Table A13 List of the file, **behavior.new.** This is the completed file after the GENOPT user's completion of the "GENTEXT" interactive session. The FORTRAN statements in this file become the **skeletal behavior.new** library. GENOPT does this automatically. **See Table a31 for a list of the "howto.behavior" file.** In this particular application of GENOPT the GENOPT user decided not to "flesh out" any of the BEHXi, i=1,14 subroutines, but instead to perform all the computations in SUBROUTINE STRUCT. The reason for this decision is that the "behaviors" (such as buckling, stress, displacement) are computed by BIGBOSOR4, and one execution of BIGBOSOR4 yields more than one "behavior". For example, one execution of BIGBOSOR4 generates skin and stiffener stresses and buckling load factors for both Region 1 and Region 2. It would take much more computer time if BIGBOSOR4 had to be executed inside each of the 14 BEHXi subroutines to yield a particular "behavior" that is the "responsibility" of that particular BEHXi subroutine. In the "equivellipse" application of GENOPT there are 14 BEHXi subroutines, two groups of 7. The first group of 7 BEHXi subroutines, BEHXi, i=1,7, corresponds to the first group of 7 "bundles" of Role 4,5,6 variables listed in Table 2 = those "bundles" with variable names that contain the digit "1" (CLAPS1, GENBK1, SKNBK1, STFBK1, SKNST1, STFST1, and WAPEX1). The second group of 7 BEHXi subroutines, BEHXi, i=8,14, corresponds to the second group of 7 "bundles" of Role 4,5,6 variables listed in Table 2 = those "bundles" with variable names that contain the digit "2" (CLAPS2, GENBK2, SKNBK2, STFBK2, SKNST2, STFST2, and WAPEX2). The digits, "1" and "2", denote, respectively, "shell with a "mode 1" axisymmetric buckling modal imperfection" and "shell with a "mode 2" axisymmetric buckling modal imperfection". (More generally, instead of "mode 1" read "odd-numbered mode" and instead of "mode 2" read "even-numbered mode").

```
==================================================================
C=DECK      BEHAVIOR.NEW
C  This library contains the skeletons of
C  subroutines called SUBROUTINE BEHXn, n = 1,
C  2, 3, . . . that will yield predictions
C  of behavioral responses of various systems
C  to environments (loads).
C
C  You may complete the subroutines by writing
C  algorithms that yield the responses,
C  each of which plays a part in constraining
C  the design to a feasible region. Examples
C  of responses are: stress, buckling, drag,
C  vibration, deformation, clearances, etc.
C
C  A skeleton routine called SUBROUTINE OBJECT
C  is also provided for any objective function
C  (e.g. weight, deformation, conductivity)
C  you may wish to create.
C
```

```
C  A skeleton routine called SUBROUTINE USRCON
C  is also provided for any user-written
C  constraint condition you may wish to write:
C  This is an INEQUALITY condition that
C  involves any program variables.  However,
C  note that this kind of thing is done
C  automatically in the program DECIDE, so
C  try DECIDE first to see if your particular
C  constraint conditions can be accommodated
C  more easily there.
C
C  Please note that you do not have to modify
C  BEHAVIOR.NEW in any way, but may instead
C  prefer to insert your subroutines into the
C  skeletal libraries ADDCODEn.NEW, n=1,2,...
C  and appropriate common blocks, dimension
C  and type statements and calls to these
C  subroutines in the library STRUCT.NEW.
C  This strategy is best if your FORTRAN
C  input to GENOPT contains quite a bit
C  of software previously written by
C  yourself or others, and/or the generation
C  of behavioral constraints is more easily
C  accomplished via another architecture
C  than that provided for in the
C  BEHAVIOR.NEW library. (See instructions
C  in the libraries ADDCODEn.NEW and
C  STRUCT.NEW for this procedure.)
C
C  The two test cases provided with GENOPT
C  provide examples of each method:
C   PLATE (test case 1): use of BEHAVIOR.NEW
C   PANEL (test case 2): use of ADDCODEn.NEW
C                        and STRUCT.NEW.
C
C   SEVEN ROLES THAT VARIABLES IN THIS SYSTEM OF PROGRAMS PLAY
C
C    A variable can have one of the following roles:
C
C    1 = a possible decision variable for optimization,
C        typically a dimension of a structure.
C    2 = a constant parameter (cannot vary as design evolves),
C        typically a control integer or material property,
C        but not a load, allowable, or factor of safety,
C        which are asked for later.
C    3 = a parameter characterizing the environment, such
C        as a load component or a temperature.
C    4 = a quantity that describes the response of the
```

```
C        structure, (e.g. stress, buckling load, frequency)
C    5 = an allowable, such as maximum allowable stress,
C        minimum allowable frequency, etc.
C    6 = a factor of safety
C    7 = the quantity that is to be minimized or maximized,
C        called the "objective function" (e.g. weight).
C ==========================================================
C
C  NAMES, DEFINITIONS, AND ROLES OF THE VARIABLES:

C YOU ARE USING WHAT I HAVE CALLED "GENOPT" TO GENERATE AN
C OPTIMIZATION PROGRAM FOR A PARTICULAR CLASS OF PROBLEMS.
C THE NAME YOU HAVE CHOSEN FOR THIS CLASS OF PROBLEMS IS: equivellipse

C "GENOPT" (GENeral OPTimization) was written during 1987-1988
C by Dr. David Bushnell, Dept. 93-30, Bldg. 251, (415)424-3237
C    Lockheed Missiles and Space Co., 3251 Hanover St.,
C    Palo Alto, California, USA  94304

C The optimizer used in GENOPT is called ADS, and was
C written by G. Vanderplaats [3]. It is based on the method
C of feasible directions [4].

C                       ABSTRACT

C "GENOPT" has the following purposes and properties:
C     1. Any relatively simple analysis is "automatically"
C        converted into an optimization of whatever system
C        can be analyzed with fixed properties. Please note
C        that GENOPT is not intended to be used for problems
C        that require elaborate data-base management systems
C        or large numbers of degrees of freedom.

C     2. The optimization problems need not be in fields nor
C        jargon familiar to me, the developer of GENOPT.
C        Although all of the example cases (See the cases
C        in the directories under genopt/case)
C        are in the field of structural analysis, GENOPT is
C        not limited to that field.


C     3. GENOPT is a program that writes other programs. These
C        programs, WHEN AUGMENTED BY USER-SUPPLIED CODING,
C        form a program system that should be user-friendly in
C        the GENOPT-user"s field. In this instance the user
C        of GENOPT must later supply FORTRAN coding that
C        calculates behavior in the problem class called "equivellipse".
```

```
C     4. Input data and textual material are elicited from
C        the user of GENOPT in a general enough way so that
C        he or she may employ whatever data, definitions, and
C        "help" paragraphs will make subsequent use of the
C        program system thus generated easy by those less
C        familiar with the class of problems "equivellipse" than
C        the GENOPT user.

C     5. The program system generated by GENOPT has the same
C        general architecture as previous programs written for
C        specific applications by the developer [7 - 16]. That
C        is, the command set is:

C           BEGIN     (User supplies starting design, loads,
C                      control integers, material properties,
C                      etc. in an interactive-help mode.)

C           DECIDE    (User chooses decision and linked
C                      variables and inequality constraints
C                      that are not based on behavior.)

C           MAINSETUP (User chooses output option, whether
C                      to perform analysis of a fixed design
C                      or to optimize, and number of design
C                      iterations.)

C           OPTIMIZE  (The program system performs, in a batch
C                      mode, the work specified in MAINSETUP.)

C           SUPEROPT  (Program tries to find the GLOBAL optimum
C                      design as described in Ref.[11] listed
C                      below (Many OPTIMIZEs in one run.)

C           CHANGE    (User changes certain parameters)

C           CHOOSEPLOT (User selects which quantities to plot
C                       vs. design iterations.)

C           DIPLOT    (User generates plots)

C           CLEANSPEC (User cleans out unwanted files.)

C     A typical runstream is:
C       GENOPTLOG   (activate command set)
C       BEGIN       (provide starting design, loads, etc.)
C       DECIDE      (choose decision variables and bounds)
C       MAINSETUP   (choose print option and analysis type)
C       OPTIMIZE    (launch batch run for n design iterations)
```

```
C       OPTIMIZE    (launch batch run for n design iterations)
C       OPTIMIZE    (launch batch run for n design iterations)
C       OPTIMIZE    (launch batch run for n design iterations)
C       OPTIMIZE    (launch batch run for n design iterations)
C       CHANGE      (change some variables for new starting pt)
C       OPTIMIZE    (launch batch run for n design iterations)
C       OPTIMIZE    (launch batch run for n design iterations)
C       OPTIMIZE    (launch batch run for n design iterations)
C       OPTIMIZE    (launch batch run for n design iterations)
C       OPTIMIZE    (launch batch run for n design iterations)
C       CHOOSEPLOT  (choose which variables to plot)
C       DIPLOT      (plot variables v. iterations)
C       CHOOSEPLOT  (choose additional variables to plot)
C       DIPLOT      (plot more variables v design iterations)
C       CLEANSPEC   (delete extraneous files for specific case)

C  IMPORTANT:  YOU MUST ALWAYS GIVE THE COMMAND "OPTIMIZE"
C              SEVERAL TIMES IN SUCCESSION IN ORDER TO OBTAIN
C              CONVERGENCE! AN EXPLANATION OF WHY YOU MUST DO
C              THIS IS GIVEN ON P 580-582 OF THE PAPER "PANDA2,
C              PROGRAM FOR MINIMUM WEIGHT DESIGN OF STIFFENED,
C              COMPOSITE LOCALLY BUCKLED PANELS", Computers and
C              Structures, Vol. 25, No. 4, pp 469-605 (1987).

C Due to introduction of a "global" optimizer, SUPEROPT,
C described in Ref.[11], you can now use the runstream

C      BEGIN       (provide starting design, loads, etc.)
C      DECIDE      (choose decision variables and bounds)
C      MAINSETUP   (choose print option and analysis type)
C      SUPEROPT    (launch batch run for "global" optimization)
C      CHOOSEPLOT  (choose which variables to plot)
C      DIPLOT      (plot variables v. iterations)

C "Global" is in quotes because SUPEROPT does its best to find
C a true global optimum design. The user is strongly urged to
C execute SUPEROPT/CHOOSEPLOT several times in succession in
C order to determine an optimum that is essentially just as
C good as the theoretical true global optimum. Each execution
C of the series,
C      SUPEROPT
C      CHOOSEPLOT

C does the following:

C 1. SUPEROPT executes many sets of the two processors,
C    OPTIMIZE and AUTOCHANGE (AUTOCHANGE gets a new random
C    "starting" design), in which each set does the following:
```

```
C      OPTIMIZE          (perform k design iterations)
C      OPTIMIZE          (perform k design iterations)
C      OPTIMIZE          (perform k design iterations)
C      OPTIMIZE          (perform k design iterations)
C      OPTIMIZE          (perform k design iterations)
C      AUTOCHANGE        (get new starting design randomly)

C    SUPEROPT keeps repeating the above sequence until the
C    total number of design iterations reaches about 270.
C    The number of OPTIMIZEs per AUTOCHANGE is user-provided.

C 2. CHOOSEPLOT allows the user to plot stuff and resets the
C    total number of design iterations from SUPEROPT to zero.
C    After each execution of SUPEROPT the user MUST execute
C    CHOOSEPLOT: before the next execution of SUPEROPT the
C    total number of design iterations MUST be reset to zero.

C                     REFERENCES

C [1] Bushnell, D., "GENOPT--A program that writes
C user-friendly optimization code", International
C Journal of Solids and Structures, Vol. 26, No. 9/10,
C pp. 1173-1210, 1990. The same paper is contained in a
C bound volume of papers from the International Journal of
C Solids and Structures published in memory of Professor
C Charles D. Babcock, formerly with the California Institute
C of Technology.

C [2] Bushnell, D., "Automated optimum design of shells of
C revolution with application to ring-stiffened cylindrical
C shells with wavy walls", AIAA paper 2000-1663, 41st
C AIAA Structures Meeting, Atlanta, GA, April 2000. Also see
C Lockheed Martin report, same title, LMMS P525674, November
C 1999

C [3] Vanderplaats, G. N., "ADS--a FORTRAN program for
C automated design synthesis, Version 2.01", Engineering
C Design Optimization, Inc, Santa Barbara, CA, January, 1987

C [4] Vanderplaats, G. N. and Sugimoto, H., "A general-purpose
C optimization program for engineering design", Computers
C and Structures, Vol. 24, pp 13-21, 1986

C [5] Bushnell, D., "BOSOR4: Program for stress, stability,
C and vibration of complex, branched shells of revolution",
C in STRUCTURAL ANALYSIS SYSTEMS, Vol. 2, edited by A.
