PATRAN–STAGS TRANSLATOR (PATSTAGS)

By Neil Otte

Structures and Dynamics Laboratory
Science and Engineering Directorate

February 1990
This document presents a computer program used to translate PATRAN finite element model data into STAGS (Structural Analysis of General Shells) input data. The program supports translation of nodal, nodal constraints, element, force, and pressure data. The subroutine UPRESS required for the readings of live pressure data into STAGS is also presented.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>FILES</td>
<td>1</td>
</tr>
<tr>
<td>NODAL DATA</td>
<td>2</td>
</tr>
<tr>
<td>ELEMENT DATA</td>
<td>4</td>
</tr>
<tr>
<td>FORCE DATA</td>
<td>4</td>
</tr>
<tr>
<td>PRESSURE DATA</td>
<td>4</td>
</tr>
<tr>
<td>APPENDIX A—SUBROUTINE UPRESS</td>
<td>7</td>
</tr>
<tr>
<td>APPENDIX B—PATSTAGS.FOR</td>
<td>11</td>
</tr>
</tbody>
</table>
PATRAN-STAGS TRANSLATOR (PATSTAGS)

INTRODUCTION

PATSTAGS will translate PATRAN finite element model data into STAGS (Structural Analysis of General Shells) input records. PATSTAGS reads the data from a PATRAN neutral file and writes STAGS input records into a STAGS input file and a UPRESS data file. The translator will not translate all PATRAN neutral file packets, nor will it write a complete STAGS input deck. The STAGS input deck must be edited after translation and the appropriate control records added as specified by the STAGS user manual. The following PATRAN neutral file packets are read and translated to the corresponding STAGS records.

<table>
<thead>
<tr>
<th>PATRAN neutral file packet</th>
<th>To</th>
<th>STAGS input record</th>
</tr>
</thead>
<tbody>
<tr>
<td>25 - Title Card</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26 - Summary Data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1  - Node Data</td>
<td></td>
<td>S-1 User Point Record</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S-2 Auxiliary Coordinate Record</td>
</tr>
<tr>
<td>2  - Element Data</td>
<td></td>
<td>T-2 Beam Element Record</td>
</tr>
<tr>
<td></td>
<td></td>
<td>T-3 Triangular Element Record</td>
</tr>
<tr>
<td></td>
<td></td>
<td>T-4 Quadrilateral Element Record</td>
</tr>
<tr>
<td>6  - Distributed Loads</td>
<td></td>
<td>UPRESS input record</td>
</tr>
<tr>
<td>7  - Node Forces</td>
<td></td>
<td>U-3 Load Definition Record</td>
</tr>
<tr>
<td>8  - Node Displacements</td>
<td></td>
<td>S-1 User Point Record</td>
</tr>
</tbody>
</table>

FILES

PATSTAGS uses three files: the PATRAN neutral file to be translated, a STAGS input file, and a STAGS pressure data file. PATSTAGS will prompt the user for the name of the neutral file to be translated and the desired names of the STAGS files to be created. The file names may be up to 40 characters in length. The STAGS input file created will contain the STAGS S-1, S-2, T-2, T-3, T-4, and U-3 input records. The pressure data file created will contain the element live pressure data used by the STAGS subroutine UPRESS.
NODAL DATA

Nodal data is read from the PATRAN neutral file packets 1 (Node Data) and 8 (Node Displacements). The data is written as S-1 cards in the STAGS input file. S-2 cards are also created if cylindrical coordinate system 1 is used for the node coordinates and the node constraints in the PATRAN model. Up to 5,000 nodes may be translated. To translate a larger model, the array dimensions will need to be increased in PATSTAGS.FOR and the program relinked.

The translator has the capability to define an auxiliary cylindrical coordinate system on a S-2 card for use in applying boundary constraints. The following requirements must be met before an S-2 card is written.

1. The node and nodal constraints must both be defined in PATRAN in cylindrical coordinate system 1.

2. The global X axis must coincide with the cylindrical Z axis, i.e., the model must be a shell of revolution about the global X axis.

