SSC-228

TANKER TRANSVERSE STRENGTH ANALYSIS PROGRAMMER'S MANUAL

SHIP STRUCTURE COMMITTEE

1972

SHIP STRUCTURE COMMITTEE

AN INTERAGENCY ADVISORY
COMMITTEE DEDICATED TO IMPROVING
THE STRUCTURE OF SHIPS

MEMBER AGENCIES:

UNITED STATES COAST GUARD
NAVAL SHIP SYSTEMS COMMAND
MILITARY SEALIFT COMMAND
MARITIME ADMINISTRATION
AMERICAN BUREAU OF SHIPPING

ADDRESS CORRESPONDENCE TO:

SECRETARY
SHIP STRUCTURE COMMITTEE
U.S. COAST GUARD HEADQUARTERS
WASHINGTON, D.C. 20591

SR-196
1972

Dear Sir:

One of the most important goals of the Ship Structure Committee is the improvement of methods for design and analysis of ship hull structures. This report is the last in a sequence of four Ship Structure Committee reports on a project directed towards development of an accurate, but less expensive, computer aided structural analysis method.

This report contains the Programmer's Manual for the transverse strength analysis portion of the program. Other reports of this project are:

SSC-225 - Structural Analysis of Longitudinally Framed Ships

SSC-226 - Tanker Longitudinal Strength Analysis--User's Manual and Computer Program

SSC-227 - Tanker Transverse Strength Analysis--User's Manual

Comments on this report would be welcomed.

Sincerely,

W. F. REA, III
Rear Admiral, U. S. Coast Guard
Chairman, Ship Structure Committee

SSC-228

Final Report

on

Project SR-196, "Computer Design of Longitudinally Framed Ships"

to the

Ship Structure Committee

TANKER TRANSVERSE STRENGTH ANALYSIS

PROGRAMMER'S MANUAL

by

R. Nielson, P. Y. Chang, and L. C. Deschamps
COM/CODE Corporation

under

Department of the Navy
Naval Ship Engineering Center
Contract No. N00024-70-C-5219

U. S. Coast Guard Headquarters
Washington, D. C.
1972

ABSTRACT

This report, the last in a sequence of four Ship Structure Committee Reports on a method for performing structural analysis of a tanker hull, contains the Programmer's Manual for the transverse strength analysis portion of the program.

CONTENTS

SHIP STRUCTURE COMMITTEE

The SHIP STRUCTURE COMMITTEE is constituted to prosecute a research program to improve the hull structures of ships by an extension of knowledge pertaining to design, materials and methods of fabrication.

RADM W. F. Rea, III, USCG, Chairman
Chief, Office of Merchant Marine Safety
U. S. Coast Guard Headquarters

Capt. J. E. Rasmussen, USN
Head, Ship Systems Engineering
and Design Department
Naval Ship Engineering Center
Naval Ship Systems Command

Mr. K. Morland, Vice President
American Bureau of Shipping

Mr. E. S. Dillon
Chief
Office of Ship Construction
Maritime Administration

Capt. L. L. Jackson, USN
Maintenance and Repair Officer
Military Sealift Command

SHIP STRUCTURE SUBCOMMITTEE

The SHIP STRUCTURE SUBCOMMITTEE acts for the Ship Structure Committee on technical matters by providing technical coordination for the determination of goals and objectives of the program, and by evaluating and interpreting the results in terms of ship structural design, construction and operation.

NAVAL SHIP ENGINEERING CENTER

Mr. P. M. Palermo - Chairman
Mr. J. B. O'Brien - Contract Administrator
Mr. G. Sorkin - Member
Mr. H. S. Sayre - Alternate
Mr. I. Fioriti - Alternate

U. S. COAST GUARD

LCDR C. S. Loosmore, USCG - Secretary
CAPT C. R. Thompson, USCG - Member
CDR J. W. Kime, USCG - Alternate
CDR J. L. Coburn, USCG - Alternate

MARITIME ADMINISTRATION

Mr. F. Dashnaw - Member
Mr. A. Maillar - Member
Mr. R. Falls - Alternate
Mr. R. F. Coombs - Alternate

MILITARY SEALIFT COMMAND

Mr. R. R. Askren - Member
LTJG E. T. Powers, USNR - Member

AMERICAN BUREAU OF SHIPPING

Mr. S. G. Stiansen - Member
Mr. F. J. Crum - Member

OFFICE OF NAVAL RESEARCH

Mr. J. M. Crowley - Member
Dr. W. G. Rauch - Alternate

NAVAL SHIP RESEARCH & DEVELOPMENT CENTER

Mr. A. B. Stavovy - Alternate

NATIONAL ACADEMY OF SCIENCES - Ship Research Committee

Mr. R. W. Rumke, Liaison
Prof. R. A. Yagle, Liaison

SOCIETY OF NAVAL ARCHITECTS & MARINE ENGINEERS

Mr. T. M. Buermann, Liaison

BRITISH NAVY STAFF

Dr. V. Flint, Liaison
CDR P. H. H. Ablett, RCNC, Liaison

WELDING RESEARCH COUNCIL

Mr. K. H. Koopman, Liaison
Mr. C. Larson, Liaison

INTRODUCTION

The tanker transverse analysis has been written in FORTRAN IV for use on the UNIVAC 1108 and the Control Data Corporation 6600 computers.

This Programmer's Manual contains details concerning the program's core requirements, data files, execution times, and general instruction logic.

Details on the data input may be found in the User's Manual.

CORE REQUIREMENTS AND DATA FILES

On the CDC 6600, the entire program, including system loader, requires 154, 101 octal words of core (55,361 decimal words). If the user desires to run the program on a computer with a smaller core size he may do so by overlaying. The program is suitably structured to be easily overlayed for this purpose.

Table 1 provides core requirements for each of the routines and COMMON areas; all starting addresses are given in Octal units.

Six drum or disk files are used for storing large blocks of data generated by the program at various stages of the execution. Three of these files (200,000 decimal words each) are defined for the storage and manipulation of the finite element stiffness matrices. The three remaining files (100,000 words each) are for the storage of miscellaneous loading and longitudinal data.

Table 1. CORE Requirements

-PROGRAM-	-ADDRESS-	--LABELED---COMMON--	
TANKER	027704	INFLU	000100
		SHIP	005545
		SAFE	006537
		MATRL	025253
		WORK	025260
REORD	061470		
BEGIN	061652	MATRL	025253
		SHIP	005545
		SAFE	006537
TRV	062355	WORK	025260
		SAFE	006537
		MATRL	025253
NOD	067076		
NODET	067137		
SWTCH	067217		
SORT	067234		
LONGI	067312	MATRL	025253
		SHIP	005545
		WORK	025260
		INFLU	000100
		SAFE	006537
LOADS	070374	WORK	025260
		MATRL	025253
		SHIP	005545
		SAFE	006537
		SHLWEB	070145
		INFLU	000100
SETOC	071707		
OCCM	072376		
TMATT	072714		
MMULT	073217		
IDENT	073322		
EQUAL	073400		
CONSI	073451		
MULT	073523		
SHIP1	073617		
NODEIN	075032		
MEMBER	075250		
TRANS	075423		
EIGEN	075501		
MATINS	077231		
TEMPCO	077673		
MULTRD	077743		
READIN	100100		
TRAMPY	100146		
DIRCOS	100244		
MEM1	100307		
MEM2	100545		
MEM5	101626		
SHIP2	101770	INFLU	000100
		SAFE	006537
		SHIP	005545
SHIP3	103162	SAFE	006537
SHIP4	103616	SAFE	006537
SR4A	104371		
SR14	104442		
SR15	104555	SAFE	006537
MEMB1	104635		
MEMB2	105042		
SHELL	105216	SHLWEB	070145
		WORK	025260
		SAFE	006537
MEMB5	105355		
GETBA	105452		
SIO$	105471		
SYSTEM$	107035		
ACGOER$	110020		
BACKSP$	110033	Blank common = 023667	
ENDFIL$	110360		
INPUTB$	110426		
INPUTC$	110710		
KODER$	111026		
KRAKER$	112422		
OUTPTB$	114114		
OUTPTC$	114365		
REWINM$	114461		
ATANE	114531		
SINCOSE	114612		
SQRTE	114667		

ROUTINE CALLING SEQUENCE

Table 2 provides the calling sequence for all program routines and subroutines. This mapping will assist the programmer in devising an overlaying scheme if such is required.

Table 2. Calling Sequence

--ENTRY--	--ADDRESS-	Calling routine
TANKER	050140	
INPUTΞ	027704	
OUTPUTΞ	031726	
TAPE5Ξ	027704	BEGIN
		TRV
		LONGI
		LOADS
		SHELL
TAPE6Ξ	031726	REORD
		BEGIN
		TRV
		LONGI
		LOADS
		SHIP1
		NODEIN
		MEMBER
		MEM2
		SHIP2
		SHIP3
		SHIP4
		MEMB1
		MEMB2
		SHELL
		MEMB5
TAPE1Ξ	033750	
TAPE3Ξ	035772	
TAPE4Ξ	040014	
TAPE19Ξ	042036	
TAPE20Ξ	044060	
TAPE21Ξ	046102	
REORD	061472	TANKER
BEGIN	061654	TANKER
TRV	062357	TANKER
NOD	067100	LONGI
NODET	067141	TRV
		LONGI
		LOADS
SWTCH	067221	TRV
		LONGI
		LOADS
SORT	067236	TRV
		SETOC

LONGI	067314	TANKER
LOADS	070376	TANKER
SETOC	071711	LOADS
OCCM	072400	TMATT
TMATT	072716	LOADS
MMULT	073221	LOADS OCCM TMATT
IDENT	073324	OCCM TMATT
EQUAL	073402	OCCM TMATT
COMSI	073453	LOADS
MULT	073525	TANKER
SHIP1	073621	TANKER
NODEIN	075034	SHIP1
MEMBER	075252	SHIP1
TRANS	075425	TANKER
EIGEN	075503	TANKER
MATINS	077233	TANKER MEM2 SHIP2
TEMPCO	077675	MEM1 MEM2 MEM5
MULTRD	077745	MEM1 MEM2
READIN	100102	MULTRD
TRAMPY	100150	SHIP1
DIRCOS	100245	MEM1 MEM2
MEM1	100310	SHIP1
MEM2	100546	SHIP1
MEM5	101627	SHIP1
SHIP2	101772	TANKER
SHIP3	103163	TANKER
SHIP4	103617	TANKER
SR4A	104373	MEMB1 MEMB2
SR14	104444	MEMB1 MEMB2 MEMB5
SR15	104557	SHIP4
MEMB1	104636	SHIP4
MEMB2	105043	SHIP4
SHELL	105217	TANKER
MEMB5	105356	SHIP4
GETBA	105453	SIO$ BACKSP$ ENDFIL$ INPUTB$ OUTPTB$ REWINM$
CIO1.	106534	BACKSP$ INPUTB$ OUTPTB$ REWINM$
RCL1.	106555	SYSTEM$ BACKSP$ INPUTB$ REWINM$
DAT.	105473	SYSTEM$ INPUTC$ KODER$ KRAKER$ OUTPTB$ OUTPTC$
SIO.CTL	105721	SYSTEM$ BACKSP$ ENDFIL$ OUTPTB$ REWINM$
INITL.	105742	SYSTEM$ BACKSP$ ENDFIL$ INPUTB$ INPUTC$ OUTPTB$ OUTPTC$ REWINM$
SIO.	106006	SYSTEM$ INPUTB$ INPUTC$ OUTPTB$ OUTPTC$
SIO.END	106507	SYSTEM$ BACKSP$ ENDFIL$ REWINM$
OPEN.	106564	SYSTEM$ INPUTB$ OUTPTB$
RDPRU.	106625	BACKSP$
BKSPRU.	106642	
ADVIN.	106647	SYSTEM$ BACKSP$ ENDFIL$ REWINM$
POSFI.	106675	ENDFIL$ OUTPTB$
MVWDS.	107022	INPUTB$ OUTPTB$
QENTRY.	107041	TANKER
END.	107055	TANKER
EXIT$	107077	
STOP.	107105	TANKER BEGIN TRV LONGI LOADS SHIP1 NODEIN MEM2 SHIP2

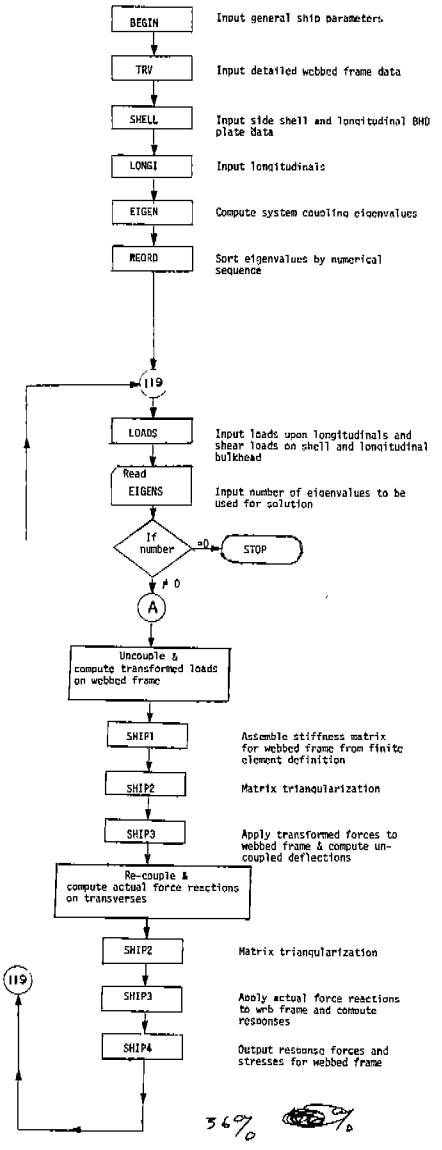
ABNORM.	107116	ACGOER$
		BACKSP$
		ENDFIL$
		INPUTB$
		INPUTC$
		KODER$
		KRAKER$
		OUTPTB$
		OUTPTC$
		REWINM$
SYSTEME	107135	
SYSTEM$	107161	
SYSTEM.	107165	
SYSTEM:	107217	ACGOER$
		BACKSP$
		ENDFIL$
		INPUTB$
		INPUTC$
		KODER$
		KRAKER$
		OUTPTB$
		OUTPTC$
		REWINM$
SYS1:	107405	KRAKER$
SYS2:	107401	KRAKER$
LOT:	107612	KRAKER$
DBGFET.	107632	
ACGOER.	110021	LOADS
		OCCM
		SHIP1
		SHIP2
		SHIP4
BACKSP.	110041	SHIP3
ENDFIL.	110370	TRV
		LOADS
IPUTBI.	110462	TANKER
		LOADS
		SHIPI
		SHIP2
		SHIP3
		SHIP4
INPUTB.	110521	TANKER
		LOADS
		SHIP1
		SHIP2
		SHIP3
		SHIP4
IPUTCI.	110750	TANKER
		BEGIN
		TRV
		LONGI
		LOADS
		SHIP1
		NODEIN
		MEMBER
		SHELL
INPUTC.	110773	TANKER
		BEGIN
		TRV
		LONGI
		LOADS
		SHIP1
		NODEIN
		MEMBER
		MEM2
		SHIP2
		SHIP3
		SHIP4
		MEMB1
		MEMB2
		SHELL
		MEMB5
KODER.	111027	OUTPTC$
KRAKER.	112423	INPUTC$
ERRSET$	113133	
OPUTBI.	114134	LOADS
		SHIP1
		SHIP2
OUTPTB.	114172	LOADS
		SHIP1
		SHIP2
OPUTCI.	114416	TANKER
		REORD
		BEGIN
		TRV
		LONGI
		LOADS
		SHIP1
		NODEIN
		MEMBER
		MEM2
		SHIP2
		SHIP3
		SHIP4
		MEMB1
		MEMB2
		SHELL
		MEMB5

OUTPTC.	114425	TANKER
		RECORD
		BEGIN
		TRV
		LONGI
		LOADS
		SHIP1
		NODEIN
		MEMBER
		MEM2
		SHIP2
		SHIP3
		SHIP4
		MEMB1
		MEMB2
		SHELL
		MEMB5
REWINM.	114470	TANKER
		TRV
		LOADS
		SHIP1
		SHIP2
		SHIP3
		SHIP4
ATAN.	114531	MEMB1
SIN.	114612	TANKER
		SHIP2
COS.	114614	TANKER
		SHIP2
SQRT.	114671	TANKER
		EIGEN
		DIRCOS
		MEM5
		MEMB1

GENERAL PROGRAM LOGIC

Figure 1 is a flow chart of the general logic of the tanker transverse analysis program. The programmer is directed to examine the subroutines in their source listing form for more detailed information provided by comment statements included in the text.

Fig. 1. General Logic Diagram for Transverse Analysis



PROGRAM EXECUTION TIMES

Execution time varies according to the structural and loading definitions. Table 3-4 provides sample computer times for the CDC 6600 and UNIVAC 1108 (Exec II) computers.

Table 3. CDC 6600 Program Execution Times

	Number Transverses	No. Frame Elements	Number Longitudinals	Number Equivalent Elements	Number Eigenvalues Used ***	CP Seconds	System Seconds
Data check only:	28	699	94	22,298	-0-	16.008	25.458
Complete analysis:	29	618	95	20,772	1	129.656	975.956
" "	28	699	95	22,327	1	129.052	1038.452
" "	29	699	95	23,121	1	129.029	1043.879
" "	3	66	11	242	1	14.409	72.059
*Stripped package:	3	36	11	141	1	3.403	20.303
**CDC/EASE:	-quarter model-		11	92	N.A.	10.977	41.071

* This stripped package does not have input routines amenable for convenient use of the transverse analysis capabilities.

** Since the problem analyzed was symmetrical fore-n-aft as well as about the hull centerline, the CDC/EASE Analysis could be simplified to a quarter-hull model. The tanker transverse program, on the other hand, analyzed the whole half-section. Nevertheless, note the significant difference in compute times between the two analyses. The additional computer time required for execution of the complete tanker program (as opposed to the stripped version) is mainly due to the extensive input processing which greatly facilitates the problem definition.

*** If the EIGENS card is omitted, the program will function in data checking only. This procedure is recommended before a complete stress analysis is attempted.

Table 4. UNIVAC 1108 (EXEC II) Program Execution Times

Complete analysis	Using UNIVAC 1108 (EXEC II), the CTU time for complete analysis for approximately 23,000 equivalent elements equals 363.34 seconds (including system time).

