March 1, 2011

First, read the paper by David Bushnell:
GENOPT-A PROGRAM THAT WRITES USER-FRIENDLY OPTIMIZATION CODE,
International Journal of Solids and Structures, Vol. 26. No. 9/10, pp. 1173-1210, 1990 (a special issue of invited papers compiled in memory of Professor Charles Babcock, formerly at California Institute of Technology)

Also, read the file, .../genopt/doc/getting.started

SAMPLE RUN STREAM FOR GENOPT

Commands from the user are in **16 pt bold face**. Note: the string, **"bush->"**, is not part of the command typed by the user.

TABLE OF CONTENTS FOR THE GENOPT RUN STREAM
FOR THE CASE WITH THE GENERIC NAME "plate"
AND THE SPECIFIC NAME, "plate1"

First, the tasks performed by the GENOPT user (PARTs 1-9)

Second, the tasks performed by the END user (PARTs 10 - 21)

PART 21 THE END USER EXECUTES "CLEANGEN" IN ORDER TO
        CLEAN UP THE FILES WITH THE GENERIC CASE NAME, "plate"

PART 1 INTRODUCTION

This sample run is for the same case, "plate/plate1",
described in the following paper:

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

There are several differences in the case because GENOPT
has been changed quite a lot since that paper was
published in 1990.

Generic case name = plate
Specific case name = plate1

Now start processing...

PART 2 ACTIVATE THE GENOPT SET OF COMMANDS

bush-> genoptlog

GENOPT commands have been activated.

```
  gentext      GENOPT user generates a prompt file.
  genprograms  GENOPT user generates (makes) executables:
               begin, decide, mainsetup, optimize,
               change, chooseplot, and diplot.
  begin        End user provides starting data.
  decide       End user chooses decision variables, bounds,
               linked variables, and inequality constraints.
  mainsetup    End user sets up strategy parameters.
  optimize     End user performs optimization.
  change       End user changes some parameters.
  autochange   New values for decision variables randomly
  superopt     End user find global optimum (autochange/optimize)...
  chooseplot   End user chooses which variable to plot vs.
               iterations.
  diplot       End user plots variables vs. iterations.
  insert       GENOPT user adds parameters to the problem.
  cleangen     GENOPT user cleans up GENeric case files.
```

```
  cleanspec    End user cleans up SPECific case files.

Please consult the following sources for more about GENOPT:
  1.  genopt.story, howto.run, and genopt.news (in the
      .../genopt/doc directory)
  2.  Sample cases, plate.cas and panel.cas (in the
      .../genopt/case directory)
  3.  NAME.DEF file, where NAME is the generic case name.
  4.  GENOPT.HLP file   (Type "helpg".)
```

PART 3 EXECUTE GENTEXT (a long interactive session)

bush-> gentext

```
  ENTER THE GENERIC CASE NAME: plate
  ARE YOU CORRECTING, ADDING TO, OR USING plate.INP ? (TYPE y  OR  n): n

 The purpose of GENOPT (GENeral OPTimization) is to generate a
 user-friendly system of programs for optimizing anything you
 want. GENOPT is designed to handle problems with small data
 bases, not large finite element models.  Before you start
 using GENOPT please read the file GENOPT.STORY and study the
 cases, located in the subdirectories under genopt/case.

 You have chosen the following name for this case:  temp.
 Henceforth, this will be called the "generic" name. After you
 have completed your tasks in GENOPT, you and other users
 will be able to optimize specific things that fit within the
 class of optimization problems that you have called temp.

 Your tasks in GENOPT are:
    1. To provide variable names, definitions and helping
       paragraphs that will make it easy for others to
       optimize things that fit within the problem class temp

    2. To complete subroutines BEHX1, BEHX2, BEHX3,. . .BEHXn
       which calculate temp behavior for a given design;
       to complete SUBROUTINE OBJECT, which calculates the
       objective to be minimized; and/or to add any other
       subroutines, common blocks, utilities, etc. that you
       may wish.

    2b.Possibly "flesh out" SUBROUTINE STRUCT (struct.new
       library) instead of or in addition to "fleshing out"
       the "behavioral" subroutines, BEHXi. See Ref. [2]
       listed in temp.DEF for an example in which both SUB.
       STRUCT and the "behavior" routines, BEHXi, were
       "fleshed out". See Ref. [2b] in temp.DEF for an
       example in which SUBROUTINE STRUCT was "fleshed out"
       but the "behavioral" routines, BEHXi, were left in
       their skeletal forms just as GENOPT automatically
       created them.
```