The auxiliary coordinate system for each node is defined on S-2 cards with two points. The points are defined as follows:

1. The node radius is calculated as

   \[ \text{Rad}_n = \sqrt{Y_n^2 + Z_n^2} \]

   where \( Y_n \) and \( Z_n \) are the global y and z coordinates of the node.

2. Point 1 coordinates are calculated as

   \[ X_1 = 0.0 \]

   \[ Y_1 = \frac{Y_n}{\text{Rad}_n} \]

   \[ Z_1 = \frac{Z_n}{\text{Rad}_n} \]

   where subscript 1 refers to the cylindrical coordinate system point 1, and subscript \( n \) refers to the node.
3. Point 2 coordinates are calculated as

\[ X_2 = 0.0 \]

\[ Y_2 = -\frac{Z_n}{\text{Rad}_{n}} \]

\[ Z_2 = \frac{Y_n}{\text{Rad}_{n}} \]

where subscript 2 refers to cylindrical coordinate system point 2 and subscript \( n \) refers to the node.

Figure 1 shows the coordinate systems attached to four nodes in the different quadrants.

The coding for the auxiliary coordinate system is contained in the 9200 block of the PATSTAGS FORTRAN file. This can be easily changed if a different axis of revolution is desired for the model.

Nodal constraints only are supported. Specified nodal displacements or rotations which would require a U-3 record to be written are not currently supported.

Figure 1. Auxiliary coordinate systems.
ELEMENT DATA

Element data is read from the PATRAN neutral file packet 2 (Element Data). The data is written as T-2, T-3, and T-4 cards in the STAGS input file. Bar, triangular, and quadrilateral elements are supported. Up to 5,000 elements may be translated. To translate a larger model, the array dimensions in PATSTAGS.FOR must be increased and the program relinked.

The program will interactively write the number of elements which have been read and will prompt the user for the desired values of ILIN, which governs geometric nonlinearity, and IPLAS, which governs material nonlinearity. The entered response will be used in all T-2, T-3, and T-4 records written. The property ID numbers entered in PATRAN will be used for ICROSS in the T-2 card and IWALL in the T-3 and T-4 cards. XSI, EC4, and ECQ in the T-2 records, ZETA and ECZ in the T-3 records, and ZETA, ECZ, INTEG, and IPENL in the T-4 records are all set to zero. Any of these defaults can be changed by editing the appropriate format card in block 9400 of the PATSTAGS.FOR file and relinking the program. The program will next interactively write the number of beam elements read and prompt the user for the desired beam element code number. The triangular and quadrilateral elements are handled similarly.

At the end of each element record in the STAGS input file, the record type, PATRAN element number, and STAGS element number are written for the user's information.

The X-Y plane of all bar elements must be defined using the node option in PATRAN. The vector and grid option are not supported.

FORCE DATA

Force data is read from the PATRAN neutral file packet 7 (Node Forces). The data is written as U-3 cards in the STAGS input file.

The translator will support only one force component per node, per load set. If more than one force component is needed on a node, they should be defined in PATRAN as belonging to different load sets (i.e., the X component in load set 1, the Y component in load set 2, the Z component in load set 3, etc.).

PRESSURE DATA

Pressure data is read from the PATRAN neutral file packet 6 (Distributed Loads). The data is written into a pressure data file, which is then used with the UPRESS subroutine. The translator interactively prompts the user for the desired name of the pressure data file.
The translator will support only one pressure component per element, per load set. If more than one pressure component is needed on an element, they should be defined in PATRAN as belonging to different load sets (i.e., the $X$ component in load set 1, the $Y$ component in load set 2, the $Z$ component in load set 3, etc.).

This data file is formatted to be used with the subroutine UPRESS listed in appendix A. This subroutine will need to be linked to STAGS before running the analysis. The pressure data file is called by the subroutine UPRESS as unit 17, so an assign statement is needed to assign the pressure data file to unit 17 before running the analysis.
APPENDIX A

SUBROUTINE UPRESS
SUBROUTINE UPRESS(T,PA,PB,IUNIT,IELT,X,Y,Z,LIVE,PRESS)

C THIS SUBROUTINE, WHEN LINKED WITH STAGS, WILL READ LIVE
C PRESSURE DATA FROM THE FORMATED PRESSURE FILE CREATED
C BY PATSTAGS.