FORTRAN Source Listing

```
      PROGRAM TANKER(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1,TAPE3
     1,TAPE4,TAPE19,TAPE20,TAPE21)
CZZ7ZI   FR5  MAIN,MAIN,MAIN
C...PRIMARY TRANSVERSE STRENGTH ROUTINE
      DIMENSION JT(5),LO(100),IND(50,3)
      DIMENSION UNITS(4),ND(6),NONO(25),N1(25),IPO(10),PIO(10)
      DIMENSION R(2100)
      DIMENSION  AP(50,50),BP(50,50),CP(50,50)
      DIMENSION DP(50,50),RL(5,100),PM(5,100)
      DIMENSION DQX(50,100),DQY(50,100),CX(100),CY(100)
      COMMON K1,K2,K3,K4,ID,NORO,NNI,
     1        UNITS ,ND    ,NONO  ,N1    ,IPO   ,PIO ,
     2  NUMFO,R
      COMMON/INFLU/AF(50,50),EIG(50),DY(50),NFILE,NF,LNOD(100),
     1 LROW(100),DX(50),MF
      COMMON/SHIP/NOLO,LNO(100),SFX(100),SFY(100),PHI(100),NOTR,
     1 ZTR(50),ZLEN,P(50),XI,XA,NSEC
       COMMON/SAFE/BP,CP,DP
      COMMON /MATRL/E,G,GNU,ALPHA,CONVF
      COMMON /WORK/DUMY(1300)
         EQUIVALENCE (DQX(1,1),R(1)),(DQY(1,1),BP(1,1))
      DATA  CFIG /6HEIGENS  /
      NFILE=21
      NCARD=19
      NSCR=20
      CALL BEGIN(IB,JT)
      CALL TRV(NCARD)
              CALL SHELL
      CALL LONGI(NFILE)
 104  FORMAT(1X,6E12.4)
  101  FORMAT (10F7.2)
  777  PF=0.
      WRITE(6,703)
 703  FORMAT(1H1,5X,48HINFLUENCE COEEFICIENTS FOR STANDARD LONGITUDINAL)
      DO   110  I=1,NOTR
        PF=PF+ AF(I,I)/FLOAT(NOTR)
      WRITE(6,104)(AF(I,J),J=1,NOTR)
  110   P(I)=SQRT(P(I))
      DO   112  I=1,NOTR
      DO   112  J=1,NOTR
        IF (I.GT.J)  GO TO 111
       AP(I,J)=AF(I,J)*P(I)*P(J)/PF
            GO TO 112
  111  AP(I,J)=AP(J,I)
 112  DP(I,J)=AP(I,J)*PF
       CALL EIGEN(AP,CP,EIG,NOTR,50,1)
        CALL REORD (EIG,50,NOTR,AP,BP)
        DO 216 I=2,NOTR
         J=I-1
        IF (EIG(I).GT.EIG(J)) GO TO 226
  216  CONTINUE
         GO TO 236
  226       WRITE (6,181)
  181   FORMAT (//28H  EIGENVALUES IN BAD ORDER   //)
          GO TO 1000
  236   DO 113     I=1,NOTR
  113     EIG(I)=EIG(I)*PF
        CALL TRANS(BP,AP,50,NOTR)
      CALL MULT(CP,BP,AP,50,NOTR)
        CALL MULT (CP,DP,AP,50,NOTR)
        CALL MULT(DP,BP,CP,50,NOTR)
         WRITE (6,114)
 114 FORMAT(/10X,12H EIGENVALUES //)
       WRITE (6,105)(EIG(I),I=1,NOTR)
       WRITE (6,105)(DP(I,I),I=1,NOTR)
      NNN=NF
       DO  115 I=1,NNN
            PE=EIG(I)
           PD=(PE-DP(I,I))
            PE=PE*.01
           PD=ABS(PD)
        IF (PD.GT.PE)  GO TO 116
  115  CONTINUE
        GO TO 118
  116  WRITE (6,117)
  117 FORMAT (10X,39H EIGENVALUE ERROR-CHECK PROGRAM PLEASE//)
         GO TO 1000
  118  CONTINUE
      NUM=NOTR
  119  CONTINUE
      REWIND NFILE
```

```
      CALL LOADS(NFILE,NSCR)
      WRITE(6,702)
  702 FORMAT(/,5X,32HNUMBER OF EIGENVALUES TO BE USED)
      READ(5,1954)CARD,ANF
      WRITE(6,1955)CARD,ANF
      NF=ANF
 1954 FORMAT(A6,4X,F10.0)
 1955 FORMAT(1X,A6,4X,F10.0)
      IF(CARD-CEIG)1956,1958,1956
 1956 WRITE(6,1957)
 1957 FORMAT(1X,21H**ERROR IN ABOVE CARD)
      STOP
 1958 CONTINUE
        REWIND NFILE
        IF (NOTR.GT.NF)  NUM=NF
        DO 130 I=1,NOLO
        READ (NFILE) NROW,NODE,LOADC,SFX(I),SFY(I),PHI(I),DX,DY
        ANG=PHI(I)
       DO 129  J=1,NUM
       CX(J)=0.
       CY(J)=0.
        IF (LOADC.EQ.0)  GO TO 149
        DO 127 K=1,NOTR
       CX(J)=CX(J)+AP(K,J)*DX(K)*P(K)
  127  CY(J)=CY(J)+AP(K,J)*DY(K)*P(K)
        BL(J,I)=SFX(I)*CX(J)*COS(ANG)-SFY(I)*CY(J)*SIN(ANG)
        BM(J,I)=SFX(I)*CX(J)*SIN(ANG)+SFY(I)*CY(J)*COS(ANG)
        GO TO 129
  149  BL(J,I)=0.
       BM(J,I)=0.
  129 CONTINUE
      LNOD(I)=NODE
      LROW(I)=NROW
  130  CONTINUE
       DO 421 J=1,NUM
        WRITE (6,424) J
        WRITE (6,105) (BL(J,I),I=1,NOLO)
        WRITE (6,105) (BM(J,I),I=1,NOLO)
  421   CONTINUE
  424 FORMAT(///26H TRANSFORMED LOADS OF THE  I2,12H TH MODE
  100  FORMAT(//20H TRANSFORMED FORCES    ,215)
         CALL SHIP1(NCARD)
       DO 2001 L=1,NOLO
         KR=LROW(L)
       LO(L)=LNOD(L)-NONO(KR)
       DO 2001 J=1,KR
 2001  LO(L)=LO(L)+NONO(J)
        NTO=0
       WRITE (6,109) (LO(I),I=1,NOLO)
        DO 201 N=1,NORO
  201  NTO=NTO+NONO(N)*2
         DO 304 NL=1,NUM
      SPRING=1./EIG(NL)
        CALL SHIP2(SPRING,1)
       DO 202 L=1,NTO
  202  P(L)=0.
       DO 203 L=1,NOLO
       J=(LO(L)-1)*2
       J1=J+1
       J2=J+2
      P(J1)=BL(NL,L)/EIG(NL)
  203  P(J2)=BM(NL,L)/EIG(NL)
         WRITE (6,105) (BL(NL,L),L=1,NOLO)
       WRITE (6,105) (BM(NL,L),L=1,NOLO)
        CALL SHIP3
      DO 205 L=1,NOLO
       J=(LO(L)+1)*2
       J1=J+1
       J2=J+2
        BL(NL,L)=P(J1)
  205   BM(NL,L)=P(J2)
       REWIND K3
       REWIND K1
  304  CONTINUE
       WRITE (6,206) (K,K=1,NOLO)
  206  FORMAT (// 20H NODAL DISPLACEMENT  ,//7I10 )
       DO 305 I=1,NUM
       WRITE (6,105) EIG(I)
       WRITE (6,105) (BL(I,L),L=1,NOLO)
  305  WRITE (6,105) (BM(I,L),L=1,NOLO)
      CALL MATINS(AF,50,NOTR,CP,50,0,DE,ID,IND)
       IF (ID.EQ.1) GO TO 60
        WRITE (6,307) ID
  307  FORMAT (//20H MATRIX SINGULAR     I5)
         GO TO 1000
   60     REWIND NFILE
      DO 405 J=1,NOLO
       READ (NFILE) NROW,NODE,LOADC,SFX(J),SFY(J),PHI(J),DX,DY
         AN=PHI(J)
       DO 401 I=1,NOTR
```

```
      CX(I)=0.
     CY(I)=0.
      DO 402 K=1,NUM
      CX(I)=CX(I)+AP(I,K)*BL(K,J)/P(I)
  402 CY(I)=CY(I)+AP(I,K)*BM(K,J)/P(I)
  401 CONTINUE
      WRITE (6,107) J
      WRITE (6,105) (CX(M),M=1,NOTR,2)
      WRITE (6,105) (CY(M),M=1,NOTR,2)
       DO 403 I=1,NOTR
       DQX(I,J)=0.
       DQY(I,J)=0.
       DO 403 K=1,NOTR
       DQX(I,J)=DQX(I,J)+AF(I,K)*(DX(K)-CX(K)*COS(AN)-CY(K)*SIN
     1 (AN))*SFX(J)
  403  DQY(I,J)=DQY(I,J)+AF(I,K)*(DY(K)+CX(K)*SIN(AN)-CY(K)*COS
     1 (AN))*SFY(J)
  405  CONTINUE
       WRITE (6,420)
       DO 404 I=1,NOTR
     WRITE(6,106)I
     WRITE(6,800)
  800 FORMAT(/,5X,31HX-DEFLECTIONS FOR LONGITUDINALS,/)
      WRITE (6,105) (DQX(I,N),N=1,NOLO)
     WRITE(6,802)
  802 FORMAT(/,5X,31HY-DEFLECTIONS FOR LONGITUDINALS,/)
  404  WRITE (6,105) (DQY(I,N),N=1,NOLO)
  106 FORMAT(1H1,5X,16HTRANSVERSE NO.      I5,/,1X,75(1H*))
  107 FORMAT(//31H DEFLECTION OF THE NODAL POINTS I5//)
       DO 430 I=1,IB
         L=JT(I)
       DO 431 J=1,NOLO
        AN=PHI(J)
        BL(I,J)=DQX(L,J)*COS(AN)-DQY(L,J)*SIN(AN)
       BM(I,J)=DQX(L,J)*SIN(AN)+DQY(L,J)*COS(AN)
  431  CONTINUE
      WRITE (6,106) L
      WRITE (6,420)
  420 FORMAT(// 33H REAL LOADS UPON THE TRANSVERSES      //)
     WRITE(6,804)
  804 FORMAT(/,5X,13HREAL X-FORCES,/)
      WRITE (6,105) (BL(I,J),J=1,NOLO)
     WRITE(6,806)
  806 FORMAT(/,5X,13HREAL Y-FORCES,/)
      WRITE (6,105) (BM(I,J),J=1,NOLO)
  430 CONTINUE
       DO 450 I=1,IB
        IF (I.EQ.1)  GO TO 441
     REWIND NCARD
     CALL TRV(NCARD)
     REWIND NACARD
       CALL SHIP1(NCARD)
  441  CALL SHIP2(0.,0)
       DO 435 M=1,NTO
  435  R(M)=0.
       DO 440 M=1,NOLO
        AN=PHI(M)
       J=(LO(M)-1)*2
       J1=J+1
       J2=J+2
       R(J1)=BL(I,M)
  440  R(J2)=BM(I,M)
       WRITE (6,105) (R(JU),JU=1,NTO)
       CALL SHIP3
       CALL SHIP4
  450 CONTINUE
  109 FORMAT (10I5)
  105 FORMAT(1X,12E11.4)
 1000 STOP
       END
CZZZZIE  FR5 REORD,REORD,REORD
      SUBROUTINE  REORD    (ELAM,N,NTV,BP,BL)
C...ROUTINE ARRANGES EIGENVALUES IN DESCENDING ORDER WITH
C...THE CORRESPONDING RE-ARRANGING OF THE EIGENVECTORS
       DIMENSION ELAM(N),BP(N,N),BL(N,N)
        WRITE (6,100)
  100  FORMAT (// 30H  EIGENVALUES AND EIGENVECTORS  //)
      WRITE (6,101) (ELAM(J),J=1,NTV)
         DO 21 J=1,NTV
   21  WRITE (6,101) (BP(I,J),I=1,NTV)
       NTV1=NTV-1
  101  FORMAT (1X,12E11.4)
   10   DO 20 I=1,NTV1
       IF (ELAM(I)-ELAM(I+1)) 30,20,20
   20  CONTINUE
       GO TO 50
   30   SAVE=ELAM(I)
       ELAM(I)=ELAM(I+1)
        ELAM(I+1)=SAVE
       DO 40 J=1,NTV
```

```
      BL(J,I)=BP(J,I)
      BP(J,I)=BP(J,I+1)
      BP(J,I+1)=BL(J,I)
   40  CONTINUE
      GO TO 10
   50  WRITE (6,100)
       WRITE (6,101) (FLAM(J),J=1,NTV)
      RETURN
      END
CZZZZI   FR5  BEGIN
      SUBROUTINE BEGIN(IB,JT)
C...ROUTINE INPUT BASIC SHIP PARAMETERS
      DIMENSION JT(5)
      DIMENSION AJT(6)
      COMMON /MATRL/E,G,GNU,ALPHA,CONVF
      COMMON /SHIP/NOLO,LNO(100),SFX(100),SFY(100),PHI(100),
     +NOTP,ZTR(50),ZLEN,P(50),XI,XA,NSEC
      COMMON /SAFE/ UNITS(4)
      DATA FACTOR,ALENG,SPACE,STIFF,ANALY /6HFACTOR,6HLENGTH,6HSPACIN,
     1  6HSTIFFN , 6HANALYZ    /
      DATA AMATER /6HMATERI  /
      IERR=0
      WRITE(6,100)
 100  FORMAT(1HI,5X,47HTRANSVERSE STRENGTH ANALYSIS OF LONGITUDINALLY
     U12HFRAMED SHIPS,/,50X,23HBY COM/CODE CORPORATION,/,1X,75(1H*))
      WRITE(6,237)
 237  FORMAT(/,5X,47HCONVERSION FACTOR IS APPLIED TO ALL DIMENSIONAL
     116H UNITS OF LENGTH)
      READ(5,330)CARD,CONVF
 330  FORMAT(A6,4X,6F10.0)
      WRITE(6,331)CARD,CONVF
 331  FORMAT(1X,A6,4X,6F10.3)
      IF(CARD-FACTOR)300,400,300
 341  FORMAT(1X,21H**ERROR IN ABOVE CARD)
 300  WRITE(6,341)
      IERR=1
 400  CONTINUE
      READ(5,505)CARD,(UNITS(I),I=1,4)
 505  FORMAT(A6,4X,4A5)
      WRITE(6,506)CARD,(UNITS(I),I=1,4)
 506  FORMAT(1X,A6,4X,4A6)
      WRITE(6,401)
 401  FORMAT(/,1X,5X,37HLENGTH OF SHIP SECTION TO BE ANALYZED)
      READ(5,330)CARD,ZLEN
      WRITE(6,331)CARD,ZLEN
      ZLEN=ZLEN*CONVF
      IF(CARD-ALENG)402,403,402
 402  WRITE(6,341)
      IERR=1
 403  WRITE(6,404)
 404  FORMAT(/,5X,30HYOUNGS MODULUS, POISSONS RATIO)
      READ(5,330)CARD,E,GNU
      WRITE(6,332)CARD,E,GNU
 332  FORMAT(1X,A6,4X,E10.3,F10.3)
      ALPHA=0.
      G=E/2./(1.+GNU)
      E=E/CONVF/CONVF/CONVF
      G=G/CONVF/CONVF/CONVF
      IF(CARD-AMATER)405,406,405
 405  WRITE(6,341)
      IERR=1
 406  WRITE(6,407)
 407  FORMAT(/,5X,27HSPACING BETWEEN TRANSVERSES)
      READ(5,330)CARD,Y
      WRITE(6,331)CARD,Y
      Y=Y*CONVF
         NOTR=IFIX(ZLEN/Y +.5)-1
      YSUM=0.
      DO 25 K=1,NOTR
      ZTR(K)=YSUM+Y
  25  YSUM=YSUM+Y
      IF(CARD-SPACE)408,409,408
 408  WRITE(6,341)
      IERR=1
 409  WRITE(6,410)
 410  FORMAT(/,5X,45HSTIFFNESS FACTORS OF ALL TRANSVERSES IN ORDER
     1 11H FROM STERN  //)
      K=0
 415  READ(5,330)CARD,(AJT(I),I=1,6)
      DO 420  I=1,6
      K=K+1
      II=I
      P(K)=AJT(I)
      IF(K-NOTR)420,425,425
 420  CONTINUE
 425  WRITE(6,331)CARD,(AJT(I),I=1,II)
      IF(CARD-STIFF)426,427,426
 426  WRITE(6,341)
      IERR=1
      GO TO 450
```

```
 427  IF(K-NOTR)415,450,450
 450  WRITE(6,451)
 451  FORMAT(/,5X,50HLIST TRANSVERSE BY POSITION FROM STERN THAT ARE TO
     1 12H BE ANALYZED)
      READ(5,330)CARD,(AJT(I),I=1,5)
      IB=0
      DO 34 I=1,5
      IF(AJT(I))34,34,32
  32  IB=IB+1
      JT(IB)=IFIX(AJT(I))
  34  CONTINUE
      WRITE(6,331)CARD,(AJT(I),I=1,IB)
      IF(CARD-ANALY)465,470,465
 465  WRITE(6,341)
      IERR=1
 470  IF(IERR)480,480,475
 475  STOP
 480  RETURN
      END
CZZZZI   FR5  TRV
      SUBROUTINE TRV(NCARD)
C...ROUTINE INPUTS DEFINITION OF TRANSVERSE AND GENERATES
C...ALL FINITE ELEMENT DATA
      INTEGER DEFIN,BLANK
      COMMON /WORK/XC(42),YC(28),NONO(25),NXC,NYC,
     +DEFIN,NODE(40,25),LROW(100),LNOD(100)
      COMMON /SAFE/UNITS(4),IT1(50),IT2(50),
     +JT1(50),JT2(50),THK(50),AX(100),IB1(100),IB2(100),
     +JB1(100),JB2(100),IO(100),JO(100),NCR(51),NOB(25)
      COMMON /SAFE/IERR,XLBHD,DECL,DESH,
     +I,J,JROW1,JROW2,ICOL1,ICOL2,BLANK,ID,NORO,
     +NOMAT,NN1,ITEMP,NOBB,MCOM,NOBO,MCON,NN(25),
     +X,Y,Z,SLOPE,NTA,K,NBAR,IFSF,NOUT,NOMAX,JRO1,JRO2,
     +ICOL,JROW,NXCM1,IEGNU,MEMNO,JP1,IP1,NI1,NI2,NI3,NI4,
     +T,AX1,AX2,AX3,AX4,AX5,J1,J2,JJ,II,IFI,IFJ,IFK,IFL,
     +NUL,P,MEMTYP,MEMTO,N
      COMMON /MATRL/E,G,GNU,ALPHA,CONVF
      DATA  DEFIN,BLANK /1H*,1H  /
      DATA XCOL,YROW,VOID,CEND /6HXCOORD ,6HYCOORD ,6HVOID   ,
     1  6HEND    /
      DATA BCCAR,WEIGH' / 6HBC      , 6HWEIGHT    /
      DATA PLATE,BAR / 6HPLATE  , 6HBAR    /
      DATA  SOLV,ANODE,BNODE /6HSOLVE    , 6HNODE    ,  6HNODES     /
      IERR=0
      WRITE(6,200)
 200  FORMAT(1H1,5X,39HFINITE ELEMENT DEFINITION OF TRANSVERSE,/,1X,
     1   75(1H*))
 330  FORMAT(A6,4X,6F10.0)
 331  FORMAT(1X,A6,4X,6F10.3)
      READ(5,330)CARD,DECL
      WRITE(6,331)CARD,DECL
      READ(5,330)CARD,DESH
      WRITE(6,331)CARD,DESH
      READ(5,330)CARD,XLBHD
      WRITE(6,331)CARD,XLBHD
C**
C**SET UP BASIC GRID COORDINATES
      NXC=0
      NYC=0
 335  READ(5,330)CARD,COORD
      WRITE(6,331)CARD,COORD
      IF(CARD-CEND)336,350,336
 336  IF(CARD-XCOL)338,337,338
 337  NXC=NXC+1
      XC(NXC)=COORD
      GO TO 335
 338  IF(CARD-YROW)340,339,340
 339  NYC=NYC+1
      YC(NYC)=COORD
      GO TO 335
 340  WRITE(6,341)
 341  FORMAT(1X,21H**ABOVE CARD IN ERROR)
      IERR=1
      GO TO 335
 350  CONTINUE
      NXC=NXC+1
      XC(NXC)=XLBHD
      CALL SORT(XC,NXC)
      YC(NYC+1)=DECL
      YC(NYC+2)=DESH
      NYC=NYC+2
      CALL SORT(YC,NYC)
      DO 100 I=1,NXC
      DO 100 J=1,NYC
 100  NODE(I,J)=DEFIN
C**
C**DEFINE VOID AREAS WITHIN TRANSVERSE
      WRITE(6,205)
 205  FORMAT(1H1,5X,38HDEFINITION OF VOID AREAS OF TRANSVERSE,/,1X,
     1   75(1H*))
```

```
      WRITE(6,333)
  333 FORMAT(/,13X,5HSTART,6X,3HEND,6X,5HSTART,6X,3HEND,/,
     1  14X,3HROW,7X,3HROW,7X,3HCOL,7X,3HCOL)
  110 READ(5,330)CARD,YR1,YR2,XC1,XC2
      WRITE(6,332)CARD,YR1,YR2,XC1,XC2
  332 FORMAT(1X,A6,4X,4(F6.0,4X))
      IF(CARD-CEND)111,150,111
  111 IF(CARD-VOID)112,113,112
  112 WRITE(6,341)
      IERR=1
      GO TO 110
  113 JROW1=YR1
      JROW2=YR2
      ICOL1=XC1
      ICOL2=XC2
      CALL SWTCH(JROW1,JROW2)
      CALL SWTCH(ICOL1,ICOL2)
  130 DO 135  J=JROW1,JROW2
      DO 135  I=ICOL1,ICOL2
  135 NODE(I,J)=BLANK
      GO TO 110
C--PLOT TRANSVERSE PROFILE
  150 DO 155 J=1,NYC
  155 NN(J)=J
      WRITE(6,207)(NN(J),J=1,NYC)
  207 FORMAT(//,1X,1HC,/,1X,1HO,5X,3HROW,/,1X,1HL,25I3,/)
      DO 140 I=1,NXC
      WRITE(6,206)I,(NODE(I,J),J=1,NYC)
  206 FORMAT(/,I3,1X,25(A1,2X))
  140 CONTINUE
C**DEFINE NUMBER OF NODES PER ROW
      DO 410  J=1,NYC
      NONO(J)=0
      DO 405  I=1,NXC
      IF(NODE(I,J)-DEFIN)405,404,405
  404 NONO(J)=NONO(J)+1
  405 CONTINUE
  410 CONTINUE
C**
C**BEGIN WRITING ON NCARD FILE
      ID=7777
      NORO=NYC
      NOMAT=1
      NN1=2
      ITEMP=0
      REWIND NCARD
C--NCARD
      WRITE(NCARD,212)ID,NORO,NN1,NOMAT,ITEMP,(UNITS(I),I=1,4)
  212 FORMAT(5I5,4A6)
C--DEFINITION OF BOUNDARY CONDITIONS
      NOBO=0
      DO 166  J=1,NYC
      IF(NODE(NXC,J)-DEFIN)166,167,166
  167 NOBO=NOBO+1
      NOB(NOBO)=J
  166 CONTINUE
      MCON=1
      DO 168  I=1,NXC
      IF(XC(I)-XLBHD)168,169,169
  168 CONTINUE
  169 MOB=NODET(I,NODE,DEFIN,1,NXC,NONO(1))
      MOBC=I
      WRITE(6,272)
  272 FORMAT(1H1,5X,19HBOUNDARY CONDITIONS,/,1X,75(1H*))
      WRITE(6,273)
  273 FORMAT(/,10X,27HRESTRICTED X-DEFLECTION = 1,/,10X,
     1 27HRESTRICTED Y-DEFLECTION = 0,//,13X,4HC.L.,5X,6HBOTTOM,/,11X,
     2  2(8HSUPPORTS,2X))
      READ(5,330)CARD,ANOBB,AMCON
