ(J)**2/OMEG(II)
      DO 30 I=1,M
      DEN = AMASS(I)*( ( FREQ(I)**2-OMEG(II)**2 )**2 + OMEG(II)**2*(
     1 AMU(I) )**2 )
      D(I) = ( FREQ(I)**2-OMEG(II)**2 )/DEN
      E(I) = OMEG(II)*AMU(I)/DEN
 30   DD(I)=D(I)**2+E(I)**2
      WRITE(6,9002) ( I, DD(I),I=1,M )
      CALL WRTETP(DD,1,NAME,1,M,B,NFILE,NMAT,NTAP2,IRR)
      IF ( IRR .NE. 0 ) GO TO 9986
 40   CONTINUE
C
C
      END FILE NTAP2
      END FILE NTAP11
      RETURN
 9985 WRITE(6,9990) IRR
      CALL EXIT
 9986 WRITE(6,9991) IRR
      CALL EXIT
      RETURN
 9000 FORMAT(1H1,40X,32HADMITTANCE SCALARS ( D**2+E**2 )//// )
 9001 FORMAT(1H0,10HFREQUENCY=E14.7,9H(RAD/SEC)  )
 9002 FORMAT(1X,I5,5X,E14.7 )
 9990 FORMAT(28H ERROR IN READTP-ERROR CODE=I5)
 9991 FORMAT(28H ERROR IN WRTETP-ERROR CODE=I5)
      END
```

SUBROUTINE ADMIT2

```
$IBFTC ADMT2*  DECK
       SUBROUTINE ADMIT2
C                *                                          *
C               * A D M I T T A N C E   S C A L A R S       *
C              *                                             *
C              *  USED IN CALCULATING THE DEFLECTION         *
C               * RESPONSE CROSS POWER SPECTRAL DENSITY      *
C                *  OR CPSD.                                *
C                 *                                        *
C
C
C
C   OPTION 2     BROAD BAND EXCITATION, DAMPING COEFFICIENTS PROPORTIONAL
C                TO A LINEAR COMBINATION OF THE MASS INERTIA AND STIFFNESS
C                COEFFICIENTS, ( CROSS MODAL COUPLING IS INCLUDED )
C
C
C
C   D(I)         ADMITTANCE SCALAR -  DIAGONAL MATRIX, REAL FACTOR IN
C                                     ADMITTANCE MATRIX.
C   E(I)         ADMITTANCE SCALAR -  DIAGONAL MATRIX, IMAGINARY FACTOR
C                                     IN ADMITTANCE MATRIX.
C
C
C                THE FREQUENCIES(FREQ), GENERALIZED MASS(AMASS) AND
C                MODE SHAPES(PHI) COME FROM THE EIGENVECTOR-EIGENVALUE
C                ROUTINE TV-105W.
C
C   N            NUMBER OF RETAINED DEGREES OF FREEDOM
C   NF           NUMBER OF FREQUEUNCIES TO CALCULATE THE CPSD
C   G            STRUCTURAL DAMPING
C   ALAM         A DAMPING PROPORTIONALITY FACTOR
C   CMU          A DAMPING PROPORTIONALITY FACTOR
C
C
      DIMENSION D(25),E(25),DD(25),OMEG(100),B(16),SC1(25,25),SC2(25,25)
     1          ,DDEE(25,25),SMAT(3),DE(25,25),ED(25,25),AMU(25),AMASS(
     2          25),FREQ(25),PHI(100,25),PHIM(100),NSTO(4),KPARAM(24)
      COMMON /BLK1/ FREQ,AMASS,OMEG
      COMMON /BLK2/ M,N,G,ALAM,CMU,K,NF
      NTAP1 = 10
      NTAP2 = 2
      NTAP16 = 16
      REWIND NTAP1
      REWIND NTAP2
      REWIND NTAP16
      MM = M-1
C
C***FORM PARAMETER MATRIX NSTO
      NSTO(1) = MM
      NSTO(2) = K
      NSTO(3) = N
      NSTO(4) = NF
      NAME = 0
      NMAT = 0
```

```
      NFILE = 0
      CALL WRTETP(NSTO,1,NAME,4,1,B,NFILE,NMAT,NTAP16,IRR)
      IF( IRR .NE. 0 ) GO TO 9999
C***FORM PARAMETER K MATRIX TO TEST FOR NUMBER OF OFF-DIAGONAL TERMS
      MK = M-K
      DO 10 I = 1,MK
 10   KPARAM(I) = 0
      MK = MK + 1
      IF ( MK .GT. MM ) GO TO 30
      DO 20 I = MK,MM
 20   KPARAM(I) = -1
 30   NAME = 0
      NMAT = 0
      NFILE = 0
      IF ( K .EQ. 1 ) GO TO 35
      GO TO 38
 35   KPARAM(M) = -1
      MM = M
 38   CONTINUE
      CALL WRTETP(KPARAM,1,NAME,MM,1,B,NFILE,NMAT,NTAP16,IRR)
      IF( IRR .NE. 0 ) GO TO 9999
      IF ( K .EQ. 1 ) MM = M-1
      NAME = 0
      NMAT =  1
      NFILE = 2
C
C               THE MODE SHAPES ARE READ
C
      CALL READTP(PHI,100,NAME,N,M,B,NFILE,NMAT,NTAP1,IRR)
      IF( IRR .NE. 0 ) GO TO 9998
      NAME = 0
      NMAT = 0
      NFILE = 0
      DO 70 II = 1,MM
      J1 = II
      ML = II+K
      IF( ML .GT. M ) ML = M
      DO 70 J = J1,ML
      DO 60 I = 1,N
 60   PHIM(I) = PHI(I,J)
      CALL WRTETP( PHIM,1,NAME,N,1,B,NFILE,NMAT,NTAP16,IRR )
      IF ( IRR .NE. 0 ) GO TO 9999
 70   CONTINUE
      END FILE NTAP16
C
C****
C               THE ADMITTANCE SCALARS FOR OPTION 2 ARE FORMED
C  SMAT(1)  =  D(I)*D(J) + E(I)*E(J)
C  SMAT(2)  =  D(I)*E(J) - D(J)*E(I)
C  SMAT(3)  =  D(J)*E(I) - D(I)*E(J)
C
C
      END FILE NTAP2
C
      DO 1000 II=1,NF
      DO 100 J=1,M
 100  AMU(J)=CMU+ALAM*FREQ(J)**2+G*FREQ(J)**2/OMEG(II)
C
      DO 200 I=1,M
      DEN = AMASS(I)*( ( FREQ(I)**2-OMEG(II)**2 )**2 + OMEG(II)**2*(AMU
```