C WRITTEN BY NEIL OTTE
C MARSHALL SPACE FLIGHT CENTER
C ED-24 STRUCTURAL STRENGTH BRANCH
C (205) 544-7231

C DIMENSION STID(5000),EPRESS(5000)
INTEGER STID
IF (M.EQ.0)THEN
READ (17,10)NPRESS
  FORMAT (I5)
  DO 100 I=1,NPRESS
    READ (17,11) STID(I),EPRESS(I)
      FORMAT (I5,F10.4)
    CONTINUE
  WRITE (6,12)
    FORMAT (1X,'SUBROUTINE UPRESS WRITEN BY NEIL OTTE',/1X,
      'HAS BEEN USED')
ELSE
  CONTINUE
ENDIF
  K = 1
200 CONTINUE
IF (STID(K).EQ.IELT)THEN
  PRESS = EPRESS
  LIVE = 1
ELSE
  K = K+1
  GO TO 200
ENDIF
M = 99
RETURN
END
PATRAN - STAGS TRANSLATOR

WRITTEN BY: NEIL OTTE

STRUCTURAL STRENGTH BRANCH ED-24

MARSHALL SPACE FLIGHT CENTER

(205) 544-7231

THIS PROGRAM WILL READ A PATRAN NEUTRAL FILE AND CREATE A STAGS INPUT DECK, AS WELL AS A LIVE PRESSURE DATA FILE TO BE READ BY THE UPRESS SUBROUTINE. THIS PROGRAM WILL NOT TRANSLATE ALL PATRAN NEUTRAL FILE PACKETS NOR WILL IT WRITE A COMPLETE STAGS INPUT DECK. SEE THE USERS MANUAL FOR MORE INFORMATION.

DIMENSION AND INITIALIZATION BLOCK

DIMENSION NID(5000), X(5000), Y(5000), Z(5000), CID(5000), XD(5000),
* YD(5000), ZD(5000), UX(5000), UY(5000), UZ(5000), NDCID(5000),
* NDID(5000), EBPID(5000), EBLNODE1(5000), EBLNODE2(5000),
* EBNODER(5000), ETPID(5000), ETLNODE1(5000), ETLNODE2(5000),
* ETLNODE1(5000), EQPID(5000), EQLNODE1(5000), EQLNODE2(5000),
* EQLNODE3(5000), EQLNODE4(5000), BID(5000), TID(5000),
* QID(5000), STID(5000), LID(5000), PDATA(5000), LE(5000),
* LD(5000), FN(5000), FCID(5000), FD(5000), FDA(5000),
* FLAX(5000)

INTEGER PT, EID, EIV, EKC, EN1, EN2, EBNODES, EBCONFIG, EBPID, EBCID,
* EBLNODE1, EBLNODE2, EBNODER, ETNODEx, ETCONFIG, ETPID, ETCID,
* ETLNODE1, ETLNODE2, ETLNODE3, EEPNODEs, EECONFIG, EEPID, EECID,
* EQLNODE1, EQLNODE2, EQLNODE3, EQLNODE4, BID, TID, QID, STID, CID,
* CONFIG, PSPC1, PSPC2, PSPC3, PSPC4, PSPC5, PSPC6, XD, YD, ZD,
* UX, UY, UZ, FD, FN, FCID, FICOMP1, FICOMP2, FICOMP3, FICOMP4,
* FICOMP5, FICOMP6, FE, FLAX

CHARACTER GTYPE*10, INFILE*40, OUTFILE*40, TITLE*80, DATE*12, TIME*8,
* VERSION*12, PRESSURE*40

INITIALIZATION

IB = 0
IT = 0
IQ = 0

FILE SETUP BLOCK

PROMPT FOR INPUT AND OUTPUT FILE NAMES

WRITE (5, 9100)
9100 FORMAT (2X, 'ENTER THE NAME OF THE NEUTRAL FILE TO BE TRANSLATED')
READ (6, 9110) INFILE
9110 FORMAT (A40)