      WRITE(6,331)CARD,ANOBB,AMCON
        M=CON=AMCON
      NOBB=ANOBB
      IF(CARD-BCCAR)391,288,391
  391 WRITE(6,341)
      IERR=1
  288 CONTINUE
C--NCARD
      WRITE(NCARD,274)NOBO,MCON,NOBB,MCOM
  274 FORMAT(20I3)
      WRITE(NCARD,274)(NOB(I),I=1,NOBO)
      WRITE(NCARD,274)MOB
      READ(5,330)CARD,FA
      WRITE(6,331)CARD,FA
      IF(CARD-WEIGH)393,394,393
  393 WRITE(6,341)
      IERR=1
  394 CONTINUE
      WRITE(NCARD,214)FA
      WRITE(6,311)NOBO
  311 FORMAT(//,5X, 9HTHERE ARE, I3,14H C.L. SUPPORTS)
```

```
      WRITE(6,312)(NOB(I),I=1,NOBO)
  312 FORMAT(//,5X,34HC.L. SUPPORTS ARE DEFINED FOR ROWS,/,25I3)
      WRITE(6,314)MOBC,MOB
  314 FORMAT(//,5X,36HSUPPORT  AT BOTTOM IS LOCATED ON COL,I3,
     1 6H (NODE,I2,1H))
      WRITE(NCARD,214)E,GNU,ALPHA
  214 FORMAT(E10.2,F7.2,E10.2)
C--NCARD
      WRITE(NCARD,215)(NONO(J),J=1,NORO)
  215 FORMAT(25I3)
      WRITE(6,282)
  282 FORMAT(1H1,5X,20HROW NUMBERING SYSTEM,/,1X,75(1H*))
      WRITE(6,207)(NN(J),J=1,NYC)
      DO 406  I=1,NXC
      DO 407  J=1,NYC
      NCR(J)=0
      IF(NODE(I,J)-DEFIN)407,408,407
  408 NCR(J)=NODET(I,NODE,DEFIN,J,NXC,NONO(J))
  407 CONTINUE
      WRITE(6,251)I,(NCR(J),J=1,NYC)
  406 CONTINUE
C**
C**DEFINE COORDINATES FOR NODES
      Z=0.
      SLOPE=(DECL-DESH)/XL8HD
      DO 430  J=1,NORO
      Y=YC(J)*CONVF
      DO 420  I=1,NXC
      IF(NODE(I,J)-DEFIN)420,411,420
  411 X=XC(I)
      IF(J-NORO)415,412,412
  412 IF(X-XL8HD)413,415,415
  413 Y=DESH+SLOPE*X
      IF(Y-YC(J-1))414,414,416
  414 Y=Y+0.01*(YC(J)-YC(J-1))
  416 Y=Y*CONVF
  415 X=X*CONVF
C--NCARD
      WRITE(NCARD,216)X,Y,Z
  216 FORMAT(3F10.2)
  420 CONTINUE
  430 CONTINUE
C**
C**DEFINE AREAS OF THE TRANSVERSE FOR DIFFERENT PLATE THICKNESSES
      WRITE(6,220)
  220 FORMAT(1H1,5X,31HDEFINITION OF PLATE THICKNESSES,/,1X,75(1H*))
      WRITE(6,222)
  222 FORMAT(/,23X,5HSTART,6X,3HEND,6X,5HSTART,6X,3HEND,6X,5HPLATE,/,
     1   11X,9HTHICKNESS,4X,3HROW,7X,3HROW,7X,3HCOL,7X,3HCOL,7X,3HNO.)
      NTA=0
  440 JNTA=NTA+1
      READ(5,330)CARD,XTK,YR1,YR2,XC1,XC2
      WRITE(6,444)CARD,XTK,YR1,YR2,XC1,XC2,JNTA
  444 FORMAT(1X,A6,4X,F10.3,4(F6.0,4X),1H(,I2,1H))
      IF(CARD-CEND)441,450,441
  441 IF(CARD-PLATE)442,443,442
  442 WRITE(6,341)
      IFRR=1
      GO TO 440
  443 NTA=NTA+1
      THK(NTA)=XTK*CONVF
      JT1(NTA)=YR1
      JT2(NTA)=YR2
      IT1(NTA)=XC1
      IT2(NTA)=XC2
      CALL SWTCH(JT1(K),JT2(K))
      CALL SWTCH(IT1(K),IT2(K))
      GO TO 440
  450 CONTINUE
      WRITE(6,221)NTA
  221 FORMAT(//,5X,41HNO. OF AREAS OF COMMON THICKNESSES (50) =,I2)
      WRITE(6,220)
      WRITE(6,207)(NN(J),J=1,NYC)
      DO 456  I=1,NXC
      DO 458  J=1,NYC
      NCR(J)=0
      IF(NODE(I,J)-DEFIN)458,457,458
  457 DO 455  K=1,NTA
      IF(JT1(K)-J)451,451,455
  451 IF(JT2(K)-J)455,452,452
  452 IF(IT1(K)-I)453,453,455
  453 IF(IT2(K)-I)455,454,454
  454 NCR(J)=K
  455 CONTINUE
  458 CONTINUE
  251 FORMAT(/,I3,I2,24I3)
  456 WRITE(6,251)I,(NCR(J),J=1,NYC)
C**
C**DEFINE BAR ELEMENTS
      WRITE(6,223)
```

```
  223  FORMAT(1H1,5X,26HDEFINITION OF BAR ELEMENTS,/,1X,75(1H*))
  459  WRITE(6,225)
  225  FORMAT(/,23X,5HSTART,6X,3HEND,6X,5HSTART,6X,3HEND,/,14X,
     1    2HAX,8X,3HROW,7X,3HROW,7X,3HCOL,7X,3HCOL)
       NBAR=0
   445  NNBAR=NBAR+1
       READ(5,330)CARD,AAX,YR1,YR2,XC1,XC2
       WRITE(6,444)CARD,AAX,YR1,YR2,XC1,XC2 ,NNBAR
       IF(CARD-CEND)446,470,446
  446  IF(CARD-BAR)447,448,447
  447  WRITE(6,341)
       IERR=1
       GO TO 445
  448  NBAR=NBAR+1
       AX(NBAR)=AAX*CONVF*CONVF
       JB1(NBAR)=YR1
       JB2(NBAR)=YR2
       IB1(NBAR)=XC1
       IB2(NBAR)=XC2
       K=NBAR
       IF(IB1(K)-IB2(K))461,460,461
  460  CALL SWTCH(JB1(K),JB2(K))
       GO TO 469
  461  IF(JB1(K)-JB2(K))463,462,463
  462  CALL SWTCH(IB1(K),IB2(K))
       GO TO 469
  463  IF(IABS(JB2(K)-JB1(K))-IABS(IB2(K)-IB1(K)))464,465,464
  464  IERR=1
       WRITE(6,262)
  262  FORMAT(1X,50H**ERROR-ABOVE BAR ELEMENT INTERSECTS BETWEEN NODES)
       GO TO 469
  465  IF(JB1(K)-JB2(K))469,469,466
  466  CALL SWTCH(JB1(K),JB2(K))
       ITEMP=IB1(K)
       IB1(K)=IB2(K)
       IB2(K)=ITEMP
  469  CONTINUE
       GO TO 445
  470  CONTINUE
       WRITE(6,224)NBAR
  224  FORMAT(//,1X,22HNO. BAR ELEMENTS (100), I4)
C**
C**DEFINE OUTPUT REQUIREMENTS
  484  WRITE(6,226)
  226  FORMAT(1H1,5X,21HOUTPUT SPECIFICATIONS,/,1X,75(1H*))
       WRITE(6,228)
  228  FORMAT(//,5X,20HNODE FORE ONLY    = 1,/,5X,
     1            20HFORCE AND STRESS = 2,/,5X,
     2            20HSTRESS ONLY      = 3,/)
       READ(5,330)CARD,AIFSF
       WRITE(6,331)CARD,AIFSF
       IFSF=AIFSF
       IF(CARD-SOLV)478,479,478
  478  WRITE(6,341)
       IERR=1
  479  CONTINUE
       NOMAX=100
       NOUT=0
       WRITE(6,227)
  227  FORMAT(/,5X,25HNODES SELECTED FOR OUTPUT,/,14X,3HROW,7X,3HCOL)
  471  READ(5,330)CARD,YR1,XC1
       WRITE(6,332)CARD,YR1,XC1
       IF(CARD-CEND)472,485,472
  472  IF(CARD-ANODE)473,474,473
  473  WRITE(6,341)
       IERR=1
       GO TO 471
  474  NOUT=NOUT+1
       IF(NOUT-NOMAX)477,477,475
  475  WRITE(6,476)NOMAX
  476  FORMAT(1X,22H**MAX.NODES FOR OUTPUT,I8)
       IERR=1
       GO TO 471
  477  IO(NOUT)=XC1
       JO(NOUT)=YR1
       GO TO 471
  485  CONTINUE
       WRITE(6,1912)
 1912  FORMAT(1H1)
C**
C**BEGIN SETTING UP ELEMENTS BY ROW
       NXCM1=NXC-1
       IFGNU=0
       MEMTO=0
       DO 1000  J=1,NORO
       MEMNO=0
       JP1=J+1
       DO 950  I=1,NXCM1
       IP1=I+1
       NT1=NODET(I,NODE,DEFIN,J,NXC,NONO(J))
```

```
      NI2=NODET(IP1,NODE,DEFIN,J,NXC,NONO(J))
      NI3=NODET(I,NODE,DEFIN,JP1,NXC,NONO(JP1))
      NI4=NODET(IP1,NODE,DEFIN,JP1,NXC,NONO(JP1))
      IF(J-NORO)605,675,675
C**
C**TEST FOR THICKNESS AREA
 605  DO 650  K=1,NTA
      IF(JT1(K)-J)610,610,650
 610  IF(JT2(K)-J)650,650,615
 615  IF(IT1(K)-I)620,620,650
 620  IF(IT2(K)-I)650,650,625
 625  T=THK(K)
      GO TO 675
 650  CONTINUE
      T=0.
      IF(NODE(I,J)-DEFIN)651,675,651
 651  IF(NODE(IP1,JP1)-DEFIN)675,652,675
 652  IF(NODE(I,JP1)-DEFIN)675,653,675
 653  IF(NODE(IP1,J)-DEFIN)675,655,675
 655  WRITE(6,235)I,J
 235  FORMAT(1X,35HERROR-THICKNESS NOT DEFINED FOR ROW,I3,4H COL,I3)
      IERR=1
C**
C**TEST FOR BAR ELEMENTS
 675  AX1=0.
      AX2=0.
      AX3=0.
      AX4=0.
      AX5=0.
      IF(NBAR)770,770,676
 676  DO 750  K=1,NBAR
C
C--HORIZONTAL BARS.....JB1=JB2
      IF(JB1(K)-JB2(K))688,680,688
C----BOTTOM BAR
 680  IF(JB1(K)-J)750,682,750
 682  IF(IB1(K)-I)684,684,750
 684  IF(IB2(K)-IP1)750,686,686
 686  AX2=AX(K)
      GO TO 750
 688  IF(J-NORO)696,750,750
C
C--VERTICAL BARS.....IB1=IB2
 696  IF(IB1(K)-IB2(K))714,698,714
C----LEFT HAND BAR
 698  IF(IB1(K)-I)706,700,706
 700  IF(JB1(K)-J)702,702,750
 702  IF(JB2(K)-JP1)750,704,704
 704  AX1=AX(K)
      GO TO 750
C----RIGHT HAND BAR
 706  IF(IB1(K)-IP1)750,708,750
 708  IF(JB1(K)-J)710,710,750
 710  IF(JB2(K)-JP1)750,712,712
 712  AX5=AX(K)
      GO TO 750
C
C--DIAGONAL BARS
 714  J1=JB1(K)
      J2=JB2(K)
      IF(IB1(K)-IB2(K))716,716,732
C----BAR BOTTOM LEFT TO TOP RIGHT
 716  IF(IB1(K)-I)718,718,750
 718  IF(IB2(K)-IP1)750,720,720
 720  II=IB1(K)-1
      DO 730  JJ=J1,J2
      II=II+1
      IF(JJ-J)730,722,750
 722  IF(II-I)750,724,750
 724  AX3=AX(K)
      GO TO 750
 730  CONTINUE
      GO TO 750
C----BAR TOP LEFT TO BOTTOM RIGHT
 732  IF(IB1(K)-IP1)750,734,734
 734  IF(IB2(K)-I)736,736,750
 736  II=IB1(K)+1
      DO 740  JJ=J1,J2
      II=II-1
      IF(JJ-J)750,738,740
 738  IF(II-IP1)750,739,750
 739  AX4=AX(K)
 740  CONTINUE
 750  CONTINUE
C
C--CHECK FOR LAST COLUMN
      IF(IP1-NXC)755,770,770
C--DO NOT INCLUDE RIGHT VERTICAL BAR UNLESS LAST COLUMN
 755  AX5=0.
```

```
C
C--CHECK NODES FOR OUTPUT SELECTION
 770  IFI=0
      IFJ=0
      IFK=0
      IFL=0
      NUL=0
      P=0.
      DO 800 K=1,NOUT
      IF(JO(K)-J)800,772,780
 772  IF(IO(K)-I)800,774,776
 774  IFI=1
      GO TO 800
 776  IF(IO(K)-IP1)800,778,800
 778  IFJ=1
      GO TO 800
 780  IF(JO(K)-JP1)800,782,800
 782  IF(IO(K)-I)800,784,786
 784  IFK=1
      GO TO 800
 786  IF(IO(K)-IP1)800,788,800
 788  IFL=1
 800  CONTINUE
      IF(J-NORO)805,900,900
C
C--CHECK NODE DEFINITIONS FOR VOIDS
 805  IF(NODE(I,J)-DEFIN)830,810,830
 810  IF(NODE(IP1,J)-DEFIN)840,812,840
 812  IF(NODE(I,JP1)-DEFIN)850,814,850
 814  IF(NODE(IP1,JP1)-DEFIN)860,816,860
C
C--QUADRILATERAL PLATE ELEMENT
 816  MEMNO=MEMNO+1
      MEMTYP=2
      WRITE(NCARD,250)MEMNO,MEMTYP,IEGNU,IFSF,IFI,IFJ,IFK,IFL,
     +J,NI1,J,NI2,JP1,NI3,JP1,NI4,T,P,P,P,P
 250  FORMAT(I3,7I1,8I2,6E15.7)
      WRITE(6,250)MEMNO,MEMTYP,IEGNU,IFSF,IFI,IFJ,IFK,IFL,
     +J,NI1,J,NI2,JP1,NI3,JP1,NI4,T
      GO TO 900
C-- J,IP1 VOID - NO PLATE POSSIBLE
 822  AX2=0.
      AX4=0.
      AX5=0.
      GO TO 900
C-- JP1,I VOID - NO PLATE POSSIBLE
 824  AX1=0.
      AX4=0.
      GO TO 900
C-- JP1,IP1 VOID - NO PLATE POSSIBLE
 826  AX3=0.
      AX5=0.
      GO TO 900
C
C..(I,J) VOID
 830  AX1=0.
      AX2=0.
      AX3=0.
      IF(NODE(IP1,J)-DEFIN)822,832,822
 832  IF(NODE(IP1,JP1)-DEFIN)826,834,826
 834  IF(NODE(I,JP1)-DEFIN)824,836,824
C--TRI-PLATE UPPER RIGHT
 836  MEMNO=MEMNO+1
      MEMTYP=1
      WRITE(NCARD,250)MEMNO,MEMTYP,IEGNU,IFSF,IFJ,IFK,IFL,NUL,
     +J,NI2,JP1,NI3,JP1,NI4,NUL,NUL,T,P,P,P,P
      WRITE(6,250)MEMNO,MEMTYP,IEGNU,IFSF,IFJ,IFK,IFL,NUL,
     +J,NI2,JP1,NI3,JP1,NI4,NUL,NUL,T
      GO TO 900
C..(I,J) DEF., (IP1,J) VOID
 840  AX2=0.
      AX4=0.
      AX5=0.
      IF(NODE(IP1,JP1)-DEFIN)826,842,826
 842  IF(NODE(I,JP1)-DEFIN)824,844,824
C--TRI-PLATE UPPER LEFT
 844  MEMNO=MEMNO+1
      MEMTYP=1
      WRITE(NCARD,250)MEMNO,MEMTYP,IEGNU,IFSF,IFI,IFK,IFL,NUL,
     +J,NI1,JP1,NI3,JP1,NI4,NUL,NUL,T,P,P,P,P
      WRITE(6,250)MEMNO,MEMTYP,IEGNU,IFSF,IFI,IFK,IFL,NUL,
     +J,NI1,JP1,NI3,JP1,NI4,NUL,NUL,T
      GO TO 900
C..(I,J),(IP1,J) DEF., (I,JP1) VOID
 850  AX1=0.
      AX4=0.
      IF(NODE(IP1,JP1)-DEFIN)826,852,826
C--TRI-PLATE LOWER RIGHT
 852  MEMNO=MEMNO+1
      MEMTYP=1
```

```
      WRITE(NCARD,250)MEMNO,MEMTYP,IEGNU,IFSF,IFI,IFJ,IFL,NUL,
     +J,NIl,J,NI2,JP1,NI4,NUL,NUL,T,P,P,P,P
      WRITE(6,250)MEMNO,MEMTYP,IEGNU,IFSF,IFI,IFJ,IFL,NUL,
     +J,NIl,J,NI2,JP1,NI4,NUL,NUL,T
      GO TO 900
C--TRI-PLATE LOWER LEFT
  860 MEMNO=MEMNO+1
      AX3=0.
      AX5=0.
      MEMTYP=1
      WRITE(NCARD,250)MEMNO,MEMTYP,IEGNU,IFSF,IFI,IFJ,IFK,NUL,
     +J,NIl,J,NI2,JP1,NI3,NUL,NUL,T,P,P,P,P
      WRITE(6,250)MEMNO,MEMTYP,IEGNU,IFSF,IFI,IFJ,IFK,NUL,
     +J,NIl,J,NI2,JP1,NI3,NUL,NUL,T
      GO TO 900
C
C--BAR ELEMENTS
  900 MEMTYP=5
      IF(AX1)904,904,902
  902 MEMNO=MEMNO+1
      WRITE(NCARD,250)MEMNO,MEMTYP,IEGNU,IFSF,IFI,IFK,NUL,NUL,
     +J,NIl,JP1,NI3,NUL,NUL,NUL,NUL,AX1,P,P,P,P
      WRITE(6,250)MEMNO,MEMTYP,IEGNU,IFSF,IFI,IFK,NUL,NUL,
     +J,NIl,JP1,NI3,NUL,NUL,NUL,NUL,AX1
  904 IF(AX2)908,908,906
  906 MEMNO=MEMNO+1
      WRITE(NCARD,250)MEMNO,MEMTYP,IEGNU,IFSF,IFI,IFJ,NUL,NUL,
     +J,NIl,J,NI2,NUL,NUL,NUL,NUL,AX2,P,P,P,P
      WRITE(6,250)MEMNO,MEMTYP,IEGNU,IFSF,IFI,IFJ,NUL,NUL,
     +J,NIl,J,NI2,NUL,NUL,NUL,NUL,AX2
  908 IF(AX3)912,912,910
  910 MEMNO=MEMNO+1
      WRITE(NCARD,250)MEMNO,MEMTYP,IEGNU,IFSF,IFI,IFL,NUL,NUL,
     +J,NIl,JP1,NI4,NUL,NUL,NUL,NUL,AX3,P,P,P,P
      WRITE(6,250)MEMNO,MEMTYP,IEGNU,IFSF,IFI,IFL,NUL,NUL,
     +J,NIl,JP1,NI4,NUL,NUL,NUL,NUL,AX3
  912 IF(AX4)916,916,914
  914 MEMNO=MEMNO+1
      WRITE(NCARD,250)MEMNO,MEMTYP,IEGNU,IFSF,IFJ,IFK,NUL,NUL,
     +J,NI2,JP1,NI3,NUL,NUL,NUL,NUL,AX4,P,P,P,P
      WRITE(6,250)MEMNO,MEMTYP,IEGNU,IFSF,IFJ,IFK,NUL,NUL,
     +J,NI2,JP1,NI3,NUL,NUL,NUL,NUL,AX4
  916 IF(AX5)950,950,918
  918 MEMNO=MEMNO+1
      WRITE(NCARD,250)MEMNO,MEMTYP,IEGNU,IFSF,IFJ,IFL,NUL,NUL,
     +J,NI2,JP1,NI4,NUL,NUL,NUL,NUL,AX5,P,P,P,P
      WRITE(6,250)MEMNO,MEMTYP,IEGNU,IFSF,IFJ,IFL,NUL,NUL,
     +J,NI2,JP1,NI4,NUL,NUL,NUL,NUL,AX5
  950 CONTINUE
      MEMTO=MEMTO+MEMNO
      IF(MEMNO)975,975,1000
  975 WRITE(NCARD,250)NUL,NUL,NUL,NUL,NUL,NUL,NUL,NUL,
     +J,NUL,NUL,NUL,NUL,NUL,NUL,NUL,P,P,P,P,P,P
      WRITE(6,250)NUL,NUL,NUL,NUL,NUL,NUL,NUL,NUL,
     +J,NUL,NUL,NUL,NUL,NUL,NUL,NUL,P,P
 1000 CONTINUE
      J=NORO+1
      WRITE(NCARD,250)NUL,NUL,NUL,NUL,NUL,NUL,NUL,NUL,
     +J,NUL,NUL,NUL,NUL,NUL,NUL,NUL,P,P,P,P,P,P
      WRITE(6,250)NUL,NUL,NUL,NUL,NUL,NUL,NUL,NUL,
     +J,NUL,NUL,NUL,NUL,NUL,NUL,NUL,P,P
      END FILE NCARD
      REWIND NCARD
      WRITE(6,281)MEMTO
  281 FORMAT(1H1,5X,23H TOU H VE JUST GENERATED,I8, 8HELEMENTS)
      IF(IERR)9999,999 ,9996
 9998 STOP
 9999 RETURN
      END
CZZZZI   FRS  NOD
      SUBROUTINE NOD(NONO,NODE,NROW,L)
C...ROUTINE COMPUTES THE LONGITUDINAL NUMBER FOR GIVEN
C...ROW AND NODE
      DIMENSION NONO   )
      L=NODE-NONO(NR   )
      DO 1 I=1,NROW
    1 L=L+NONO(I)
      RETURN
      END
CZZZZI  FRS  NODET
      FUNCTION NODET(I,NODE,DEFIN,J,NXC,NONO)
C...ROUTINE COMPUTES NODE NUMBER FOR GIVEN ROW AND COLUMN
      INTEGER DEFIN
      DIMENSION NODE(40,25)
      IF(NONO-NXC)20,10,10
   10 NODET=I
      RETURN
   20 NODET=0
      DO 50 II=1,I
      IF(NODE(II,J)-DEFIN)50,30,50
   30 NODET=NODET+1
```

```
  50  CONTINUE
      RETURN
      END
CZZZZI   FR5  SWTCH
      SUBROUTINE SWTCH(I1,I2)
      IF(I1-I2)99,99,10
  10  I=I1
      I1=I2
      I2=I
  99  RETURN
      END
CZZZZI   FR5  SORT
      SUBROUTINE SORT(X,N)
C...ROUTINE ARRANGES X-ARRAY IN ASCENDING ORDER AND THROWS
C...OUT DUPLICATE VALUES
      DIMENSION X(N)
   5  NM1=N-1
      IF(NM1)99,99,10
 10   DO 15 I=1,NM1
      IF(X(I)-X(I+1))15,20,30
  15  CONTINUE
  99  RETURN
  20  X(I)=X(N)
      N=N-1
      GO TO 5
  30  SAVE=X(I)
      X(I)=X(I+1)
      X(I+1)=SAVE
      GO TO 10
      END
CZZZZI   FR5  LONGI
      SUBROUTINE LONGI(NFILE)
C...ROUTINE INPUTS LONGITUDINAL DATA
      COMMON /MATRL/E,G,GNU,ALPHA,CONVF
      COMMON /SHIP/NOLO,LNO(100),SFX(100),SFY(100),PHI(100),
     +NOTR,ZTR(50),ZLEN,P(50),XI,XA,NSEC
      INTEGER DEFIN
      COMMON /WORK/XC(42),YC(28),NONO(25),NXC,NYC,
     +DEFIN,NODE(40,25),LROW(100),LNOD(100)
      COMMON /INFLU/AF(50,50),EIG(50)
      COMMON /SAFE/ UNITS(4)
      COMMON /SAFE/NN(25),NCR(51)
      DATA  XLONG,YLONG,CEND/6HXLONG  ,6HYLONG  ,6HEND    /
      DATA STAND /6HSTANDA  /
      IERR=0
      NOLO=0
      LOMAX=100
      WRITE(6,276)
 276  FORMAT(1H1,5X,27HDEFINITION OF LONGITUDINALS,/,1X,75(1H*))
      WRITE(6,250)
 250  FORMAT(/,14X,2HI ,8X,2HAX)
      READ(5,330)CARD,XI,XA
      WRITE(6,331)CARD,XI,XA
      NSEC=1
      XI=XI*(CONVF**4)
      XA=XA*(CONVF**2)
      IF(CARD-STAND)251,252,251
 251  WRITE(6,341)
      IERR=1
 252  CONTINUE
      WRITE(6,278)
 278  FORMAT(43X,3HROW,5X,8HSTRT COL,2X,7HEND COL,/,14X,
     12HIX,8X,2HIY,  16X     ,5H(COL),3X,10H(STRT ROW),9H(END ROW))
2000  READ(5,330)CARD,XIX,XIY,C1,C2,C3
 330  FORMAT(A6,4X,6F10.0)
      WRITE(6,444)CARD,XIX,XIY,C1,C2,C3