```
     1       (I) )**2 )
      D(I) = ( FREQ(I)**2-OMEG(II)**2 )/DEN
  200 E(I)=OMEG(II)*AMU(I)/DEN
C
      DO 300 I=1,M
      DO 300 J=1,M
      DE(I,J)=D(I)*E(J)
      ED(J,I)=DE(I,J)
  300 DDEE(I,J)=D(I)*D(J)+E(I)*E(J)
C
      WRITE(6,9000)
      WRITE(6,9001)
C
      DO 4100 I = 1,M
      WRITE(6,9002)I
      WRITE(6,9003) ( J,DE(I,J),DDEE(I,J),J=1,M)
      WRITE(6,9004)
 4100 CONTINUE
C
      DO 400 I=1,MM
      M1=I+1
      DO 400 J=M1,M
      SC1(I,J) = DE(I,J)-DE(J,I)
  400 SC2(I,J)=-SC1(I,J)
      NAME = 0
      NMAT = 0
      NFILE = 0
      DO 500 I=1,MM
      M1=I+1
      ML = I + K
      IF( ML .GT. M ) ML = M
      DO 500 J = M1,ML
      SMAT(1) = DDEE(I,J)
      SMAT(2) = SC1(I,J)
      SMAT(3) = SC2(I,J)
      CALL WRTETP(SMAT,1,NAME,3,1,B,NFILE,NMAT,NTAP2,IRR)
      IF(IRR .NE. 0)GO TO 9999
  500 CONTINUE
      END FILE NTAP2
 1000 CONTINUE
C
C
C
      END FILE 2
      RETURN
 9998 WRITE(6,9010)IRR
      CALL EXIT
 9999 WRITE(6,9020)IRR
      CALL EXIT
      RETURN
 9000 FORMAT(1H1,50X,21HADMITTANCE SCALARS   ////)
 9001 FORMAT(1H0,10X,9HD(I)*E(J),20X,22HD(I)*D(J) + E(I)*E(J)  ///)
 9002 FORMAT(1H0,3H I=I3)
 9003 FORMAT(7X,I3,1X,E14.7,10X,E14.7)
 9004 FORMAT(1H0)
 9010 FORMAT( 28H ERROR IN READTP-ERROR CODE=I5 )
 9020 FORMAT(28H ERROR IN WRTETP ERROR CODE=I5)
      END
```

SUBROUTINE CQCPSD

```
$IBFTC CQCPS*  DECK
       SUBROUTINE CQCPSD
       COMMON/BLK2/M,N,G,ALAM,CMU,K,NF
          DIMENSION SMAT(3),CFW(90,90),QFW(90,90),B(16)
C***ROUTINE THAT STORES THE EXCITATIONS FOR THE CROSS POWER SPECTRAL
C            DENSITY ON TAPE
C
C    FSF     FORWARD SPACE FILES
C            CALLING SEQUENCE        CALL FSF(NFILE,NTAP18,LERROR)
C                                 WHERE
C                                      NFILE=NO OF FILES TO FORWARD SPACE
C                                     NTAP18=LOGICAL TAPE UNIT
C                                     LERROR= 0     SUCCESS
C                                         NON-ZERO NO OF UNSPACED FILES
C
C    BSF     BACKSPACE FILES         SAME CALLING SEQUENCE AS FSF
C
       NTAP2 = 2
       NTAP15= 15
       NTAP18 = 14
       REWIND NTAP2
       REWIND NTAP15
       REWIND NTAP18
       IQ = 0
C
       MM = M-1
       DO 600 IJ = 1,2
       IF ( IJ .EQ. 2 ) IQ = 1
C***LOOP NO OF FREQUENCIES
       DO 500 II = 1,NF
       NFILE = 1
       CALL FSF( NFILE,NTAP2,LERROR )
       DO 200 I = 1,MM
       M1 = I + 1
       ML = I+K
       IF( ML .GT. M ) ML = M
       DO 200 J = M1,ML
       NAME = 0
       NMAT = 0
       NFILE = 0
C***READ IN THE ADMITTANCE SCALARS
       CALL READTP(SMAT,1,NAME,NR,NC,B,NFILE,NMAT,NTAP2,IRR)
       IF( IRR .NE. 0 ) GO TO 9998
C***READ THE CO-POWER SPECTRAL DENSITY(CFW)
       NAME = 0
       NMAT = 0
       NFILE = 1
       CALL READTP(CFW,90,NAME,N,N,B,NFILE,NMAT,NTAP18,IRR)
       IF( IRR .NE. 0 ) GO TO 9998
C***READ IN THE QUAD-POWER SPECORAL DENSITY(QFW)
       NAME = 0
       NMAT = 0
       NFILE = 0
       CALL READTP(QFW,90,NAME,N,N,B,NFILE,NMAT,NTAP18,IRR)
       IF( IRR .NE. 0 ) GO TO 9998
       IF ( IQ .NE. 0 ) GO TO 54
C
```

```
C
C***FORM THE EXCITATION MATRIX TO BE STORED ON TAPE 15
      DO 50 K1 = 1,N
      DO 50 L1 = 1,N
 50   CFW(K1,L1) = CFW(K1,L1)*SMAT(1) + QFW(K1,L1)*SMAT(2)
C
C
      GO TO 80
 54   CONTINUE
C***READ THE QUAD-POWER SPECTRAL DENSITY(QFW)
      DO 60 K1 = 1,N
      DO 60 L1 = 1,N
 60   CFW(K1,L1) = CFW(K1,L1)*SMAT(3) + QFW(K1,L1)*SMAT(1)
 80   CONTINUE
      NAME = 0
      NMAT = 0
      NFILE = 0
      CALL WRTETP(CFW,90,NAME,N,N,B,NFILE,NMAT,NTAP15,IRR)
      IF( IRR .NE. 0 ) GO TO 9999
      NFILE = 1
      CALL BSF(NFILE,NTAP18,LERROR)
      IF( LERROR .NE. 0 ) GO TO 9997
 200  CONTINUE
C
C
      IF ( I .EQ. MM ) GO TO 500
      NFILE = 1
      CALL FSF(NFILE,NTAP18,LERROR )
 500  CONTINUE
C
C
      REWIND NTAP2
      REWIND NTAP18
 600  CONTINUE
      END FILE NTAP15
C
      RETURN
C
C
 9997 WRITE(6,9005) LERROR
      CALL EXIT
 9998 WRITE(6,9010) IRR
      CALL EXIT
 9999 WRITE(6,9020) IRR
      CALL EXIT
      RETURN
 9005 FORMAT(67H ERROR IN BSF ROUTINE(BACKSPACE FILES)-NUMBER OF UNSPACE
     1D FILES IS I5 )
 9010 FORMAT(28H ERROR IN READTP-ERROR CODE=I5)
 9020 FORMAT(28H ERROR IN WRTETP-ERROR CODE=I5)
      END
```

SUBROUTINE SUM3