3. To compile and collect all pertinent software that
   both you and GENOPT have written in order to generate
   the absolute elements: BEGIN.EXE, DECIDE.EXE,
   MAINSETUP.EXE, OPTIMIZE.EXE, and CHANGE.EXE.

```
  STARTING PROMPT INDEX (integer from 1 to 10.  Try 5):5
  K=      5
  INCREMENT FOR PROMPT INDEX (integer from 1 to 10. Try 5):5
  DK=      5

  CHOOSE: 0  or  1:
  (0 means "introductory explanatory text")
  (1 means "one-line input datum prompt")
  CHOOSE: 0  or  1:   TYPE=      0
  TYPE=      0
  PROVIDE INTRODUCTORY EXPLANATORY TEXT
PROGRAM FOR OPTIMIZATION OF A RECTANGULAR PLATE
 PROGRAM FOR OPTIMIZATION OF A RECTANGULAR PLATE
  ANY MORE LINES IN THIS PARAGRAPH? (y  or  <cr> or  n): y
  ANSWER=y
SUBJECTED TO SEVERAL LOADING ENVIRONMENTS AND
 SUBJECTED TO SEVERAL LOADING ENVIRONMENTS AND
  ANY MORE LINES IN THIS PARAGRAPH? (y  or  <cr> or  n): y
  ANSWER=y
CONSTRAINTS ON STRESS, BUCKLING, DISPLACEMENTS, AND FREQUENCY.
 CONSTRAINTS ON STRESS, BUCKLING, DISPLACEMENTS, AND FREQUENCY.
  ANY MORE LINES IN THIS PARAGRAPH? (y  or  <cr> or  n): n
  ANSWER=n
  CHOOSE: 0  or  1:
  (0 means "introductory explanatory text")
  (1 means "one-line input datum prompt")
  CHOOSE: 0  or  1: 0
  TYPE=      0
  PROVIDE INTRODUCTORY EXPLANATORY TEXT
FIRST, PROVIDE ALL VARIABLES THAT CAN BE DECISION VARIABLES,
 FIRST, PROVIDE ALL VARIABLES THAT CAN BE DECISION VARIABLES,
  ANY MORE LINES IN THIS PARAGRAPH? (y  or  <cr> or  n): y
  ANSWER=y
THAT IS, VARIABLES THAT CAN CHANGE DURING OPTIMIZATION
 THAT IS, VARIABLES THAT CAN CHANGE DURING OPTIMIZATION
  ANY MORE LINES IN THIS PARAGRAPH? (y  or  <cr> or  n): y
  ANSWER=y
ITERATIONS (ROLE TYPE 1), AND FIXED VARIABLES (ROLE TYPE 2).
 ITERATIONS (ROLE TYPE 1), AND FIXED VARIABLES (ROLE TYPE 2).
  ANY MORE LINES IN THIS PARAGRAPH? (y  or  <cr> or  n): n
  ANSWER=n
  CHOOSE: 0  or  1:
  (0 means "introductory explanatory text")
  (1 means "one-line input datum prompt")
  CHOOSE: 0  or  1: 1
  TYPE=      1


 ============================================================
 You will next be asked to provide information
 about a new variable that will play a role in your program.
 The following items relative to this new variable will be
 asked of you:
```

```
 1. A name of the variable (six characters or less).
 2. The role of the variable in your program.
 3. Is the variable an array? (If yes, give number of rows,
    NROWS and columns, NCOLS.)
 4. A one-line definition of the variable.
 5. Do you want to include a "help" paragraph that explains
    more about the variable than the one-line definition?
 6. If you answer 5. with Y, you provide a help paragraph.

 Hit RETURN.


The variable can have one of the following roles:

   1 = a possible decision variable for optimization,
       typically a dimension of a structure.
   2 = a constant parameter (cannot vary as design evolves),
       typically a control integer or a material property,
       but not a load, allowable, or factor of safety,
       which are asked for later.
   3 = a parameter characterizing the environment, such
       as a load component or a temperature.
   4 = a quantity that describes the response of the
       structure, (e.g. stress, buckling load, frequency)
   5 = an allowable, such as maximum allowable stress,
       minimum allowable frequency, etc.
   6 = a factor of safety
   7 = the design objective (e.g. weight)

NOTE: ALWAYS START WITH A ROLE = 1 OR A ROLE = 2 VARIABLE!
============================================================

 PROVIDE A NAME FOR THE VARIABLE (7 or less characters, CAPS):THICK
 VNAME=THICK
 IDENTIFY ROLE OF  THICK (1  or  2  or  3  or  4  or  5  or  6  or  7):
 1 = decision variable candidate(e.g.length, width, thickness)
 2 = fixed parameter (e.g. control integer, material property)
 3 = environmental factor (e.g. load, temperature)
 4 = response quantity (e.g. stress, buckling load, frequency)
 5 = allowable (e.g. maximum stress, minimum frequency)
 6 = factor of safety