  444  FORMAT (1X,A6,2E15.5,3(F6.0,4X))
 331  FORMAT(1X,A6,4X,2F10.3,3(F6.0,4X))
      IF(CARD-CEND)2001,2050,2001
2001  IF(CARD-XLONG)2002,2004,2002
2002  IF(CARD-YLONG)2003,2020,2003
2003  WRITE(6,341)
 341  FORMAT(1X,21H**ABOVE CARD IN ERROR)
      IERR=1
      GO TO 2000
2004  JROW=C1
      ICOL1=C2
      ICOL2=C3
      CALL SWTCH(ICOL1,ICOL2)
2005  IF(XIX)2007,2007,2006
2006  IF(XIY)2007,2007,2010
2007  IERR=1
      WRITE(6,2008)
2008  FORMAT(1X,53H**ERROR-MOMENT OF INERTIA FOR ABOVE LONGITUDINAL NOT
     1 7HDEFINED)
      GO TO 2000
2010  DO 2015   I=ICOL1,ICOL2
      IF(NOLO-LOMAX)2011,2016,2016
2011  NOLO=NOLO+1
      N=NODET(I,NODE,DEFIN,JROW,NXC,NONO(JROW))
```

```
      LROW(NOLO)=JROW
      LNOD(NOLO)=N
      CALL NOD(NONO,N,JROW,LNO(NOLO))
      SFX(NOLO)=XIX*(CONVF**4)/XI
      SFY(NOLO)=XIY*(CONVF**4)/XI
      PHI(NOLO)=0.
 2015 CONTINUE
      GO TO 2000
 2016 WRITE(6,288)LOMAX
  288 FORMAT(1X,30HMAX. NUMBER OF LONGITUDINALS =, I8)
      IERR=1
      GO TO 2000
 2020 ICOL=C1
      JRO1=C2
      JRO2=C3
      CALL SWTCH(JRO1,JRO2)
 2035 IF(XIX)2037,2037,2036
 2036 IF(XIY)2037,2037,2040
 2037 IERR=1
      WRITE(6,2008)
      GO TO 2000
 2040 DO 2045   J=JRO1,JRO2
      IF(NOLO-LOMAX)2041,2047,2047
 2041 NOLO=NOLO+1
      N=NODET(ICOL,NODE,DEFIN,J,NXC,NONO(J))
      LROW(NOLO)=J
      LNOD(NOLO)=N
      CALL NOD(NONO,N,J,LNO(NOLO))
      SFX(NOLO)=XIX*(CONVF**4)/XI
      SFY(NOLO)=XIY*(CONVF**4)/XI
      PHI(NOLO)=0.
 2045 CONTINUE
      GO TO 2000
 2047 WRITE(6,288)LOMAX
      IERR=1
      GO TO 2000
 2050 WRITE(6,287)NOLO
  287 FORMAT(///,1X,22H**THERE ARE A TOTAL OF,I6,14H LONGITUDINALS)
      DO 3000   J=1,NYC
 3000 NN(J)=J
      WRITE(6,3001)
 3001 FORMAT(1H1,5X,29HLONGITUDINAL NUMBERING SYSTEM,/,1X,75(1H*))
      WRITE(6,3002)(NN(J),J=1,NYC)
 3002 FORMAT(//,1X,1HC,/,1X,1HO,5X,3HROW,/,1X,1HL,25I3,/)
      DO 3050 I=1,NXC
      DO 3045  J=1,NYC
         NCR(J)=1
      IF(NODE(I,J)-DEFIN)3045,3010,3045
 3010 NCR(J)=0
      N=NODET(I,NODE,DEFIN,J,NXC,NONO(J))
      DO 3040  L=1,NOLO
      IF(LROW(L)-J)3040,3015,3040
 3015 IF(LNOD(L)-N)3040,3020,3040
 3020 NCR(J)=LNO(L)
      GO TO 3045
 3040 CONTINUE
 3045 CONTINUE
      WRITE(6,3047)(NCR(J),J=1,NYC)
 3047 FORMAT(/,2X,25I3)
 3050 CONTINUE
      IF(IERR)2060,2060,2055
 2055 STOP
 2060 N=NOTR+1
C--
C--COMPUTE INFLUENCE COEFFICIENTS FOR STANDARD LONGITUDINAL
      EI=E*XI
        Y=ZLEN/FLOAT(N)
        DO 2 I=1,NOTR
        DO 2 J=1,NOTR
        IF (I.GT.J)  GO TO 1
        A=Y*FLOAT(J)
        X=Y*FLOAT(I)
       B=ZLEN-A
      GK=0.
      IF(XA.NE.0.)GK=X*B/XA/G/ZLEN
       AF(I,J)=B*X/6./EI*(ZLEN*ZLEN-B*B-X*X)/ZLEN+GK
        GO TO 2
    1  AF(I,J)=AF(J,I)
    2  CONTINUE
      RETURN
      END
      SUBROUTINE SHELL
      COMMON /SHLWEB/I8HD,B(50),T(50,2)
      COMMON /WORK/XC(42),YC(28 ),NONO(25),NXC,NYC
      COMMON /SAFE/UNITS(4),IT1(50),IT2(50),
     +JT1(50),JT2(50),THK(50),AX(100),IB1(100),IB2(100),
     +J91(100),J82(100),IO(100),JO(100),NCR(51),NOB(25)
      COMMON /SAFE/IERR,XL8HD,DECL,DESH
       DATA  CEND,CSHEL /6HEND     , 6HSHELL
      DO 5 II=1,NYC
```

```
      T(II,1)=0.0
      T(II,2)=0.0
      B(II)=0.0
    5 CONTINUE
      DO 6 II=1,NXC
          IF(XC(II)-XLBHD)6,7,7
    6 CONTINUE
    7 IBHD=II
      WRITE(6,8)
    8 FORMAT(1H1,21X,5HPLATE,5X,5HSHELL,7X,3HBHD,/,13X,3HROW,5X,
     *  6HLENGTH,5X,5HTHICK,6X,5HTHICK)
   10 READ(5,11)CARD,YROW,BL,TSHL,TLBHD
   11 FORMAT(A6,4X,4F10.0)
      WRITE(6,12)CARD,YROW,BL,TSHL,TLBHD
   12 FORMAT(1X,A6,4X,4F10.3)
      IF(CARD-CEND)15,50,15
   15 IF(CARD-CSHEL)16,20,16
   16 WRITE(6,17)
   17 FORMAT(1X,21H**ABOVE CARD IN ERROR)
      IERR=1
      GO TO 10
   20 IROW=YROW
      IF(IROW)16,16,22
   22 IF(IROW-NYC)26,26,16
   26 MNODE=MNODE+1
      T(IROW,1)=TSHL
      T(IROW,2)=TLBHD
      B(IROW)=BL
      GO TO 10
   50 CONTINUE
   60 RETURN
      END
CZZZZI  FR5  LOADS
      SUBROUTINE LOADS(NFILE,NSCR)
C...ROUTINE TO INPUT LOADING CONDITION
      INTEGER DEFIN
      COMMON /WORK/XC(42),YC(28),NONO(25),NXC,NYC,
     +DEFIN,NODE(40,25),LROW(100),LNOD(100)
      COMMON /MATRL/E,G,GNU,ALPHA,CONVF
      COMMON /SHIP/NOLO,LNO(100),SFX(100),SFY(100),PHI(100),
     +NOTR,ZTR(50),ZLEN,P(50),XI,XA,NSEC
      COMMON /SAFE/ UNITS(4)
      COMMON /SAFE/LOADC(100),DX(50),DY(50),DBASE(50),
     +ZP(50),PC(50),ZO(20),Q(20),ZPL(100)
      COMMON /SAFE/TM(5,5),TR(5,5),SI(5,1),SX(5,1),ODATA(8,50)
      COMMON /SAFE/ZI(20),EYE(20)
      COMMON /SAFE/DXL(50),DYL(50)
          COMMON /SAFE/ ST(50,25,2),A(2),R(50,2)

         COMMON /SHLWEB/IBHD,B(50),T(50,2)
      COMMON /INFLU/AF(50,50)
      DATA CEND,XUNIF,YUNIF,XFOR,YFOR,SHEAR /6HEND   ,6HXUNIFO,6HYUNIFO
     1 ,  6HXFORCE,6HYFORCE,6HSHEAR  /
      DATA  XLONG,YLONG  /6HXLONG ,6HYLONG  /
      DATA  CSHR  /6HSHEAR    /
      ZI(1)=0.
      EYE(1)=XI
      NSEC=1
   50 NREC=0
      NMISC=0
          DO 52 L=1,NOLO
   52  LOADC(L)=0
      IERR=0
      WRITE(6,55)
   55 FORMAT(1H1,5X,17HLOADING CONDITION,/,1X,75(1H*))
      NP=0
      NQ=0
      NDONE=0
   60 READ(5,61)CARD,(ZPL(IL),IL=1,6)
   61 FORMAT(A6,4X,6F10.0)
   62 FORMAT(1X,A6,4X,6F10.3)
      IF(CARD-CEND)65,1000,65
   65 IF(CARD-XUNIF)66,160,66
   66 IF(CARD-YUNIF)67,162,67
   67 IF(CARD-XFOR)68,164,68
   68 IF(CARD-YFOR)190,166,190
  160 NDIR=1
      GO TO 170
  162 NDIR=2
      GO TO 170
  164 NDIR=1
      GO TO 180
  166 NDIR=2
      GO TO 180
  170 DO 175 IL=1,3
      ILL=2*IL-1
      Z=ZPL(ILL)
      QL=ZPL(ILL+1)
      IF(Z)172,172,173
  172 IF(NQ)173,173,176
```

```
 173  NQ=NQ+1
      N=2*IL
      Q(NQ)=QL/CONVF
      ZQ(NQ)=Z*CONVF
 175  CONTINUE
 176  WRITE(6,62)CARD,(ZPL(IL),IL=1,N)
      NDONE=0
      GO TO 60
 180  DO 185 IL=1,3
      ILL=2*IL-1
      Z=ZPL(ILL)
      PL=ZPL(ILL+1)
      IF(Z)182,182,183
 182  IF(NP)183,183,186
 183  NP=NP+1
      N=2*IL
      PC(NP)=PL
      ZP(NP)=Z*CONVF
 185  CONTINUE
 186  WRITE(6,62)CARD,(ZPL(IL),IL=1,N)
      NDONE=0
      GO TO 60
C
 190  IF(CARD-XLONG)192,191,192
 191  JRO1=ZPL(1)
      JRO2=JRO1
      ICOL1=ZPL(2)
      ICOL2=ZPL(3)
      LDIR=2
      IF(ICOL2)194,194,195
 194  ICOL2=ICOL1
      GO TO 195
 192  IF(CARD-YLONG)200,193,200
 193  ICOL1=ZPL(1)
      ICOL2=ICOL1
      JRO1=ZPL(2)
      JRO2=ZPL(3)
      LDIR=1
      IF(JRO2)187,187,195
 187  JRO2=JRO1
 195  SFL=ZPL(4)
      IF(SFL)197,196,197
 196  SFL=1.0
 197  IF(NDONE)270,270,205
C---------------
C THIS SECTION COMPUTES DEFLECTIONS FOR BASIC LONGITUDINAL
 270  SFA=0.
      SFS=1.0
      CALL SETOC(NSEC,ZI,EYE,NQ,ZQ,Q,NP,ZP,PC,ZLEN,NOCC,ODATA)
      NCON=0
      CALL TMATT(XB,NB,ZLEN,TM,NOCC,NCON,ODATA,SFS,1.0,SFA,E,G)
      CALL COMSI(TM,SI)
      NCON=0
      DO 500 K=1,NOTR
      CALL TMATT(XB,NB,ZTR(K),TM,NOCC,NCON,ODATA,SFS,1.0,SFA,E,G)
      CALL MMULT(TM,SI,SX,5,5,1)
      DBASE(K)=SX(1,1)
 500  CONTINUE
      NDONE=1
      NP=0
      NQ=0
      WRITE(6,501)
 501  FORMAT(/)
C-------------
 205  CONTINUE
      WRITE(6,198)CARD,(ZPL(IL),IL=1,4)
 198  FORMAT(1X,A6,4X,3(F6.0,4X),F10.3)
      CALL SWTCH(JRO1,JRO2)
      CALL SWTCH(ICOL1,ICOL2)
      J=JRO1
      I=ICOL1
 225  IF(NODE(I,J)-DEFIN)800,230,800
 230  N=NODET(I,NODE,DEFIN,J,NXC,NONO(J))
C--CHECK IF NODE GIVEN HAS BEEN DEFINED AS A LONGITUDINAL
      DO 250  L=1,NOLO
      IF(J-LROW(L))250,240,250
 240  IF(N-LNOD(L))250,260,250
 250  CONTINUE
      IERR=1
      WRITE(6,252)J,I
 252  FORMAT(1X,19H**ERROR-NODE ON ROW,I4,4H COL,I4,8H HAS NOT,
     1  31H BEEN DEFINED AS A LONGITUDINAL)
         GO TO 800
 260  LOADC(L)=1
      NREC=NREC+1
      DO 265   K=1,NOTR
      GO TO (262,264),NDIR
 262  DX(K)=DBASE(K)*SFL/SFX(L)
      DY(K)=0.
      GO TO 265
 264  DY(K)=DBASE(K)*SFL/SFY(L)
```

```
      DX(K)=0.
  265 CONTINUE
      WRITE(NSCR)J,N,DX,DY
  800 GO TO (810,820),LDIR
  810 J=J+1
      IF(J-JR02)225,225,60
  820 I=I+1
      IF(I-ICOL2)225,225,60
  200 WRITE(6,62)CARD
      WRITE(6,202)
  202 FORMAT(1X,21H**ABOVE CARD IN ERROR)
      IFRR=1
      GO TO 60
 1000 CONTINUE
      WRITE(6,62)CARD
C--SHEAR LOADS AT TRANSVERSES.....WEB LENGTH B AND THICKNESS
      WRITE(6,1021)
 1021 FORMAT(1H1,26HSHEAR LOADS ON TRANSVERSES)
      DO 1023 K=1,NOTR
      DO 1023 NLMEM=1,2
 1023 R(K,NLMEM)=0.0
      A(1)=0.
      A(2)=0.
          DO 1025 J=1,NYC
      A(1)=A(1)+T(J,1)*B(J)
      A(2)=A(2)+T(J,2)*B(J)
 1025 CONTINUE
          NLOO=2
        IF(A(2)) 1091,1092,1091
 1092     NLOO=1
 1091  WRITE (6,1027)
 1027 FORMAT(22X,5HSHELL,6X,3HBHD,/,13X,7HTRANSV.,2X,5HSHEAR,
     *   5X,5HSHEAR)
 1028 READ(5,61    )CARD,ATRAN,RSH,RBH
        WRITE (6,2020) CARD,ATRAN,RSH,RBH
 2020  FORMAT (1X,A6,5E13.5)
      IF(CARD-CEND)1029,1032,1029
 1029 IF(CARD-CSHR)1030,1031,1030
 1030 WRITE(6, 202    )
      IFRR=1
      GO TO 1028
 1031 K=ATRAN
      R(K,1)=RSH
      R(K,2)=RBH
      GO TO 1028
 1032 CONTINUE
      DO 1036  K=1,NOTR
          DO 1035 NLMEM=1,NLOO
      S=-R(K,NLMEM)/A(NLMEM)
      DO 1033 J=1,NYC
 1033 ST(K,J,NLMEM)=S*T(J,NLMEM)*B(J)
 1035 CONTINUE
 1036 CONTINUE
      DO 1038 K=1,NOTR
 1038 DX(K)=0.
      NDIR=2
      SFL=1.0
              DO 1042 NLMEM=1,NLOO
      GO TO (1041,1044),NLMEM
 1041 I=1
      GO TO 1046
 1044 I=IBHD
 1046 CONTINUE
      DO 1040 J=1,NYC
      IF(T(J,NLMEM))1040,1040,1037
 1037 N=NODET(I,NODE,DEFIN,J,NXC,NONO(J))
      DO 720  L=1,NOLO
      IF(J-LROW(L))720,710,720
  710 IF(N-LNOD(L))720,730,720
  720 CONTINUE
      IFRR=1
      WRITE(6,252)J,I
      GO TO 1040
  730 NREC=NREC+1
          LOADC(L)=1
      DO 735 K=1,NOTR
      DY(K)=0.
      DO 734  M=1,NOTR
      DY(K)=DY(K)+AF(K,M)*ST(M,J,NLMEM)/SFY(L)
  734 CONTINUE
  735 CONTINUE
      WRITE(NSCR)J,N,DX,DY
 1040 CONTINUE
 1042 CONTINUE
 1045 END FILE NSCR
      IF(IERR)1048,1048,50
 1048 REWIND NFILE
      WRITE(6,740)
      DO 2000  L=1,NOLO
      DO 1050  K=1,NOTR
```

```
      DXL(K)=0.
      DYL(K)=0.
1050  CONTINUE
      REWIND NSCR
 740  FORMAT(1H1,37HLONGITUDINAL DEFLECTIONS DUE TO LOADS)
      DO 1500  NR=1,NREC
      READ(NSCR)J,N,DX,DY
      IF(J-LROW(L))1500,1100,1500
1100  IF(N-LNOD(L))1500,1200,1500
1200  DO 1250  K=1,NOTR
      DXL(K)=DXL(K)+DX(K)
      DYL(K)=DYL(K)+DY(K)
1250  CONTINUE
      WRITE(6,741)J,N
 741  FORMAT(1X,3HROW,I3,5H NODE,I3)
      WRITE(6,742)
 742  FORMAT(10X,13HX-DEFLECTIONS)
 300  FORMAT(1X,5E12.4)
      WRITE(6,300)(DX(K),K=1,NOTR)
      WRITE(6,743)
 743  FORMAT(10X,13HY-DEFLECTIONS)
      WRITE(6,300)(DY(K),K=1,NOTR)
1500  CONTINUE
      WRITE(NFILE)LROW(L),LNOD(L),LOADC(L),SFX(L),SFY(L),
     +PHI(L),DXL,DYL
2000  CONTINUE
      END FILE NFILE
      REWIND NFILE
9999  RETURN
9998  STOP
      END
CZZZZI   FRS  SETOCS,SETOCS,SETOC
      SUBROUTINE SETOC(NSEC,ZI,EYE,NQ,ZQ,Q,NP,ZP,P,ZLEN,NOCC,ODATA)
C--ROUTINE TO SET UP OCCURRENCE DATA VECTORS
      DIMENSION ZI(20),EYE(20),ZQ(20),Q(20),ZP(50),P(50),ODATA(8,50),
     +ZOCC(70)
C
C--DETERMINE LOCATIONS FOR ALL OCCURRENCE CHANGES
      ZOCC(1)=0.
      ZOCC(2)=ZLEN
      NOCC=2
C--FIRST ARRANGE X-SECTION CHANGES IN ASCENDING ORDER LEFT TO RIGHT
      ZI(NSEC+1)=ZLEN
  10  DO 20 N=1,NSEC
      IF(ZI(N)-ZI(N+1))20,20,30
  20  CONTINUE
      GO TO 40
  30  SAVE=ZI(N)
      ZI(N)=ZI(N+1)
      ZI(N+1)=SAVE
      SAVE=EYE(N)
      EYE(N)=EYE(N+1)
      EYE(N+1)=SAVE
      GO TO 10
  40  DO 50 N=1,NSEC
      NOCC=NOCC+1
  50  ZOCC(NOCC)=ZI(N)
C
      IF(NQ)110,110,60
  60  ZQ(NQ+1)=ZLEN
C--ARRANGE UNIFORM LOADS IN ASCENDING ORDER LEFT TO RIGHT
  70  DO 80 N=1,NQ
      IF(ZQ(N)-ZQ(N+1))80,80,90
  80  CONTINUE
      GO TO 95
  90  SAVE=ZQ(N)
      ZQ(N)=ZQ(N+1)
      ZQ(N+1)=SAVE
      SAVE=Q(N)
      Q(N)=Q(N+1)
      Q(N+1)=SAVE
      GO TO 70
  95  DO 100 N=1,NQ
      NOCC=NOCC+1
 100  ZOCC(NOCC)=ZQ(N)
 110  IF(NP)200,200,120
 120  ZP(NP+1)=ZLEN
C--ARRANGE CONCENTRATED LOADS IN ASCENDING ORDER LEFT TO RIGHT
 130  DO 140  N=1,NP
      IF(ZP(N)-ZP(N+1))140,140,150
 140  CONTINUE
      GO TO 160
 150  SAVE=ZP(N)
      ZP(N)=ZP(N+1)
      ZP(N+1)=SAVE
      SAVE=P(N)
      P(N)=P(N+1)
      P(N+1)=SAVE
      GO TO 130
 160  DO 170 N=1,NP
```

```
      NOCC=NOCC+1
 170  ZOCC(NOCC)=ZP(N)
C--ARRANGE OCCURRENCE LOCATIONS IN ASCENDING ORDER LEFT TO RIGHT
 200  CALL SORT(ZOCC,NOCC)
C
C--INSERT OCCURRENCE DATA
      NOCC=NOCC-1
      DO 500 J=1,NOCC
      ODATA(6,J)=ZOCC(J)
      ODATA(7,J)=ZOCC(J+1)
      ODATA(8,J)=1.
C-- ODATA(8,J) INDICATES IF A CONCENTRATED CONDITION OCCURRS AT THE LEFT
C  AN OCCURRENCE FIELD.  IF SET TO 1, NONE EXISTS.  IF SET TO 2, ONE DOE
      DO 210 I=1,5
 210  ODATA(I,J)=0.
      DO 230 N=1,NSEC
      IF(ZI(N)-ZOCC(J))220,240,250
 220  IF(ZI(N+1)-ZOCC(J))230,230,240
 230  CONTINUE
 240  ODATA(1,J)=EYE(N)
 250  IF(NQ)300,300,260
 260  DO 280 N=1,NQ
      IF(ZQ(N)-ZOCC(J))270,290,300
 270  IF(ZQ(N+1)-ZOCC(J))280,280,290
 280  CONTINUE
 290  ODATA(3,J)=Q(N)
 300  IF(NP)400,400,310
 310  DO 320 N=1,NP
      IF(ZP(N)-ZOCC(J))320,315,400
 315  ODATA(5,J)=P(N)
      ODATA(8,J)=2.
 320  CONTINUE
 400  CONTINUE
 500  CONTINUE
      RETURN
      END
CZZZZI    FR5  OCCMS,OCCMS,OCCM
      SUBROUTINE OCCM(T,PROP,IDF,X,INFLU,SFS,SFL,SFA,E,G)
C--OCCURRENCE MATRIX DEVELOPMENT
      DIMENSION T(5,5),PROP(5),TR(5,5),TC(5,5)
      CALL IDENT(T,5)
      EI=E*PROP(1)*SFS
      GA=G*PROP(2)*SFA
      S=0.
      IF(GA)120,120,110
 110  S=1./GA
 120  T(1,2)=-X
      T(1,3)=-X*X/2./EI
      T(2,3)=+X/EI
      T(1,4)=-X*X*X/6./EI + S*X
      T(2,4)=+X*X/2./EI
      T(3,4)=+X
      IF(INFLU)150,150,999
 150  Q=PROP(3)*SFL
      RQ=PROP(4)
      T(1,5)=-X*X*(Q*(-X*X/24./EI+S/2.)+RQ*X*(-X*X/120./EI+S/6.))
      T(2,5)=-X*X*X*(Q+RQ*X/4.)/6./EI
      T(3,5)=-X*X*(Q+RQ*X/3.)/2.
      T(4,5)=-X*(Q+RQ*X/2.)
      GO TO (999,200),IDF
 200  CALL IDENT(TC,5)
      CF=PROP(5)*SFL
      TC(4,5)=-CF
      CALL MMULT(T,TC,TR,5,5,5)
      CALL EQUAL(T,TR,5,5)
 999  RETURN
      END
CZZZZI    FR5  TMATS,TMATS,TMATT
      SUBROUTINE TMATT(XB,NB,X,TM,NOCC,NCON,ODATA,SFS,SFL,SFA,E,G)
C...ROUTINE TO COMPUTE OCCURRENCE MATRIX FROM LOCATION
C...ZER TO X
      DIMENSION TM(5,5),TO(5,5),TR(5,5),PROP(5),ODATA(8,50)
      INFLU=0
      IF(NCON)110,110,100
 100  IF(X-XB)110,120,120
 110  CALL IDENT(TM,5)
      NB=1
      XB=0.
      NCON=1
 120  NBB=NB
      DO 300 N=NBB,NOCC
      X1=ODATA(6,N)
      X2=ODATA(7,N)
      IDF=ODATA(8,N)
      DO 130 I=1,5
 130  PROP(I)=ODATA(I,N)
C--CHECK IF POINT MATRIX HAS BEEN USED IN 1ST OCCURENCE
      IF(N-NBB)140,140,160
 140  IF(XB-X1)160,160,150
 150  IDF=1
```

```
 160  NR=N
      IF(X-X2)180,170,190
 170  NR=N+1
 180  NSTOP=1
      Y=X-XR
      XR=X
      GO TO 295
 190  NSTOP=0
      Y=X2-XR
      XR=X2
      GO TO 295
 295  CALL OCCM(TO,PROP,IDF,Y,INFLU,SFS,SFL,SFA,E,G)
      CALL MMULT(TO,TM,TR,5,5,5)
      CALL EQUAL(TM,TR,5,5)
      IF(NSTOP)300,300,400
 300  CONTINUE
 400  RETURN