```
$IBFTC SUM3*   DECK
      SUBROUTINE SUM3( ITP1,ITP2,ITP3)
C
C
C                    CROSS MODAL EFFECT OF CROSS-PSD MATRICES WILL
C                      BE SUMMED AT EACH FREQUENCY FOR REAL AND
C                      IMAGINARY PART.
C
C
C
      COMMON/BLK1/FREQ,AMASS,OMEGA
      COMMON /BLK2/M,N,G,ALAM,CMU,K,NF,NPLATE,NBEAMS
      DIMENSION AMAT(90,90),BMAT(90,90),B(16),IPARAM(2)
      DIMENSION FREQ(25),AMASS(25),OMEGA(100)
      REWIND ITP1
      REWIND ITP2
      REWIND ITP3
      IPARAM(1) = NPLATE + 2*NBEAMS
      IPARAM(2) = 2*NF
      NAME = 0
      NMAT = 0
      NFILE = 0
      CALL WRTETP(IPARAM,1,NAME,1,2,B,NFILE,NMAT,ITP3,IRR)
      IF ( IRR .NE. 0 ) GO TO 9986
      NAME = 0
      NFILE = 0
C
      DO 500 II = 1,NF
      WRITE(6,9000) OMEGA(II)
      IF ( II .EQ. 1 ) GO TO 10
      NMAT = 0
      GO TO 20
 10   NMAT = 1
 20   CALL READTP(AMAT,90,NAME,N,N,B,NFILE,NMAT,ITP1,IRR)
      CALL READTP(BMAT,90,NAME,N,N,B,NFILE,NMAT,ITP2,IRR)
      IF ( IRR .NE. 0 ) GO TO 9985
      DO 50 I = 1,N
      DO 50 J = 1,N
 50   AMAT(I,J) = AMAT(I,J) + BMAT(I,J)
      DO 450 I = 1,N
      WRITE(6,9001)I,(AMAT(I,J),J=1,N)
 450  CONTINUE
C
      NMAT = 0
      CALL WRTETP(AMAT,90,NAME,N,N,B,NFILE,NMAT,ITP3,IRR)
      IF ( IRR .NE. 0 ) GO TO 9986
 500  CONTINUE
C
C
      DO 600 II = 1,NF
      WRITE(6,9002) OMEGA(II)
      NAME = 0
      NMAT = 0
      NFILE = 0
      CALL READTP (AMAT,90,NAME,N,N,B,NFILE,NMAT,ITP2,IRR)
      IF ( IRR .NE. 0 ) GO TO 9985
      NAME = 0
```

```
      NMAT = 0
      NFILE = 0
      CALL WRTETP(AMAT,90,NAME,N,N,B,NFILE,NMAT,ITP3,IRR)
      IF ( IRR .NE. 0 ) GO TO 9986
      DO 550 I = 1,N
      WRITE(6,9001)I,(AMAT(I,J),J=1,N)
  550 CONTINUE
  600 CONTINUE
C
      END FILE ITP3
      RETURN
 9985 WRITE(6,9990) IRR
      CALL EXIT
 9986 WRITE(6,9991) IRR
      CALL EXIT
      RETURN
 9000 FORMAT(1H0,33HDEFLECTION CO-POWER, FREQUENCY = E14.7,11H  (RAD/SEC
     1)  //)
 9001 FORMAT(1H0,I5,1P7E16.6/(E22.6,6E16.6) )
 9002 FORMAT(1H0,35HDEFLECTION QUAD-POWER, FREQUENCY = E14.7,11H  (RAD/S
     1EC)  //)
 9990 FORMAT(28H ERROR IN READTP-ERROR CODE=I5)
 9991 FORMAT(28H ERROR IN WRTETP-ERROR CODE=I5)
      END
```

SUBROUTINE PRINTE

```
$IBFTC PRNTE*  DECK
       SUBROUTINE PRINTE
C
C
C                    THE STRESS-PSD MATRICES ARE PRINTED FOR
C                      OPTION 2 IN THIS SUBROUTINE.
C
C
C
       COMMON/BLK1/FREQ,AMASS,OMEGA
       COMMON/BLK2/M,N,G,ALAM,CMU,K,NF,NPLATE,NBEAMS
       COMMON/BLK3/FLG1,FLG2,FLG3,FLG4
       INTEGER FLG1,FLG2,FLG3,FLG4
       DIMENSION FREQ(25),AMASS(25),OMEGA(100),S(8,8),B(16),SB(6,6)
       NTAPE = 15
       REWIND NTAPE
       NAME = 0
       NMAT = 0
       NFILE = 0
C
C***IF NPLATE EQUALS 0 - SKIP PLATES PRINTOUT
       IF ( NPLATE .EQ. 0 ) GO TO 200
C
C
C
       DO 100 III = 1,NPLATE
       WRITE(6,9000)
C
       DO 60 II = 1,2
       IF ( II .EQ. 2 ) GO TO 10
       WRITE(6,9001)
       GO TO 20
 10    WRITE(6,9002)
C
 20    DO 50 IJ = 1,NF
       WRITE(6,9003) OMEGA(IJ)
       CALL READTP(S,8,NAME,NR,NC,B,NFILE,NMAT,NTAPE,IRR)
       IF ( IRR .NE. 0 ) GO TO 9985
       DO 15 I = 1,8
       WRITE(6,9004)I,(S(I,J),J=1,8)
 15    CONTINUE
 50    CONTINUE
C
 60    CONTINUE
C
C
 100   CONTINUE
C
C
C
C
C***IF NBEAMS EQUALS 0 - SKIP BEAMS PRINTOUT
 200   IF ( NBEAMS .EQ. 0 ) RETURN
C
C
       DO 1000 III = 1,NBEAMS
       WRITE(6,9005)III
```