C Niku-Lari, pp. 25-54, (1986)
```

```
C [6] Bushnell, D., "BOSOR5: Program for buckling of complex,
C branched shells of revolution including large deflections,
C plasticity and creep," in STRUCTURAL ANALYSIS SYSTEMS, Vol.
C 2,  edited by A. Niku-Lari, pp. 55-67, (1986)

C [7] Bushnell, D., "PANDA2--program for minimum weight
C design of stiffened, composite, locally buckled panels",
C COMPUTERS AND STRUCTURES, vol. 25, No. 4, pp 469-605, 1987

C [8] Bushnell, D., "Improved optimum design of dewar
C supports", COMPUTERS and STRUCTURES, Vol. 29, No. 1,
C pp. 1-56 (1988)

C [9] Bushnell, D., "SPHERE - Program for minimum weight
C design of isogrid-stiffened spherical shells under uniform
C external pressure", Lockheed Report F372046, January, 1990

C [10] Bushnell, D., "Optimum design of isogrid-stiffened
C torispherical head", written and placed in the file
C ..genopt/case/torisph/readme.torisph, October 2005

C [11] Bushnell, D., "Recent enhancements to PANDA2", AIAA
C paper 96-1337-CP, Proc. 37th AIAA SDM Meeting, April 1996
C pp. 126-182, in particular, pp. 127-130

C [12] Bushnell, D., the file ..genopt/doc/getting.started

C [13] Bushnell, D., the case ..genopt/case/torisph

C [14] Bushnell, D., the case ..genopt/case/cylinder

C [15] Bushnell, D., the case ..genopt/case/wavycyl

C [16] Bushnell, D., the case ..genopt/case/plate

C [17] Bushnell, D., the case ..genopt/case/sphere

C=============================================================
C                TABLE 1      "GENOPT" COMMANDS
C=============================================================
C     HELPG        (get information on GENOPT.)
C     GENTEXT      (GENOPT user generate a prompt file, program
C                   fragments [see TABLE 5], programs [see
C                   TABLE 4]., and this and other files
C                   [see TABLE 5 and the rest of this file.])
C     GENPROGRAMS  (GENOPT user generate absolute elements:
C                   BEGIN.EXE, DECIDE.EXE, MAINSETUP.EXE,
```

```
C                   OPTIMIZE.EXE, CHANGE.EXE, STORE.EXE,
C                   CHOOSEPLOT.EXE, DIPLOT.EXE.)

C     BEGIN         (end user provide starting data.)
C     DECIDE        (end user choose decision variables, bounds,
C                    linked variables,inequality constraints.)
C     MAINSETUP     (end user set up strategy parameters.)
C     OPTIMIZE      (end user perform optimization, batch mode.)
C     SUPEROPT      (Program tries to find the GLOBAL optimum
C                    design as described in Ref.[11] listed
C                    above (Many OPTIMIZEs in one run.)

C     CHANGE        (end user change some parameters.)
C     CHOOSEPLOT    (end user choose which variables to plot v.
C                    design iterations.)
C     DIPLOT        (end user obtain plots.)
C     INSERT        (GENOPT user add parameters to the problem.)
C     CLEANGEN      (GENOPT user cleanup your GENeric files.)
C     CLEANSPEC     (end user cleanup your SPECific case files)

C   Please consult the following sources for more
C   information about GENOPT:
C        1.  GENOPT.STORY  and  HOWTO.RUN  and  GENOPT.NEWS
C        2.  Sample cases: (in the directory, genopt/case)
C        3.  NAME.DEF file, where NAME is the name chosen by
C            the GENOPT-user for a class of problems. (In this
C            case  NAME = equivellipse)
C        4.  GENOPT.HLP file    (type HELPG)
C=============================================================

C=============================================================
C   TABLE 2   GLOSSARY OF VARIABLES USED IN "equivellipse"
C=============================================================
C  ARRAY  NUMBER OF         PROMPT
C    ?   (ROWS,COLS)  ROLE  NUMBER   NAME                DEFINITION OF
VARIABLE
C                         (equivellipse.PRO)
C=============================================================
C    n   (   0,   0)    2      10   npoint   = number of x-coordinates
C    n   (   0,   0)    2      15   Ixinpu   = vector element number for
xinput
C    y   (  21,   0)    2      20   xinput   = x-coordinates for ends of
segment
C    n   (   0,   0)    2      25   ainput   = length of semi-major axis
C    n   (   0,   0)    2      30   binput   = length of semi-minor axis
of elli
C    n   (   0,   0)    2      35   nodes    = number of nodal points per
segmen
```

```
C    n   (   0,   0)    2      40   xlimit   = max. x-coordinate for x-
coordinat
C    y   (  21,   0)    1      45   THKSKN   = skin thickness at xinput
C    y   (  21,   0)    1      50   HIGHST   = height of isogrid members
at xinp
C    n   (   0,   0)    1      55   SPACNG   = spacing of the isogrid
members
C    n   (   0,   0)    1      60   THSTIF   = thickness of an isogrid
stiffenin
C    n   (   0,   0)    2      65   THKCYL   = thickness of the
cylindrical shel
C    n   (   0,   0)    2      70   RADCYL   = radius of the cylindrical
shell
C    n   (   0,   0)    2      75   LENCYL   = length of the cylindrical
segment
C    n   (   0,   0)    2      80   WIMP     = amplitude of the
axisymmetric imp
C    n   (   0,   0)    2      85   EMATL    = elastic modulus
C    n   (   0,   0)    2      90   NUMATL   = Poisson ratio of material
C    n   (   0,   0)    2      95   DNMATL   = mass density of material
C    n   (   0,   0)    2     100   IMODE    = strategy control for
imperfection
C    n   (   0,   0)    2     105   NCASES   = Number of load cases
(number of e
C    y   (  20,   0)    3     110   PRESS    = uniform external pressure
C    y   (  20,   0)    4     115   CLAPS1   = collapse pressure with
imperfecti
C    y   (  20,   0)    5     120   CLAPS1A  = allowable pressure for
axisymmetr
C    y   (  20,   0)    6     125   CLAPS1F  = factor of safety for
axisymmetric
C    y   (  20,   0)    4     130   GENBK1   = general buckling load
factor, mod
C    y   (  20,   0)    5     135   GENBK1A  = allowable general buckling
load f
C    y   (  20,   0)    6     140   GENBK1F  = factor of safety for
general buck
C    n   (   0,   0)    2     145   JSKNBK1  = number of regions for
computing b
C    y   (  20,  10)    4     150   SKNBK1   = local skin buckling load
factor,
C    y   (  20,  10)    5     155   SKNBK1A  = allowable buckling load
factor
C    y   (  20,  10)    6     160   SKNBK1F  = factor of safety for skin
bucklin
C    y   (  20,  10)    4     165   STFBK1   = buckling load factor,
isogrid mem
C    y   (  20,  10)    5     170   STFBK1A  = allowable for isogrid
```

```
stiffener b
C    y   (  20,  10)    6     175   STFBK1F  = factor of safety for
isogrid stif
C    y   (  20,  10)    4     180   SKNST1   = maximum stress in the shell
skin,
C    y   (  20,  10)    5     185   SKNST1A  = allowable stress for the
shell sk
C    y   (  20,  10)    6     190   SKNST1F  = factor of safety for skin
stress
C    y   (  20,  10)    4     195   STFST1   = maximum stress in isogrid
stiffen
C    y   (  20,  10)    5     200   STFST1A  = allowable stress in isogrid
stiff
C    y   (  20,  10)    6     205   STFST1F  = factor of safety for stress
in is
C    y   (  20,   0)    4     210   WAPEX1   = normal (axial) displacement
at ap
C    y   (  20,   0)    5     215   WAPEX1A  = allowable normal (axial)
displace
C    y   (  20,   0)    6     220   WAPEX1F  = factor of safety for WAPEX
C    y   (  20,   0)    4     225   CLAPS2   = collapse pressure with
imperfecti
C    y   (  20,   0)    5     230   CLAPS2A  = allowable pressure for
axisymmetr
C    y   (  20,   0)    6     235   CLAPS2F  = factor of safety for
axisymmetric
C    y   (  20,   0)    4     240   GENBK2   = general buckling load
factor, mod
C    y   (  20,   0)    5     245   GENBK2A  = allowable general buckling
load f
C    y   (  20,   0)    6     250   GENBK2F  = factor of safety for
general buck
C    n   (   0,   0)    2     255   JSKNBK2  = number of regions for
computing b
C    y   (  20,  10)    4     260   SKNBK2   = local skin buckling load
factor,
C    y   (  20,  10)    5     265   SKNBK2A  = allowable skin buckling
load fact
C    y   (  20,  10)    6     270   SKNBK2F  = factor of safety for local
skin b
C    y   (  20,  10)    4     275   STFBK2   = buckling load factor for
isogrid
C    y   (  20,  10)    5     280   STFBK2A  = allowable for isogrid
stiffener b
C    y   (  20,  10)    6     285   STFBK2F  = factor of safety for
isogrid stif
C    y   (  20,  10)    4     290   SKNST2   = maximum stress in the shell
skin,
```

```
C    y   (  20,  10)    5     295   SKNST2A  = allowable stress for the
shell sk
C    y   (  20,  10)    6     300   SKNST2F  = factor of safety for skin
stress
C    y   (  20,  10)    4     305   STFST2   = maximum stress in isogrid
stiffen
C    y   (  20,  10)    5     310   STFST2A  = allowable stress in isogrid
stiff
C    y   (  20,  10)    6     315   STFST2F  = factor of safety for stress
in is
C    y   (  20,   0)    4     320   WAPEX2   = normal (axial) displacement
at ap
C    y   (  20,   0)    5     325   WAPEX2A  = allowable normal (axial)
displace
C    y   (  20,   0)    6     330   WAPEX2F  = factor of safety for WAPEX
C    n   (   0,   0)    7     335   WEIGHT   = weight of the equivalent
ellipsoi
C
C=DECK      BEHX1
      SUBROUTINE BEHX1
     1 (IFILE,NPRINX,IMODX,IFAST,ILOADX,PHRASE)
C
C   PURPOSE: OBTAIN collapse pressure with imperfection mode 1
C
C   YOU MUST WRITE CODE THAT, USING
C   THE VARIABLES IN THE LABELLED
C   COMMON BLOCKS AS INPUT, ULTIMATELY
C   YIELDS THE RESPONSE VARIABLE FOR
C   THE ith LOAD CASE, ILOADX:
C
C     CLAPS1(ILOADX)
C
C   AS OUTPUT. THE ith CASE REFERS
C   TO ith ENVIRONMENT (e.g. load com-
C   bination).