      END
CZZZZI   FR5  MMULTS,MMULTS,MMULT
      SUBROUTINE MMULT(A,B,C,N1,N2,N3)
      DIMENSION A(N1,N2),B(N2,N3),C(N1,N3)
      DO 1 I=1,N1
      DO 1 J=1,N3
      C(I,J)=0.
      DO 1 K=1,N2
   1  C(I,J)=C(I,J)+A(I,K)*B(K,J)
      RETURN
      END
CZZZZI   FR5  IDENTS,IDENTS,IDENT
      SUBROUTINE IDENT(T,N)
      DIMENSION T(N,N)
      DO 5 I=1,N
      DO 4 J=1,N
   4  T(I,J)=0.
   5  T(I,I)=1.
      RETURN
      END
CZZZZI   FR5  EQUALS,EQUALS,EQUAL
      SUBROUTINE EQUAL(T,TR,NR,NC)
      DIMENSION T(NR,NC),TR(NR,NC)
      DO 5 I=1,NR
      DO 5 J=1,NC
   5  T(I,J)=TR(I,J)
      RETURN
      END
CZZZZI   FR5  COMSIS,COMSIS,COMSI
      SUBROUTINE COMSI(TM,SI)
C...ROUTINE TO COMPUTE INITIAL PARAMETERS OF BEAM MEMBER
      DIMENSION TM(5,5),SI(5,1)
      DEL=TM(1,2)*TM(3,4)-TM(3,2)*TM(1,4)
      SI(1,1)=0.
      SI(3,1)=0.
      SI(5,1)=1.
      SI(2,1)=(TM(3,5)*TM(1,4)-TM(1,5)*TM(3,4))/DEL
      SI(4,1)=(TM(3,2)*TM(1,5)-TM(1,2)*TM(3,5))/DEL
      RETURN
      END
CZZZZIE  FR5 MULT,MULT,MULT
       SUBROUTINE  MULT (A,B,C,L,M)
       DIMENSION  A(L,L),B(L,L),C(L,L)
       DO  2  I=1,M
         DO 2 J=1,M
          A(I,J)=0.0
       DO  1  K=1,M
    1  A(I,J)=A(I,J)+B(I,K)*C(K,J)
    2 CONTINUE
      RETURN
      END
CZZZZIE  FR5 SHIP1,SHIP1,SHIP1
      SUBROUTINE SHIP1(NCARD)
C...ROUTINE DEVELOPS FINITE ELEMENT STIFFNESS MATRICES
C...FOR THE TRANSVERSE MEMBER
C     FORMATION OF STIFFNESS MATRICES
      DIMENSION BK(084,084),ALPHAI(4)
      DIMENSION UNITS(4),ND(6),NONO(25),NI(25),IPQ(10),PIQ(10)
      DIMENSION X(25,40),Y(25,40),EI(4),GNUI(4),DC(2,2),
     1SK(6,6), DI(6,6),AI(6,6),AJ(6,6),AK(6,6),AL(6,6),SKAI(6,6),
     2SKAJ(6,6),SKAK(6,6),SKAL(6,6),A1(6,6,4),A2(6,6,4),SKA1(6,6,4),
     3 SKA2(6,6,4)
        DIMENSION NOB,25),MOB(40)
      DIMENSION IM(4),JM(4),ZAI(6),ZAJ(6),ZAK(6),ZAL(6) ,XI(6)
      COMMON  K1,K2  , K3     , K4     , ID     , NORO   , NNI
      COMMON  UNITS  , ND     , NONO   , NI     , IPQ    , PIQ
      COMMON  X      , Y      , Z      , E      , EI     , GNU
      COMMON  GNUI   , MEMNO  , MEMTYP , IEGNU  , IFSF   , IFI
      COMMON  IFJ    , IFK    , IFL    , INI    , JNI    , INJ
      COMMON  JNJ    , INK    , JNK    , INL    , JNL    , PI
      COMMON  P2     , P3     , P4     , P5     , P6     , XJ
      COMMON  YK     , XL     , YL     , DC     , SK     , DI
      COMMON  AI     , AJ     , AK     , AL     , SKAI   , SKAJ
```

```
      COMMON  SKAK    , SKAL    , A1      , A2      , SKA1    , SKA2
      COMMON  IZ      , NC      , XK      , NOMEM   , ICOUNT  , BK
      COMMON  IM      , JM      , NA1     , NA2     , ZAI     , ZAJ
      COMMON  ZAK     , ZAL     , ITEMP   , ALPHA   , XI
C      IPQ(I) = TEMPERATURE FLAG, ITEMP
C      PIQ(I) IS USED TO PASS THE TIME OF DAY
      READ(NCARD,100         )ID,NORO,NN1, NOMAT,ITEMP,(UNITS(I),I=1,4)
         IF (NORO.GT.25)  GO TO 331
          IF (NOMAT.GT.4) GO TO 331
  100 FORMAT(5I5,4A6)
       K1=1
       K3=3
       K4=4
         READ(NCARD,110         ) NOBO,MCON,NOBB,MCOM
         IF (NORO.GT.NORO) GO TO 331
          IF (NOBB.GT.1)  GO TO 331
          IF (MCOM.GT.1) GO TO 331
C       NOBO=NUMBER OF ROWS WITH BOUNDARY CONDITIONS
C      NOB(I) IMPLIES BOUNDARY CONDITION AT THE END OF NOB(I)TH ROW
C       NOBB=1,FIXED IN X-DIRECTION,NOBB=0,FIXED IN Y DIRECTION
C       MCON=NUMBER OF BOUNDARY CONDITION AT 1ST ROW
C       MOB(I) IMPLIES BOUNDARY CONDITION AT MOB(I) TH NODE
C       MCOM SAME AS NOBB BUT FOR BOUNDARY CONDITIONS OF 1ST ROW
         READ(NCARD,110         ) (NOB(I),I=1,NOBO)
         READ(NCARD,110         ) (MOB(I),I=1,MCON)
  110  FORMAT(20I3)
       READ(NCARD,103         ) FA
      REWIND K1
      REWIND K3
      REWIND K4
      IPQ(1)=ITEMP
      WRITE (6,102)ID
  102 FORMAT (24H1DATA FOR PROBLEM NUMBER,I6)
       ND(1)=3
       ND(2)=5
       ND(3)=6
       ND(4)=6
       ND(5)=1
      WRITE(6,100)ID,NORO,NN1,NOMAT,ITEMP,(UNITS(I),I=1,4)
      WRITE(6,110)NOBO,MCON,NOBB,MCOM
      WRITE(6,110)(NOB(I),I=1,NOBO)
      WRITE(6,110)(MOB(I),I=1,MCON)
      WRITE(6,103)FA
      READ(NCARD,103)E1(1),GNU1(1),ALPHA1(1)
  103 FORMAT(E10.2,F7.2,E10.2)
        GO TO 333
  331  WRITE (6,332)
  332  FORMAT (//25H INPUT ERRORS IN SHIP1      //)
         STOP
  333   CALL NODEIN(NCARD)
      NOMEM=0
      ICOUNT=1
      N1(1)=0
      ICOL1=NN1*NONO(1)
      ICOL2=NN1*NONO(2)
      DO 9 I=1,ICOL1
      DO 9 J=I,ICOL1
    9 BK(I,J)=0.0
   11 CALL MEMBER(NCARD)
      IEGNU=IEGNU
      MEMTYP=MEMTYP
      INI=INI
      JNI=JNI
    6 IF(INI-ICOUNT) 12,12,13
   12 IF(MEMTYP) 13,13,26
   26 IMAT=IEGNU+1
      E=E1(IMAT)
      GNU=GNU1(IMAT)
      ALPHA=ALPHA1(IMAT)
      IZ=ND(MEMTYP)
      DO 20 K=1,4
      IM(K)=0
      JM(K)=0
      DO 20 I=1,IZ
      DO 20 J=1,NN1
      A1(I,J,K)=0.0
      A2(I,J,K)=0.0
      SKA1(I,J,K)=0.0
   20 SKA2(I,J,K)=0.0
      NA1=0
      NA2=0
      GO TO (1,2,5,5,5),MEMTYP
    1 CALL MEM1
      GO TO 10
    2 CALL MEM2
      GO TO 10
    5 CALL MEM5
C       IFSF = 1        FORCE ONLY
C       IFSF= 2         FORCE AND STRESS
C       IFSF = 3        STRESS ONLY
```

```
C       FOR THERMAL STRESSE PROBLEM IFSF MUST BE GREATER
C       THAN ZERO FOR ALL MEMBERS
   10 IF (IFSF) 22,22,23
   23 WRITE (K4)MEMNO,MEMTYP,INI,JNI,INJ,JNJ,INK,JNK,INL,JNL,IFSF, IFI,I
     1FJ,IFK,IFL,AI,AJ,AK,AL,SKAI,SKAJ,SKAK,SKAL,DI,NC,ZAI,ZAJ,ZAK,ZAL,X
     2I
      N1(INI)=N1(INI)+1
   22 DO 30 I=1,NA1
       IMM=IM(I)
       DO 30 J=1,NA1
       JMM=IM(J)
   30 CALL TRAMPY(A1,I,IZ,NC,SKA1,J,IMM,JMM)
      WRITE (K1)IZ,NC,NA1,NA2,A1,A2,SKA2,IM,JM
      NOMEM=NOMEM+1
      GO TO 11
   13 IRO=INI-1
  400  FORMAT (4E15.4)
       DO 49 IP=1,NOBO
         IF (IRO.NE.NOB(IP)) GO TO 49
        IK=NONO(IRO)*2-NOBB
         BK(IK,IK)=BK(IK,IK)*10000000.*FA
   49   CONTINUE
         IF (IRO.NE.1) GO TO 233
         DO  232 IP=1,MCON
        JK=MOB(IP)*2-MCOM
  232   BK(JK,JK)=BK(JK,JK)*10000000.*FA
  233 WRITE (K3)IRO,ICOL1,ICOL2,((BK(I,J),I=1,ICOL1),J=1,ICOL1)
   17 IF(INI-NORO) 14,14,999
   14 REWINDK1
      DO 24 I=1,ICOL1
      DO 24 J=1,ICOL2
   24 BK(I,J)=0.0
   18 DO 15 II=1,NOMEM
      READ (K1)IZ,NC,NA1,NA2,A1,A2,SKA2,IM,JM
      IF(NA2) 15,15,31
   31 DO 35 I=1,NA1
      IMM=IM(I)
      DO 35 J=1,NA2
      JMM=JM(J)
   35 CALL TRAMPY(A1,I,IZ,NC,SKA2,J,IMM,JMM)
   15 CONTINUE
      WRITE (K3)IRO,ICOL1,ICOL2,((BK(I,J),I=1,ICOL1),J=1,ICOL2)
      REWINDK1
      DO 25 I=1,ICOL2
      DO 25 J=1,ICOL2
   25 BK(I,J)=0.0
   19 DO 16 II=1,NOMEM
      READ (K1)IZ,NC,NA1,NA2,A1,A2,SKA2,IM,JM
      IF ( NA2) 16,16,32
   32 DO 36 I=1,NA2
      IMM=JM(I)
      DO 36 J=1,NA2
      JMM=JM(J)
   36 CALL TRAMPY(A2,I,IZ,NC,SKA2,J,IMM,JMM)
   16 CONTINUE
      REWIND K1
      ICOL1=NN1*NONO(INI)
        IF (INI.EQ.NORO) GO TO 70
      ICOL2=NN1*NONO(INI+1)
           GO TO 71
   70    ICOL2=ICOL1
   71    ICOUNT=INI
      N1(INI)=0
            NOMEM=0
      IF(JNI) 11,11,12
  999 REWIND K3
      REWINDK1
      REWINDK4
      RETURN
      END
CZZZZIE  FR5 NODEIN,NODEIN,NODEIN
      SUBROUTINE NODEIN(NCARD)
      DIMENSION UNITS(4),ND(6),NONO(25),N1(25),IPQ(10),PIQ(10)
      DIMENSION X(25,40),Y(25,40),E1(4),GNU1(4),DC(2,2),
     1SK(6,6), DI(6,6),AI(6,6),AJ(6,6),AK(6,6),AL(6,6),SKAI(6,6),
     2SKAJ(6,6),SKAK(6,6),SKAL(6,6),A1(6,6,4),A2(6,6,4),SKA1(6,6,4),
     3 SKA2(6,6,4)
      COMMON  K1,K2  , K3     , K4     , ID     , NORO   , NN1
      COMMON  UNITS  , ND     , NONO   , N1     , IPQ    , PIQ
      COMMON  X      , Y      , Z      , E      , E1     , GNU
      COMMON  GNU1   , MEMNO  , MEMTYP , IEGNU  , IFSF   , IFI
      COMMON  IFJ    , IFK    , IFL    , INI    , JNI    , INJ
      COMMON  JNJ    , INK    , JNK    , INL    , JNL    , P1
      COMMON  P2     , P3     , P4     , P5     , P6     , XJ
      COMMON  YK     , XL     , YL     , DC     , SK     , DI
      COMMON  AI     , AJ     , AK     , AL     , SKAI   , SKAJ
      COMMON  SKAK   , SKAL   , A1     , A2     , SKA1   , SKA2
      COMMON  IZ     , NC
      WRITE (6,103)
  103 FORMAT(18H1NODE  COORDINATES//14H    ROW   NODE,13X,7HX-COORD,
```

```
     1 13X,7HY-COORD,13X,7HZ-COORD//1H )
      READ(NCARD,100        )(NONO(I),I=1,NORO)
        DO 300 I=1,NORO
      IF (NONO(I).GT.40) GO TO 200
  300  CONTINUE
  100 FORMAT(25I3)
      DO 1 I=1,NORO
      JOE=NONO(I)
      DO 1 J=1,JOE
    1 READ(NCARD,101         )X(I,J),Y(I,J)
  101 FORMAT(3F10.2)
      DO 2 I=1,NORO
      JOE=NONO(I)
      DO 2 J=1,JOE
    2 WRITE (6,102)I,J,X(I,J),Y(I,J)
  102 FORMAT(1H ,I6,I7, 3F20.5)
      GO TO 223
  200  WRITE (6,221)
  221  FORMAT (//25H INPUT ERROR IN NODFIN     //)
         STOP
  223   WRITE (6,104)
  104 FORMAT (13H1MEMBER  DATA//
     150H0   M  M  I  I  I  I  I  I  J  I  J  I  J  I  J,
     211X,2HP1,11X,2HP2,11X,2HP3,11X,2HP4,11X,2HP5,11X,2HP6/
     350H    E  E  E  F  F  F  F  F  N  N  N  N  N  N  N  N/
     450H    M  M  G  S  I  J  K  L  I  I  J  J  K  K  L  L/
     514H    N  T  N  F/
     611H    O  Y  U/
     7 AH       P)
      RETURN
      END
CZZZZIE  FR5 MEMBER,MEMBER,MEMBER
      SUBROUTINE MEMBER(NCARD)
      DIMENSION UNITS(4),ND(6),NONO(25),N1(25),IPQ(10),PIQ(10)
      DIMENSION X(25,40),Y(25,40),E1(4),GNU1(4),DC(2,2),
     1SK(6,6), DI(6,6),AI(6,6),AJ(6,6),AK(6,6),AL(6,6),SKAI(6,6),
     2SKAJ(6,6),SKAK(6,6),SKAL(6,6),A1(6,6,4),A2(6,6,4),SKA1(6,6,4),
     3 SKA2(6,6,4)
      COMMON  K1,K2   , K3     , K4     , ID     , NORO   , NN1
      COMMON  UNITS   , ND     , NONO   , N1     , IPQ    , PIQ
      COMMON  X       , Y      , Z      , E      , E1     , GNU
      COMMON  GNU1    , MEMNO  , MEMTYP , IEGNU  , IFSF   , IFI
      COMMON  IFJ     , IFK    , IFL    , INI    , JNI    , INJ
      COMMON  JNJ     , INK    , JNK    , INL    , JNL    , P1
      COMMON  P2      , P3     , P4     , P5     , P6     , XJ
      COMMON  YK      , XL     , YL     , DC     , SK     , DI
      COMMON  AI      , AJ     , AK     , AL     , SKAI   , SKAJ
      COMMON  SKAK    , SKAL   , A1     , A2     , SKA1   , SKA2
      COMMON  IZ      , NC
      READ(NCARD,100         )MEMNO,MEMTYP,IEGNU,IFSF,IFI,IFJ,IFK,IFL,
     1INI,JNI,INJ,JNJ,INK,JNK,INL,JNL,P1,P2,P3,P4,P5,P6
  100 FORMAT(I3,7I1,8I2,6E15.7)
      WRITE (6,101)MEMNO,MEMTYP,IEGNU,IFSF,IFI,IFJ,IFK,IFL, INI,JNI,INJ,
     1JNJ,INK,JNK,INL,JNL,P1,P2,P3,P4,P5,P6
  101 FORMAT (1H ,I4,15I3,6F13.5)
      RETURN
      END
CZZZZIE  FR5 TRANS,TRANS,TRANS
        SUBROUTINE TRANS(A,B,M,N)
       DIMENSION A(M,M),B(M,M)
       DO 1 I=1,N
       DO 1 J=1,N
    1  A(I,J)=B(J,I)
       RETURN
       END
CZZZZIE  FR5 EIGEN,EIGEN,EIGEN
       SUBROUTINE EIGEN(A,B,VALU,N,L,M)
C...ROUTINE COMPUTES EIGENVALUES FROM A MATRIX OF INFLUENCE
C...COEFFICENTS AND ALSO GENERATES THE EIGENVECTORS
      DIMENSION A(L,L),B(L,L),VALU(50),DIAG(50),SUPERD(49),
     1 Q(49),VALL(50),S(49),C(49),D(50),IND(50)
       DIMENSION U(20)
C     CALCULATE NORM OF MATRIX
    3 ANORM2=0.0
    4 DO 6 I=1,N
    5 DO 6 J=1,N
    6 ANORM2=ANORM2+A(I,J)**2
    7 ANORM=SQRT(ANORM2)
C     GENERATE IDENTITY MATRIX
    9 IF (M) 10, 45, 10
   10 DO 40 I=1,N
   12 DO 40 J=1,N
   20 IF(I-J) 35, 25, 35
   25 B(I,J)=1.0
   30 GO TO 40
   35 B(I,J)=0.0
   40 CONTINUE
C     PERFORM ROTATIONS TO REDUCE MATRIX TO JACOBI FORM
   45 IEXIT=1
   50 NN=N-2
```

```
   52 IF (NN) 890, 170, 55
   55 DO 160 I=1,NN
   60 II=I+2
   65 DO 160 J=II,N
   70 T1=A(I,I+1)
   75 T2=A(I,J)
      IF(T2)910,160,910
  910 T=SQRT(T1**2+T2**2)
      COS=T1/T
      SIN=T2/T
   90 DO 105 K=I,N
   95 T2=COS*A(K,I+1)+SIN*A(K,J)
  100 A(K,J)=COS*A(K,J)-SIN*A(K,I+1)
  105 A(K,I+1)=T2
  110 DO 125 K=I,N
  115 T2=COS*A(I+1,K)+SIN*A(J,K)
  120 A(J,K)=COS*A(J,K)-SIN*A(I+1,K)
  125 A(I+1,K)=T2
  128 IF (M) 130, 160, 130
  130 DO 150 K=1,N
  135 T2=COS*B(K,I+1)+SIN*B(K,J)
  140 B(K,J)=COS*B(K,J)-SIN*B(K,I+1)
  150 B(K,I+1)=T2
  160 CONTINUE
C     MOVE JACORI FORM ELEMENTS AND INITIALIZE EIGENVALUE BOUNDS
  170 DO 200 I=1,N
  180 DIAG(I)=A(I,I)
  190 VALU(I)=ANORM
  200 VALL(I)=-ANORM
  210 DO 230 I=2,N
  220 SUPERD(I-1)=A(I-1,I)
  230 Q(I-1)=(SUPERD(I-1))**2
C     DETERMINE SIGNS OF PRINCIPAL MINORS
  235 TAU=0.0
  240 I=1
  260 MATCH=0
  270 T2=0.0
  275 T1=1.0
  277 DO 450 J=1,N
  280 P=DIAG(J)-TAU
  290 IF(T2) 300, 330, 300
  300 IF(T1) 310, 370, 310
  310 T=P*T1-Q(J-1)*T2
  320 GO TO 410
  330 IF(T1) 335, 350, 350
  335 T1=-1.0
  340 T=-P
  345 GO TO 410
  350 T1=1.0
  355 T=P
  360 GO TO 410
  370 IF(Q(J-1)) 380, 350, 380
  380 IF(T2) 400, 390, 390
  390 T=-1.0
  395 GO TO 410
  400 T=1.0
C     COUNT AGREEMENTS IN SIGN
  410 IF(T1) 425, 420, 420
  420 IF(T) 440, 430, 430
  425 IF(T) 430, 440, 440
  430 MATCH=MATCH+1
  440 T2=T1
  450 T1=T
C     ESTABLISH TIGHTER BOUNDS ON EIGENVALUES
  460 DO 530 K=1,N
  465 IF (K-MATCH) 470, 470, 520
  470 IF(TAU-VALL(K)) 530, 530, 480
  480 VALL(K)=TAU
  490 GO TO 530
  520 IF(TAU-VALU(K)) 525, 530, 530
  525 VALU(K)=TAU
  530 CONTINUE
  540 IF(VALU(I)-VALL(I)-5.0E-8) 570, 570, 550
  550 IF(VALU(I)) 560, 580, 560
  560 IF(ABS(VALL(I)/VALU(I)-1.0)-5.0E-8) 570, 570, 580
  570 I=I+1
  575 IF(I-N) 540, 540, 590
  580 TAU=(VALL(I)+VALU(I))/2.0
  585 GO TO 260
C     JACOBI EIGENVECTORS BY ROTATIONAL TRIANGULARIZATION
  590 IF (M) 593, 890, 593
  593 IFXIT=2
  595 DO 610 I=1,N
  600 DO 610 J=1,N
  610 A(I,J)=0.0
  615 DO 850 I=1,N
  620 IF (I-1) 625, 625, 621
  621 IF (VALU(I-1)-VALU(I)-5.0E-7) 730, 730, 622
  622 IF (VALU(I-1)) 623, 625, 623
  623 IF (ABS(VALU(I)/VALU(I-1)-1.0)-5.0E-7) 730, 730, 625
```

```
  625 COS=1.0
  628 SIN=0.0
  630 DO 700 J=1,N
  635 IF(J-1) 680, 680, 640
 640  T=SQRT(T1**2+T2**2)
      COS=T1/T
      SIN=T2/T
  650 S(J-1)=SIN
  660 C(J-1)=COS
  670 D(J-1)=T1*COS+T2*SIN
  680 T1=(DIAG(J)-VALU(I))*COS-BETA*SIN
  690 T2=SUPERD(J)
  700 BETA=SUPERD(J)*COS
  710 D(N)=T1
  720 DO 725 J=1,N
  725 IND(J)=0
  730 SMALLD=ANORM
  735 DO 780 J=1,N
  740 IF (IND(J)-1) 750, 780, 780
  750 IF (ABS(SMALLD)-ABS(D(J)))780, 780, 760
  760 SMALLD=D(J)
  770 NN=J
  780 CONTINUE
  790 IND(NN)=1
  800  PRODS=1.0
  805 IF (NN-1) 810, 850, 810
  810 DO 840 K=2,NN
  820 II=NN+1-K
  830 A(II+1,I)=C(II)*PRODS
  840 PRODS=-PRODS*S(II)
  850 A(1,I)=PRODS
C     FORM MATRIX PRODUCT OF ROTATION MATRIX WITH JACOBI VECTOR MATRIX
  855 DO 885 J=1,N
  860 DO 865 K=1,N
  865 U(K)=A(K,J)
  870 DO 885 I=1,N
  875 A(I,J)=0.0
  880 DO 885 K=1,N
  885 A(I,J)=B(I,K)*U(K)+A(I,J)
  890 RETURN
      END
       SUBROUTINE MATINS(A,NR,N1,B,NC,M1,DETERM,ID,INDEX)
CZZZZIE  FR5 MATINS,MATINS,MATINS
C        PIVOT METHOD                                                    M
C         MATRIX  INVERSION WITH ACCOMPANYING SOLUTION OF MAP4
C     PIVOT METHOD
C     FORTRAN IV SINGLE PRECISION WITH ADJUSTABLE DIMENSION
C           DIMENSION A(NR,NR), B(NR,NC), INDEX(NR,3)