 7 = objective (e.g. minimum weight, minimum cost)

 IDENTIFY ROLE OF  THICK (1  or  2  or  3  or  4  or  5  or  6  or  7):
 IROLEV=      1
 Is the variable  THICK   an array?   ANSWER=n
 PROVIDE A DEFINITION FOR  THICK.  (LESS THAN 60 CHARACTERS!)
THICK = thickness of the plate
 DO YOU WANT TO INCLUDE A "HELP" PARAGRAPH? (y  or  n):   ANSWER=n
 ANY MORE DECISION VARIABLE CANDIDATES     (ROLE 1 VARIABLES)
    OR    FIXED PARAMETERS (e.g. material) (ROLE 2 VARIABLES)?   ANSWER=y


 CHOOSE ANOTHER DECISION VARIABLE CANDIDATE (ROLE 1 VARIABLE)
       OR       FIXED PARAMETER             (ROLE 2 VARIABLE).


 CHOOSE: 0  or  1:
```

```
 (0 means "introductory explanatory text")
 (1 means "one-line input datum prompt")
 CHOOSE: 0  or  1:   TYPE=      1

 PROVIDE A NAME FOR THE VARIABLE (7 or less characters, CAPS):LENGTH
 VNAME=LENGTH

 IDENTIFY ROLE OF  THICK (1  or  2  or  3  or  4  or  5  or  6  or  7):
 1 = decision variable candidate(e.g.length, width, thickness)
 2 = fixed parameter (e.g. control integer, material property)
 3 = environmental factor (e.g. load, temperature)
 4 = response quantity (e.g. stress, buckling load, frequency)
 5 = allowable (e.g. maximum stress, minimum frequency)
 6 = factor of safety
 7 = objective (e.g. minimum weight, minimum cost)

 IDENTIFY ROLE OF  LENGTH (1  or  2  or  3  or  4  or  5  or  6  or  7):
 IROLEV=      1
 Is the variable  LENGTH   an array?   ANSWER=n
 PROVIDE A DEFINITION FOR  LENGTH.  (LESS THAN 60 CHARACTERS!)
LENGTH = Length of the plate
 DO YOU WANT TO INCLUDE A "HELP" PARAGRAPH? (y  or  n):   ANSWER=n
 ANY MORE DECISION VARIABLE CANDIDATES     (ROLE 1 VARIABLES)
    OR    FIXED PARAMETERS (e.g. material) (ROLE 2 VARIABLES)?   ANSWER=y


 CHOOSE ANOTHER DECISION VARIABLE CANDIDATE (ROLE 1 VARIABLE)
       OR       FIXED PARAMETER             (ROLE 2 VARIABLE).


 CHOOSE: 0  or  1:
 (0 means "introductory explanatory text")
 (1 means "one-line input datum prompt")
 CHOOSE: 0  or  1:   TYPE=      1

 PROVIDE A NAME FOR THE VARIABLE (7 or less characters, CAPS): WIDTH
 VNAME=WIDTH

 IDENTIFY ROLE OF  THICK (1  or  2  or  3  or  4  or  5  or  6  or  7):
 1 = decision variable candidate(e.g.length, width, thickness)
 2 = fixed parameter (e.g. control integer, material property)
 3 = environmental factor (e.g. load, temperature)
 4 = response quantity (e.g. stress, buckling load, frequency)
 5 = allowable (e.g. maximum stress, minimum frequency)
 6 = factor of safety
 7 = objective (e.g. minimum weight, minimum cost)

 IDENTIFY ROLE OF  WIDTH (1  or  2  or  3  or  4  or  5  or  6  or  7):
 IROLEV=      1
 Is the variable  WIDTH   an array?   ANSWER=n
 PROVIDE A DEFINITION FOR  WIDTH.  (LESS THAN 60 CHARACTERS!)
WIDTH = Width of the plate
 DO YOU WANT TO INCLUDE A "HELP" PARAGRAPH? (y  or  n):   ANSWER=n
 ANY MORE DECISION VARIABLE CANDIDATES     (ROLE 1 VARIABLES)
    OR    FIXED PARAMETERS (e.g. material) (ROLE 2 VARIABLES)?   ANSWER=y


 CHOOSE ANOTHER DECISION VARIABLE CANDIDATE (ROLE 1 VARIABLE)
```

```
       OR       FIXED PARAMETER             (ROLE 2 VARIABLE).

 CHOOSE: 0  or  1:
 (0 means "introductory explanatory text")
 (1 means "one-line input datum prompt")
 CHOOSE: 0  or  1:   TYPE=      1

 PROVIDE A NAME FOR THE VARIABLE (7 or less characters, CAPS): E
 VNAME=E

 IDENTIFY ROLE OF  THICK (1  or  2  or  3  or  4  or  5  or  6  or  7):
 1 = decision variable candidate(e.g.length, width, thickness)
 2 = fixed parameter (e.g. control integer, material property)
 3 = environmental factor (e.g. load, temperature)
 4 = response quantity (e.g. stress, buckling load, frequency)
 5 = allowable (e.g. maximum stress, minimum frequency)
 6 = factor of safety
 7 = objective (e.g. minimum weight, minimum cost)

 IDENTIFY ROLE OF  E (1  or  2  or  3  or  4  or  5  or  6  or  7):
 IROLEV=      2
 IDENTIFY TYPE FOR VARIABLE  E (1  or  2):
 ITYPEV=      2
 Is the variable  E   an array?   ANSWER=n
 PROVIDE A DEFINITION FOR  E.  (LESS THAN 60 CHARACTERS!)
E = Young's modulus of the plate material