```
C
      DO 999 II = 1,2
      WRITE(6,9006)  II
C
      DO 998 IJ = 1,2
      IF ( IJ .EQ. 2 ) GO TO 70
      WRITE(6,9007)
      GO TO 80
 70   WRITE(6,9008)
 80   DO 995 I1 = 1,NF
      WRITE(6,9009) OMEGA(I1)
      CALL READTP(SB,6,NAME,NR,NC,B,NFILE,NMAT,NTAPE,IRR)
      IF ( IRR .NE. 0 ) GO TO 9985
C
      DO 90 I = 1,6
      WRITE(6,9010)I,(SB(I,J),J=1,6)
 90   CONTINUE
C
 995  CONTINUE
C
 998  CONTINUE
C
C
 999  CONTINUE
C
C
 1000 CONTINUE
C
C
C
C
      REWIND NTAPE
      RETURN
 9985 WRITE(6,9990) IRR
      CALL EXIT
      RETURN
 9000 FORMAT(1H0,6HPLATE I3)
 9001 FORMAT(10X,9HREAL PART///)
 9002 FORMAT(10X,9HIMAG PART///)
 9003 FORMAT(20X,10HFREQUENCY=E14.7)
 9004 FORMAT(1H0,14X,I5,8E14.5/(20X,8E14.5) )
 9005 FORMAT(1H0,6H BEAM I3///)
 9006 FORMAT(10X,4HEND I3//)
 9007 FORMAT(1H0,17X,10HREAL PART I5//)
 9008 FORMAT(1H0,17X,10HIMAG PART I5//)
 9009 FORMAT(1H0,30X,10HFREQUENCY=E14.7///)
 9010 FORMAT(1H0,30X,I5,6E14.5/(36X,6E14.5) )
 9990 FORMAT(28H ERROR IN READTP-ERROR CODE=I5)
      END
```

SUBROUTINE DSECM3

```
$IBFTC DSECM*  DECK
      SUBROUTINE DSECM3
C                   THE DEFLECTION SECOND SPECTRAL MOMENTS ARE
C                     FORMED FOR OPTION 3.
C                     THE DEFLECTION CO-VARIANCE MATRICES ARE MULTIPLIED
C                     BY FREQ**2 AND SUMMED OVER M NORMAL MODES.
      COMMON/BLK1/FREQ
      COMMON/BLK2/M,N,G,ALAM,CMU,K,NF,NPLATE,NBEAMS
      DIMENSION AMAT(90,90 ),B(16),SUM(90,90),FREQ(25),IPARAM(2)
      NTAPE = 8
      REWIND NTAPE
      DO 10 I = 1,N
      DO 10 J = 1,N
 10   SUM(I,J) = 0.
      NAME = 0
      NMAT = 0
      NFILE = 0
      DO 100 II = 1,M
      CALL READTP(AMAT,90,NAME,N,N,B,NFILE,NMAT,NTAPE,IRR)
      IF ( IRR .NE. 0 ) GO TO 9985
      DO 50 I = 1,N
      DO 50 J = 1,N
 50   SUM(I,J) = SUM(I,J) + AMAT(I,J)*FREQ(II)**2
 100  CONTINUE
      NTAPE = 3
      REWIND NTAPE
      NO = 1
      NAME = 0
      NMAT = 0
      NFILE = 0
C***FORM PARAMETER MATRIX IPARAM
      IPARAM(1) = NPLATE + 2*NBEAMS
      IPARAM(2) = NO
      CALL WRTETP(IPARAM,1,NAME,1,2,B,NFILE,NMAT,NTAPE,IRR)
      IF ( IRR .NE. 0 ) GO TO 9986
      NAME = 0
      NMAT = 0
      NFILE = 0
      CALL WRTETP(SUM,90,NAME,N,N,B,NFILE,NMAT,NTAPE,IRR)
      IF ( IRR .NE. 0 ) GO TO 9986
      WRITE(6,9000)
      DO 200 I = 1,N
      WRITE(6,9001)I,(SUM(I,J),J=1,N)
 200  CONTINUE
      END FILE NTAPE
      RETURN
 9985 WRITE(6,9990) IRR
      CALL EXIT
 9986 WRITE(6,9991) IRR
      CALL EXIT
      RETURN
 9000 FORMAT(1H1,30X,53HDEFLECTION SECOND SPECTRAL MOMENT MATRIX (REAL P
     1ART)  ///)
 9001 FORMAT(1H0,I5,1P7E16.6/(E22.6,6E16.6) )
 9990 FORMAT(28H ERROR IN READTP-ERROR CODE=I5)
 9991 FORMAT(28H ERROR IN WRTETP-ERROR CODE=I5)
      END
```

```
8150B                                                   0B      LOAD + EXECUTE
          8350B                                         0B      PARTITION CORE
          -80000008                                     0B      REW 8(SCRATCH)
          -11000000B                                    0B      REW 11(DIAG SCALARS)
          -17000000B                                    0B      REW 17(CFW-NXN)
110000008           8000B                               0B      READ PARAMETER M
110000008           8004B                               0B      READ DIAG SCALARS
110000008           8032B                               0B      READ MODES(PHI)
170000008           1B                                  0B      READ CFW
8032B     1B        2B                                  8B      PHI(TRANSP)*CFW
2B        8032B     1B                                  6B      PHI(TRANSP)*CFW*PHI
8007B               8136B                               27B     READY NEXT  SCALAR
-1011B                              1B                  27B
8136B     1B        8138B                               5B      SCALAR MULT
8141B     8032B     2B                                  5B      SCALAR MULT
2B        8032B     3B                                  7B      POST MULT BY PHI(TR)
3B                  80000008                            0B      STORE ON TAPE 8
-10B                                8003B               0B      CYCLE M TIMES
          80000008                                      0B      WRITE EOF ON 8
                                                                                      +

8010B                                                   0B      LOAD + EXECUTE
          8200B                                         0B      PARTITION CORE
          -30000008                                     0B      REW 3 (JD)
          -100000008                                    0B      REW 10 (STRESSES)
          -15000000B                                    0B      REW 15
30000008            8000B                               0B      READ PARAM(CYCLE CNT)
30000008            1B                                  0B      READ JD
          10001000B                                     0B      SKIP 1 EOF ON 10
100000008           2B                                  0B      READ STRESSES(S)
2B        1B        3B                                  6B      (S)*(JD)
3B        2B        4B                                  7B      (S)*(JD)*(S)TRANSP
4B                  150000008                           0B      STORE ON TAPE 15
-4B                                 8003B               0B      CYCLE
          150000008                                     0B      WRITE EOF ON 15
                                                                                      +