PRECEDING PAGE BLANK NOT FILMED

PAGE 12 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED
WRITE (5,9101)
9101 FORMAT (2X,'ENTER THE DESIRED NAME OF THE STAGS INPUT FILE')
READ (6,9110) INFILE
WRITE (5,9102)
9102 FORMAT (2X, 'ENTER THE DESIRED NAME OF THE STAGS PRESSURE FILE')
READ (6,9110) PRESSURE
C
OPEN THE INPUT AND OUTPUT FILES
C
OPEN (UNIT=7,FILE=INFILE,STATUS='UNKNOWN')
OPEN (UNIT=8,FILE=OUTFILE,STATUS='UNKNOWN')
OPEN (UNIT=10,FILE=PRESSURE,STATUS='UNKNOWN')
C
TITLE CARD BLOCK
C
READ PACKET TYPE 25 - TITLE CARD
C
READ (7,250) PT,ID,IV,KC,TITLE
250 FORMAT (I2,318/,A80)
C
SUMMARY DATA BLOCK
C
READ PACKET TYPE 26 - SUMMARY DATA
C
READ (7,260) PT,ID,IV,KC,N1,N2,N3,N4,N5,DATE,TIME,VERSION
260 FORMAT (I2,818/,12A,8A,12A)
C
NODE COORDINATE BLOCK
C
READ PACKET TYPE 01 - NODE COORD. DATA
C
DO 100 I=1,N1
   READ (7,10) PT,NID(I),NIV,NKC,X(I),Y(I),Z(I),ICF,GTYPE,NDF,
* CONFIG,CID(I),PSPC1,PSPC2,PSPC3,PSPC4,PSPC5,PSPC6
10 FORMAT (I2,318/,3E16.9/,I1,1A1,318,2X,6I1)
100 CONTINUE
C
ELEMENT DATA BLOCK
C
READ PACKET TYPE 02 - ELEMENT DATA
C
DO 200 I=1,N2
   READ (7,20) PT,EID,EIV,EKC,EN1,EN2
20 FORMAT (I2,818)
C
READ BEAM ELEMENT DATA
IF (EIV.EQ.2) THEN
  IB = IB+1
  BID(IB) = EID
  READ (7,21) EBNODES, EBCONFIG, EBPID(IB), EBCEID, EBTH1, EBTH2,
          EBTH3, EBLNODE1(IB), EBLNODE2(IB)
  FORMAT (418, 3E16.9, /, 218)
  EBNODER(IB) = EN2
  ELSE
    CONTINUE
  ENDIF

READ TRIANGULAR ELEMENT DATA

IF (EIV.EQ.3) THEN
  IT = IT+1
  TID(IT) = EID
  READ (7,22) ETNODES, ETCONFIG, ETPID(IT), ETCEID, ETTH1, ETTH2, ETTH3,
          ETLNODE1(IT), ETLNODE2(IT), ETLNODE3(IT)
  FORMAT (418, 3E16.9, /, 318)
  ELSE
    CONTINUE
  ENDIF

READ QUADRALATERAL ELEMENT DATA

IF (EIV.EQ.4) THEN
  IQ = IQ+1
  QID(IQ) = EID
  READ (7,23) EQNODES, EQCONFIG, EQPID(IQ), EQCEID, EQUH1, EQUH2, EQUH3,
          EQLNODE1(IQ), EQLNODE2(IQ), EQLNODE3(IQ), EQLNODE4(IQ)
  FORMAT (418, 3E16.9, /, 418)
  ELSE
    CONTINUE
  ENDIF

CREATE MATRIX OF STAGS ELEMENT ID VS. PATRAN ELEMENT ID

IF (IB.GT.0) THEN
  DO 201 L = 1, IB
    STID(L) = BID(L)
  201 CONTINUE
  ELSE
    CONTINUE
  ENDIF

IF (IT.GT.0) THEN
  M = IB+1
  N = IB+IT
  DO 202 L = M, N
    K = L-IB
    STID(L) = TID(K)
  202 CONTINUE
  ELSE
    CONTINUE
  ENDIF
IF (IQ.GT.0) THEN
  M = IT+IB+1
  DO 203 L = M, N2
  K = L-IB-IT
  STID(L) = QID(K)
203  CONTINUE
ELSE
  CONTINUE
ENDIF