 DO YOU WANT TO INCLUDE A "HELP" PARAGRAPH? (y  or  n):   ANSWER=n
 ANY MORE DECISION VARIABLE CANDIDATES     (ROLE 1 VARIABLES)
    OR    FIXED PARAMETERS (e.g. material) (ROLE 2 VARIABLES)?   ANSWER=y


 CHOOSE ANOTHER DECISION VARIABLE CANDIDATE (ROLE 1 VARIABLE)
       OR       FIXED PARAMETER             (ROLE 2 VARIABLE).


 CHOOSE: 0  or  1:   TYPE=      1
 PROVIDE A NAME FOR THE VARIABLE (7 or less characters, CAPS): NU
 VNAME=NU

 IDENTIFY ROLE OF  THICK (1  or  2  or  3  or  4  or  5  or  6  or  7):
 1 = decision variable candidate(e.g.length, width, thickness)
 2 = fixed parameter (e.g. control integer, material property)
 3 = environmental factor (e.g. load, temperature)
 4 = response quantity (e.g. stress, buckling load, frequency)
 5 = allowable (e.g. maximum stress, minimum frequency)
 6 = factor of safety
 7 = objective (e.g. minimum weight, minimum cost)

 IDENTIFY ROLE OF  NU (1  or  2  or  3  or  4  or  5  or  6  or  7):
 IROLEV=      2
 IDENTIFY TYPE FOR VARIABLE  NU (1  or  2):
 ITYPEV=      2
 Is the variable  NU   an array?   ANSWER=n
 PROVIDE A DEFINITION FOR  NU.  (LESS THAN 60 CHARACTERS!)
NU = Poisson's ratio of the plate material
 DO YOU WANT TO INCLUDE A "HELP" PARAGRAPH? (y  or  n):   ANSWER=n
 ANY MORE DECISION VARIABLE CANDIDATES     (ROLE 1 VARIABLES)
    OR    FIXED PARAMETERS (e.g. material) (ROLE 2 VARIABLES)?   ANSWER=y
```

```
 CHOOSE ANOTHER DECISION VARIABLE CANDIDATE (ROLE 1 VARIABLE)
       OR       FIXED PARAMETER              (ROLE 2 VARIABLE).


 CHOOSE: 0  or  1:   TYPE=      1
 PROVIDE A NAME FOR THE VARIABLE (7 or less characters, CAPS): RHO
 VNAME=RHO

 IDENTIFY ROLE OF  THICK (1  or  2  or  3  or  4  or  5  or  6  or  7):
 1 = decision variable candidate(e.g.length, width, thickness)
 2 = fixed parameter (e.g. control integer, material property)
 3 = environmental factor (e.g. load, temperature)
 4 = response quantity (e.g. stress, buckling load, frequency)
 5 = allowable (e.g. maximum stress, minimum frequency)
 6 = factor of safety
 7 = objective (e.g. minimum weight, minimum cost)

 IDENTIFY ROLE OF  RHO (1  or  2  or  3  or  4  or  5  or  6  or  7):
 IROLEV=      2
 IDENTIFY TYPE FOR VARIABLE  RHO (1  or  2):
 ITYPEV=      2
 Is the variable  RHO   an array?   ANSWER=n
 PROVIDE A DEFINITION FOR  RHO.  (LESS THAN 60 CHARACTERS!)
RHO = Weight density (e.g. lb/in**3) of the plate material
 DO YOU WANT TO INCLUDE A "HELP" PARAGRAPH? (y  or  n):   ANSWER=n
 ANY MORE DECISION VARIABLE CANDIDATES     (ROLE 1 VARIABLES)
    OR    FIXED PARAMETERS (e.g. material) (ROLE 2 VARIABLES)?   ANSWER=y


 CHOOSE ANOTHER DECISION VARIABLE CANDIDATE (ROLE 1 VARIABLE)
       OR       FIXED PARAMETER              (ROLE 2 VARIABLE).


 CHOOSE: 0  or  1:   TYPE=      0
 TYPE=      0
 PROVIDE INTRODUCTORY EXPLANATORY TEXT
THE FOLLOWING TABLE (FROM ROARK, 3RD EDITION, TABLE XVI,
 THE FOLLOWING TABLE (FROM ROARK, 3RD EDITION, TABLE XVI,
 ANY MORE LINES IN THIS PARAGRAPH? (y  or  <cr> or  n): y
 ANSWER=y
ENTRY NO. 1, PAGE 312), GIVES THE RELATIONSHIP OF PLATE
 ENTRY NO. 1, PAGE 312), GIVES THE RELATIONSHIP OF PLATE
 ANY MORE LINES IN THIS PARAGRAPH? (y  or  <cr> or  n): y
 ANSWER=y
(LENGTH/WIDTH) TO A COEFFICIENT FOR BUCKLING UNDER UNIFORM
 (LENGTH/WIDTH) TO A COEFFICIENT FOR BUCKLING UNDER UNIFORM
 ANY MORE LINES IN THIS PARAGRAPH? (y  or  <cr> or  n): y
 ANSWER=y
AXIAL COMPRESSION.
 AXIAL COMPRESSION.
 ANY MORE LINES IN THIS PARAGRAPH? (y  or  <cr> or  n): n
 ANSWER=n
 CHOOSE: 0  or  1:   TYPE=      1
 PROVIDE A NAME FOR THE VARIABLE (7 or less characters, CAPS): AOBAXL
 VNAME=AOBAXL
 IDENTIFY ROLE OF  AOBAXL (1  or  2  or  3  or  4  or  5  or  6  or  7):
```

```
 IROLEV=       2
 IDENTIFY TYPE FOR VARIABLE  AOBAXL (1  or  2):
 ITYPEV=       2
 Is the variable  AOBAXL   an array?   ANSWER=y
 Do you want to establish new dimension(s) for AOBAXL?
  If you answer "no", GENOPT will use the dimensions of the last
  array (       ) for which dimension(s) were established.