C
C   DEFINITIONS OF INPUT DATA:
C    IMODX  = DESIGN CONTROL INTEGER:
C     IMODX = 0 MEANS BASELINE DESIGN
C     IMODX = 1 MEANS PERTURBED DESIGN
C     IFAST = 0 MEANS FEW  SHORTCUTS FOR PERTURBED DESIGNS
C     IFAST = 1 MEANS MORE SHORTCUTS FOR PERTURBED DESIGNS
C    IFILE = FILE FOR OUTPUT LIST:
C    NPRINX= OUTPUT CONTROL INTEGER:
C     NPRINX=0 MEANS SMALLEST AMOUNT
C     NPRINX=1 MEANS MEDIUM AMOUNT
C     NPRINX=2 MEANS LOTS OF OUTPUT
C
```

```
C     ILOADX = ith LOADING COMBINATION
C     PHRASE = collapse pressure with imperfection mode 1
C
C   OUTPUT:
C
C     CLAPS1(ILOADX)
C
       CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      COMMON/FV01/xinput(21),Ixinpu
      REAL xinput
      COMMON/FV02/ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      REAL ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      COMMON/FV05/THKSKN(21),HIGHST(21)
      REAL THKSKN,HIGHST
      COMMON/FV16/PRESS(20)
      REAL PRESS
      COMMON/FV19/CLAPS1(20),CLAPS1A(20),CLAPS1F(20)
      REAL CLAPS1,CLAPS1A,CLAPS1F
      COMMON/FV22/GENBK1(20),GENBK1A(20),GENBK1F(20)
      REAL GENBK1,GENBK1A,GENBK1F
      COMMON/FV25/SKNBK1(20,10),JSKNBK1,SKNBK1A(20,10),SKNBK1F(20,10)
      REAL SKNBK1,SKNBK1A,SKNBK1F
      COMMON/FV28/STFBK1(20,10),STFBK1A(20,10),STFBK1F(20,10)
      REAL STFBK1,STFBK1A,STFBK1F
      COMMON/FV31/SKNST1(20,10),SKNST1A(20,10),SKNST1F(20,10)
      REAL SKNST1,SKNST1A,SKNST1F
      COMMON/FV34/STFST1(20,10),STFST1A(20,10),STFST1F(20,10)
      REAL STFST1,STFST1A,STFST1F
      COMMON/FV37/WAPEX1(20),WAPEX1A(20),WAPEX1F(20)
      REAL WAPEX1,WAPEX1A,WAPEX1F
      COMMON/FV40/CLAPS2(20),CLAPS2A(20),CLAPS2F(20)
      REAL CLAPS2,CLAPS2A,CLAPS2F
      COMMON/FV43/GENBK2(20),GENBK2A(20),GENBK2F(20)
      REAL GENBK2,GENBK2A,GENBK2F
      COMMON/FV46/SKNBK2(20,10),JSKNBK2,SKNBK2A(20,10),SKNBK2F(20,10)
      REAL SKNBK2,SKNBK2A,SKNBK2F
      COMMON/FV49/STFBK2(20,10),STFBK2A(20,10),STFBK2F(20,10)
      REAL STFBK2,STFBK2A,STFBK2F
      COMMON/FV52/SKNST2(20,10),SKNST2A(20,10),SKNST2F(20,10)
      REAL SKNST2,SKNST2A,SKNST2F
      COMMON/FV55/STFST2(20,10),STFST2A(20,10),STFST2F(20,10)
      REAL STFST2,STFST2A,STFST2F
      COMMON/FV58/WAPEX2(20),WAPEX2A(20),WAPEX2F(20)
      REAL WAPEX2,WAPEX2A,WAPEX2F
      COMMON/IV01/npoint,nodes,IMODE
      INTEGER npoint,nodes,IMODE
      COMMON/FV11/LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
```

```
      REAL LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
C
C
C  INSERT SUBROUTINE STATEMENTS HERE.
C
C
C
C
      RETURN
      END
C
C
C
C
C=DECK      BEHX2
      SUBROUTINE BEHX2
     1 (IFILE,NPRINX,IMODX,IFAST,ILOADX,PHRASE)
C
C   PURPOSE: OBTAIN general buckling load factor, mode 1
C
C   YOU MUST WRITE CODE THAT, USING
C   THE VARIABLES IN THE LABELLED
C   COMMON BLOCKS AS INPUT, ULTIMATELY
C   YIELDS THE RESPONSE VARIABLE FOR
C   THE ith LOAD CASE, ILOADX:
C
C     GENBK1(ILOADX)
C
C   AS OUTPUT. THE ith CASE REFERS
C   TO ith ENVIRONMENT (e.g. load com-
C   bination).
C
C   DEFINITIONS OF INPUT DATA:
C    IMODX  = DESIGN CONTROL INTEGER:
C     IMODX = 0 MEANS BASELINE DESIGN
C     IMODX = 1 MEANS PERTURBED DESIGN
C     IFAST = 0 MEANS FEW  SHORTCUTS FOR PERTURBED DESIGNS
C     IFAST = 1 MEANS MORE SHORTCUTS FOR PERTURBED DESIGNS
C    IFILE = FILE FOR OUTPUT LIST:
C    NPRINX= OUTPUT CONTROL INTEGER:
C     NPRINX=0 MEANS SMALLEST AMOUNT
C     NPRINX=1 MEANS MEDIUM AMOUNT
C     NPRINX=2 MEANS LOTS OF OUTPUT
C
C     ILOADX = ith LOADING COMBINATION
C     PHRASE = general buckling load factor, mode 1
C
C   OUTPUT:
```

```
C
C     GENBK1(ILOADX)
C
       CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      COMMON/FV01/xinput(21),Ixinpu
      REAL xinput
      COMMON/FV02/ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      REAL ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      COMMON/FV05/THKSKN(21),HIGHST(21)
      REAL THKSKN,HIGHST
      COMMON/FV16/PRESS(20)
      REAL PRESS
      COMMON/FV19/CLAPS1(20),CLAPS1A(20),CLAPS1F(20)
      REAL CLAPS1,CLAPS1A,CLAPS1F
      COMMON/FV22/GENBK1(20),GENBK1A(20),GENBK1F(20)
      REAL GENBK1,GENBK1A,GENBK1F
      COMMON/FV25/SKNBK1(20,10),JSKNBK1,SKNBK1A(20,10),SKNBK1F(20,10)
      REAL SKNBK1,SKNBK1A,SKNBK1F
      COMMON/FV28/STFBK1(20,10),STFBK1A(20,10),STFBK1F(20,10)
      REAL STFBK1,STFBK1A,STFBK1F
      COMMON/FV31/SKNST1(20,10),SKNST1A(20,10),SKNST1F(20,10)
      REAL SKNST1,SKNST1A,SKNST1F
      COMMON/FV34/STFST1(20,10),STFST1A(20,10),STFST1F(20,10)
      REAL STFST1,STFST1A,STFST1F
      COMMON/FV37/WAPEX1(20),WAPEX1A(20),WAPEX1F(20)
      REAL WAPEX1,WAPEX1A,WAPEX1F
      COMMON/FV40/CLAPS2(20),CLAPS2A(20),CLAPS2F(20)
      REAL CLAPS2,CLAPS2A,CLAPS2F
      COMMON/FV43/GENBK2(20),GENBK2A(20),GENBK2F(20)
      REAL GENBK2,GENBK2A,GENBK2F
      COMMON/FV46/SKNBK2(20,10),JSKNBK2,SKNBK2A(20,10),SKNBK2F(20,10)
      REAL SKNBK2,SKNBK2A,SKNBK2F
      COMMON/FV49/STFBK2(20,10),STFBK2A(20,10),STFBK2F(20,10)
      REAL STFBK2,STFBK2A,STFBK2F
      COMMON/FV52/SKNST2(20,10),SKNST2A(20,10),SKNST2F(20,10)
      REAL SKNST2,SKNST2A,SKNST2F
      COMMON/FV55/STFST2(20,10),STFST2A(20,10),STFST2F(20,10)
      REAL STFST2,STFST2A,STFST2F
      COMMON/FV58/WAPEX2(20),WAPEX2A(20),WAPEX2F(20)
      REAL WAPEX2,WAPEX2A,WAPEX2F
      COMMON/IV01/npoint,nodes,IMODE
      INTEGER npoint,nodes,IMODE
      COMMON/FV11/LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
      REAL LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
C
C
C  INSERT SUBROUTINE STATEMENTS HERE.
```

```
C
C
C
C
      RETURN
      END
C
C
C
C
C=DECK      BEHX3
      SUBROUTINE BEHX3
     1 (IFILE,NPRINX,IMODX,IFAST,ILOADX,JCOL,PHRASE)
C
C   PURPOSE: OBTAIN local skin buckling load factor, mode 1
C
C   YOU MUST WRITE CODE THAT, USING
C   THE VARIABLES IN THE LABELLED
C   COMMON BLOCKS AS INPUT, ULTIMATELY
C   YIELDS THE RESPONSE VARIABLE FOR
C   THE ith LOAD CASE, ILOADX:
C
C     SKNBK1(ILOADX,JCOL)
C
C   AS OUTPUT. THE ith CASE REFERS
C   TO ith ENVIRONMENT (e.g. load com-
C   bination).
C   THE jth COLUMN  (JCOL)
C   INDEX IS DEFINED AS FOLLOWS:
C     number of regions for computing behavior
C
C   DEFINITIONS OF INPUT DATA:
C    IMODX  = DESIGN CONTROL INTEGER:
C     IMODX = 0 MEANS BASELINE DESIGN
C     IMODX = 1 MEANS PERTURBED DESIGN
C     IFAST = 0 MEANS FEW  SHORTCUTS FOR PERTURBED DESIGNS
C     IFAST = 1 MEANS MORE SHORTCUTS FOR PERTURBED DESIGNS
C    IFILE = FILE FOR OUTPUT LIST:
C    NPRINX= OUTPUT CONTROL INTEGER:
C     NPRINX=0 MEANS SMALLEST AMOUNT
C     NPRINX=1 MEANS MEDIUM AMOUNT
C     NPRINX=2 MEANS LOTS OF OUTPUT
C
C     ILOADX = ith LOADING COMBINATION
C     JCOL   = jth column of SKNBK1
C     JCOL   = number of regions for computing behavior
C     PHRASE = local skin buckling load factor, mode 1
C
```

```
C   OUTPUT:
C
C     SKNBK1(ILOADX,JCOL)
C
       CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      COMMON/FV01/xinput(21),Ixinpu
      REAL xinput
      COMMON/FV02/ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      REAL ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      COMMON/FV05/THKSKN(21),HIGHST(21)
      REAL THKSKN,HIGHST
      COMMON/FV16/PRESS(20)
      REAL PRESS
      COMMON/FV19/CLAPS1(20),CLAPS1A(20),CLAPS1F(20)
      REAL CLAPS1,CLAPS1A,CLAPS1F
      COMMON/FV22/GENBK1(20),GENBK1A(20),GENBK1F(20)
      REAL GENBK1,GENBK1A,GENBK1F
      COMMON/FV25/SKNBK1(20,10),JSKNBK1,SKNBK1A(20,10),SKNBK1F(20,10)
      REAL SKNBK1,SKNBK1A,SKNBK1F
      COMMON/FV28/STFBK1(20,10),STFBK1A(20,10),STFBK1F(20,10)
      REAL STFBK1,STFBK1A,STFBK1F
      COMMON/FV31/SKNST1(20,10),SKNST1A(20,10),SKNST1F(20,10)
      REAL SKNST1,SKNST1A,SKNST1F
      COMMON/FV34/STFST1(20,10),STFST1A(20,10),STFST1F(20,10)
      REAL STFST1,STFST1A,STFST1F
      COMMON/FV37/WAPEX1(20),WAPEX1A(20),WAPEX1F(20)
      REAL WAPEX1,WAPEX1A,WAPEX1F
      COMMON/FV40/CLAPS2(20),CLAPS2A(20),CLAPS2F(20)
      REAL CLAPS2,CLAPS2A,CLAPS2F
      COMMON/FV43/GENBK2(20),GENBK2A(20),GENBK2F(20)
      REAL GENBK2,GENBK2A,GENBK2F
      COMMON/FV46/SKNBK2(20,10),JSKNBK2,SKNBK2A(20,10),SKNBK2F(20,10)
      REAL SKNBK2,SKNBK2A,SKNBK2F
      COMMON/FV49/STFBK2(20,10),STFBK2A(20,10),STFBK2F(20,10)
      REAL STFBK2,STFBK2A,STFBK2F
      COMMON/FV52/SKNST2(20,10),SKNST2A(20,10),SKNST2F(20,10)
      REAL SKNST2,SKNST2A,SKNST2F
      COMMON/FV55/STFST2(20,10),STFST2A(20,10),STFST2F(20,10)
      REAL STFST2,STFST2A,STFST2F
      COMMON/FV58/WAPEX2(20),WAPEX2A(20),WAPEX2F(20)
      REAL WAPEX2,WAPEX2A,WAPEX2F
      COMMON/IV01/npoint,nodes,IMODE
      INTEGER npoint,nodes,IMODE
      COMMON/FV11/LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
      REAL LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
C
C
```

```
C  INSERT SUBROUTINE STATEMENTS HERE.
C
C
C
C
      RETURN
      END
C
C
C
C
C=DECK      BEHX4
      SUBROUTINE BEHX4
     1 (IFILE,NPRINX,IMODX,IFAST,ILOADX,JCOL,PHRASE)
C
C   PURPOSE: OBTAIN buckling load factor, isogrid member, mode 1
C
C   YOU MUST WRITE CODE THAT, USING
C   THE VARIABLES IN THE LABELLED
C   COMMON BLOCKS AS INPUT, ULTIMATELY
C   YIELDS THE RESPONSE VARIABLE FOR
C   THE ith LOAD CASE, ILOADX:
C
C     STFBK1(ILOADX,JCOL)
C
C   AS OUTPUT. THE ith CASE REFERS
C   TO ith ENVIRONMENT (e.g. load com-
C   bination).