C

               BLOCK 30               
C

C

               SKIP BLOCK
C

C

               SKIP PACKET TYPES 3, 4, AND 5
C

C

300  CONTINUE
READ (7,30) PT,ID,IV,KC
30  FORMAT (I2,818)
IF (PT.EQ.3) THEN
  DO 301 I=I,KC
  READ (7,*)
301  CONTINUE
  GO TO 300
ELSE
  CONTINUE
ENDIF
IF (PT.EQ.4) THEN
  DO 400 I=I,KC
  READ (7,*)
400  CONTINUE
  GO TO 300
ELSE
  CONTINUE
ENDIF
IF (PT.EQ.5) THEN
  DO 500 I=I,KC
  READ (7,*)
500  CONTINUE
  GO TO 300
ELSE
  CONTINUE
ENDIF
IF (PT.EQ.6) THEN

C

               BLOCK 60               
C

C

               PRESSURE BLOCK
C

C

               READ PACKET TYPE 6 - PRESSURE LOAD
C

K = I
600  CONTINUE
LID(K) = ID
READ (7,61) LTYPE,LEFLAG,LGFLAG,LICOMP1,LICOMP2,LICOMP3,LICOMP4,
* LICOMP5,LICOMP6,LNODE1,LNODE2,LNODE3,LNODE4,LNODE5,LNODE6,
* LNODE7,LNODE8,LNFE

61 FORMAT (17I1,I2)
READ (7,62) PDATA(K)
62 FORMAT (5E16.9)

C
C FIND STAGS ELEMENT ID FOR THIS LOAD
C
I = 0
601 CONTINUE
I = I+1
IF (STID(I).EQ.LID(K))THEN
LE(K) = I
ELSE
GO TO 601
ENDIF

C
C FIND PROPER LOAD DIRECTION
C
IF (LICOMP1.EQ.1)LD(K)=1
IF (LICOMP2.EQ.1)LD(K)=2
IF (LICOMP3.EQ.1)LD(K)=3
IF (LICOMP4.EQ.1)LD(K)=4
IF (LICOMP5.EQ.1)LD(K)=5
IF (LICOMP6.EQ.1)LD(K)=6
KMAX = K

C
C READ NEXT HEADER
C
READ (7,60) PT,ID,IV,KC
60 FORMAT (I2,818)
IF (PT.EQ.6) THEN
K=K+I
GO TO 600
ELSE
CONTINUE
ENDIF
ELSE
CONTINUE
ENDIF

C
C FORCE BLOCK
C
READ PACKET TYPE 7 - FORCES
C
IF (PT.EQ.7)THEN
M = 1
700 CONTINUE
FN(M) = ID
READ (7,70) FCID(M),FICOMP1,FICOMP2,FICOMP3,FICOMP4,FICOMP5,
* FICOMP6
70 FORMAT (I8,611)
READ (7,71) FDATA(M)
SET UP THE PROPER COORDINATE NUMBER

IF (FCID(M).EQ.1) THEN
   FLAX(M) = 0
ELSE
   FLAX(M) = 1
END IF

FIND THE PROPER LOAD DIRECTION

IF (FICOMP1.EQ.1) FD(M)=1
IF (FICOMP2.EQ.1) FD(M)=2
IF (FICOMP3.EQ.1) FD(M)=3
IF (FICOMP4.EQ.1) FD(M)=4
IF (FICOMP5.EQ.1) FD(M)=5
IF (FICOMP6.EQ.1) FD(M)=6
MMAX = M

READ NEXT HEADER

READ (7,72) PT,ID,IV,KC
IF (PT.EQ.7) THEN
   M = M+1
   GO TO 700
ELSE
   CONTINUE
ENDIF
ELSE
   CONTINUE
ENDIF
IF (PT.EQ.8) THEN
   *** NODE CONSTRAINT BLOCK ***
   READ PACKET TYPE 8 - NODE CONSTRAINTS
   CONTINUE
   I = ID
   NDID(I) = I
   IF (KC.EQ.3) THEN
      READ (7,81) NDCID(I),XD(I),YD(I),ZD(I),UX(I),UY(I),UZ(I),
      DDATA1,DDATA2,DDATA3,DDATA4,DDATA5,DDATA6
      FORMAT (I8,6I1,/,5E16.9,/,5E16.9)
   ELSE
      READ (7,82) NDCID(I),XD(I),YD(I),ZD(I),UX(I),UY(I),UZ(I),
      DDATA1,DDATA2,DDATA3,DDATA4,DDATA5
      FORMAT (I8,6I1,/,5E16.9)
   END IF
   READ (7,80) PT,ID,IV,KC
   CONTINUE
   FORMAT (I2,818)
IF (PT.EQ.8) THEN
  GO TO 800
ELSE
  CONTINUE
ENDIF
ELSE
  CONTINUE
ENDIF