  The row index would be named "-------", with the definition:
   "-------".
  In this case AOBAXL would be dimensioned AOBAXL( 0).
  If you want either or both of the following:
     1. new dimension(s) for AOBAXL and/or
     2. new name(s) and definition(s) of the array indices
  then answer "yes".  If you are in doubt, then answer "yes".

 Do you want to establish new dimension(s) for AOBAXL?  ANSWER=y
 Array AOBAXL is of type 1 or 2:
  1 = vector (1-dim. array),  2 = matrix (2-d array): NDIMEN=  NDIMEN=      1
 The row index of AOBAXL is called IAOBAXL .
 PROVIDE A DEFINITION FOR  IAOBAXL.  (LESS THAN 60 CHARACTERS!)
Number of entries in the table of axial buckling v. a/b
 Maximum allowable number of rows in the array,  AOBAXL. NROWS=  NROWS=     30
 PROVIDE A DEFINITION FOR  AOBAXL.  (LESS THAN 60 CHARACTERS!)
AOBAXL = (plate length, LENGTH)/(plate width, WIDTH)
 DO YOU WANT TO INCLUDE A "HELP" PARAGRAPH? (y  or  n):   ANSWER=n
 ANY MORE DECISION VARIABLE CANDIDATES     (ROLE 1 VARIABLES)
    OR    FIXED PARAMETERS (e.g. material) (ROLE 2 VARIABLES)?   ANSWER=y


 CHOOSE ANOTHER DECISION VARIABLE CANDIDATE (ROLE 1 VARIABLE)
       OR       FIXED PARAMETER             (ROLE 2 VARIABLE).


 CHOOSE: 0  or  1:   TYPE=      1
 PROVIDE A NAME FOR THE VARIABLE (7 or less characters, CAPS): KAXL
 VNAME=KAXL
 IDENTIFY ROLE OF  KAXL (1  or  2  or  3  or  4  or  5  or  6  or  7):
 IROLEV=       2
 IDENTIFY TYPE FOR VARIABLE  KAXL (1  or  2):
 ITYPEV=       2
 Is the variable  KAXL   an array?   ANSWER=y
 Do you want to establish new dimension(s) for KAXL?
  If you answer "no", GENOPT will use the dimensions of the last
  array (AOBAXL ) for which dimension(s) were established.
  The row index would be named "IAOBAXL", with the definition:
   "Number of entries in the table of axial buckling v. a/b".
  In this case KAXL would be dimensioned KAXL(30).
  If you want either or both of the following:
     1. new dimension(s) for KAXL and/or
     2. new name(s) and definition(s) of the array indices
  then answer "yes".  If you are in doubt, then answer "yes".

 Do you want to establish new dimension(s) for KAXL?  ANSWER=n
 PROVIDE A DEFINITION FOR  KAXL.  (LESS THAN 60 CHARACTERS!)
KAXL = buckling coefficient: uniform axial compression
 DO YOU WANT TO INCLUDE A "HELP" PARAGRAPH? (y  or  n):   ANSWER=n
 ANY MORE DECISION VARIABLE CANDIDATES     (ROLE 1 VARIABLES)
    OR    FIXED PARAMETERS (e.g. material) (ROLE 2 VARIABLES)?   ANSWER=y
```

```
 CHOOSE ANOTHER DECISION VARIABLE CANDIDATE (ROLE 1 VARIABLE)
       OR       FIXED PARAMETER             (ROLE 2 VARIABLE).


 CHOOSE: 0  or  1:   TYPE=      0
 TYPE=      0
 PROVIDE INTRODUCTORY EXPLANATORY TEXT
THE FOLLOWING TABLE (FROM ROARK, 3RD EDITION, TABLE XVI,
 THE FOLLOWING TABLE (FROM ROARK, 3RD EDITION, TABLE XVI,
 ANY MORE LINES IN THIS PARAGRAPH? (y  or  <cr> or  n): y
 ANSWER=y
ENTRY NO. 10, PAGE 312), GIVES THE RELATIONSHIP OF PLATE
 ENTRY NO. 10, PAGE 312), GIVES THE RELATIONSHIP OF PLATE
 ANY MORE LINES IN THIS PARAGRAPH? (y  or  <cr> or  n): y
 ANSWER=y
(LENGTH/WIDTH) TO A COEFFICIENT FOR BUCKLING UNDER UNIFORM
 (LENGTH/WIDTH) TO A COEFFICIENT FOR BUCKLING UNDER UNIFORM
 ANY MORE LINES IN THIS PARAGRAPH? (y  or  <cr> or  n): y
 ANSWER=y
IN-PLANE SHEAR.
 IN-PLANE SHEAR.
 ANY MORE LINES IN THIS PARAGRAPH? (y  or  <cr> or  n): n
 ANSWER=n
 CHOOSE: 0  or  1:   TYPE=      1
 PROVIDE A NAME FOR THE VARIABLE (7 or less characters, CAPS): AOBSHR
 VNAME=AOBSHR

 IDENTIFY ROLE OF  FREQ (1  or  2  or  3  or  4  or  5  or  6  or  7):
 1 = decision variable candidate(e.g.length, width, thickness)
 2 = fixed parameter (e.g. control integer, material property)
 3 = environmental factor (e.g. load, temperature)
 4 = response quantity (e.g. stress, buckling load, frequency)
 5 = allowable (e.g. maximum stress, minimum frequency)
 6 = factor of safety
 7 = objective (e.g. minimum weight, minimum cost)
 IDENTIFY ROLE OF  AOBSHR (1  or  2  or  3  or  4  or  5  or  6  or  7):
 IROLEV=      2