C   THE jth COLUMN  (JCOL)
C   INDEX IS DEFINED AS FOLLOWS:
C     number of regions for computing behavior
C
C   DEFINITIONS OF INPUT DATA:
C    IMODX  = DESIGN CONTROL INTEGER:
C     IMODX = 0 MEANS BASELINE DESIGN
C     IMODX = 1 MEANS PERTURBED DESIGN
C     IFAST = 0 MEANS FEW  SHORTCUTS FOR PERTURBED DESIGNS
C     IFAST = 1 MEANS MORE SHORTCUTS FOR PERTURBED DESIGNS
C    IFILE = FILE FOR OUTPUT LIST:
C    NPRINX= OUTPUT CONTROL INTEGER:
C     NPRINX=0 MEANS SMALLEST AMOUNT
C     NPRINX=1 MEANS MEDIUM AMOUNT
C     NPRINX=2 MEANS LOTS OF OUTPUT
C
C     ILOADX = ith LOADING COMBINATION
C     JCOL   = jth column of STFBK1
C     JCOL   = number of regions for computing behavior
C     PHRASE = buckling load factor, isogrid member, mode 1
```

```
C
C   OUTPUT:
C
C     STFBK1(ILOADX,JCOL)
C
       CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      COMMON/FV01/xinput(21),Ixinpu
      REAL xinput
      COMMON/FV02/ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      REAL ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      COMMON/FV05/THKSKN(21),HIGHST(21)
      REAL THKSKN,HIGHST
      COMMON/FV16/PRESS(20)
      REAL PRESS
      COMMON/FV19/CLAPS1(20),CLAPS1A(20),CLAPS1F(20)
      REAL CLAPS1,CLAPS1A,CLAPS1F
      COMMON/FV22/GENBK1(20),GENBK1A(20),GENBK1F(20)
      REAL GENBK1,GENBK1A,GENBK1F
      COMMON/FV25/SKNBK1(20,10),JSKNBK1,SKNBK1A(20,10),SKNBK1F(20,10)
      REAL SKNBK1,SKNBK1A,SKNBK1F
      COMMON/FV28/STFBK1(20,10),STFBK1A(20,10),STFBK1F(20,10)
      REAL STFBK1,STFBK1A,STFBK1F
      COMMON/FV31/SKNST1(20,10),SKNST1A(20,10),SKNST1F(20,10)
      REAL SKNST1,SKNST1A,SKNST1F
      COMMON/FV34/STFST1(20,10),STFST1A(20,10),STFST1F(20,10)
      REAL STFST1,STFST1A,STFST1F
      COMMON/FV37/WAPEX1(20),WAPEX1A(20),WAPEX1F(20)
      REAL WAPEX1,WAPEX1A,WAPEX1F
      COMMON/FV40/CLAPS2(20),CLAPS2A(20),CLAPS2F(20)
      REAL CLAPS2,CLAPS2A,CLAPS2F
      COMMON/FV43/GENBK2(20),GENBK2A(20),GENBK2F(20)
      REAL GENBK2,GENBK2A,GENBK2F
      COMMON/FV46/SKNBK2(20,10),JSKNBK2,SKNBK2A(20,10),SKNBK2F(20,10)
      REAL SKNBK2,SKNBK2A,SKNBK2F
      COMMON/FV49/STFBK2(20,10),STFBK2A(20,10),STFBK2F(20,10)
      REAL STFBK2,STFBK2A,STFBK2F
      COMMON/FV52/SKNST2(20,10),SKNST2A(20,10),SKNST2F(20,10)
      REAL SKNST2,SKNST2A,SKNST2F
      COMMON/FV55/STFST2(20,10),STFST2A(20,10),STFST2F(20,10)
      REAL STFST2,STFST2A,STFST2F
      COMMON/FV58/WAPEX2(20),WAPEX2A(20),WAPEX2F(20)
      REAL WAPEX2,WAPEX2A,WAPEX2F
      COMMON/IV01/npoint,nodes,IMODE
      INTEGER npoint,nodes,IMODE
      COMMON/FV11/LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
      REAL LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
C
```

```
C
C  INSERT SUBROUTINE STATEMENTS HERE.
C
C
C
C
      RETURN
      END
C
C
C
C
C=DECK      BEHX5
      SUBROUTINE BEHX5
     1 (IFILE,NPRINX,IMODX,IFAST,ILOADX,JCOL,PHRASE)
C
C   PURPOSE: OBTAIN maximum stress in the shell skin, mode 1
C
C   YOU MUST WRITE CODE THAT, USING
C   THE VARIABLES IN THE LABELLED
C   COMMON BLOCKS AS INPUT, ULTIMATELY
C   YIELDS THE RESPONSE VARIABLE FOR
C   THE ith LOAD CASE, ILOADX:
C
C     SKNST1(ILOADX,JCOL)
C
C   AS OUTPUT. THE ith CASE REFERS
C   TO ith ENVIRONMENT (e.g. load com-
C   bination).
C   THE jth COLUMN  (JCOL)
C   INDEX IS DEFINED AS FOLLOWS:
C     number of regions for computing behavior
C
C   DEFINITIONS OF INPUT DATA:
C    IMODX  = DESIGN CONTROL INTEGER:
C     IMODX = 0 MEANS BASELINE DESIGN
C     IMODX = 1 MEANS PERTURBED DESIGN
C     IFAST = 0 MEANS FEW  SHORTCUTS FOR PERTURBED DESIGNS
C     IFAST = 1 MEANS MORE SHORTCUTS FOR PERTURBED DESIGNS
C    IFILE = FILE FOR OUTPUT LIST:
C    NPRINX= OUTPUT CONTROL INTEGER:
C     NPRINX=0 MEANS SMALLEST AMOUNT
C     NPRINX=1 MEANS MEDIUM AMOUNT
C     NPRINX=2 MEANS LOTS OF OUTPUT
C
C     ILOADX = ith LOADING COMBINATION
C     JCOL   = jth column of SKNST1
C     JCOL   = number of regions for computing behavior
```

```
C     PHRASE = maximum stress in the shell skin, mode 1
C
C   OUTPUT:
C
C     SKNST1(ILOADX,JCOL)
C
       CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      COMMON/FV01/xinput(21),Ixinpu
      REAL xinput
      COMMON/FV02/ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      REAL ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      COMMON/FV05/THKSKN(21),HIGHST(21)
      REAL THKSKN,HIGHST
      COMMON/FV16/PRESS(20)
      REAL PRESS
      COMMON/FV19/CLAPS1(20),CLAPS1A(20),CLAPS1F(20)
      REAL CLAPS1,CLAPS1A,CLAPS1F
      COMMON/FV22/GENBK1(20),GENBK1A(20),GENBK1F(20)
      REAL GENBK1,GENBK1A,GENBK1F
      COMMON/FV25/SKNBK1(20,10),JSKNBK1,SKNBK1A(20,10),SKNBK1F(20,10)
      REAL SKNBK1,SKNBK1A,SKNBK1F
      COMMON/FV28/STFBK1(20,10),STFBK1A(20,10),STFBK1F(20,10)
      REAL STFBK1,STFBK1A,STFBK1F
      COMMON/FV31/SKNST1(20,10),SKNST1A(20,10),SKNST1F(20,10)
      REAL SKNST1,SKNST1A,SKNST1F
      COMMON/FV34/STFST1(20,10),STFST1A(20,10),STFST1F(20,10)
      REAL STFST1,STFST1A,STFST1F
      COMMON/FV37/WAPEX1(20),WAPEX1A(20),WAPEX1F(20)
      REAL WAPEX1,WAPEX1A,WAPEX1F
      COMMON/FV40/CLAPS2(20),CLAPS2A(20),CLAPS2F(20)
      REAL CLAPS2,CLAPS2A,CLAPS2F
      COMMON/FV43/GENBK2(20),GENBK2A(20),GENBK2F(20)
      REAL GENBK2,GENBK2A,GENBK2F
      COMMON/FV46/SKNBK2(20,10),JSKNBK2,SKNBK2A(20,10),SKNBK2F(20,10)
      REAL SKNBK2,SKNBK2A,SKNBK2F
      COMMON/FV49/STFBK2(20,10),STFBK2A(20,10),STFBK2F(20,10)
      REAL STFBK2,STFBK2A,STFBK2F
      COMMON/FV52/SKNST2(20,10),SKNST2A(20,10),SKNST2F(20,10)
      REAL SKNST2,SKNST2A,SKNST2F
      COMMON/FV55/STFST2(20,10),STFST2A(20,10),STFST2F(20,10)
      REAL STFST2,STFST2A,STFST2F
      COMMON/FV58/WAPEX2(20),WAPEX2A(20),WAPEX2F(20)
      REAL WAPEX2,WAPEX2A,WAPEX2F
      COMMON/IV01/npoint,nodes,IMODE
      INTEGER npoint,nodes,IMODE
      COMMON/FV11/LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
      REAL LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
```

```
C
C
C  INSERT SUBROUTINE STATEMENTS HERE.
C
C
C
C
      RETURN
      END
C
C
C
C
C=DECK      BEHX6
      SUBROUTINE BEHX6
     1 (IFILE,NPRINX,IMODX,IFAST,ILOADX,JCOL,PHRASE)
C
C   PURPOSE: OBTAIN maximum stress in isogrid stiffener, mode 1
C
C   YOU MUST WRITE CODE THAT, USING
C   THE VARIABLES IN THE LABELLED
C   COMMON BLOCKS AS INPUT, ULTIMATELY
C   YIELDS THE RESPONSE VARIABLE FOR
C   THE ith LOAD CASE, ILOADX:
C
C     STFST1(ILOADX,JCOL)
C
C   AS OUTPUT. THE ith CASE REFERS
C   TO ith ENVIRONMENT (e.g. load com-
C   bination).
C   THE jth COLUMN  (JCOL)
C   INDEX IS DEFINED AS FOLLOWS:
C     number of regions for computing behavior
C
C   DEFINITIONS OF INPUT DATA:
C    IMODX  = DESIGN CONTROL INTEGER:
C     IMODX = 0 MEANS BASELINE DESIGN
C     IMODX = 1 MEANS PERTURBED DESIGN
C     IFAST = 0 MEANS FEW  SHORTCUTS FOR PERTURBED DESIGNS
C     IFAST = 1 MEANS MORE SHORTCUTS FOR PERTURBED DESIGNS
C    IFILE = FILE FOR OUTPUT LIST:
C    NPRINX= OUTPUT CONTROL INTEGER:
C     NPRINX=0 MEANS SMALLEST AMOUNT
C     NPRINX=1 MEANS MEDIUM AMOUNT
C     NPRINX=2 MEANS LOTS OF OUTPUT
C
C     ILOADX = ith LOADING COMBINATION
C     JCOL   = jth column of STFST1
```

```
C     JCOL   = number of regions for computing behavior
C     PHRASE = maximum stress in isogrid stiffener, mode 1
C
C   OUTPUT:
C
C     STFST1(ILOADX,JCOL)
C
       CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      COMMON/FV01/xinput(21),Ixinpu
      REAL xinput
      COMMON/FV02/ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      REAL ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      COMMON/FV05/THKSKN(21),HIGHST(21)
      REAL THKSKN,HIGHST
      COMMON/FV16/PRESS(20)
      REAL PRESS
      COMMON/FV19/CLAPS1(20),CLAPS1A(20),CLAPS1F(20)
      REAL CLAPS1,CLAPS1A,CLAPS1F
      COMMON/FV22/GENBK1(20),GENBK1A(20),GENBK1F(20)
      REAL GENBK1,GENBK1A,GENBK1F
      COMMON/FV25/SKNBK1(20,10),JSKNBK1,SKNBK1A(20,10),SKNBK1F(20,10)
      REAL SKNBK1,SKNBK1A,SKNBK1F
      COMMON/FV28/STFBK1(20,10),STFBK1A(20,10),STFBK1F(20,10)
      REAL STFBK1,STFBK1A,STFBK1F
      COMMON/FV31/SKNST1(20,10),SKNST1A(20,10),SKNST1F(20,10)
      REAL SKNST1,SKNST1A,SKNST1F
      COMMON/FV34/STFST1(20,10),STFST1A(20,10),STFST1F(20,10)
      REAL STFST1,STFST1A,STFST1F
      COMMON/FV37/WAPEX1(20),WAPEX1A(20),WAPEX1F(20)
      REAL WAPEX1,WAPEX1A,WAPEX1F
      COMMON/FV40/CLAPS2(20),CLAPS2A(20),CLAPS2F(20)
      REAL CLAPS2,CLAPS2A,CLAPS2F
      COMMON/FV43/GENBK2(20),GENBK2A(20),GENBK2F(20)
      REAL GENBK2,GENBK2A,GENBK2F
      COMMON/FV46/SKNBK2(20,10),JSKNBK2,SKNBK2A(20,10),SKNBK2F(20,10)
      REAL SKNBK2,SKNBK2A,SKNBK2F
      COMMON/FV49/STFBK2(20,10),STFBK2A(20,10),STFBK2F(20,10)
      REAL STFBK2,STFBK2A,STFBK2F
      COMMON/FV52/SKNST2(20,10),SKNST2A(20,10),SKNST2F(20,10)
      REAL SKNST2,SKNST2A,SKNST2F
      COMMON/FV55/STFST2(20,10),STFST2A(20,10),STFST2F(20,10)
      REAL STFST2,STFST2A,STFST2F
      COMMON/FV58/WAPEX2(20),WAPEX2A(20),WAPEX2F(20)
      REAL WAPEX2,WAPEX2A,WAPEX2F
      COMMON/IV01/npoint,nodes,IMODE
      INTEGER npoint,nodes,IMODE
      COMMON/FV11/LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
```

```
      REAL LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
C
C
C  INSERT SUBROUTINE STATEMENTS HERE.