8100B                                                   0B      LOAD + EXECUTE
          8200B                                         0B      PARTITION CORE
          -30000008                                     0B      REW 3 (JD)
          -100000008                                    0B      REW 10 (STRESSES)
          -15000000B                                    0B      REW 15
30000008            8000B                               0B      READ PARAM(CYCLE CNT)
30000008            1B                                  0B      READ JD
          10001000B                                     0B      SKIP 1 EOF ON 10
100000008           2B                                  0B      READ STRESSES(S)
2B        1B        3B                                  6B      (S)*(JD)
3B        2B        4B                                  7B      (S)*(JD)*(S)TRANSP
4B                  150000008                           0B      STORE ON TAPE 15
-4B                                 8003B               0B      CYCLE
          150000008                                     0B      WRITE EOF ON 15
                                                                                      +


8150B                                                   0B      LOAD + EXECUTE
          8350B                                         0B      PARTITION CORE
          -80000008                                     0B      REW 8 (SCRATCH)
          -20000008                                     0B      REW 2 (SCALARS)
          -11000000B                                    0B      REW 11(MODES)
          -17000000B                                    0B      REW 17(CFW)
```

```
20000008              8000B                    0B        READ M,NF
20000008              8006B                    0B        READ SCALARS
          -11000000B                           0B        REW 11(MODES)
          17001000B                            0B        SKIP EOF ON 17
11000000B             8034B                    0B        READ MODES(PHI)
17000000B             1B                       0B        READ CFW
8034B     1B          2B                       8B        PHI(TRANSP)*CFW
2B        8034B       1B                       6B        PHI(TRANSP)*CFW*PHI
8009B                 8137B                    27B       READY NEXT SCALAR
-1011B                              1B         27B
8137B     1B          8140B                    5B        SCALAR MULT
8143B     8034B       2B                       5B        SCALAR MULT
8034B                 3B                       9B        PHI(TRANSP)
2B        3B          4B                       6B        MULT
          -17001000B                           0B        BACKSPACE 1 EOF
4B                    80000008                 0B        STORE ON TAPE 8
-12B                                8003B      0B        CYCLE M TIMES
-1091B                              8003B      33B       RESTORE SCALAR CELLS
-17B                                8005B      0B        CYCLE NFREQ TIMES
                                                                              +

8010B                                          0B        LOAD + EXECUTE
          8200B                                0B        PARTITION CORE
          -30000000B                           0B        REW 3 (JD)
          -15000000B                           0B        REW 15
30000008              8000B                    0B        READ PARAM(CYCLE CNT)
          -10000000B                           0B        REW 10 (STRESSES)
30000008              1B                       0B        READ JD
          10001000B                            0B        SKIP 1 EOF ON 10
10000000B             2B                       0B        READ STRESSES(S)
2B        1B          3B                       6B        (S)*(JD)
3B        2B          4B                       7B        (S)*(JD)*(S)TRANSP
4B                    15000000B                0B        STORE ON TAPE 15
-4B                                 8003B      0B        CYCLE
-8B                                 8004B      0B        CYCLE NFREQ
          15000000B                            0B        WRITE EOF ON 15
                                                                              +


8150B                                          0B        LOAD + EXECUTE
          8350B                                0B        PARTITION CORE
          -8000000B                            0B        REW 8(SCRATCH)
          -11000000B                           0B        REW 11(DIAG SCALARS)
          -17000000B                           0B        REW 17(CFW-NXN)
11000000B             8000B                    0B        READ PARAMETER M
11000000B             8004B                    0B        READ DIAG SCALARS
11000000B             8032B                    0B        READ MODES(PHI)
17000000B             1B                       0B        READ CFW
8032B     1B          2B                       8B        PHI(TRANSP)*CFW
2B        8032B       1B                       6B        PHI(TRANSP)*CFW*PHI
8007B                 8136B                    27B       READY NEXT  SCALAR
-1011B                              1B         27B
8136B     1B          8138B                    5B        SCALAR MULT
8141B     8032B       2B                       5B        SCALAR MULT
2B        8032B       3B                       7B        POST MULT BY PHI(TR)
3B                    80000008                 0B        STORE ON TAPE 8
-10B                                8003B      0B        CYCLE M TIMES
          80000008                             0B        WRITE EOF ON 8
                                                                              +
```

```
83508                                                   0B      LOAD + EXECUTE
          87508                                         0B      PARTITION CORE
          -80000008                                     0B      REW 8(SCRATCH)
          -15000000B                                    0B      REW 15(EXCITATIONS)
          -16000000B                                    0B      REW 16(MODE SHAPES)
160000008           80008                             0B      READ M-1,K,N
160000008           80078                             0B      READ K TERMS
80048               80068                             27B     COPY K
160000008           80418                             0B      READ PHI(I)
82478                               1038              29B     CLEAR 103 CELLS
82488                               18                27B     SET ROW DIM
82498                               80058             27B     SET COL DIM
80068                               80108             27B     READY NEXT K
-10148                              18                27B
160000008           81448                             0B      READ PHI(J)
150000008           18                                0B      READ CQ(I,J) MATRIX
80418     18        28                                8B      PHI(I)TRANSP*CQ(I,J)
28        8144B     38                                6B      MULT BY PHI(J)
8144B               48                                9B      PHI(J)TRANSP
38        48        8247B                             30B     MULT + ADD
-68                                 80068             0B      CYCLE J=I+1,I+K
80418     8247B     18                                6B      MULT SUM J  BY PHI(I)
18                  80000008                          0B      STORE SUM OF J ON 8
-158                                80038             0B      CYCLE I=1,M-1
          -16000000B                                    0B      REW 16
          160000028                                     0B      SKIP 2 MATRICES ON 16
-1144B                              80038             33B     REDUCE K LOC
-208                                28                0B      CYCLE BACK FOR IMAG
          80000008                                      0B      WRITE EOF ON 8
                                                                                        +