 IDENTIFY TYPE FOR VARIABLE  AOBSHR (1  or  2):
 ITYPEV=      2
 Is the variable  AOBSHR   an array?   ANSWER=y
 Do you want to establish new dimension(s) for AOBSHR?
  If you answer "no", GENOPT will use the dimensions of the last
  array (AOBAXL ) for which dimension(s) were established.
  The row index would be named "IAOBAXL", with the definition:
   "Number of entries in the table of axial buckling v. a/b".
  In this case AOBSHR would be dimensioned AOBSHR(30).
  If you want either or both of the following:
     1. new dimension(s) for AOBSHR and/or
     2. new name(s) and definition(s) of the array indices
  then answer "yes".  If you are in doubt, then answer "yes".

 Do you want to establish new dimension(s) for AOBSHR?  ANSWER=y
 Array AOBSHR is of type 1 or 2:
  1 = vector (1-dim. array),  2 = matrix (2-d array): NDIMEN=  NDIMEN=      1
 The row index of AOBSHR is called IAOBSHR .
 PROVIDE A DEFINITION FOR  IAOBSHR.  (LESS THAN 60 CHARACTERS!)
```

```
Number of entries in the table of shear buckling v. a/b
 Maximum allowable number of rows in the array,  AOBSHR. NROWS=  NROWS=     20
 PROVIDE A DEFINITION FOR  AOBSHR.  (LESS THAN 60 CHARACTERS!)
AOBSHR = (plate length, LENGTH)/(plate width, WIDTH)
 DO YOU WANT TO INCLUDE A "HELP" PARAGRAPH? (y  or  n):   ANSWER=n
 ANY MORE DECISION VARIABLE CANDIDATES     (ROLE 1 VARIABLES)
    OR    FIXED PARAMETERS (e.g. material) (ROLE 2 VARIABLES)?   ANSWER=y


 CHOOSE ANOTHER DECISION VARIABLE CANDIDATE (ROLE 1 VARIABLE)
       OR       FIXED PARAMETER             (ROLE 2 VARIABLE).


 CHOOSE: 0  or  1:   TYPE=      1
 PROVIDE A NAME FOR THE VARIABLE (7 or less characters, CAPS): KSHR
 VNAME=KSHR
 IDENTIFY ROLE OF  KSHR (1  or  2  or  3  or  4  or  5  or  6  or  7):
 IROLEV=      2
 IDENTIFY TYPE FOR VARIABLE  KSHR (1  or  2):
 ITYPEV=      2
 Is the variable  KSHR   an array?   ANSWER=y
 Do you want to establish new dimension(s) for KSHR?
  If you answer "no", GENOPT will use the dimensions of the last
  array (AOBSHR ) for which dimension(s) were established.
  The row index would be named "IAOBSHR", with the definition:
   "Number of entries in the table of shear buckling v. a/b".
  In this case KSHR would be dimensioned KSHR(20).
  If you want either or both of the following:
     1. new dimension(s) for KSHR and/or
     2. new name(s) and definition(s) of the array indices
  then answer "yes".  If you are in doubt, then answer "yes".

 Do you want to establish new dimension(s) for KSHR?  ANSWER=n
 PROVIDE A DEFINITION FOR  KSHR.  (LESS THAN 60 CHARACTERS!)
KSHR = buckling coefficient: uniform in-plane shear
 DO YOU WANT TO INCLUDE A "HELP" PARAGRAPH? (y  or  n):   ANSWER=n
 ANY MORE DECISION VARIABLE CANDIDATES     (ROLE 1 VARIABLES)
    OR    FIXED PARAMETERS (e.g. material) (ROLE 2 VARIABLES)?   ANSWER=n


 CHOOSE AN ENVIRONMENTAL PARAMETER (load)   (ROLE 3 VARIABLE).
 CHOOSE: 0  or  1:   TYPE=      0
 TYPE=      0
 PROVIDE INTRODUCTORY EXPLANATORY TEXT
NEXT, PROVIDE ALL ENVIRONMENTAL PARAMETERS (LOADS, TEMPERATURES)
 NEXT, PROVIDE ALL ENVIRONMENTAL PARAMETERS (LOADS, TEMPERATURES)
 ANY MORE LINES IN THIS PARAGRAPH? (y  or  <cr> or  n): n
 ANSWER=n
 CHOOSE: 0  or  1:   TYPE=      1
 PROVIDE A NAME FOR THE VARIABLE (7 or less characters, CAPS): Nx
 VNAME=Nx

 IDENTIFY ROLE OF  FREQ (1  or  2  or  3  or  4  or  5  or  6  or  7):
 1 = decision variable candidate(e.g.length, width, thickness)
 2 = fixed parameter (e.g. control integer, material property)
 3 = environmental factor (e.g. load, temperature)
 4 = response quantity (e.g. stress, buckling load, frequency)
 5 = allowable (e.g. maximum stress, minimum frequency)
```

```
 6 = factor of safety
 7 = objective (e.g. minimum weight, minimum cost)
 IDENTIFY ROLE OF  Nx (1  or  2  or  3  or  4  or  5  or  6  or  7):
 IROLEV=      3
 PROVIDE A DEFINITION FOR  Nx.  (LESS THAN 60 CHARACTERS!)
Nx = Axial tension/width of the plate (lb/in)
 DO YOU WANT TO INCLUDE A "HELP" PARAGRAPH? (y  or  n):   ANSWER=y
 PROVIDE HELP PARAGRAPH TO EXPLAIN INPUT