C
C
C
C
      RETURN
      END
C
C
C
C
C=DECK      BEHX7
      SUBROUTINE BEHX7
     1 (IFILE,NPRINX,IMODX,IFAST,ILOADX,PHRASE)
C
C   PURPOSE: OBTAIN normal (axial) displacement at apex, mode 1
C
C   YOU MUST WRITE CODE THAT, USING
C   THE VARIABLES IN THE LABELLED
C   COMMON BLOCKS AS INPUT, ULTIMATELY
C   YIELDS THE RESPONSE VARIABLE FOR
C   THE ith LOAD CASE, ILOADX:
C
C     WAPEX1(ILOADX)
C
C   AS OUTPUT. THE ith CASE REFERS
C   TO ith ENVIRONMENT (e.g. load com-
C   bination).
C
C   DEFINITIONS OF INPUT DATA:
C    IMODX  = DESIGN CONTROL INTEGER:
C     IMODX = 0 MEANS BASELINE DESIGN
C     IMODX = 1 MEANS PERTURBED DESIGN
C     IFAST = 0 MEANS FEW  SHORTCUTS FOR PERTURBED DESIGNS
C     IFAST = 1 MEANS MORE SHORTCUTS FOR PERTURBED DESIGNS
C    IFILE = FILE FOR OUTPUT LIST:
C    NPRINX= OUTPUT CONTROL INTEGER:
C     NPRINX=0 MEANS SMALLEST AMOUNT
C     NPRINX=1 MEANS MEDIUM AMOUNT
C     NPRINX=2 MEANS LOTS OF OUTPUT
C
C     ILOADX = ith LOADING COMBINATION
C     PHRASE = normal (axial) displacement at apex, mode 1
C
C   OUTPUT:
```

```
C
C     WAPEX1(ILOADX)
C
       CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      COMMON/FV01/xinput(21),Ixinpu
      REAL xinput
      COMMON/FV02/ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      REAL ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      COMMON/FV05/THKSKN(21),HIGHST(21)
      REAL THKSKN,HIGHST
      COMMON/FV16/PRESS(20)
      REAL PRESS
      COMMON/FV19/CLAPS1(20),CLAPS1A(20),CLAPS1F(20)
      REAL CLAPS1,CLAPS1A,CLAPS1F
      COMMON/FV22/GENBK1(20),GENBK1A(20),GENBK1F(20)
      REAL GENBK1,GENBK1A,GENBK1F
      COMMON/FV25/SKNBK1(20,10),JSKNBK1,SKNBK1A(20,10),SKNBK1F(20,10)
      REAL SKNBK1,SKNBK1A,SKNBK1F
      COMMON/FV28/STFBK1(20,10),STFBK1A(20,10),STFBK1F(20,10)
      REAL STFBK1,STFBK1A,STFBK1F
      COMMON/FV31/SKNST1(20,10),SKNST1A(20,10),SKNST1F(20,10)
      REAL SKNST1,SKNST1A,SKNST1F
      COMMON/FV34/STFST1(20,10),STFST1A(20,10),STFST1F(20,10)
      REAL STFST1,STFST1A,STFST1F
      COMMON/FV37/WAPEX1(20),WAPEX1A(20),WAPEX1F(20)
      REAL WAPEX1,WAPEX1A,WAPEX1F
      COMMON/FV40/CLAPS2(20),CLAPS2A(20),CLAPS2F(20)
      REAL CLAPS2,CLAPS2A,CLAPS2F
      COMMON/FV43/GENBK2(20),GENBK2A(20),GENBK2F(20)
      REAL GENBK2,GENBK2A,GENBK2F
      COMMON/FV46/SKNBK2(20,10),JSKNBK2,SKNBK2A(20,10),SKNBK2F(20,10)
      REAL SKNBK2,SKNBK2A,SKNBK2F
      COMMON/FV49/STFBK2(20,10),STFBK2A(20,10),STFBK2F(20,10)
      REAL STFBK2,STFBK2A,STFBK2F
      COMMON/FV52/SKNST2(20,10),SKNST2A(20,10),SKNST2F(20,10)
      REAL SKNST2,SKNST2A,SKNST2F
      COMMON/FV55/STFST2(20,10),STFST2A(20,10),STFST2F(20,10)
      REAL STFST2,STFST2A,STFST2F
      COMMON/FV58/WAPEX2(20),WAPEX2A(20),WAPEX2F(20)
      REAL WAPEX2,WAPEX2A,WAPEX2F
      COMMON/IV01/npoint,nodes,IMODE
      INTEGER npoint,nodes,IMODE
      COMMON/FV11/LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
      REAL LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
C
C
C  INSERT SUBROUTINE STATEMENTS HERE.
```

```
C
C
C
C
      RETURN
      END
C
C
C
C
C=DECK      BEHX8
      SUBROUTINE BEHX8
     1 (IFILE,NPRINX,IMODX,IFAST,ILOADX,PHRASE)
C
C   PURPOSE: OBTAIN collapse pressure with imperfection mode 2
C
C   YOU MUST WRITE CODE THAT, USING
C   THE VARIABLES IN THE LABELLED
C   COMMON BLOCKS AS INPUT, ULTIMATELY
C   YIELDS THE RESPONSE VARIABLE FOR
C   THE ith LOAD CASE, ILOADX:
C
C     CLAPS2(ILOADX)
C
C   AS OUTPUT. THE ith CASE REFERS
C   TO ith ENVIRONMENT (e.g. load com-
C   bination).
C
C   DEFINITIONS OF INPUT DATA:
C    IMODX  = DESIGN CONTROL INTEGER:
C     IMODX = 0 MEANS BASELINE DESIGN
C     IMODX = 1 MEANS PERTURBED DESIGN
C     IFAST = 0 MEANS FEW  SHORTCUTS FOR PERTURBED DESIGNS
C     IFAST = 1 MEANS MORE SHORTCUTS FOR PERTURBED DESIGNS
C    IFILE = FILE FOR OUTPUT LIST:
C    NPRINX= OUTPUT CONTROL INTEGER:
C     NPRINX=0 MEANS SMALLEST AMOUNT
C     NPRINX=1 MEANS MEDIUM AMOUNT
C     NPRINX=2 MEANS LOTS OF OUTPUT
C
C     ILOADX = ith LOADING COMBINATION
C     PHRASE = collapse pressure with imperfection mode 2
C
C   OUTPUT:
C
C     CLAPS2(ILOADX)
C
       CHARACTER*80 PHRASE
```

```
C  INSERT ADDITIONAL COMMON BLOCKS:
      COMMON/FV01/xinput(21),Ixinpu
      REAL xinput
      COMMON/FV02/ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      REAL ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      COMMON/FV05/THKSKN(21),HIGHST(21)
      REAL THKSKN,HIGHST
      COMMON/FV16/PRESS(20)
      REAL PRESS
      COMMON/FV19/CLAPS1(20),CLAPS1A(20),CLAPS1F(20)
      REAL CLAPS1,CLAPS1A,CLAPS1F
      COMMON/FV22/GENBK1(20),GENBK1A(20),GENBK1F(20)
      REAL GENBK1,GENBK1A,GENBK1F
      COMMON/FV25/SKNBK1(20,10),JSKNBK1,SKNBK1A(20,10),SKNBK1F(20,10)
      REAL SKNBK1,SKNBK1A,SKNBK1F
      COMMON/FV28/STFBK1(20,10),STFBK1A(20,10),STFBK1F(20,10)
      REAL STFBK1,STFBK1A,STFBK1F
      COMMON/FV31/SKNST1(20,10),SKNST1A(20,10),SKNST1F(20,10)
      REAL SKNST1,SKNST1A,SKNST1F
      COMMON/FV34/STFST1(20,10),STFST1A(20,10),STFST1F(20,10)
      REAL STFST1,STFST1A,STFST1F
      COMMON/FV37/WAPEX1(20),WAPEX1A(20),WAPEX1F(20)
      REAL WAPEX1,WAPEX1A,WAPEX1F
      COMMON/FV40/CLAPS2(20),CLAPS2A(20),CLAPS2F(20)
      REAL CLAPS2,CLAPS2A,CLAPS2F
      COMMON/FV43/GENBK2(20),GENBK2A(20),GENBK2F(20)
      REAL GENBK2,GENBK2A,GENBK2F
      COMMON/FV46/SKNBK2(20,10),JSKNBK2,SKNBK2A(20,10),SKNBK2F(20,10)
      REAL SKNBK2,SKNBK2A,SKNBK2F
      COMMON/FV49/STFBK2(20,10),STFBK2A(20,10),STFBK2F(20,10)
      REAL STFBK2,STFBK2A,STFBK2F
      COMMON/FV52/SKNST2(20,10),SKNST2A(20,10),SKNST2F(20,10)
      REAL SKNST2,SKNST2A,SKNST2F
      COMMON/FV55/STFST2(20,10),STFST2A(20,10),STFST2F(20,10)
      REAL STFST2,STFST2A,STFST2F
      COMMON/FV58/WAPEX2(20),WAPEX2A(20),WAPEX2F(20)
      REAL WAPEX2,WAPEX2A,WAPEX2F
      COMMON/IV01/npoint,nodes,IMODE
      INTEGER npoint,nodes,IMODE
      COMMON/FV11/LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
      REAL LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
C
C
C  INSERT SUBROUTINE STATEMENTS HERE.
C
C
C
C
```

```
      RETURN
      END
C
C
C
C
C=DECK      BEHX9
      SUBROUTINE BEHX9
     1 (IFILE,NPRINX,IMODX,IFAST,ILOADX,PHRASE)
C
C   PURPOSE: OBTAIN general buckling load factor, mode 2
C
C   YOU MUST WRITE CODE THAT, USING
C   THE VARIABLES IN THE LABELLED
C   COMMON BLOCKS AS INPUT, ULTIMATELY
C   YIELDS THE RESPONSE VARIABLE FOR
C   THE ith LOAD CASE, ILOADX:
C
C     GENBK2(ILOADX)
C
C   AS OUTPUT. THE ith CASE REFERS
C   TO ith ENVIRONMENT (e.g. load com-
C   bination).