80108                                                   0B      LOAD + EXECUTE
          82008                                         0B      PARTITION CORE
          -30000008                                     0B      REW 3 (JD)
          -15000000B                                    0B      REW 15
30000008            80008                             0B      READ PARAM(CYCLE CNT)
          -10000000B                                    0B      REW 10 (STRESSES)
30000008            18                                0B      READ JD
          100010008                                     0B      SKIP 1 EOF ON 10
100000008           28                                0B      READ STRESSES(S)
28        18        38                                6B      (S)*(JD)
38        28        48                                7B      (S)*(JD)*(S)TRANSP
48                  150000008                         0B      STORE ON TAPE 15
-48                                 80038             0B      CYCLE
-88                                 80048             0B      CYCLE
          150000008                                     0B      WRITE EOF ON 15
                                                                                        +

81508                                                   0B      LOAD + EXECUTE
          83508                                         0B      PARTITION CORE
          -80000008                                     0B      REW 8(SCRATCH)
          -11000000B                                    0B      REW 11(DIAG SCALARS)
          -17000000B                                    0B      REW 17(CFW-NXN)
110000008           80008                             0B      READ PARAMETER M
110000008           80048                             0B      READ DIAG SCALARS
110000008           80328                             0B      READ MODES(PHI)
170000008           18                                0B      READ CFW
80328     18        28                                8B      PHI(TRANSP)*CFW
28        80328     18                                6B      PHI(TRANSP)*CFW*PHI
80078               81368                             27B     READY NEXT  SCALAR
```

```
-10118                                  1B          278
8136B       1B          8138B                       58          SCALAR MULT
8141B       8032B       2B                          58          SCALAR MULT
2B          8032B       3B                          78          POST MULT BY PHI(TR)
3B                      80000008                    08          STORE ON TAPE 8
-10B                                    8003B       08          CYCLE M TIMES
            80000008                                08          WRITE EOF ON 8
                                                                                    +


8350B                                               08          LOAD + EXECUTE
            8750B                                   08          PARTITION CORE
            -8000000B                               08          REW 8(SCRATCH)
            -150000008                              08          REW 15(EXCITATIONS)
            -160000008                              08          REW 16(MODE SHAPES)
160000008               8000B                       08          READ M-1,K,N
160000008               8007B                       08          READ K TERMS
8004B                   8006B                       278         COPY K
160000008               8041B                       08          READ PHI(I)
8247B                               103B            298         CLEAR 103 CELLS
8248B                               1B              278         SET ROW DIM
8249B                               8005B           278         SET COL DIM
8006B                               8010B           278         READY NEXT K
-1014B                              1B              278
160000008               8144B                       08          READ PHI(J)
150000008               1B                          08          READ CQ(I,J) MATRIX
8041B       1B          2B                          88          PHI(I)TRANSP*CQ(I,J)
2B          8144B       3B                          68          MULT BY PHI(J)
8144B                   4B                          98          PHI(J)TRANSP
3B          4B          8247B                       308         MULT + ADD
-6B                                     8006B       08          CYCLE J=I+1,I+K
8041B       8247B       1B                          68          MULT SUM J  BY PHI(I)
1B                      80000008                    08          STORE SUM OF J ON 8
-15B                                    8003B       08          CYCLE I=1,M-1
            -160000008                              08          REW 16
            160000028                               08          SKIP 2 MATRICES ON 16
-1144B                                  8003B       338         REDUCE K LOC
-20B                                    2B          08          CYCLE BACK FOR IMAG
            80000008                                08          WRITE EOF ON 8
                                                                                    +


8010B                                               08          LOAD + EXECUTE
            8200B                                   08          PARTITION CORE
            -3000000B                               08          REW 3 (JD)
            -150000008                              08          REW 15
30000008                8000B                       08          READ PARAM(CYCLE CNT)
            -100000008                              08          REW 10 (STRESSES)
3000000B                1B                          08          READ JD
            100010008                               08          SKIP 1 EOF ON 10
100000008               2B                          08          READ STRESSES(S)
2B          1B          3B                          68          (S)*(JD)
3B          2B          4B                          78          (S)*(JD)*(S)TRANSP
4B                      150000008                   08          STORE ON TAPE 15
-4B                                     8003B       08          CYCLE
-8B                                     8004B       08          CYCLE NF TIMES
            150000008                               08          WRITE EOF ON 15
                                                                                    +


8150B                                               08          LOAD + EXECUTE
            8350B                                   08          PARTITION CORE
```

```
          -80000008                          0B        REW 8 (SCRATCH)
          -20000008                          0B        REW 2 (SCALARS)
          -11000000B                         0B        REW 11(MODES)
          -17000000B                         0B        REW 17(CFW)
20000008            8000B                    0B        READ M,NF
20000008            8006B                    0B        READ SCALARS
          -11000000B                         0B        REW 11(MODES)
          17001000B                          0B        SKIP EOF ON 17
11000000B           8034B                    0B        READ MODES(PHI)
17000000B           1B                       0B        READ CFW
8034B     1B        2B                       8B        PHI(TRANSP)*CFW
2B        8034B     1B                       6B        PHI(TRANSP)*CFW*PHI
8009B               8137B                    27B       READY NEXT SCALAR
-10011B                           1B         27B
8137B     1B        8140B                    5B        SCALAR MULT
8143B     8034B     2B                       5B        SCALAR MULT
8034B               3B                       9B        PHI(TRANSP)
2B        3B        4B                       6B        MULT
          -17001000B                         0B        BACKSPACE 1 EOF
4B                  80000008                 0B        STORE ON TAPE 8
-12B                              8003B      0B        CYCLE M TIMES
-1091B                            8003B      33B       RESTORE SCALAR CELLS
-17B                              8005B      0B        CYCLE NFREQ TIMES
                                                                          +