NOTE:  Nx must be negative for axial compression!
 ANSWER=n
 ANY MORE ENVIRONMENTAL PARAMETERS (loads) (ROLE 3 VARIABLES)?   ANSWER=y


 CHOOSE ANOTHER ENVIRONMENTAL PARAMETER     (ROLE 3 VARIABLE).


 CHOOSE: 0  or  1:   TYPE=      1
 PROVIDE A NAME FOR THE VARIABLE (7 or less characters, CAPS): Ny
 VNAME=Ny
 IDENTIFY ROLE OF  Ny (1  or  2  or  3  or  4  or  5  or  6  or  7):
 IROLEV=      3
 PROVIDE A DEFINITION FOR  Ny.  (LESS THAN 60 CHARACTERS!)
Ny = Transverse tension/length of the plate (lb/in)
 DO YOU WANT TO INCLUDE A "HELP" PARAGRAPH? (y  or  n):   ANSWER=n
 ANY MORE ENVIRONMENTAL PARAMETERS (loads) (ROLE 3 VARIABLES)?   ANSWER=y


 CHOOSE ANOTHER ENVIRONMENTAL PARAMETER     (ROLE 3 VARIABLE).


 CHOOSE: 0  or  1:   TYPE=      1
 PROVIDE A NAME FOR THE VARIABLE (7 or less characters, CAPS): Nxy
 VNAME=Nxy
 IDENTIFY ROLE OF  Nxy (1  or  2  or  3  or  4  or  5  or  6  or  7):
 IROLEV=      3
 PROVIDE A DEFINITION FOR  Nxy.  (LESS THAN 60 CHARACTERS!)
Nxy = In-plane shear/length applied to the plate
 DO YOU WANT TO INCLUDE A "HELP" PARAGRAPH? (y  or  n):   ANSWER=n
 ANY MORE ENVIRONMENTAL PARAMETERS (loads) (ROLE 3 VARIABLES)?   ANSWER=y


 CHOOSE ANOTHER ENVIRONMENTAL PARAMETER     (ROLE 3 VARIABLE).


 CHOOSE: 0  or  1:   TYPE=      1
 PROVIDE A NAME FOR THE VARIABLE (7 or less characters, CAPS): PRESS
 VNAME=PRESS
 IDENTIFY ROLE OF  PRESS (1  or  2  or  3  or  4  or  5  or  6  or  7):
 IROLEV=      3
 PROVIDE A DEFINITION FOR  PRESS.  (LESS THAN 60 CHARACTERS!)
PRESS = Uniform normal pressure on the plate
 DO YOU WANT TO INCLUDE A "HELP" PARAGRAPH? (y  or  n):   ANSWER=n
 ANY MORE ENVIRONMENTAL PARAMETERS (loads) (ROLE 3 VARIABLES)?   ANSWER=n


 CHOOSE A RESPONSE PARAMETER (e.g. stress)  (ROLE 4 VARIABLE).
 CHOOSE: 0  or  1:   TYPE=      0
 TYPE=      0
```

```
 PROVIDE INTRODUCTORY EXPLANATORY TEXT
NEXT, PROVIDE RESPONSE PARAMETERS, ALLOWABLES, AND FACTORS OF
 NEXT, PROVIDE RESPONSE PARAMETERS, ALLOWABLES, AND FACTORS OF
 ANY MORE LINES IN THIS PARAGRAPH? (y  or  <cr> or  n): y
 ANSWER=y
SAFETY. THE ORDER IN WHICH YOU MUST PROVIDE THESE DATA IS:
 SAFETY. THE ORDER IN WHICH YOU MUST PROVIDE THESE DATA IS:
 ANY MORE LINES IN THIS PARAGRAPH? (y  or  <cr> or  n): y
 ANSWER=y
A,B,C; A,B,C; A,B,C; A,B,C; etc, in which
 A,B,C; A,B,C; A,B,C; A,B,C; etc, in which
 ANY MORE LINES IN THIS PARAGRAPH? (y  or  <cr> or  n): y
 ANSWER=y
A = RESPONSE,   B = ALLOWABLE,   C = FACTOR OF SAFETY.
 A = RESPONSE,   B = ALLOWABLE,   C = FACTOR OF SAFETY.
 ANY MORE LINES IN THIS PARAGRAPH? (y  or  <cr> or  n): n
 ANSWER=n
 CHOOSE: 0  or  1:   TYPE=      1
 PROVIDE A NAME FOR THE VARIABLE (7 or less characters, CAPS): STRESS
 VNAME=STRESS

 IDENTIFY ROLE OF  FREQ (1  or  2  or  3  or  4  or  5  or  6  or  7):
 1 = decision variable candidate(e.g.length, width, thickness)
 2 = fixed parameter (e.g. control integer, material property)
 3 = environmental factor (e.g. load, temperature)
 4 = response quantity (e.g. stress, buckling load, frequency)
 5 = allowable (e.g. maximum stress, minimum frequency)
 6 = factor of safety
 7 = objective (e.g. minimum weight, minimum cost)
 IDENTIFY ROLE OF  STRESS (1  or  2  or  3  or  4  or  5  or  6  or  7):
 IROLEV=      4
STRESS is an array with the number of rows equal to 20 .
 Each row corresponds to a load case.
 Do you want to reset the number of columns in STRESS?
 In this case, if you answer "no" STRESS would be dimensioned STRESS(20).
 Do you want to reset the number of columns in STRESS?  ANSWER=n
 PROVIDE A DEFINITION FOR  STRESS.  (LESS THAN 60 CHARACTERS!)
STRESS = Maximum effective (von Mises) stress
 DO YOU WANT TO INCLUDE A "HELP" PARAGRAPH? (y  or  n):   ANSWER=n


 CHOOSE AN ALLOWABLE FOR STRESS (ROLE 5 VARIABLE).

 CHOOSE: 0  or  1:   TYPE=      1
 PROVIDE A NAME FOR THE VARIABLE (7 or less characters, CAPS): MAXSTR
 VNAME=MAXSTR

 IDENTIFY ROLE OF  FREQ (1  or  2  or  3  or  4  or  5  or  6  or  7):
 1 = decision variable candidate(e.g.length, width, thickness)