C

NODE DATA WRITE BLOCK

WRITE NODAL DATA IN STAGS FORMAT

DO 9200 I=1,N1
  IF (XD(I).EQ.0) THEN
    XD(I) = 1
  ELSE
    XD(I) = 0
  ENDIF
  IF (YD(I).EQ.0) THEN
    YD(I) = 1
  ELSE
    YD(I) = 0
  ENDIF
  IF (ZD(I).EQ.0) THEN
    ZD(I) = 1
  ELSE
    ZD(I) = 0
  ENDIF
  IF (UX(I).EQ.0) THEN
    UX(I) = 1
  ELSE
    UX(I) = 0
  ENDIF
  IF (UY(I).EQ.0) THEN
    UY(I) = 1
  ELSE
    UY(I) = 0
  ENDIF
  IF (UZ(I).EQ.0) THEN
    UZ(I) = 1
  ELSE
    UZ(I) = 0
  ENDIF
  WRITE (8,9201) NID(I),0,0,0,X(I),Y(I),Z(I),XD(I),YD(I),ZD(I),
            UX(I),UY(I),UZ(I),CID(I)
  FORMAT (I5,1X,I 1,1X,I 1,1X,F9.4,1X,F9.4,1X,F9.4,1X,F9.4,1X,
    10X,'$ S-I')
  IF (CID(I).EQ.1) THEN
    RAD = SQRT(Y(I)**2+Z(I)**2)
    IF (RAD.EQ.0) THEN
      XAY = 1.0
      XAZ = 0.0
    ENDIF
$$\begin{align*}
YAY &= 0.0 \\
YAZ &= 1.0 \\
\text{ELSE} \\
XAY &= (Y(I)/RAD) \\
XAZ &= (Z(I)/RAD) \\
YAY &= -(Z(I)/RAD) \\
YAZ &= (Y(I)/RAD) \\
\text{ENDIF}
\end{align*}$$

WRITE (8,9202) 0.0,XAY,XAZ,0.0,YAY,YAZ

9202 FORMAT (1X,F9.4,,F9.4,,F9.4,1X,F9.4,,F9.4,*,
\*,,F9.4,5X,$ S-2$)

ELSE
\text{CONTINUE}
\text{ENDIF}

9200 CONTINUE

C

INTERACTIVE PROMPTS BLOCK

C

INTERACTIVE PROMPTS

C

PROMPT FOR ANALYSIS TYPE

C

WRITE (6,9300) N2

9300 FORMAT (2X,'DATA FOR',I7,2X,'ELEMENTS HAVE BEEN READ.',/,,2X,
\* 'PLEASE ENTER 0 FOR NONLINEAR STRAIN-DISPLACEMENT RELATIONS',
\* /,2X,'OR 1 FOR LINEAR STRAIN-DISPLACEMENT RELATIONS.')

READ (5,9310) ILIN

9310 FORMAT (I10)

WRITE (6,9301)

9301 FORMAT (2X,'PLEASE ENTER 0 FOR ELASTIC BEHAVIOR OR 1 FOR',/2X,
\* 'PLASTICITY EFFECTS')

READ (5,9310) IPLAS

C

PROMPT FOR BEAM ELEMENT TYPE

C

IF (IB.GT.0)THEN
\text{WRITE (6,9302) IB}

9302 FORMAT (2X,'DATA FOR',I7,2X,'BEAM ELEMENTS HAVE BEEN READ.',/2X,
\* 'PLEASE ENTER THE DESIRED BEAM ELEMENT CODE NUMBER.')