C          N      IS THE ORDER OF A
C          M      IS THE NUMBER OF COLUMN VECTORS IN B(MAY BE 0)
C         DETERM WILL CONTAIN DETERMINANT ON EXIT
C          ID     WILL BE SET BY ROUTINE TO 2 IF MATRIX A IS SINGULAR
C                 1 IF INVERSION WAS SUCCESSFUL
C          A      THE INPUT MATRIX WILL BE REPLACED BY A INVERSEE
C          B      THE COLUMN VECTORS WILL BE REPLACED BY CORRESPONDING
C                 SOLUTION VECTORS
C          INDEX WORKING STORAGE ARRAY
C         IF IT IS DESIRED TO SCALE THE DETERMINANT CARD      MAY BE
C         DELETED AND DETERM PRESET BEFORE ENTERING THE ROUTINE
C
      EQUIVALENCE (IROW,JROW), (ICOLUM,JCOLUM), (AMAX, T, SWAP)
                          DIMENSION  A(NR,NR),  B(NR,NC),  INDEX(NR,3)
C
C      INITIALIZATION
C
       N=N1
      M=M1
       DETERM= 1.0E-08
       DO 20 J=1,N
   20 INDEX(J,3) = 0
       DO 550 I=1,N
C
C     SEARCH FOR PIVOT ELEMENT
C
      AMAX = 0.0
      DO 105 J=1,N
      IF(INDEX(J,3)-1) 60, 105, 60
   60 DO 100 K=1,N
      IF(INDEX(K,3)-1) 80, 100, 715
   80 IF (     AMAX -ABS (A(J,K))) 85, 100, 100
   85 IROW=J
      ICOLUM =K
      AMAX = ABS (A(J,K))
  100 CONTINUE
  105 CONTINUE
      INDEX(ICOLUM,3) = INDEX(ICOLUM,3) +1
      INDEX(I,1)=IROW
      INDEX(I,2)=ICOLUM
C
C     INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL
C
```

```
      IF (IROW-ICOLUM) 140, 310, 140
  140 DETERM=-DETERM
      DO 200 L=1,N
      SWAP=A(IROW,L)
      A(IROW,L)=A(ICOLUM,L)
  200 A(ICOLUM,L)=SWAP
      IF(M) 310, 310, 210
  210 DO 250 L=1, M
      SWAP=B(IROW,L)
      B(IROW,L)=B(ICOLUM,L)
  250 B(ICOLUM,L)=SWAP
C
C     DIVIDE PIVOT ROW BY PIVOT ELEMENT
C
  310 PIVOT   =A(ICOLUM,ICOLUM)
      DETERM=DETERM*PIVOT
  330 A(ICOLUM,ICOLUM)=1.0
      DO 350 L=1,N
  350 A(ICOLUM,L)=A(ICOLUM,L)/PIVOT
      IF(M) 380, 380, 360
  360 DO 370 L=1,M
  370 B(ICOLUM,L)=B(ICOLUM,L)/PIVOT
C
C     REDUCE NON-PIVOT ROWS
C
  380 DO 550 L1=1,N
      IF(L1-ICOLUM) 400, 550, 400
  400 T=A(L1,ICOLUM)
      A(L1,ICOLUM)=0.0
      DO 450 L=1,N
  450 A(L1,L)=A(L1,L)-A(ICOLUM,L)*T
      IF(M) 550, 550, 460
  460 DO 500 L=1,M
  500 B(L1,L)=B(L1,L)-B(ICOLUM,L)*T
  550 CONTINUE
C
C     INTERCHANGE COLUMNS
C
      DO 710 I=1,N
      L=N+1-I
      IF (INDEX(L,1)-INDEX(L,2)) 630, 710, 630
  630 JROW=INDEX(L,1)
      JCOLUM=INDEX(L,2)
      DO 705 K=1,N
      SWAP=A(K,JROW)
      A(K,JROW)=A(K,JCOLUM)
      A(K,JCOLUM)=SWAP
  705 CONTINUE
  710 CONTINUE
      DO 730 K = 1,N
      IF(INDEX(K,3) -1) 715,720,715
  720  CONTINUE
  730  CONTINUE
       ID = 1
  810  RETURN
  715  ID = 2
       GO TO 810
       END
CZZZZIE  FRS TEMPCO,TEMPCO,TEMPCO
       SUBROUTINE TEMPCO(      NC,IZ,SKA,XI,ZA)
      DIMENSION SKA(6,6),XI(6),ZA(6)
      DO 2 I=1,NC
      ZA(I)=0.0
      DO 2 J=1,IZ
    2 ZA(I)=ZA(I)+SKA(J,I)*XI(J)
      RETURN
      END
CZZZZIE  FRS MULTRD,MULTRD,MULTRD
      SUBROUTINE MULTRD(AA,IN,JN,SKA)
C     PREMULTIPLIES AA BY SK THEN READS AA INTO A1 OR A2 AND
C     SKA INTO SKA1 OR SKA2
      DIMENSION UNITS(4),ND(6),NONO(25),N1(25),IPQ(10),PIQ(10)
      DIMENSION X(25,40),Y(25,40),E1(4),GNU1(4),DC(2,2),
     1SK(6,6), DI(6,6),AI(6,6),AJ(6,6),AK(6,6),AL(6,6),SKAI(6,6),
     2SKAJ(6,6),SKAK(6,6),SKAL(6,6),A1(6,6,4),A2(6,6,4),SKA1(6,6,4),
     3 SKA2(6,6,4)
      DIMENSION BK(084,084)
      DIMENSION IM(4),JM(4)
           DIMENSION AA(6,6),SKA(6,6)
      COMMON  K1,K2  , K3     , K4     , ID     , NORO   , NN1
      COMMON  UNITS  , ND     , NONO   , N1     , IPQ    , PIQ
      COMMON  X      , Y      , Z      , E      , E1     , GNU
      COMMON  GNU1   , MEMNO  , MEMTYP , IEGNU  , IFSF   , IFI
      COMMON  IFJ    , IFK    , IFL    , INI    , JNI    , INJ
      COMMON  JNJ    , INK    , JNK    , INL    , JNL    , P1
      COMMON  P2     , P3     , P4     , P5     , P6     , XJ
      COMMON  YK     , XL     , YL     , DC     , SK     , DI
      COMMON  AI     , AJ     , AK     , AL     , SKAI   , SKAJ
      COMMON  SKAK   , SKAL   , A1     , A2     , SKA1   , SKA2
      COMMON  IZ     , NC     , XK     , NOMEM  , ICOUNT , BK
```

```
      COMMON  IM     , JM     , NA1    , NA2
      DO 1 I=1,IZ
           DO 1 J=1,NC
      SKA(I,J)=0.0
      DO 1 K=1,IZ
    1 SKA(I,J)=SKA(I,J)+SK(I,K)*AA(K,J)
        JI=NN1*(JN-1)
      IF(IN-INI) 2,2,3
    2 NA1=NA1+1
      IM(NA1)=JI
      CALL READIN(AA,A1,IZ,NC,NA1)
      CALL READIN(SKA,SKA1,IZ,NC,NA1)
      GO TO 4
    3 NA2=NA2+1
      JM(NA2)=JI
      CALL READIN(AA,A2,IZ,NC,NA2)
      CALL READIN(SKA,SKA2,IZ,NC,NA2)
    4 RETURN
      END
CZZZZIE  FR5 READIN,READIN,READIN
      SUBROUTINE READIN(A,B,IZ,NC,K)
C      READS MATRIX A INTO LAYER K OF MATRIX B
      DIMENSION A(6,6),B(6,6,4)
          DO 1 I=1,IZ
      DO 1 J=1,NC
    1 B(I,J,K)=A(I,J)
      RETURN
      END
CZZZZIE  FR5 TRAMPY,TRAMPY,TRAMPY
      SUBROUTINE TRAMPY(A,KA,IZ,NC,SKA,KB,IMM,JMM)
        DIMENSION A(6,6,4),SKA(6,6,4),BK(84,84),DUMMY(3069)
       COMMON DUMMY,BK
      DO 1 I=1,NC
      DO 1 J=1,NC
      DO 1 K=1,IZ
      IBK=IMM+I
      JBK=JMM+J
    1 BK(IBK,JBK)=BK(IBK,JBK)+A(K,I,KA)*SKA(K,J,KB)
      RETURN
      END
CZZZZIE FR5 DIRCOS,RIDCOS,RIDCOS
      SUBROUTINE DIRCOS
C     DIRECTION COSINE SUBROUTINE FOR PLATE
      DIMENSION UNITS(4),ND(6),NONO(25),N1(25),IPQ(10),PIQ(10)
      DIMENSION X(25,40),Y(25,40),E1(4),GNU1(4),DC(2,2),
     1SK(6,6), DI(6,6),AI(6,6),AJ(6,6),AK(6,6),AL(6,6),SKAI(6,6),
     2SKAJ(6,6),SKAK(6,6),SKAL(6,6),A1(6,6,4),A2(6,6,4),SKA1(6,6,4),
     3 SKA2(6,6,4)
      COMMON  K1,K2  , K3     , K4     , ID     , NORO   , NN1
      COMMON  UNITS  , ND     , NONO   , N1     , IPQ    , PIQ
      COMMON  X      , Y      , Z      , E      , E1     , GNU
      COMMON  GNU1   , MEMNO  , MEMTYP , IEGNU  , IFSF   , IFI
      COMMON  IFJ    , IFK    , IFL    , INI    , JNI    , INJ
      COMMON  JNJ    , INK    , JNK    , INL    , JNL    , PI
      COMMON  P2     , P3     , P4     , P5     , P6     , XJ
      COMMON  YK     , XL     , YL     , DC     , SK     , DI
      COMMON  AI     , AJ     , AK     , AL     , SKAI   , SKAJ
      COMMON  SKAK   , SKAL   , A1     , A2     , SKAI   , SKA2
      COMMON  IZ     , NC     , XK
      INI=INI
      JNI=JNI
      INJ=INJ
      JNJ=JNJ
      INK=INK
      JNK=JNK
      X1=X(INJ,JNJ)-X(INI,JNI)
      X2=Y(INJ,JNJ)-Y(INI,JNI)
      R1=X(INK,JNK)-X(INI,JNI)
      R2=Y(INK,JNK)-Y(INI,JNI)
      XJ=SQRT(X1*X1+X2*X2)
      DC(1,1)=X1/XJ
      DC(1,2)=X2/XJ
        DC(2,1)=-DC(1,2)
        DC(2,2)=DC(1,1)
      XK=R1*DC(1,1)+R2*DC(1,2)
      YK=R1*DC(2,1)+R2*DC(2,2)
      RETURN
      END
CZZZZIE  FR5 MEM1,MEM1,MEM1
      SUBROUTINE MEM1
C     TRIANGULAR PLATE SUBMATRIX SUBROUTINE
      DIMENSION UNITS(4),ND(6),NONO(25),N1(25),IPQ(10),PIQ(10)
      DIMENSION X(25,40),Y(25,40),E1(4),GNU1(4),DC(2,2),
     1SK(6,6), DI(6,6),AI(6,6),AJ(6,6),AK(6,6),AL(6,6),SKAI(6,6),
     2SKAJ(6,6),SKAK(6,6),SKAL(6,6),A1(6,6,4),A2(6,6,4),SKA1(6,6,4),
     3 SKA2(6,6,4)
      DIMENSION BK(084,084),IM(4),JM(4),ZAI(6),ZAJ(6),ZAK(6),ZAL(6),
     1 XI(6)
      COMMON  K1,K2  , K3     , K4     , ID     , NORO   , NN1
      COMMON  UNITS  , ND     , NONO   , N1     , IPQ    , PIQ
```

```
      COMMON  X       , Y       , Z       , E       , E1      , GNU
      COMMON  GNU1    , MEMNO   , MEMTYP  , IEGNU   , IFSF    , IFI
      COMMON  IFJ     , IFK     , IFL     , INI     , JNI     , INJ
      COMMON  JNJ     , INK     , JNK     , INL     , JNL     , P1
      COMMON  P2      , P3      , P4      , P5      , P6      , XJ
      COMMON  YK      , XL      , YL      , DC      , SK      , DI
      COMMON  AI      , AJ      , AK      , AL      , SKAI    , SKAJ
      COMMON  SKAK    , SKAL    , A1      , A2      , SKA1    , SKA2
      COMMON  IZ      , NC      , XK      , NOMEM   , ICOUNT  , BK
      COMMON  IM      , JM      , NA1     , NA2     , ZAI     , ZAJ
      COMMON  ZAK     , ZAL     , ITEMP   , ALPHA   , XI
      CALL DIRCOS
      CA=(1.0-GNU)/2.0
      CB=F*P1/(2.0*(1.0-GNU*GNU)*XJ*YK)
      SK(1,1)=CB*(YK*YK+CA*XK*XK)
      SK(1,2)=-CB*CA*XJ*XK
      SK(1,3)=CB*GNU*XJ*YK
      SK(2,1)=SK(1,2)
      SK(2,2)=CB*CA*XJ*XJ
      SK(2,3)=0.0
      SK(3,1)=SK(1,3)
      SK(3,2)=0.0
      SK(3,3)=CB*XJ*XJ
      CA=YK/XJ
      CB=XK/XJ
      CC=CB-1.0
      DO 9 I = 1,6
      DO 9 J =1,6
       AI(I,J)=0.
      AJ(I,J) = 0.0
    9 AK(I,J) = 0.0
      DO 1 I=1,3
      AI(1,I)=-DC(1,I)
      AJ(1,I)=DC(1,I)
      AK(1,I)=0.0
      AI(2,I)=-DC(1,I)-CA*DC(2,I)
      AJ(2,I)=CA*DC(2,I)
      AK(2,I)=DC(1,I)
      AI(3,I)=CC*DC(2,I)
      AJ(3,I)=-CB*DC(2,I)
    1 AK(3,I)=DC(2,I)
      IF(NN1-3) 4,5,5
    4 NC=NN1
      GO TO 6
    5 NC=3
    6 CALL MULTRD(AI,INI,JNI,SKAI)
      CALL MULTRD(AJ,INJ,JNJ,SKAJ)
      CALL MULTRD(AK,INK,JNK,SKAK)
      IF(IFSF) 3,3,2
    2 CA=E/(1.0-GNU*GNU)
      CB=E/((1.0+GNU)*(2.0*YK))
      DI(1,1)=CA/XJ
      DI(1,2)=0.0
      DI(1,3)=CA*GNU/YK
      DI(2,1)=CA*GNU/XJ
      DI(2,2)=0.0
      DI(2,3)=CA/YK
      DI(3,1)=-CB*XK/XJ
      DI(3,2)=CB
      DI(3,3)=0.0
      DO 7 I=1,3
      DO 7 J=1,3
    7 SKAL(I,J)=SK(I,J)
    3 IF(ITEMP) 8,8,10
   10 XI(1)=XJ*ALPHA
      XI(2)=XK*ALPHA
      XI(3)=YK*ALPHA
      CALL TEMPCO(       NC,IZ,SKAI,XI,ZAI)
      CALL TEMPCO(       NC,IZ,SKAJ,XI,ZAJ)
      CALL TEMPCO(       NC,IZ,SKAK,XI,ZAK)
    8 RETURN
      END
CZZZZIE  FR5 MEM2,MEM2,MEM2
      SUBROUTINE MEM2
C     QUADRILATERAL PLATE SUBMATRIX SUBROUTINE
      DIMENSION F(6,6),INDEX(6,3)
      DIMENSION UNITS(4),ND(6),NONO(25),N1(25),IPQ(10),PIQ(10)
      DIMENSION X(25,40),Y(25,40),E1(4),GNU1(4),DC(2,2),
     1SK(6,6), DI(6,6),AI(6,6),AJ(6,6),AK(6,6),AL(6,6),SKAI(6,6),
     2SKAJ(6,6),SKAK(6,6),SKAL(6,6),A1(6,6,4),A2(6,6,4),SKA1(6,6,4),
     3 SKA2(6,6,4)
      DIMENSION BK(084,084),IM(4),JM(4),ZAI(6),ZAJ(6),ZAK(6),ZAL(6),
     1 XI(6)
      COMMON  K1,K2   , K3      , K4      , ID      , NORO    , NN1
      COMMON  UNITS   , ND      , NONO    , N1      , IPQ     , PIQ
      COMMON  X       , Y       , Z       , E       , E1      , GNU
      COMMON  GNU1    , MEMNO   , MEMTYP  , IEGNU   , IFSF    , IFI
      COMMON  IFJ     , IFK     , IFL     , INI     , JNI     , INJ
      COMMON  JNJ     , INK     , JNK     , INL     , JNL     , P1
      COMMON  P2      , P3      , P4      , P5      , P6      , XJ
      COMMON  YK      , XL      , YL      , DC      , SK      , DI
```

```
      COMMON  AI      , AJ      , AK      , AL      , SKAI    , SKAJ
      COMMON  SKAK    , SKAL    , A1      , A2      , SKA1    , SKA2
      COMMON  IZ      , NC      , XK      , NOMEM   , ICOUNT  , BK
      COMMON  IM      , JM      , NA1     , NA2     , ZAI     , ZAJ
      COMMON  ZAK     , ZAL     , ITEMP   , ALPHA   , XI
      CALL DIRCOS
      INI=INI
      JNI=JNI
      INL=INL
      JNL=JNL
      XXL=X(INL,JNL)-X(INI,JNI)
      YYL=Y(INL,JNL)-Y(INI,JNI)
      XL=XXL*DC(1,1)+YYL*DC(1,2)
      YL=XXL*DC(2,1)+YYL*DC(2,2)
      F(1,1)=XJ
      F(1,2)=0.0
      F(1,3)=-GNU*XJ
      F(1,4)=.5*XJ*F(1,3)
      F(1,5)=0.0
      F(2,1)=XK
      F(2,2)=XK*YK-.5*XJ*YK
      F(2,3)=-GNU*XK
      F(2,4)=-.5*(GNU*XK*XK+YK*YK)
      F(2,5)=2.0*(1.0+GNU)*YK
      F(3,1)=-GNU*YK
      F(3,2)=.5*(XK*XJ-XK*XK-GNU*YK*YK)
      F(3,3)=YK
      F(3,4)=XK*YK
      F(3,5)=0.0
      F(4,1)=XL
      F(4,2)=XL*YL-.5*XJ*YL
      F(4,3)=-GNU*XL
      F(4,4)=-.5*(GNU*XL*XL+YL*YL)
      F(4,5)=2.0*(1.0+GNU)*YL
      F(5,1)=-GNU*YL
      F(5,2)=.5*(XL*XJ-XL*XL-GNU*YL*YL)
      F(5,3)=YL
      F(5,4)=XL*YL
      F(5,5)=0.0
      CALL MATINS(F,6,5,DI,6,0,DD,M,INDEX)
      IF(M-1) 13,13,12
   12 WRITE (6,100)MEMNO
  100 FORMAT(31H SOMETHING WRONG WITH MEMBER   ,I5,12H  TOUGH LUCK)
      STOP
   13 BA=XL
      HA=YL
      BB=XL-XJ
      HB=YL
      BC=XL-XK
      HC=YL-YK
      BD=XK
      HD=YL-YK
      BE=XK
      HE=YK
      XBA=.5*BA
      YBA=.5*HA
      XBB=XL-BB/3.0
      YBB=HB/3.0
      XBC=XK+BC/3.0
      YBC=YL-HC/3.0
      XBD=.5*XK
      YBD=YK+.5*HD
      XBE=BE/3.0
      YBE=.666666666*HE
      AA=BA*HA
      AB=.5*BB*HB
      AC=.5*BC*HC
      AD=BD*HD
      AE=.5*BE*HE
      A=AA-AB-AC-AD-AE
      XM=AA*YBA-AB*YBB-AC*YBC-AD*YBD-AE*YBE
      YM=AA*XBA-AB*XBB-AC*XBC-AD*XBD-AE*XBE
      XIA=AA*HA*HA/3.0
      XIB=AB*(HB*HB/18.0+YBB*YBB)
      XIC=AC*(HC*HC/18.0+YBC*YBC)
      XID=AD*(HD*HD/12.0+YBD*YBD)
      XIE=AE*(HE*HE/18.0+YBE*YBE)
      XI=XIA-XIB-XIC-XID-XIE
      YIA=AA*BA*BA/3.0
      YIB=AB*(BB*BB/18.0+XBB*XBB)
      YIC=AC*(BC*BC/18.0+XBC*XBC)
      YID=AD*(BD*BD/12.0+XBD*XBD)
      YIE=AE*(BE*BE/18.0+XBE*XBE)
      YI=YIA-YIB-YIC-YID-YIE
      AJ(1,1)=AA*XBA*YBA
      BJ=AB*(AB/18.0+XBB*YBB)
      CJ=AC*(AC/18.0+XBC*YBC)
      DJ=AD*XBD*YBD
      EJ=AE*(AE/18.0+XBE*YBE)
      XYJ=AJ(1,1)-BJ-CJ-DJ-EJ
```

```
      CA=E*P1
      CB=-CA*GNU
      DI(1,1)=CA*A
      DI(1,2)=CA*XM
      DI(1,3)=CB*A
      DI(1,4)=CB*YM
      DI(1,5)=0.0
      DI(2,2)=CA*XQ
      DI(2,3)=CB*XM
      DI(2,4)=CB*XYJ
      DI(2,5)=0.0
      DI(3,3)=DI(1,1)
      DI(3,4)=CA*YM
      DI(3,5)=0.0
      DI(4,4)=CA*YI
      DI(4,5)=0.0
      DI(5,5)=CA*2.0*(1.0+GNU)*A
      DO 3 I=2,5
      JOE=I-1
      DO 3 J=1,JOE
    3 DI(I,J)=DI(J,I)
      DO 9 I=1,5
      DO 9 J=1,5
      AI(I,J)=0.0
      DO 9 K=1,5
    9 AI(I,J)=AI(I,J)+DI(I,K)*F(K,J)
      DO 10 I=1,5
      DO 10 J=1,5
      SK(I,J)=0.0
      DO 10 K=1,5
   10 SK(I,J)=SK(I,J)+F(K,I)*AI(K,J)
      CA=YK/XJ
      CB=XK/XJ-1.0
      CC=YL/XJ
      CD=XL/XJ-1.0
      CE=-XK/XJ
      CF=-XL/XJ
      IF(NN1-3) 14,15,15
   14 NC=NN1
      GO TO 16
   15 NC=3
   16 DO 17 I = 1,6
      DO 17 J = 1,6
      AI(I,J) = 0.0
      AJ(I,J) = 0.0
      AK(I,J) = 0.0
   17 AL(I,J) = 0.0
      DO 5 J=1,NC
      AI(1,J)=-DC(1,J)
      AI(2,J)=-DC(1,J)-CA*DC(2,J)
      AI(3,J)=CB*DC(2,J)
      AI(4,J)=-DC(1,J)-CC*DC(2,J)
      AI(5,J)=CD*DC(2,J)
      AJ(1,J)=DC(1,J)
      AJ(2,J)=CA*DC(2,J)
      AJ(3,J)=CE*DC(2,J)
      AJ(4,J)=CC*DC(2,J)
      AJ(5,J)=CF*DC(2,J)
      AK(2,J)=DC(1,J)
      AK(3,J)=DC(2,J)
      AL(4,J)=DC(1,J)
    5 AL(5,J)=DC(2,J)
      CALL MULTRD(AI,INI,JNI,SKAI)
      CALL MULTRD(AJ,INJ,JNJ,SKAJ)
      CALL MULTRD(AK,INK,JNK,SKAK)
      CALL MULTRD(AL,INL,JNL,SKAL)
      IF(ITEMP) 19,19,20
   20 XI(1)=XJ*ALPHA
      XI(2)=XK*ALPHA
      XI(3)=YK*ALPHA
      XI(4)=XL*ALPHA
      XI(5)=YL*ALPHA
      CALL TEMPCO(      NC,IZ,SKAI,XI,ZAI)
      CALL TEMPCO(      NC,IZ,SKAJ,XI,ZAJ)
      CALL TEMPCO(      NC,IZ,SKAK,XI,ZAK)
      CALL TEMPCO(      NC,IZ,SKAL,XI,ZAL)
   19 CONTINUE
      IF(IFSF) 8,8,7
    7 DO 6 I=1,5
      DO 6 J=1,5
      SKAL(I,J)=SK(I,J)
    6 DI(I,J)=E*F(I,J)
    8 RETURN
      END
CZZZZIE  FR5 MEM5,MEM5,MEM5
      SUBROUTINE MEM5
C     PIN-ENDED BAR SUBMATRIX SUBROUTINE  P1=BAR CROSS SECTION AREA