8350B                                        0B        LOAD AND EXECUTE
          8750B                              0B        PARTITION CORE
          -80000008                          0B        REW 8(SCRATCH)
          -150000008                         0B        REW 15(EXCITATIONS)
          -160000008                         0B        REW 16(MODE SHAPES)
160000008           8000B                    0B        READ M-1,K,N,NFREQ
160000008           8008B                    0B        READ K OFF-DIAG TERMS
8004B               8007B                    27B       COPY K
160000008           8035B                    0B        READ PHI(I)
8241B                             103B       29B       CLEAR 103 CELLS
8242B                             1B         27B       ADD ROW DIM
8243B                             8005B      27B       ADD COL DIM
8007B                             8011B      27B       READY NEXT K
-1014B                            1B         27B
160000008           8138B                    0B        READ PHI(J)
150000008           1B                       0B        READ C(I,J) MATRIX
8035B     1B        2B                       8B        PHI(I)TRANSP*CQ(I,J)
2B        8138B     3B                       6B        MULT BY PHI(J)
8138B               4B                       9B        PHI(J)TRANSP
3B        4B        8241B                    30B       MULT + ADD
-6B                               8007B      0B        CYCLE J=I+1,I+K
8035B     8241B     1B                       6B        MULT BY PHI(I)
1B                  80000008                 0B        STORE SUM OF J ON 8
-15B                              8003B      0B        CYCLE I = 1,M-1
          -160000008                         0B        REW 16
          160000028                          0B        SKIP 2 MATRICES ON 16
-1144B                            8003B      33B       RESET K LOC
-20B                              8006B      0B        CYCLE NF TIMES
-21B                              2B         0B        CYCLE REAL,IMAG
          80000008                           0B        WRITE EOF ON 8
                                                                          +

8010B                                        0B        LOAD + EXECUTE
          8200B                              0B        PARTITION CORE
          -80000008                          0B        REW 8 (PARAM-CPSD)
```

```
          -100000008                                  08        REW 10 (STRESSES)
          -150000008                                  08        REW 15
80000008                  8008                        08        READ PARAM
          -80000008                                   08        REW 8
          100010008                                   08        SKIP 1 EOF ON 10
100000008                 18                          08        READ STRESS
          80000018                                    08        SKIP 1 MATRIX
80000008                  28                          08        READ CPSD
18        28              38                          68        S*CPSD
38        18              48                          78        S*CPSD*S TRANP
48                        150000008                   08        STORE ON TAPE 15
-48                                     8004B         08        CYCLE REAL,IMAG
          -80000008                                   08        REW 8
-88                                     8003B         08        CYCLE STRESSES
          150000008                                   08        WRITE EOF ON 15
                                                                                    +


80108                                                 08        LOAD + EXECUTE
          -120000008                                  08        REW 12 (OUTPUT)
          -140000008                                  08        REW 14 (PARAM)
          -150000008                                  08        REW 15 (A-REAL )
          -160000008                                  08        REW 16 ( B-IMAG )
          120000008                                   08        WRITE EOF ON 12
          8200B                                       08        PARTITION CORE
140000008                 8008                        08        READ PARAM
150000008                 18                          08        READ A
18                        18                          188       A(INVERSE)
160000008                 28                          08        READ B
18        28              38                          68        A(INV)*B
28        38              18                          68        B*A(INV)*B
150000008                 28                          08        READ A
28        18              28                          18        A+B*A(INV)*B
38                        18                          08        COPY A(INV)*B
28                        28                          188       (A+B*A(INV)*B) INV
18        28              38                          68        MULT - IMAG
28                        120000008                   08        SAVE J - REAL
38                        120000008                   08        SAVE L - IMAG
          120000008                                   08        WRITE EOF ON 12
-138                                    8005B         08        CYCLE NF TIMES
                                                                                    +


80108                                                 08        LOAD + EXECUTE
          8500B                                       08        PARTITION CORE
          -30000008                                   08        REW 3 -SCRATCH
          -120000008                                  08        REW 12 (L AND J)
          -140000008                                  08        REW 14 ( PARAM )
          -150000008                                  08        REW 15 (OUT-REAL)
          -160000008                                  08        REW 16 ( OUT-IMAG)
          -170000008                                  08        REW 17-QF AND CF MAT.
140000008                 8008                        08        READ PARAM
          120010008                                   08        SKIP 1 EOF
          170010008                                   08        SKIP PAST 1 EOF
120000008                 18                          08        READ L MATRIX
170000008                 28                          08        READ QF MATRIX
28        18              38                          68        MULT QF*L
38                        18                          08        COPY
170000008                 28                          08        READ CF MATRIX
120000008                 38                          08        READ J MATRIX
28        38              18                          308       CF*J + QF*L = X
```

```
3B          1B          2B                      6B          J*X
2B                      3000000B                0B          STORE ON TAPE 3
            -120010008                          0B          BACKSPACE 1 EOF
12000000B               2B                      0B          READ L MATRIX
2B          1B          3B                      6B          MULT L*X MATRIX
3B                      3000000B                0B          STORE ON TAPE 3
            30000008                            0B          WRITE EOF ON 3
12000000B               1B                      0B          READ J MATRIX
            -170010008                          0B          BACKSPACE PAST 1 EOF
17000000B               2B                      0B          READ QF MATRIX
2B          1B          3B                      6B          MULT QF*J
3B                      1B                      0B          COPY
1B          1B          2B                      1B          ADD QF*J + QF*J=Y
1B          2B          1B                      2B          SUBT QF*J - Y = -QF*J
17000000B               2B                      0B          READ CF MATRIX
            -120010008                          0B          BACKSPACE 1 EOF
12000000B               3B                      0B          READ L MATRIX
2B          3B          1B                      30B         CF*L - QF*J = Z
3B          1B          2B                      6B          MULT L*Z
            -30000008                           0B          REW 3
30000008                3B                      0B          READ REAL
2B          3B          2B                      1B          TOTAL REAL
2B                      150000008               0B          STORE REAL
12000000B               2B                      0B          READ J MATRIX
2B          1B          3B                      6B          MULT J*Z
30000008                1B                      0B          READ IMAG
1B          3B          1B                      2B          IMAG
1B                      160000008               0B          STORE IMAG
-37B                                80058       0B          CYCLE NF TIMES
            150000008                           0B          WRITE EOF ON 15
            160000008                           0B          WRITE EOF ON 16
                                                                          +

80108                                           0B          LOAD + EXECUTE
            8200B                               0B          PARTITION CORE
            -30000008                           0B          REW 3
            -100000008                          0B          REW 10 (STRESSES)
            -140000008                          0B          REW 14 (PARAM)
            -150000008                          0B          REW 15 (REAL-CPSD)
            -160000008                          0B          REW 16 (IMAG-CPSD)
140000008               8000B                   0B          READ PARAM
            100010008                           0B          SKIP 1 EOF ON 10
100000008               1B                      0B          READ STRESS
150000008               2B                      0B          READ REAL CPSD
1B          2B          3B                      6B          S*CPSD-REAL
3B          1B          4B                      7B          S*CPSD*S TRANP
4B                      3000000B                0B          STORE ON TAPE 3
160000008               2B                      0B          READ IMAG CPSD
1B          2B          3B                      6B          S*CPSD IMAG
3B          1B          4B                      7B          S*CPSD*S TRANP
4B                      3000000B                0B          STORE ON TAPE 3
-8B                                 80058       0B          CYCLE NF TIMES
            -150000008                          0B          REW 15 REAL CPSD
            -160000008                          0B          REW 16 IMAG CPSD
-12B                                80078       0B          CYCLE STRESSES
            30000008                            0B          WRITE EOF ON TAPE 3
                                                                          +