C
C   DEFINITIONS OF INPUT DATA:
C    IMODX  = DESIGN CONTROL INTEGER:
C     IMODX = 0 MEANS BASELINE DESIGN
C     IMODX = 1 MEANS PERTURBED DESIGN
C     IFAST = 0 MEANS FEW  SHORTCUTS FOR PERTURBED DESIGNS
C     IFAST = 1 MEANS MORE SHORTCUTS FOR PERTURBED DESIGNS
C    IFILE = FILE FOR OUTPUT LIST:
C    NPRINX= OUTPUT CONTROL INTEGER:
C     NPRINX=0 MEANS SMALLEST AMOUNT
C     NPRINX=1 MEANS MEDIUM AMOUNT
C     NPRINX=2 MEANS LOTS OF OUTPUT
C
C     ILOADX = ith LOADING COMBINATION
C     PHRASE = general buckling load factor, mode 2
C
C   OUTPUT:
C
C     GENBK2(ILOADX)
C
       CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      COMMON/FV01/xinput(21),Ixinpu
      REAL xinput
      COMMON/FV02/ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
```

```
      REAL ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      COMMON/FV05/THKSKN(21),HIGHST(21)
      REAL THKSKN,HIGHST
      COMMON/FV16/PRESS(20)
      REAL PRESS
      COMMON/FV19/CLAPS1(20),CLAPS1A(20),CLAPS1F(20)
      REAL CLAPS1,CLAPS1A,CLAPS1F
      COMMON/FV22/GENBK1(20),GENBK1A(20),GENBK1F(20)
      REAL GENBK1,GENBK1A,GENBK1F
      COMMON/FV25/SKNBK1(20,10),JSKNBK1,SKNBK1A(20,10),SKNBK1F(20,10)
      REAL SKNBK1,SKNBK1A,SKNBK1F
      COMMON/FV28/STFBK1(20,10),STFBK1A(20,10),STFBK1F(20,10)
      REAL STFBK1,STFBK1A,STFBK1F
      COMMON/FV31/SKNST1(20,10),SKNST1A(20,10),SKNST1F(20,10)
      REAL SKNST1,SKNST1A,SKNST1F
      COMMON/FV34/STFST1(20,10),STFST1A(20,10),STFST1F(20,10)
      REAL STFST1,STFST1A,STFST1F
      COMMON/FV37/WAPEX1(20),WAPEX1A(20),WAPEX1F(20)
      REAL WAPEX1,WAPEX1A,WAPEX1F
      COMMON/FV40/CLAPS2(20),CLAPS2A(20),CLAPS2F(20)
      REAL CLAPS2,CLAPS2A,CLAPS2F
      COMMON/FV43/GENBK2(20),GENBK2A(20),GENBK2F(20)
      REAL GENBK2,GENBK2A,GENBK2F
      COMMON/FV46/SKNBK2(20,10),JSKNBK2,SKNBK2A(20,10),SKNBK2F(20,10)
      REAL SKNBK2,SKNBK2A,SKNBK2F
      COMMON/FV49/STFBK2(20,10),STFBK2A(20,10),STFBK2F(20,10)
      REAL STFBK2,STFBK2A,STFBK2F
      COMMON/FV52/SKNST2(20,10),SKNST2A(20,10),SKNST2F(20,10)
      REAL SKNST2,SKNST2A,SKNST2F
      COMMON/FV55/STFST2(20,10),STFST2A(20,10),STFST2F(20,10)
      REAL STFST2,STFST2A,STFST2F
      COMMON/FV58/WAPEX2(20),WAPEX2A(20),WAPEX2F(20)
      REAL WAPEX2,WAPEX2A,WAPEX2F
      COMMON/IV01/npoint,nodes,IMODE
      INTEGER npoint,nodes,IMODE
      COMMON/FV11/LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
      REAL LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
C
C
C  INSERT SUBROUTINE STATEMENTS HERE.
C
C
C
C
      RETURN
      END
C
C
```

```
C
C
C=DECK      BEHX10
      SUBROUTINE BEHX10
     1 (IFILE,NPRINX,IMODX,IFAST,ILOADX,JCOL,PHRASE)
C
C   PURPOSE: OBTAIN local skin buckling load factor, mode 2
C
C   YOU MUST WRITE CODE THAT, USING
C   THE VARIABLES IN THE LABELLED
C   COMMON BLOCKS AS INPUT, ULTIMATELY
C   YIELDS THE RESPONSE VARIABLE FOR
C   THE ith LOAD CASE, ILOADX:
C
C     SKNBK2(ILOADX,JCOL)
C
C   AS OUTPUT. THE ith CASE REFERS
C   TO ith ENVIRONMENT (e.g. load com-
C   bination).
C   THE jth COLUMN  (JCOL)
C   INDEX IS DEFINED AS FOLLOWS:
C     number of regions for computing behavior
C
C   DEFINITIONS OF INPUT DATA:
C    IMODX  = DESIGN CONTROL INTEGER:
C     IMODX = 0 MEANS BASELINE DESIGN
C     IMODX = 1 MEANS PERTURBED DESIGN
C     IFAST = 0 MEANS FEW  SHORTCUTS FOR PERTURBED DESIGNS
C     IFAST = 1 MEANS MORE SHORTCUTS FOR PERTURBED DESIGNS
C    IFILE = FILE FOR OUTPUT LIST:
C    NPRINX= OUTPUT CONTROL INTEGER:
C     NPRINX=0 MEANS SMALLEST AMOUNT
C     NPRINX=1 MEANS MEDIUM AMOUNT
C     NPRINX=2 MEANS LOTS OF OUTPUT
C
C     ILOADX = ith LOADING COMBINATION
C     JCOL   = jth column of SKNBK2
C     JCOL   = number of regions for computing behavior
C     PHRASE = local skin buckling load factor, mode 2
C
C   OUTPUT:
C
C     SKNBK2(ILOADX,JCOL)
C
       CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      COMMON/FV01/xinput(21),Ixinpu
      REAL xinput
```

```
      COMMON/FV02/ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      REAL ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      COMMON/FV05/THKSKN(21),HIGHST(21)
      REAL THKSKN,HIGHST
      COMMON/FV16/PRESS(20)
      REAL PRESS
      COMMON/FV19/CLAPS1(20),CLAPS1A(20),CLAPS1F(20)
      REAL CLAPS1,CLAPS1A,CLAPS1F
      COMMON/FV22/GENBK1(20),GENBK1A(20),GENBK1F(20)
      REAL GENBK1,GENBK1A,GENBK1F
      COMMON/FV25/SKNBK1(20,10),JSKNBK1,SKNBK1A(20,10),SKNBK1F(20,10)
      REAL SKNBK1,SKNBK1A,SKNBK1F
      COMMON/FV28/STFBK1(20,10),STFBK1A(20,10),STFBK1F(20,10)
      REAL STFBK1,STFBK1A,STFBK1F
      COMMON/FV31/SKNST1(20,10),SKNST1A(20,10),SKNST1F(20,10)
      REAL SKNST1,SKNST1A,SKNST1F
      COMMON/FV34/STFST1(20,10),STFST1A(20,10),STFST1F(20,10)
      REAL STFST1,STFST1A,STFST1F
      COMMON/FV37/WAPEX1(20),WAPEX1A(20),WAPEX1F(20)
      REAL WAPEX1,WAPEX1A,WAPEX1F
      COMMON/FV40/CLAPS2(20),CLAPS2A(20),CLAPS2F(20)
      REAL CLAPS2,CLAPS2A,CLAPS2F
      COMMON/FV43/GENBK2(20),GENBK2A(20),GENBK2F(20)
      REAL GENBK2,GENBK2A,GENBK2F
      COMMON/FV46/SKNBK2(20,10),JSKNBK2,SKNBK2A(20,10),SKNBK2F(20,10)
      REAL SKNBK2,SKNBK2A,SKNBK2F
      COMMON/FV49/STFBK2(20,10),STFBK2A(20,10),STFBK2F(20,10)
      REAL STFBK2,STFBK2A,STFBK2F
      COMMON/FV52/SKNST2(20,10),SKNST2A(20,10),SKNST2F(20,10)
      REAL SKNST2,SKNST2A,SKNST2F
      COMMON/FV55/STFST2(20,10),STFST2A(20,10),STFST2F(20,10)
      REAL STFST2,STFST2A,STFST2F
      COMMON/FV58/WAPEX2(20),WAPEX2A(20),WAPEX2F(20)
      REAL WAPEX2,WAPEX2A,WAPEX2F
      COMMON/IV01/npoint,nodes,IMODE
      INTEGER npoint,nodes,IMODE
      COMMON/FV11/LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
      REAL LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
C
C
C  INSERT SUBROUTINE STATEMENTS HERE.
C
C
C
C
      RETURN
      END
C
```

```
C
C
C
C=DECK      BEHX11
      SUBROUTINE BEHX11
     1 (IFILE,NPRINX,IMODX,IFAST,ILOADX,JCOL,PHRASE)
C
C   PURPOSE: OBTAIN buckling load factor for isogrid member, mode 2
C
C   YOU MUST WRITE CODE THAT, USING
C   THE VARIABLES IN THE LABELLED
C   COMMON BLOCKS AS INPUT, ULTIMATELY
C   YIELDS THE RESPONSE VARIABLE FOR
C   THE ith LOAD CASE, ILOADX:
C
C     STFBK2(ILOADX,JCOL)
C
C   AS OUTPUT. THE ith CASE REFERS
C   TO ith ENVIRONMENT (e.g. load com-
C   bination).
C   THE jth COLUMN  (JCOL)
C   INDEX IS DEFINED AS FOLLOWS:
C     number of regions for computing behavior
C
C   DEFINITIONS OF INPUT DATA:
C    IMODX  = DESIGN CONTROL INTEGER:
C     IMODX = 0 MEANS BASELINE DESIGN
C     IMODX = 1 MEANS PERTURBED DESIGN
C     IFAST = 0 MEANS FEW  SHORTCUTS FOR PERTURBED DESIGNS
C     IFAST = 1 MEANS MORE SHORTCUTS FOR PERTURBED DESIGNS
C    IFILE = FILE FOR OUTPUT LIST:
C    NPRINX= OUTPUT CONTROL INTEGER:
C     NPRINX=0 MEANS SMALLEST AMOUNT
C     NPRINX=1 MEANS MEDIUM AMOUNT
C     NPRINX=2 MEANS LOTS OF OUTPUT
C
C     ILOADX = ith LOADING COMBINATION
C     JCOL   = jth column of STFBK2
C     JCOL   = number of regions for computing behavior
C     PHRASE = buckling load factor for isogrid member, mode 2
C
C   OUTPUT:
C
C     STFBK2(ILOADX,JCOL)
C
       CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      COMMON/FV01/xinput(21),Ixinpu
```

```
      REAL xinput
      COMMON/FV02/ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      REAL ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      COMMON/FV05/THKSKN(21),HIGHST(21)
      REAL THKSKN,HIGHST
      COMMON/FV16/PRESS(20)
      REAL PRESS
      COMMON/FV19/CLAPS1(20),CLAPS1A(20),CLAPS1F(20)
      REAL CLAPS1,CLAPS1A,CLAPS1F
      COMMON/FV22/GENBK1(20),GENBK1A(20),GENBK1F(20)
      REAL GENBK1,GENBK1A,GENBK1F
      COMMON/FV25/SKNBK1(20,10),JSKNBK1,SKNBK1A(20,10),SKNBK1F(20,10)
      REAL SKNBK1,SKNBK1A,SKNBK1F
      COMMON/FV28/STFBK1(20,10),STFBK1A(20,10),STFBK1F(20,10)
      REAL STFBK1,STFBK1A,STFBK1F
      COMMON/FV31/SKNST1(20,10),SKNST1A(20,10),SKNST1F(20,10)
      REAL SKNST1,SKNST1A,SKNST1F
      COMMON/FV34/STFST1(20,10),STFST1A(20,10),STFST1F(20,10)
      REAL STFST1,STFST1A,STFST1F
      COMMON/FV37/WAPEX1(20),WAPEX1A(20),WAPEX1F(20)
      REAL WAPEX1,WAPEX1A,WAPEX1F
      COMMON/FV40/CLAPS2(20),CLAPS2A(20),CLAPS2F(20)
      REAL CLAPS2,CLAPS2A,CLAPS2F
      COMMON/FV43/GENBK2(20),GENBK2A(20),GENBK2F(20)
      REAL GENBK2,GENBK2A,GENBK2F
      COMMON/FV46/SKNBK2(20,10),JSKNBK2,SKNBK2A(20,10),SKNBK2F(20,10)
      REAL SKNBK2,SKNBK2A,SKNBK2F
      COMMON/FV49/STFBK2(20,10),STFBK2A(20,10),STFBK2F(20,10)
      REAL STFBK2,STFBK2A,STFBK2F
      COMMON/FV52/SKNST2(20,10),SKNST2A(20,10),SKNST2F(20,10)
      REAL SKNST2,SKNST2A,SKNST2F
      COMMON/FV55/STFST2(20,10),STFST2A(20,10),STFST2F(20,10)
      REAL STFST2,STFST2A,STFST2F
      COMMON/FV58/WAPEX2(20),WAPEX2A(20),WAPEX2F(20)
      REAL WAPEX2,WAPEX2A,WAPEX2F
      COMMON/IV01/npoint,nodes,IMODE
      INTEGER npoint,nodes,IMODE
      COMMON/FV11/LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
      REAL LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
C
C
C  INSERT SUBROUTINE STATEMENTS HERE.