      DIMENSION UNITS(4),ND(6),NONO(25),N1(25),IPQ(10),PIQ(10)
      DIMENSION X(25,40),Y(25,40),E1(4),GNU1(4),DC(2,2),
     1SK(6,6), DI(6,6),AI(6,6),AJ(6,6),AK(6,6),AL(6,6),SKAI(6,6),
```

```
     2SKAJ(6,6),SKAK(6,6),SKAL(6,6),A1(6,6,4),A2(6,6,4),SKA1(6,6,4),
     3 SKA2(6,6,4)
      DIMENSION BK(084,084),IM(4),JM(4),ZAI(6),ZAJ(6),ZAK(6),ZAL(6),
     1 XI(6)
      COMMON  K1,K2  , K3      , K4      , ID      , NORO    , NNI
      COMMON  UNITS  , ND      , NONO    , N1      , IPQ     , PIQ
      COMMON  X      , Y       , Z       , E       , E1      , GNU
      COMMON  GNU1   , MEMNO   , MEMTYP  , IEGNU   , IFSF    , IFI
      COMMON  IFJ    , IFK     , IFL     , INI     , JNI     , INJ
      COMMON  JNJ    , INK     , JNK     , INL     , JNL     , PI
      COMMON  P2     , P3      , P4      , P5      , P6      , XJ
      COMMON  YK     , XL      , YL      , DC      , SK      , DI
      COMMON  AI     , AJ      , AK      , AL      , SKAI    , SKAJ
      COMMON  SKAK   , SKAL    , A1      , A2      , SKA1    , SKA2
      COMMON  IZ     , NC      , XK      , NOMEM   , ICOUNT  , BK
      COMMON  IM     , JM      , NA1     , NA2     , ZAI     , ZAJ
      COMMON  ZAK    , ZAL     , ITEMP   , ALPHA   , XI
       X1=X(INJ,JNJ)-X(INI,JNI)
       X2=Y(INJ,JNJ)-Y(INI,JNI)
       XJ=SQRT(X1*X1+X2*X2)
        DC(1,1)=X1/XJ
        DC(1,2)=X2/XJ
       DC(2,1)=-DC(1,2)
       DC(2,2)=DC(1,1)
    3 SKK=P1*E /XJ
      SKAL(1,1)=SKK
      IT=NN1*(JNI-1)
      IJ=NN1*(JNJ-1)
      IF(NN1-3) 4,5,5
    4 NC=NN1
      GO TO 6
    5 NC=3
    6 DO 12 I=1,6
      DO 12 J=1,NN1
      AI (I,J)=0.0
      AJ (I,J)=0.0
      SKAI(I,J)=0.0
   12 SKAJ(I,J)=0.0
      DO 9 JJ=1,NC
      A1(1,JJ,1)=-DC(1,JJ)
      SKA1(1,JJ,1)=-DC(1,JJ)*SKK
      AI(1,JJ)=-DC(1,JJ)
      SKAI(1,JJ)=SKA1(1,JJ,1)
      NA1=1
      IM(1)=II
      IF(INJ-INI) 7,7,8
    7 A1(1,JJ,2)=DC(1,JJ)
      SKA1(1,JJ,2)=SKK*DC(1,JJ)
      NA1=2
      IM(2)=IJ
      AJ(1,JJ)=DC(1,JJ)
      SKAJ(1,JJ)=SKA1(1,JJ,2)
      GO TO 9
    8 A2(1,JJ,1)=DC(1,JJ)
      SKA2(1,JJ,1)=SKK*DC(1,JJ)
      NA2=1
      JM(1)=IJ
      AJ(1,JJ)=DC(1,JJ)
      SKAJ(1,JJ)=SKA2(1,JJ,1)
    9 CONTINUE
      IF(IFSF) 11,11,10
   10 DI(1,1)=P1
   11 IF(ITEMP) 13,13,15
   15 XI(1)=XJ*ALPHA
      CALL TEMPCO(       NC,IZ,SKAI,XI,ZAI)
      CALL TEMPCO(       NC,IZ,SKAJ,XI,ZAJ)
   13 RETURN
      END
CZZZZIE  FR5 SHIP2,SHIP2,SHIP2
         SUBROUTINE SHIP2(SPRING,MOR)
CSHIP2
C     MATRIX  TRIANGULARIZATION
      DIMENSION UNITS(4),ND(6),NONO(25),N1(25),IPQ(10),PIQ(10),
     1 INDEX(084,3)
       DIMENSION BK(84,84),BK2(84,84),BTEMP(84)
      COMMON  K1,K2  , K3      , K4      , ID      , NORO    , NNI
      COMMON  UNITS  , ND      , NONO    , N1      , IPQ     , PIQ
      COMMON  BK2
        COMMON/INFLU/AF(50,50),EIG(50),DY(50),NFILE,NF,LNOD(100)
     1 ,LROW(100),DX(50),MF
       COMMON/SAFE/ BK
        COMMON/SHIP/NOLO,LNO(100),SFX(100),SFY(100),PHI(100)
      REWINDK3
      REWINDK1
      READ(K3) IRO,ICOL1,ICOL2,((BK(I,J),I=1,ICOL1),J=1,ICOL1)
  400  FORMAT (4E15.4)
      DO 35 II=1,NORO
        IF (MOR.EQ.0)  GO TO 79
        DO 74 IK=1,NOLO
         AN=PHI(IK)
```

```
      IF (II.NE.LROW(IK))  GO TO 74
        J1=LNOD(IK)*2-1
        J2=J1+1
       A=COS(AN)*SFX(IK)-SIN(AN)*SFY(IK)
        B=SIN(AN)*SFX(IK)+COS(AN)*SFY(IK)
       BK(J1,J1)=BK(J1,J1)+A*SPRING
       BK(J2,J2)=BK(J2,J2)+B*SPRING
   74  CONTINUE
   79    DD=0.
      CALL MATINS(BK,84,ICOL1,BK2,84,00,DD,M,INDEX)
      GO TO (36,38),M
   38 WRITE (6,111)IRO
  111 FORMAT(1H0,17HSINGULAR IN ROW  ,I3,14H    TOUGH LUCK)
      WRITE (6,121) II,DD
  121 FORMAT (9H ROW NO =, I10,10H  DETERM =, E20.5)
      STOP
   36 WRITE (K1)IRO,ICOL1,ICOL2,((BK(I,J),I=1,ICOL1),J=1,ICOL1)
      IF(II-NORO) 40,39,39
   40 READ (K3)IRO,ICOL1,ICOL2,((BK2(I,J),I=1,ICOL1),J=1,ICOL2)
      WRITE (K1)IRO,ICOL1,ICOL2,((BK2(I,J),I=1,ICOL1),J=1,ICOL2)
      DO 44 J=1,ICOL1
      DO 43 K=1,ICOL2
      BTEMP(K)=0.0
      DO 43 I=1,ICOL1
   43 BTEMP(K)=BTEMP(K)+BK(I,J)*BK2(I,K)
      DO 44 I=1,ICOL2
   44 BK(I,J)=BTEMP(I)
       WRITE(K1)IRO,ICOL1,ICOL2,((BK(I,J),I=1,ICOL2),J=1,ICOL1)
      DO 50 K=1,ICOL2
      DO 51 I=1,ICOL2
      BTEMP(I)=0.0
      DO 51 J=1,ICOL1
   51 BTEMP(I)=BTEMP(I)+BK(I,J)*BK2(J,K)
      DO 50 I=1,ICOL2
   50 BK2(I,K)=BTEMP(I)
       READ(K3)IRO,ICOL1,ICOL2,((BK(I,J),I=1,ICOL1),J=1,ICOL1)
      DO 35 I=1,ICOL1
      DO 35 J=1,ICOL1
   35 BK(I,J)=BK(I,J)-BK2(I,J)
   39 REWINDK1
      REWINDK3
      RETURN
      END
CZZZZIE  FR5 SHIP3,SHIP3,SHIP3
      SUBROUTINE SHIP3
CSHIP3
C     INPUT OF FORCE DATA AND BACK SUBSTITUTION FOR FINAL
C     SOLUTION OF EQUATIONS
      DIMENSION BK(084,084), VTEMP(84,1)
      DIMENSION UNITS(4),ND(6),NONO(25),N1(25),IPQ(10),PIQ(10)
      DIMENSION R(2100)
      COMMON  K1,K2  , K3     , K4     , ID     , NORO   , NNI
      COMMON  UNITS  , ND     , NONO   , N1     , IPQ    , PIQ
      COMMON  NUMFO  , R
        COMMON/SAFE/BK
       WRITE (6,110) K1,K2,K3,K4
  110  FORMAT(10I10)
      REWINDK1
C       CALL FORCIN
        NUMFO=1
      JOE=NORO-1
      KKK=0
  400  FORMAT (4E15.4)
      DO 41 II=1,JOE
      READ (K1)IRO
      READ (K1)IRO
      READ(K1)IRO,ICOL1,ICOL2,((BK(I,J),I=1,ICOL2),J=1,ICOL1)
      DO 42 I=1,ICOL2
      IIK=KKK+ICOL1+I
      DO 42 K=1,NUMFO
      DO 42 J=1,ICOL1
      IKK=KKK+J
   42 R(IIK)=R(IIK)-BK(I,J)*R(IKK)
   41 KKK=KKK+ICOL1
      DO 47 II=1,NORO
      READ(K1)IRO,ICOL1,ICOL2,((BK(I,J),I=1,ICOL1),J=1,ICOL1)
      DO 45 I=1,ICOL1
      DO 45 K=1,NUMFO
      VTEMP(I,K)=0.0
      DO 45 J=1,ICOL1
      IKK=KKK+J
   45 VTEMP(I,K)=VTEMP(I,K)+BK(I,J)*R(IKK)
      DO 46 I=1,ICOL1
      IKK=KKK+I
      DO 46 J=1,NUMFO
   46 R(IKK)=VTEMP(I,J)
      IF (II-NORO)   49,54,54
   49 BACKSPACEK1
      BACKSPACEK1
      BACKSPACEK1
```

```
      READ (K1)IRO,ICOL1,ICOL2,((BK(I,J),I=1,ICOL1),J=1,ICOL2)
      BACKSPACE K1
      BACKSPACEK1
      KKK=KKK-ICOL1
      DO 47 I=1,ICOL1
      IKK=KKK+I
      DO 47 K=1,NUMFO
      DO 47 J=1,ICOL2
   47 R(IKK)=R(IKK)-BK(I,J)*VTEMP(J,K)
   54 REWINDK1
        WRITE (6,400) (R(I),I=1,IKK)
      RETURN
      END
CZZZZIE  FRS SHIP4,SHIP4,SHIP4
      SUBROUTINE SHIP4
      DIMENSION  JACK(25,40),FORCE(25,40,2)
      DIMENSION UNITS(4),ND(6),NONO(25),N1(25),IPQ(10),PIQ(10)
      DIMENSION V(2100),UU(6),QQ(6),AI(6,6),AJ(6,6),AK(6,6),AL(6,6),
     1SKAI(6,6),SKAJ(6,6),SKAK(6,6),SKAL(6,6),DI(6,6)
      DIMENSION VTEE(6,3)
      COMMON  K1,K2  , K3     , K4     , ID     , NORO   , NN1
      COMMON  UNITS  , ND     , NONO   , N1     , IPQ    , PIQ
      COMMON  NUMFO  , V      , KK     , KKK    , III    , IZ
      COMMON  UU     , QQ     , MEMNO  , MEMTYP , INI    , JNI
      COMMON  INJ    , JNJ    , INK    , JNK    , INL    , JNL
      COMMON  IFSF   , IFI    , IFJ    , IFK    , IFL    , AI
      COMMON  AJ     , AK     , AL     , SKAI   , SKAJ   , SKAK
      COMMON  SKAL   , DI     , VTEE
        COMMON/SAFE/JACK,FORCE
      WRITE (6,110)ID
  110 FORMAT(27H1RESULTS FOR PROBLEM NUMBER,I8)
      REWINDK3
      REWINDK1
      DO 60 III=1,NUMFO
      WRITE (6,100)UNITS(1),UNITS(2),UNITS(3),UNITS(4)
  100 FORMAT(20H1MEMBER STRESSES IN ,2A6,14H  PER SQUARE  ,2A6//
     1132H LOAD SYSTEM    ROW   MEMBER TYPE AND NUMBER    X-STRESS
     2Y-STRESS   SHEAR STRESS  1ST PRINC STR 2ND PRINC STR  ANGLE  1ST P
     3RINC/44X,14H(TRIANG PLATE),58X,16HSTRESS TO X-AXIS//48X,
     48HX-STRESS,6X,7HX-GRAD ,7X,8HY-STRESS,6X,7HY-GRAD ,5X,
     512HSHEAR STRESS/46X,12H(QUAD PLATE)//48X,8HX-STRESS/50X,5H(BAR)//)
      REWINDK4
      KKK=NN1*NONO(1)
      KK=0
      DO 2 I=1,NORO
      JOE=NONO(I)
      DO 2 J=1,JOE
      JACK(I,J)=0
      DO 2 K=1,NN1
    2 FORCE(I,J,K)=0.0
      DO 59 II=1,NORO
      JOE=N1(II)
      IF (JOE) 23,23,22
   22 DO 58 JJ=1,JOE
      READ (K4)MEMNO,MEMTYP,INI,JNI,INJ,JNJ,INK,JNK,INL,JNL,IFSF,IFI,IFJ
     1,IFK,IFL,AI,AJ,AK,AL,SKAI,SKAJ,SKAK,SKAL,DI
      IF(IPQ(1)) 31,31,30
   30 READ (K3)VTEE
   31 CONTINUE
      IZ=ND(MEMTYP)
      GO TO (4,5,8,8,8),MEMTYP
    4 CALL MEMB1
      GO TO 20
    5 CALL MEMB2
      GO TO 20
    8 CALL MEMB5
   20  IF (IFSF-2) 10,10,58
   10 IF (IFI) 12,12,11
   11 CALL SR15(AI,QQ,INI,JNI,IZ,NN1)
   12 IF(IFJ) 14,14,13
   13 CALL SR15(AJ,QQ,INJ,JNJ,IZ,NN1)
   14 IF(IFK) 16,16,15
   15 CALL SR15(AK,QQ,INK,JNK,IZ,NN1)
   16 IF(IFL) 58,58,17
   17 CALL SR15(AL,QQ,INL,JNL,IZ,NN1)
   58 CONTINUE
   23 CONTINUE
      KK=KKK
   59 KKK=KKK+NN1*NONO(II+1)
      WRITE (6,101)UNITS(1),UNITS(2),UNITS(1),UNITS(2),UNITS(3),UNITS(4)
  101 FORMAT(33H1CUT NODE FORCES AND MOMENTS IN  ,2A6,7H  AND  ,4A6//
     12AH LOAD SYSTEM    ROW     NODE,9X,7HX-FORCE,7X,7HY-FORCE,7X,
     27HZ-FORCE,9X,5HX-MOM,9X,5HY-MOM,9X,5HZ-MOM///)
      REWINDK4
      REWINDK3
      DO 24 I=1,NORO
      JOE=NONO(I)
      DO 24 J=1,JOE
      IF(JACK(I,J)) 24,24,25
   25 WRITE (6,102)III,I,J,(FORCE(I,J,K),K=1,NN1)
```

```
 102 FORMAT(1H ,3I9,3X,6E14.6)
  24 CONTINUE
  26 IK=1
     IKK=NN1
     WRITE (6,103)UNITS(3),UNITS(4)
 103 FORMAT(23HNODE DISPLACEMENTS IN ,2A6,13H  AND RADIANS//
    127H LOAD SYSTEM   ROW     NODE,10X,7HX-DISP ,7X,7HY-DISP ,7X,
    27HZ-DISP ,4X,10HX-ROTATION,4X,10HY-ROTATION,4X,10HZ-ROTATION//)
     DO 27 I=1,NORO
     IN=NONO(I)
     DO 27 II=1,IN
     WRITE (6,104)III,I,II,(V(IV),IV=IK,IKK)
 104 FORMAT(1H ,I8,2I9,3X,6E14.6)
     IK=IK+NN1
  27 IKK=IKK+NN1
  60 CONTINUE
     RETURN
     END
ZZZZIE  FR5 SR4A,SR4A,SR4A
     SUBROUTINE SR4A(A,M,N,B,C)
     SR4A MATRIX MULTIPLICATION SHIP 4
     DIMENSION A(6,6),B(6),C(6)
     DO 1 I=1,M
     C(I)=0.0
     DO 1 J=1,N
   1 C(I)=C(I)+A(I,J)*B(J)
     RETURN
     END
ZZZZIE FR5 SR14,SR14,SR14
     SUBROUTINE SR14(KK,N,JN,AIJK,QU,III,KKK,IIN,IJK)
     SR14  BRANCH DISPLACEMENTS OR FORCES SHIP 4
     DIMENSION V(2100),AIJK(6,6),QU(6),W(6)
      DIMENSION UNITS(4),ND(6),NONO(25),N1(25),IPQ(10),PIQ(10)
      COMMON K1,K2,K3,K4,ID,NORO,NN1,UNITS,ND,NONO,N1,IPQ,PIQ
      COMMON NUMFO,V
     IF (IJK-IIN) 3,3,4
   3 IQ=KK*NN1*(JN-1)
     GO TO 5
   4 IQ=KKK*NN1*(JN-1)
   5 DO 1 I=1,NN1
     II=IQ+I
   1 W(I)=V(II)
     DO 2 I=1,N
     DO 2 J=1,NN1
   2 QU(I)=QU(I)+AIJK(I,J)*W(J)
     RETURN
     END
ZZZZIE  FR5 SR15,SR15,SR15
     SUBROUTINE SR15(AIJK,QQ,II,JI,N,NN1)
     SR15 NODE FORCES SHIP 4
     DIMENSION AIJK(6,6),QQ(6),FORCE(25,40,2),JACK(25,40)
       COMMON/SAFE/JACK,FORCE
     DO 1 J=1,NN1
     DO 1 I=1,N
   1 FORCE(II,JI,J)=FORCE(II,JI,J)-AIJK(I,J)*QQ(I)
     JACK(II,JI)=1
     RETURN
     END
ZZZZIE  FR5 MEMB1,MEMB1,MEMB1
     SUBROUTINE MEMB1
     BRANCH DISPLACEMENTS AND STRESSES FOR TRIANG PLATE  SHIP 4
     DIMENSION STRESS(6)
     DIMENSION UNITS(4),ND(6),NONO(25),N1(25),IPQ(10),PIQ(10)
     DIMENSION V(2100),UU(6),QQ(6),AI(6,6),AJ(6,6),AK(6,6),AL(6,6),
    1SKAI(6,6),SKAJ(6,6),SKAK(6,6),SKAL(6,6),DI(6,6)
     DIMENSION VTEE(6,3)
     COMMON  K1,K2  , K3     , K4     , ID     , NORO   , NN1
     COMMON  UNITS  , ND     , NONO   , N1     , IPQ    , PIQ
     COMMON  NUMFO  , V      , KK     , KKK    , III    , IZ
     COMMON  UU     , QQ     , MEMNO  , MEMTYP , INI    , JNI
     COMMON  INJ    , JNJ    , INK    , JNK    , INL    , JNL
     COMMON  IFSF   , IFI    , IFJ    , IFK    , IFL    , AI
     COMMON  AJ     , AK     , AL     , SKAI   , SKAJ   , SKAK
     COMMON  SKAL   , DI     , VTEE
     DO 1 I=1,3
   1 UU(I)=0.0
     CALL SR14(KK,3,JNI,AI,UU,III,KKK,INI,INI)
     CALL SR14(KK,3,JNJ,AJ,UU,III,KKK,INI,INJ)
     CALL SR14(KK,3,JNK,AK,UU,III,KKK,INI,INK)
     IF(IPQ(1)) 11,11,9
   9 IT=III
     DO 10 I=1,3
  10 UU(I)=UU(I)-VTEE(I,IT)
  11 CONTINUE
     IF(IFSF-2) 7,2,2
   2 CALL SR4A(DI,3,3,UU,STRESS)
     TA=(STRESS(1)+STRESS(2))/2.0
     TB=(STRESS(1)-STRESS(2))/2.0
     TC=SQRT(TB*TB+STRESS(3)*STRESS(3))
     PA=TA+TC
```

```
      PB=TA-TC
      ANGLE=28.6478*ATAN(STRESS(3)/TB)
      IF(TB) 3,6,6
    3 IF(STRESS(3)) 4,5,5
    4 ANGLE=ANGLE-90.0
      GO TO 6
    5 ANGLE=ANGLE+90.0
    6 WRITE (6,100)III,INI,MEMNO,(STRESS(I),I=1,3),PA,PB,ANGLE
  100 FORMAT(1H0,I9,I10,16H    TRIANG PLATE,I6,4X,5E14.6,F10.4,4H DEG)
      IF(IFSF-3) 7,8,8
    7 CALL SR4A(SKAL,3,3,UU,QQ)
    8 RETURN
      END
CZZZZIE  FR5 MEMB2,MEMB2,MEMB2
      SUBROUTINE MEMB2
C     BRANCH DEFORMATIONS AND STRESSES FOR QUAD PLATE SHIP 4
      DIMENSION STRESS(6)
      DIMENSION UNITS(4),ND(6),NONO(25),NI(25),IPQ(10),PIQ(10)
      DIMENSION V(2100),UU(6),QQ(6),AI(6,6),AJ(6,6),AK(6,6),AL(6,6),
     1SKAI(6,6),SKAJ(6,6),SKAK(6,6),SKAL(6,6),DI(6,6)
      DIMENSION VTEE(6,3)
      COMMON  K1,K2  , K3     , K4     , ID     , NORO   , NN1
      COMMON  UNITS  , ND     , NONO   , NI     , IPQ    , PIQ
      COMMON  NUMFO  , V      , KK     , KKK    , III    , IZ
      COMMON  UU     , QQ     , MEMNO  , MEMTYP , INI    , JNI
      COMMON  INJ    , JNJ    , INK    , JNK    , INL    , JNL
      COMMON  IFSF   , IFI    , IFJ    , IFK    , IFL    , AI
      COMMON  AJ     , AK     , AL     , SKAI   , SKAJ   , SKAK
      COMMON  SKAL   , DI     , VTEE
      DO 1 I=1,5
    1 UU(I)=0.0
      CALL SR14(KK,5,JNI,AI,UU,III,KKK,INI,INI)
      CALL SR14(KK,5,JNJ,AJ,UU,III,KKK,INI,INJ)
      CALL SR14(KK,5,JNK,AK,UU,III,KKK,INI,INK)
      CALL SR14(KK,5,JNL,AL,UU,III,KKK,INI,INL)
      IF(IPQ(1)) 11,11,9
    9 IT=III
      DO 10 I=1,5
   10 UU(I)=UU(I)-VTEE(I,IT)
   11 CONTINUE
      IF(IFSF-2) 3,2,2
    2 CALL SR4A(DI,5,5,UU,STRESS)
      WRITE (6,100)III,INI,MEMNO,(STRESS(I),I=1,5)
  100 FORMAT(1H0,I9,I10,14H    QUAD PLATE,I8,4X,5E14.6)
      IF(IFSF-3) 3,4,4
    3 CALL SR4A(SKAL,5,5,UU,QQ)
    4 RETURN
      END
      SUBROUTINE MEMB5
CCIE  FR5 MEMB5,MEMB5,MEMB5
C     BRANCH FORCES AND STRESS FOR PIN ENDED BAR  SHIP 4
      DIMENSION UNITS(4),ND(6),NONO(25),NI(25),IPQ(10),PIQ(10)
      DIMENSION V(2100),UU(6),QQ(6),AI(6,6),AJ(6,6),AK(6,6),AL(6,6),
     1SKAI(6,6),SKAJ(6,6),SKAK(6,6),SKAL(6,6),DI(6,6)
      DIMENSION VTEE(6,3)
      COMMON  K1,K2  , K3     , K4     , ID     , NORO   , NN1
      COMMON  UNITS  , ND     , NONO   , NI     , IPQ    , PIQ
      COMMON  NUMFO  , V      , KK     , KKK    , III    , IZ
      COMMON  UU     , QQ     , MEMNO  , MEMTYP , INI    , JNI
      COMMON  INJ    , JNJ    , INK    , JNK    , INL    , JNL
      COMMON  IFSF   , IFI    , IFJ    , IFK    , IFL    , AI
      COMMON  AJ     , AK     , AL     , SKAI   , SKAJ   , SKAK
      COMMON  SKAL   , DI     , VTEE
      DO 1 I=1,IZ
    1 UU(I)=0.0
      CALL SR14(KK,IZ,JNI,  AI,UU,III,KKK,INI,INI)
      CALL SR14(KK,IZ,JNJ,  AJ,UU,III,KKK,INI,INJ)
      IF(IPQ(1)) 11,11,9
    9 IT=III
      UU(1)=UU(1)-VTEE(1,IT)
   11 CONTINUE
      QQ(1)=SKAL(1,1)*UU(1)
      IF(IFSF=2) 3,2,2
    2 STRESS=QQ(1)/DI(1,1)
      WRITE (6,100)III,INI,MEMNO,STRESS
  100 FORMAT(1H0,I9,I10,7H    BAR,I15,4X,E14.6)
    3 RETURN
      END
```