READ (5,9310) KBM

ELSE
\text{CONTINUE}
\text{ENDIF}

C

PROMPT FOR TRIANGULAR BEAM ELEMENT TYPE

C

IF (IT.GT.0)THEN
\text{WRITE (6,9303) IT}

9303 FORMAT (2X,'DATA FOR',I7,2X,
\* 'TRIANGULAR ELEMENTS HAVE BEEN READ.',/2X,
\* 'PLEASE ENTER THE DESIRED TRIANGULAR ELEMENT CODE NUMBER.')

READ (5,9310) KTRI

ELSE
\text{CONTINUE}
\text{ENDIF}
C
C
CONTINUE
ENDIF
C
PROMPT FOR QUAD BEAM ELEMENT TYPE
C
IF (IQ.GT.0)THEN
WRITE (6,9304) IQ
9304 FORMAT (2X,'DATA FOR',I7,2X,'QUAD ELEMENTS HAVE BEEN READ.',/,
* 'PLEASE ENTER THE DESIRED QUAD ELEMENT CODE NUMBER.')
READ (5,9310)KQUAD
ELSE
CONTINUE
ENDIF
C

ELEMENT DATA WRITE BLOCK

WRITE ELEMENT DATA IN STAGS FORMAT
WRITE BEAM ELEMENT DATA

IF (IB.GT.0)THEN
DO 9400 I=I,IB
WRITE (8,9401) EBLNODE1(I),EBLNODE2(I),EBNODER(I),KBM,EBPID(I),
* ILIN,ILAS,B1D(I),I
9401 FORMAT (1X,I4,1X,I4,1X,I4,1X,I4,1X,I3,1X,'0.',1X,'0.',1X,'0.',1X,
* 11,1X,11,10X,'$ T-2',1X,I5,3X,I5)
9400 CONTINUE
ELSE
CONTINUE
ENDIF
C
WRITE TRIANGULAR ELEMENT DATA
C
IF (IT.GT.0)THEN
DO 9402 I=1,IT
M = I+IB
WRITE (8,9403) ETLNODE1(I),ETLNODE2(I),ETLNODE3(I),KTRI,ETPID(I),
* ILIN,ILAS,TID(I),M
9403 FORMAT (1X,I4,1X,I4,1X,I4,1X,I4,1X,I3,1X,'0.',1X,'0.',1X,
* 1X,I1,1X,11,10X,'$ T-3',1X,I5,3X,I5)
9402 CONTINUE
ELSE
CONTINUE
ENDIF
C
WRITE QUAD ELEMENT DATA
C
IF (IQ.GT.0)THEN
DO 9404 I=1,IQ
M = I+IB+IT
WRITE (8,9405) EQLNODE1(I),EQLNODE2(I),EQLNODE3(I),EQLNODE4(I),
* KQUAD,EQPID(I),ILIN,ILAS,QID(I),M
9405 FORMAT (1X,I4,1X,I4,1X,I4,1X,I4,1X,I4,1X,1X,'0.',1X,'0.',1X,
*I 1,1X,I1,1X,'0',1X,'0',10X,'$ T-4',1X,I5,3X,I5)
9404 CONTINUE
  ELSE
  CONTINUE
ENDIF
C
C
C
C
C
WRITE FORCES TO STAGS FORMAT
C
DO 9500 M=1,MMAX
  WRITE (8,9501) FDATA(M),FD(M),FN(M),FLAX(M)
9501 FORMAT (F10.3,1X,'I',1X,I1,1X,I5,1X,'0',1X,
* 11.5X,'$ U-3 FORCES')
9500 CONTINUE
C
C
C
C
PRESSURE WRITE BLOCK
C
C
WRITE PRESSURE DATA TO UPRESS FORMAT
C
WRITE (10,9600) KMAX
9600 FORMAT (I5)
  DO 9601 K=I,KMAX
    WRITE (10,9602) LE(K),PDATA(K)
9602 FORMAT (I5,F10.4)
9601 CONTINUE
STOP
END
The information in this report has been reviewed for technical content. Review of any information concerning Department of Defense or nuclear energy activities or programs has been made by the MSFC Security Classification Officer. This report, in its entirety, has been determined to be unclassified.

JAMES C. BLAIR
Director, Structures and Dynamics Laboratory