C
C
C
C
      RETURN
      END
```

```
C
C
C
C
C=DECK      BEHX12
      SUBROUTINE BEHX12
     1 (IFILE,NPRINX,IMODX,IFAST,ILOADX,JCOL,PHRASE)
C
C   PURPOSE: OBTAIN maximum stress in the shell skin, mode 2
C
C   YOU MUST WRITE CODE THAT, USING
C   THE VARIABLES IN THE LABELLED
C   COMMON BLOCKS AS INPUT, ULTIMATELY
C   YIELDS THE RESPONSE VARIABLE FOR
C   THE ith LOAD CASE, ILOADX:
C
C     SKNST2(ILOADX,JCOL)
C
C   AS OUTPUT. THE ith CASE REFERS
C   TO ith ENVIRONMENT (e.g. load com-
C   bination).
C   THE jth COLUMN  (JCOL)
C   INDEX IS DEFINED AS FOLLOWS:
C     number of regions for computing behavior
C
C   DEFINITIONS OF INPUT DATA:
C    IMODX  = DESIGN CONTROL INTEGER:
C     IMODX = 0 MEANS BASELINE DESIGN
C     IMODX = 1 MEANS PERTURBED DESIGN
C     IFAST = 0 MEANS FEW  SHORTCUTS FOR PERTURBED DESIGNS
C     IFAST = 1 MEANS MORE SHORTCUTS FOR PERTURBED DESIGNS
C    IFILE = FILE FOR OUTPUT LIST:
C    NPRINX= OUTPUT CONTROL INTEGER:
C     NPRINX=0 MEANS SMALLEST AMOUNT
C     NPRINX=1 MEANS MEDIUM AMOUNT
C     NPRINX=2 MEANS LOTS OF OUTPUT
C
C     ILOADX = ith LOADING COMBINATION
C     JCOL   = jth column of SKNST2
C     JCOL   = number of regions for computing behavior
C     PHRASE = maximum stress in the shell skin, mode 2
C
C   OUTPUT:
C
C     SKNST2(ILOADX,JCOL)
C
       CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
```

```
      COMMON/FV01/xinput(21),Ixinpu
      REAL xinput
      COMMON/FV02/ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      REAL ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      COMMON/FV05/THKSKN(21),HIGHST(21)
      REAL THKSKN,HIGHST
      COMMON/FV16/PRESS(20)
      REAL PRESS
      COMMON/FV19/CLAPS1(20),CLAPS1A(20),CLAPS1F(20)
      REAL CLAPS1,CLAPS1A,CLAPS1F
      COMMON/FV22/GENBK1(20),GENBK1A(20),GENBK1F(20)
      REAL GENBK1,GENBK1A,GENBK1F
      COMMON/FV25/SKNBK1(20,10),JSKNBK1,SKNBK1A(20,10),SKNBK1F(20,10)
      REAL SKNBK1,SKNBK1A,SKNBK1F
      COMMON/FV28/STFBK1(20,10),STFBK1A(20,10),STFBK1F(20,10)
      REAL STFBK1,STFBK1A,STFBK1F
      COMMON/FV31/SKNST1(20,10),SKNST1A(20,10),SKNST1F(20,10)
      REAL SKNST1,SKNST1A,SKNST1F
      COMMON/FV34/STFST1(20,10),STFST1A(20,10),STFST1F(20,10)
      REAL STFST1,STFST1A,STFST1F
      COMMON/FV37/WAPEX1(20),WAPEX1A(20),WAPEX1F(20)
      REAL WAPEX1,WAPEX1A,WAPEX1F
      COMMON/FV40/CLAPS2(20),CLAPS2A(20),CLAPS2F(20)
      REAL CLAPS2,CLAPS2A,CLAPS2F
      COMMON/FV43/GENBK2(20),GENBK2A(20),GENBK2F(20)
      REAL GENBK2,GENBK2A,GENBK2F
      COMMON/FV46/SKNBK2(20,10),JSKNBK2,SKNBK2A(20,10),SKNBK2F(20,10)
      REAL SKNBK2,SKNBK2A,SKNBK2F
      COMMON/FV49/STFBK2(20,10),STFBK2A(20,10),STFBK2F(20,10)
      REAL STFBK2,STFBK2A,STFBK2F
      COMMON/FV52/SKNST2(20,10),SKNST2A(20,10),SKNST2F(20,10)
      REAL SKNST2,SKNST2A,SKNST2F
      COMMON/FV55/STFST2(20,10),STFST2A(20,10),STFST2F(20,10)
      REAL STFST2,STFST2A,STFST2F
      COMMON/FV58/WAPEX2(20),WAPEX2A(20),WAPEX2F(20)
      REAL WAPEX2,WAPEX2A,WAPEX2F
      COMMON/IV01/npoint,nodes,IMODE
      INTEGER npoint,nodes,IMODE
      COMMON/FV11/LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
      REAL LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
C
C
C  INSERT SUBROUTINE STATEMENTS HERE.
C
C
C
C
      RETURN
```

```
      END
C
C
C
C
C=DECK      BEHX13
      SUBROUTINE BEHX13
     1 (IFILE,NPRINX,IMODX,IFAST,ILOADX,JCOL,PHRASE)
C
C   PURPOSE: OBTAIN maximum stress in isogrid stiffener, mode 2
C
C   YOU MUST WRITE CODE THAT, USING
C   THE VARIABLES IN THE LABELLED
C   COMMON BLOCKS AS INPUT, ULTIMATELY
C   YIELDS THE RESPONSE VARIABLE FOR
C   THE ith LOAD CASE, ILOADX:
C
C     STFST2(ILOADX,JCOL)
C
C   AS OUTPUT. THE ith CASE REFERS
C   TO ith ENVIRONMENT (e.g. load com-
C   bination).
C   THE jth COLUMN  (JCOL)
C   INDEX IS DEFINED AS FOLLOWS:
C     number of regions for computing behavior
C
C   DEFINITIONS OF INPUT DATA:
C    IMODX  = DESIGN CONTROL INTEGER:
C     IMODX = 0 MEANS BASELINE DESIGN
C     IMODX = 1 MEANS PERTURBED DESIGN
C     IFAST = 0 MEANS FEW  SHORTCUTS FOR PERTURBED DESIGNS
C     IFAST = 1 MEANS MORE SHORTCUTS FOR PERTURBED DESIGNS
C    IFILE = FILE FOR OUTPUT LIST:
C    NPRINX= OUTPUT CONTROL INTEGER:
C     NPRINX=0 MEANS SMALLEST AMOUNT
C     NPRINX=1 MEANS MEDIUM AMOUNT
C     NPRINX=2 MEANS LOTS OF OUTPUT
C
C     ILOADX = ith LOADING COMBINATION
C     JCOL   = jth column of STFST2
C     JCOL   = number of regions for computing behavior
C     PHRASE = maximum stress in isogrid stiffener, mode 2
C
C   OUTPUT:
C
C     STFST2(ILOADX,JCOL)
C
       CHARACTER*80 PHRASE
```

```
C  INSERT ADDITIONAL COMMON BLOCKS:
      COMMON/FV01/xinput(21),Ixinpu
      REAL xinput
      COMMON/FV02/ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      REAL ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      COMMON/FV05/THKSKN(21),HIGHST(21)
      REAL THKSKN,HIGHST
      COMMON/FV16/PRESS(20)
      REAL PRESS
      COMMON/FV19/CLAPS1(20),CLAPS1A(20),CLAPS1F(20)
      REAL CLAPS1,CLAPS1A,CLAPS1F
      COMMON/FV22/GENBK1(20),GENBK1A(20),GENBK1F(20)
      REAL GENBK1,GENBK1A,GENBK1F
      COMMON/FV25/SKNBK1(20,10),JSKNBK1,SKNBK1A(20,10),SKNBK1F(20,10)
      REAL SKNBK1,SKNBK1A,SKNBK1F
      COMMON/FV28/STFBK1(20,10),STFBK1A(20,10),STFBK1F(20,10)
      REAL STFBK1,STFBK1A,STFBK1F
      COMMON/FV31/SKNST1(20,10),SKNST1A(20,10),SKNST1F(20,10)
      REAL SKNST1,SKNST1A,SKNST1F
      COMMON/FV34/STFST1(20,10),STFST1A(20,10),STFST1F(20,10)
      REAL STFST1,STFST1A,STFST1F
      COMMON/FV37/WAPEX1(20),WAPEX1A(20),WAPEX1F(20)
      REAL WAPEX1,WAPEX1A,WAPEX1F
      COMMON/FV40/CLAPS2(20),CLAPS2A(20),CLAPS2F(20)
      REAL CLAPS2,CLAPS2A,CLAPS2F
      COMMON/FV43/GENBK2(20),GENBK2A(20),GENBK2F(20)
      REAL GENBK2,GENBK2A,GENBK2F
      COMMON/FV46/SKNBK2(20,10),JSKNBK2,SKNBK2A(20,10),SKNBK2F(20,10)
      REAL SKNBK2,SKNBK2A,SKNBK2F
      COMMON/FV49/STFBK2(20,10),STFBK2A(20,10),STFBK2F(20,10)
      REAL STFBK2,STFBK2A,STFBK2F
      COMMON/FV52/SKNST2(20,10),SKNST2A(20,10),SKNST2F(20,10)
      REAL SKNST2,SKNST2A,SKNST2F
      COMMON/FV55/STFST2(20,10),STFST2A(20,10),STFST2F(20,10)
      REAL STFST2,STFST2A,STFST2F
      COMMON/FV58/WAPEX2(20),WAPEX2A(20),WAPEX2F(20)
      REAL WAPEX2,WAPEX2A,WAPEX2F
      COMMON/IV01/npoint,nodes,IMODE
      INTEGER npoint,nodes,IMODE
      COMMON/FV11/LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
      REAL LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
C
C
C  INSERT SUBROUTINE STATEMENTS HERE.
C
C
C
C
```

```
      RETURN
      END
C
C
C
C
C=DECK      BEHX14
      SUBROUTINE BEHX14
     1 (IFILE,NPRINX,IMODX,IFAST,ILOADX,PHRASE)
C
C   PURPOSE: OBTAIN normal (axial) displacement at apex, mode 2
C
C   YOU MUST WRITE CODE THAT, USING
C   THE VARIABLES IN THE LABELLED
C   COMMON BLOCKS AS INPUT, ULTIMATELY
C   YIELDS THE RESPONSE VARIABLE FOR
C   THE ith LOAD CASE, ILOADX:
C
C     WAPEX2(ILOADX)
C
C   AS OUTPUT. THE ith CASE REFERS
C   TO ith ENVIRONMENT (e.g. load com-
C   bination).
C
C   DEFINITIONS OF INPUT DATA:
C    IMODX  = DESIGN CONTROL INTEGER:
C     IMODX = 0 MEANS BASELINE DESIGN
C     IMODX = 1 MEANS PERTURBED DESIGN
C     IFAST = 0 MEANS FEW  SHORTCUTS FOR PERTURBED DESIGNS
C     IFAST = 1 MEANS MORE SHORTCUTS FOR PERTURBED DESIGNS
C    IFILE = FILE FOR OUTPUT LIST:
C    NPRINX= OUTPUT CONTROL INTEGER:
C     NPRINX=0 MEANS SMALLEST AMOUNT
C     NPRINX=1 MEANS MEDIUM AMOUNT
C     NPRINX=2 MEANS LOTS OF OUTPUT
C
C     ILOADX = ith LOADING COMBINATION
C     PHRASE = normal (axial) displacement at apex, mode 2
C
C   OUTPUT:
C
C     WAPEX2(ILOADX)
C
       CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      COMMON/FV01/xinput(21),Ixinpu
      REAL xinput
      COMMON/FV02/ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
```

```
      REAL ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      COMMON/FV05/THKSKN(21),HIGHST(21)
      REAL THKSKN,HIGHST
      COMMON/FV16/PRESS(20)
      REAL PRESS
      COMMON/FV19/CLAPS1(20),CLAPS1A(20),CLAPS1F(20)
      REAL CLAPS1,CLAPS1A,CLAPS1F
      COMMON/FV22/GENBK1(20),GENBK1A(20),GENBK1F(20)
      REAL GENBK1,GENBK1A,GENBK1F
      COMMON/FV25/SKNBK1(20,10),JSKNBK1,SKNBK1A(20,10),SKNBK1F(20,10)
      REAL SKNBK1,SKNBK1A,SKNBK1F
      COMMON/FV28/STFBK1(20,10),STFBK1A(20,10),STFBK1F(20,10)
      REAL STFBK1,STFBK1A,STFBK1F
      COMMON/FV31/SKNST1(20,10),SKNST1A(20,10),SKNST1F(20,10)
      REAL SKNST1,SKNST1A,SKNST1F
      COMMON/FV34/STFST1(20,10),STFST1A(20,10),STFST1F(20,10)
      REAL STFST1,STFST1A,STFST1F
      COMMON/FV37/WAPEX1(20),WAPEX1A(20),WAPEX1F(20)
      REAL WAPEX1,WAPEX1A,WAPEX1F
      COMMON/FV40/CLAPS2(20),CLAPS2A(20),CLAPS2F(20)
      REAL CLAPS2,CLAPS2A,CLAPS2F
      COMMON/FV43/GENBK2(20),GENBK2A(20),GENBK2F(20)
      REAL GENBK2,GENBK2A,GENBK2F
      COMMON/FV46/SKNBK2(20,10),JSKNBK2,SKNBK2A(20,10),SKNBK2F(20,10)
      REAL SKNBK2,SKNBK2A,SKNBK2F
      COMMON/FV49/STFBK2(20,10),STFBK2A(20,10),STFBK2F(20,10)
      REAL STFBK2,STFBK2A,STFBK2F
      COMMON/FV52/SKNST2(20,10),SKNST2A(20,10),SKNST2F(20,10)
      REAL SKNST2,SKNST2A,SKNST2F
      COMMON/FV55/STFST2(20,10),STFST2A(20,10),STFST2F(20,10)
      REAL STFST2,STFST2A,STFST2F
      COMMON/FV58/WAPEX2(20),WAPEX2A(20),WAPEX2F(20)
      REAL WAPEX2,WAPEX2A,WAPEX2F
      COMMON/IV01/npoint,nodes,IMODE
      INTEGER npoint,nodes,IMODE
      COMMON/FV11/LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
      REAL LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
C
C
C  INSERT SUBROUTINE STATEMENTS HERE.