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)*

COM/CODE Corporation
Alexandria, Virginia

2a. REPORT SECURITY CLASSIFICATION

Unclassified

2b. GROUP

3 REPORT TITLE

Tanker Transverse Strength Analysis -- Programmer's Manual

4 DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

5. AUTHOR(S) *(First name, middle initial, last name)*

R. Nielson, P. Y. Chang, and L. C. Deschamps

6 REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
July 1972	43	

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
N0024-70-C-5219	
b. PROJECT NO.	
c.	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
d.	SSC-228

10. DISTRIBUTION STATEMENT

Unlimited

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Naval Ship Systems Command

13. ABSTRACT

This report, the last in a sequence of four Ship Structure Committee Reports on a method for performing structural analysis of a tanker hull, contains the Programmer's Manual for the transverse strength analysis portion of the program.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6801

UNCLASSIFIED

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT

GPO 939-061

SHIP RESEARCH COMMITTEE
Maritime Transportation Research Board
National Academy of Sciences-National Research Council

The Ship Research Committee has technical cognizance of the inter-agency Ship Structure Committee's research program:

PROF. R. A. YAGLE, Chairman, *Prof. of Naval Architecture, University of Michigan*

DR. H. N. ABRAMSON, *Director, Dept. of Mech. Sciences, Southwest Research Institute*

MR. W. H. BUCKLEY, *Coordinator of Hydrofoil Struc. Res., Naval Ship R & D Center*

MR. E. L. CRISCUOLO, *Senior Nondestructive Testing Spec., Naval Ordnance Lab.*

DR. W. D. DOTY, *Research Consultant, U. S. Steel Corporation*

PROF. J. E. GOLDBERG, *School of Civil Engineering, Purdue University*

PROF. W. J. HALL, *Prof. of Civil Engineering, Univ. of Illinois*

MR. J. E. HERZ, *Chief Structural Design Engineer, Sun Shipbuilding & Dry Dock Co.*

MR. G. E. KAMPSCHAEFER, JR., *Manager, Application Engineering, ARMCO Steel Corp.*

MR. R. C. STRASSER, *Dir. of Research, Newport News Shipbuilding & Dry Dock Co.*

CAPT R. M. WHITE, USCG, *Chief, Applied Engineering Section, U. S. Coast Guard Academy*

MR. R. W. RUMKE, *Executive Secretary, Ship Research Committee*

Advisory Group II, "Ship Structural Design" prepared the project prospectus and evaluated the proposals for this project.

MR. J. E. HERZ, Chairman, *Chief Struc. Des. Engr., Sun Shipbuilding & Dry Dock Co.*

MR. C. M. COX, *Asst. Naval Architect, Newport News Shipbuilding & Dry Dock Co.*

MR. C. R. CUSHING, *President, C. R. Cushing and Co.*

PROF. J. E. GOLDBERG, *School of Civil Engineering, Purdue University*

PROF. J. R. PAULLING, JR., *Chairman of Dept. of Naval Arch., Univ. of California*

MR. D. P. ROSEMAN, *Naval Architect, Hydronautics, Inc.*

CAPT R. M. WHITE, USCG, *Chief, Applied Engineering Section, U. S. Coast Guard Academy*

The SR-196 Project Advisory Committee provided the liaison technical guidance, and reviewed the project reports with the investigator.

MR. J. E. HERZ, Chairman, *Chief Struc. Des. Engr., Sun Shipbuilding & Dry Dock Co.*

MR. R. G. KLINE, *Associate Research Consultant, U. S. Steel Corporation*

PROF. J. R. PAULLING, JR., *Chairman of Dept. of Naval Arch., Univ. of California*

SHIP STRUCTURE COMMITTEE PUBLICATIONS

These documents are distributed by the National Technical Information Service, Springfield, Va. 22151. These documents have been announced in the Clearinghouse Journal U.S. Government Research & Development Reports (USGRDR) under the indicated AD numbers.

SSC-216, *Program "Transhíp" - A Computer Program for the Design of the Midship Section of a Transversely-Framed Dry Cargo Ship - Part Two.* by Manley St. Denis. 1972.

SSC-217, *Compressive Strength of Ship Hull Girders - Part I - Unstiffened Plates* by H. Becker, R. Goldman, and J. Pozerycki. 1971. AD 717590.

SSC-218, *Design Considerations for Aluminum Hull Structures: Study of Aluminum Bulk Carrier* by C. J. Altenburg and R. J. Scott. 1971. AD 729021.

SSC-219, *Crack Propagation and Arrest in Ship and Other Steels* by G. T. Hahn, R. G. Hoagland, P. N. Mincer, A. R. Rosenfield, and M. Sarrate. 1971.

SSC-220, *A Limited Survey of Ship Structural Damage* by S. Hawkins, G. H. Levine, and R. Taggart. 1971.

SSC-221, *Response of the Delta Test to Specimen Variables* by L. J. McGeady. 1971.

SSC-222, *Catamarans - Technological Limits to Size and Appraisal of Structural Design Information and Procedures* by N. M. Maniar and W. P. Chiang. 1971.

SSC-223, *Compressive Strength of Ship Hull Girders - Part II - Stiffened Plates* by H. Becker, A. Colao, R. Goldman, and J. Pozerycki. 1971.

SSC-224, *Feasibility Study of Glass Reinforced Plastic Cargo Ship* by R. J. Scott and J. H. Sommella. 1971.

SSC-225, *Structural Analysis of Longitudinally Framed Ships* by R. Nielson, P. Y. Chang, and L. C. Deschamps. 1972.

SSC-226, *Tanker Longitudinal Strength Analysis - User's Manual and Computer Program* by R. Nielson, P. Y. Chang, and L. C. Deschamps. 1972.

SSC-227, *Tanker Transverse Strength Analysis - User's Manual* by R. Nielson, P. Y. Chang, and L. C. Deschamps. 1972.