8200B                                           0B          LOAD + EXECUTE
```

```
          8500B                                  0B        PARTITION CORE
          -4000000B                              0B        REW 4 (CONSTANTS)
          -12000000B                             0B        REW 12 (OUTPUT)
          -14000000B                             0B        REW 14 (PARAM)
          -15000000B                             0B        REW 15 (REAL-CPSD)
          -16000000B                             0B        REW 16 (IMAG-CPSD)
140000008             8000B                      0B        READ PARAM
4000008               8010B                      0B        READ CONSTANTS
1014B                           8004B            27B
8500B                                            29B       NULL MATRIX
8501B                           8003B            27B       ADD ROW DIM
8502B                           8003B            27B       ADD COL DIM
8013B                 8199B                      27B
-1011B                          1B               27B       READY NEXT SCL
150000008             2B                         0B        READ A MATRIX
8199B       2B        2B                         5B        A*SCL
2B          1B        1B                         1B        SUM
-5B                             8005B            0B        CYCLE FREQ
1B                    120000008                  0B        STORE JD
-1071B                          8005B            33B       SET LOC AGAIN
-1061B      -1B                 10000008         27B       SET TO TAPE 16
-13B                            2B               0B        CYCLE BACK FOR IM
          120000008                                0B        WRITE EOF ON 12
                                                                                +

8010B                                            0B        LOAD + EXECUTE
          8200B                                  0B        PARTITION CORE
          -3000000B                              0B        REW 3
          -12000000B                             0B        REW 12(JD)
          -14000000B                             0B        REW 14(PARAM)
140000008             8000B                      0B        READ PARAM
          -100000008                             0B        REW 10 (STRESSES)
120000008             1B                         0B        READ JD
          10001000B                              0B        SKIP 1 EOF ON 10
100000008             2B                         0B        READ STRESS
2B          1B        3B                         6B        S*JD
3B          2B        4B                         7B        S*JD*S TRANP
4B                    3000000B                   0B        STORE ON TAPE 3
-4B                             8007B            0B        CYCLE
-8B                             2B               0B        CYCLE REAL,IMAG
          30000008                                 0B        WRITE EOF ON TAPE 3
                                                                                +

8010B                                            0B        LOAD + EXECUTE
          8200B                                  0B        PARTITION CORE
          -3000000B                              0B        REW 3
          -12000000B                             0B        REW 12 (JD)
          -14000000B                             0B        REW 14 (PARAM)
140000008             8000B                      0B        TEAD PARAM
          -100000008                             0B        REW 10 (STRESSES)
12000000B             1B                         0B        READ JD
          10001000B                              0B        SKIP 1 EOF ON 10
100000008             2B                         0B        READ STRESS
2B          1B        3B                         6B        S*JD
3B          2B        4B                         7B        S*JD*S TRANP
4B                    3000000B                   0B        STORE ON TAPE 3
-4B                             8007B            0B        CYCLE
-8B                             2B               0B        CYCLE REAL,IMAG
          30000008                                 0B        WRITE EOF ON TAPE 3
                                                                                +

$EOF
```

REFERENCES

1. L. D. Jacobs and D. R. Lagerquist, Finite-Element Analysis of Complex Panel Response to Random Loads, AFFDL-TR-68-44, Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, Ohio (in press)

2. L. D. Jacobs and D. R. Lagerquist, A Finite Element Analysis of Simple Panel Response to Turbulent Boundary Layers, AFFDL-TR-67-81, Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, Ohio, December 1967

3. D. R. Lagerquist and L. D. Jacobs, Random-Vibration Analysis System for Complex Structures, AFFDL-TR-68-43, Part I, "Engineering User's Guide," Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, Ohio (in press)

4. J. H. Wilkinson, The Algebraic Eigenvalue Problem, Oxford University Press, 1965

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*
The Boeing Company, Commercial Airplane Division
P.O. Box 707, Renton, Washington 98005

2a. REPORT SECURITY CLASSIFICATION
Unclassified

2b. GROUP

3. REPORT TITLE
Random - Vibration Analysis System for Complex Structures
Part II--Computer Program Description

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
Final Report July 1966—December 1967

5. AUTHOR(S) *(First name, middle initial, last name)*
K. Tsurusaki and F.S. Wallace

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
January 1969	482	4

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
AF 33 (615)-5155 b. PROJECT NO. 1471 c. TASK NO. 147101 d.	D6-23145, Part II 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)* AFFDL-TR-68-43, Part II

10. DISTRIBUTION STATEMENT
Distribution limited to U. S. Government agencies only; test and evaluation statement applied December 1971. Other requests for this document must be referred to AF Flight Dynamics Laboratory (FY), Wright-Patterson AFB, Ohio 45433.

Approved for public release; distribution unlimited ~~ntrols, and each transmittal to foreign e only with prior approval of the Air Force~~ ~~h AFB, Ohio 45433.~~

12. SPONSORING MILITARY ACTIVITY
Air Force Flight Dynamics Laboratory (FDD)
Wright-Patterson AFB, Ohio 45433

13. ABSTRACT

A programming description is presented for a computer program developed to aid in the design of sonic-fatigue-resistant aircraft structure. The computer program is written in FORTRAN IV and MAP for the IBM 7094 Mod. II. The program employs matrix structural analysis methods to calculate statistical measurements of response (deflection and stress) for complex structure subjected to pressure loads random in both time and space. The program is organized into two phases, each performed separately. The phases are further organized in modular form for ease of maintenance and/or modification.

DD FORM 1 NOV 65 1473

Unclassified

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Computer Program Description						
Digital Computer Program						
Matrix Structural Analysis Methods						
Finite-Element Structural Methods						
Random-Vibration Analysis System						
Acoustic Vibration Analysis System						
Structural Vibration Analysis System						
Stiffness Matrix						
Eigen values						
Random Fluctuating Pressure Loads						