C
C
C
C
      RETURN
      END
C
C
```

```
C
C
C=DECK      USRCON
      SUBROUTINE USRCON(INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,
     1  WORDCX,WORDMX,PCWORD,CPLOTX,ICARX,IFILEX)
C   PURPOSE: GENERATE USER-WRITTEN
C   INEQUALITY CONSTRAINT CONDITION
C   USING ANY COMBINATION OF PROGRAM
C   VARIABLES.
C   YOU MUST WRITE CODE THAT, USING
C   THE VARIABLES IN THE LABELLED
C   COMMON BLOCKS AS INPUT, ULTIMATELY
C   YIELDS A CONSTRAINT CONDITION,
C   CALLED "CONX" IN THIS ROUTINE.
      DIMENSION WORDCX(*),WORDMX(*),IPOINC(*),CONSTX(*)
      DIMENSION PCWORD(*),CPLOTX(*)
      CHARACTER*80 WORDCX,WORDMX,PCWORD
C  INSERT ADDITIONAL COMMON BLOCKS:
      COMMON/FV01/xinput(21),Ixinpu
      REAL xinput
      COMMON/FV02/ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      REAL ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      COMMON/FV05/THKSKN(21),HIGHST(21)
      REAL THKSKN,HIGHST
      COMMON/FV16/PRESS(20)
      REAL PRESS
      COMMON/FV19/CLAPS1(20),CLAPS1A(20),CLAPS1F(20)
      REAL CLAPS1,CLAPS1A,CLAPS1F
      COMMON/FV22/GENBK1(20),GENBK1A(20),GENBK1F(20)
      REAL GENBK1,GENBK1A,GENBK1F
      COMMON/FV25/SKNBK1(20,10),JSKNBK1,SKNBK1A(20,10),SKNBK1F(20,10)
      REAL SKNBK1,SKNBK1A,SKNBK1F
      COMMON/FV28/STFBK1(20,10),STFBK1A(20,10),STFBK1F(20,10)
      REAL STFBK1,STFBK1A,STFBK1F
      COMMON/FV31/SKNST1(20,10),SKNST1A(20,10),SKNST1F(20,10)
      REAL SKNST1,SKNST1A,SKNST1F
      COMMON/FV34/STFST1(20,10),STFST1A(20,10),STFST1F(20,10)
      REAL STFST1,STFST1A,STFST1F
      COMMON/FV37/WAPEX1(20),WAPEX1A(20),WAPEX1F(20)
      REAL WAPEX1,WAPEX1A,WAPEX1F
      COMMON/FV40/CLAPS2(20),CLAPS2A(20),CLAPS2F(20)
      REAL CLAPS2,CLAPS2A,CLAPS2F
      COMMON/FV43/GENBK2(20),GENBK2A(20),GENBK2F(20)
      REAL GENBK2,GENBK2A,GENBK2F
      COMMON/FV46/SKNBK2(20,10),JSKNBK2,SKNBK2A(20,10),SKNBK2F(20,10)
      REAL SKNBK2,SKNBK2A,SKNBK2F
      COMMON/FV49/STFBK2(20,10),STFBK2A(20,10),STFBK2F(20,10)
      REAL STFBK2,STFBK2A,STFBK2F
```

```
      COMMON/FV52/SKNST2(20,10),SKNST2A(20,10),SKNST2F(20,10)
      REAL SKNST2,SKNST2A,SKNST2F
      COMMON/FV55/STFST2(20,10),STFST2A(20,10),STFST2F(20,10)
      REAL STFST2,STFST2A,STFST2F
      COMMON/FV58/WAPEX2(20),WAPEX2A(20),WAPEX2F(20)
      REAL WAPEX2,WAPEX2A,WAPEX2F
      COMMON/IV01/npoint,nodes,IMODE
      INTEGER npoint,nodes,IMODE
      COMMON/FV11/LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
      REAL LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
C
      CONX = 0.0
C
C  INSERT USER-WRITTEN STATEMENTS
C  HERE. THE CONSTRAINT CONDITION
C  THAT YOU CALCULATE IS CALLED "CONX"
C
      IF (CONX.EQ.0.0) RETURN
      IF (CONX.LT.0.0) THEN
         WRITE(IFILEX,*)' CONX MUST BE GREATER THAN ZERO.'
         CALL EXIT
      ENDIF
C
C  DO NOT CHANGE THE FOLLOWING STATEMENTS, EXCEPT WORDC
C
      ICARX = ICARX + 1
      INUMTT = INUMTT + 1
      WORDCX(ICARX) = ' USER: PROVIDE THIS.'
      CPLOTX(ICARX) = CONX - 1.
      CALL BLANKX(WORDCX(ICARX),IENDP)
      PCWORD(ICARX)  = WORDCX(ICARX)(1:IENDP)//' -1'
      IF (IMODX.EQ.0.AND.CONX.GT.CONMAX) GO TO 200
      IF (IMODX.EQ.1.AND.IPOINC(INUMTT).EQ.0) GO TO 200
      ICONSX = ICONSX + 1
      IF (IMODX.EQ.0) IPOINC(INUMTT) = 1
      CONSTX(ICONSX) = CONX
      WORDMX(ICONSX)  = WORDCX(ICARX)(1:IENDP)//' -1'
  200 CONTINUE
C  END OF USRCON
C
C
      RETURN
      END
C
C
C
C=DECK      USRLNK
      SUBROUTINE USRLNK(VARI,I,VARIAB)
```

```
C Purpose: generate user-written
C linking conditions using any
C combination of decision variables.
C You must write conde that, using
C the variables in the subroutine
C argument VARIAB as input, ultimately
C yield a value for the linked variable
C VARI.
C
C VARI is the Ith entry of the array
C VARIAB.  You have decided that this
C is to be a linked variable with user
C defined linking.  It is linked to
C the decision variables in the array
C VARIAB.
C An example will provide the simplest
C explanation of this:
C Let"s say that the 5th decision
C variable candidate (I=5) is linked
C to the decision variable candidates
C 2 and 7. (You used DECIDE to select
C these as decision variables.
C In this case VARI is equal to
C VARIAB(I).  You then write your
C linking equation in the form
C VARI=f(VARIAB(2),VARIAB(7)).
C Use the index I in an IF statement if
C you have more than one user-defined
C linked variable.
C
C
      REAL VARI,VARIAB(50)
      INTEGER I
C
C  INSERT USER-WRITTEN DECLARATION
C  STATEMENTS HERE.
C
C  INSERT USER-WRITTEN
C  STATEMENTS HERE.
C
C
C  END OF USRLNK
      RETURN
      END
C=DECK      OBJECT
      SUBROUTINE OBJECT(IFILE,NPRINX,IMODX,OBJGEN,PHRASE)
C   PURPOSE:weight of the equivalent ellipsoidal head
C
```

```
C   YOU MUST WRITE CODE THAT, USING
C   THE VARIABLES IN THE LABELLED
C   COMMON BLOCKS AS INPUT, ULTIMATELY
C   YIELDS THE OBJECTIVE FUNCTION
C         WEIGHT
C   AS OUTPUT. MAKE SURE TO INCLUDE AT
C   THE END OF THE SUBROUTINE, THE
C   STATEMENT: OBJGEN = WEIGHT
C
C
C   DEFINITIONS OF INPUT DATA:
C    IMODX  = DESIGN CONTROL INTEGER:
C     IMODX = 0 MEANS BASELINE DESIGN
C     IMODX = 1 MEANS PERTURBED DESIGN
C     IFAST = 0 MEANS FEW  SHORTCUTS FOR PERTURBED DESIGNS
C     IFAST = 1 MEANS MORE SHORTCUTS FOR PERTURBED DESIGNS
C    IFILE = FILE FOR OUTPUT LIST:
C    NPRINX= OUTPUT CONTROL INTEGER:
C     NPRINX=0 MEANS SMALLEST AMOUNT
C     NPRINX=1 MEANS MEDIUM AMOUNT
C     NPRINX=2 MEANS LOTS OF OUTPUT
C
C   DEFINITION OF PHRASE:
C     PHRASE = weight of the equivalent ellipsoidal head
C
       CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      COMMON/FV01/xinput(21),Ixinpu
      REAL xinput
      COMMON/FV02/ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      REAL ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      COMMON/FV05/THKSKN(21),HIGHST(21)
      REAL THKSKN,HIGHST
      COMMON/FV16/PRESS(20)
      REAL PRESS
      COMMON/FV19/CLAPS1(20),CLAPS1A(20),CLAPS1F(20)
      REAL CLAPS1,CLAPS1A,CLAPS1F
      COMMON/FV22/GENBK1(20),GENBK1A(20),GENBK1F(20)
      REAL GENBK1,GENBK1A,GENBK1F
      COMMON/FV25/SKNBK1(20,10),JSKNBK1,SKNBK1A(20,10),SKNBK1F(20,10)
      REAL SKNBK1,SKNBK1A,SKNBK1F
      COMMON/FV28/STFBK1(20,10),STFBK1A(20,10),STFBK1F(20,10)
      REAL STFBK1,STFBK1A,STFBK1F
      COMMON/FV31/SKNST1(20,10),SKNST1A(20,10),SKNST1F(20,10)
      REAL SKNST1,SKNST1A,SKNST1F
      COMMON/FV34/STFST1(20,10),STFST1A(20,10),STFST1F(20,10)
      REAL STFST1,STFST1A,STFST1F
      COMMON/FV37/WAPEX1(20),WAPEX1A(20),WAPEX1F(20)
```

```
      REAL WAPEX1,WAPEX1A,WAPEX1F
      COMMON/FV40/CLAPS2(20),CLAPS2A(20),CLAPS2F(20)
      REAL CLAPS2,CLAPS2A,CLAPS2F
      COMMON/FV43/GENBK2(20),GENBK2A(20),GENBK2F(20)
      REAL GENBK2,GENBK2A,GENBK2F
      COMMON/FV46/SKNBK2(20,10),JSKNBK2,SKNBK2A(20,10),SKNBK2F(20,10)
      REAL SKNBK2,SKNBK2A,SKNBK2F
      COMMON/FV49/STFBK2(20,10),STFBK2A(20,10),STFBK2F(20,10)
      REAL STFBK2,STFBK2A,STFBK2F
      COMMON/FV52/SKNST2(20,10),SKNST2A(20,10),SKNST2F(20,10)
      REAL SKNST2,SKNST2A,SKNST2F
      COMMON/FV55/STFST2(20,10),STFST2A(20,10),STFST2F(20,10)
      REAL STFST2,STFST2A,STFST2F
      COMMON/FV58/WAPEX2(20),WAPEX2A(20),WAPEX2F(20)
      REAL WAPEX2,WAPEX2A,WAPEX2F
      COMMON/IV01/npoint,nodes,IMODE
      INTEGER npoint,nodes,IMODE
      COMMON/FV11/LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
      REAL LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
C
C
C  INSERT SUBROUTINE STATEMENTS HERE.
C
C
      OBJGEN =WEIGHT
C
C
      RETURN
      END
================================================================
```