 2 = fixed parameter (e.g. control integer, material property)
 3 = environmental factor (e.g. load, temperature)
 4 = response quantity (e.g. stress, buckling load, frequency)
 5 = allowable (e.g. maximum stress, minimum frequency)
 6 = factor of safety
 7 = objective (e.g. minimum weight, minimum cost)
 IDENTIFY ROLE OF  MAXSTR (1  or  2  or  3  or  4  or  5  or  6  or  7):
 IROLEV=      5
MAXSTR is an array with the number of rows equal to 20 .
```

```
  Each row corresponds to a load case.
  PROVIDE A DEFINITION FOR  MAXSTR.  (LESS THAN 60 CHARACTERS!)
 MAXSTR = Maximum effective (von Mises) stress allowed
  DO YOU WANT TO INCLUDE A "HELP" PARAGRAPH? (y  or  n):   ANSWER=n


  CHOOSE FACTOR OF SAFETY FOR STRESS (ROLE 6 VARIABLE).


  CHOOSE: 0  or  1:   TYPE=      1
  PROVIDE A NAME FOR THE VARIABLE (7 or less characters, CAPS): FSTRES
  VNAME=FSTRES

  IDENTIFY ROLE OF  FREQ (1  or  2  or  3  or  4  or  5  or  6  or  7):
  1 = decision variable candidate(e.g.length, width, thickness)
  2 = fixed parameter (e.g. control integer, material property)
  3 = environmental factor (e.g. load, temperature)
  4 = response quantity (e.g. stress, buckling load, frequency)
  5 = allowable (e.g. maximum stress, minimum frequency)
  6 = factor of safety
  7 = objective (e.g. minimum weight, minimum cost)
  IDENTIFY ROLE OF  FSTRES (1  or  2  or  3  or  4  or  5  or  6  or  7):
  IROLEV=      6
 FSTRES is an array with the number of rows equal to 20 .
  Each row corresponds to a load case.
  PROVIDE A DEFINITION FOR  FSTRES.  (LESS THAN 60 CHARACTERS!)
 FSTRES = Factor of safety for effective stress
  DO YOU WANT TO INCLUDE A "HELP" PARAGRAPH? (y  or  n):   ANSWER=n
  PLEASE CHOOSE  1  OR  2 OR 3  FOR THIS CONSTRAINT (STRESS):
TYPE OF CONSTRAINT=      1
  ANY MORE RESPONSE VARIABLES (e.g.buckling) (ROLE 4 VARIABLES)?   ANSWER=y


  CHOOSE ANOTHER RESPONSE VARIABLE           (ROLE 4 VARIABLE).


  CHOOSE: 0  or  1:   TYPE=      1
  PROVIDE A NAME FOR THE VARIABLE (7 or less characters, CAPS): BUCKLE
  VNAME=BUCKLE
  IDENTIFY ROLE OF  BUCKLE (1  or  2  or  3  or  4  or  5  or  6  or  7):
  IROLEV=      4
 BUCKLE is an array with the number of rows equal to 20 .
  Each row corresponds to a load case.
  Do you want to reset the number of columns in BUCKLE?
  In this case, if you answer "no" BUCKLE would be dimensioned BUCKLE(20).
  Do you want to reset the number of columns in BUCKLE?  ANSWER=n
  PROVIDE A DEFINITION FOR  BUCKLE.  (LESS THAN 60 CHARACTERS!)
 BUCKLE = Buckling load factor
  DO YOU WANT TO INCLUDE A "HELP" PARAGRAPH? (y  or  n):   ANSWER=n


  CHOOSE AN ALLOWABLE FOR BUCKLE (ROLE 5 VARIABLE).

  CHOOSE: 0  or  1:   TYPE=      1
  PROVIDE A NAME FOR THE VARIABLE (7 or less characters, CAPS): MINBUC
  VNAME=MINBUC
  IDENTIFY ROLE OF  MINBUC (1  or  2  or  3  or  4  or  5  or  6  or  7):
  IROLEV=      5
```

```
 MINBUC is an array with the number of rows equal to 20 .
  Each row corresponds to a load case.
  PROVIDE A DEFINITION FOR  MINBUC.  (LESS THAN 60 CHARACTERS!)
 MINBUC = Minimum allowable buckling load factor (use 1.0)
  DO YOU WANT TO INCLUDE A "HELP" PARAGRAPH? (y  or  n):   ANSWER=n


  CHOOSE FACTOR OF SAFETY FOR BUCKLE (ROLE 6 VARIABLE).


  CHOOSE: 0  or  1:   TYPE=      1
  PROVIDE A NAME FOR THE VARIABLE (7 or less characters, CAPS): FBUCKL
  VNAME=FBUCKL
  IDENTIFY ROLE OF  FBUCKL (1  or  2  or  3  or  4  or  5  or  6  or  7):
  IROLEV=      6
 FBUCKL is an array with the number of rows equal to 20 .
  Each row corresponds to a load case.
  PROVIDE A DEFINITION FOR  FBUCKL.  (LESS THAN 60 CHARACTERS!)
 FBUCKL = Factor of safety for buckling load factor
  DO YOU WANT TO INCLUDE A "HELP" PARAGRAPH? (y  or  n):   ANSWER=n
  PLEASE CHOOSE  1  OR  2 OR 3  FOR THIS CONSTRAINT (BUCKLE):
TYPE OF CONSTRAINT=      2
  ANY MORE RESPONSE VARIABLES (e.g.buckling) (ROLE 4 VARIABLES)?   ANSWER=y


  CHOOSE ANOTHER RESPONSE VARIABLE           (ROLE 4 VARIABLE).


  CHOOSE: 0  or  1:   TYPE=      1
  PROVIDE A NAME FOR THE VARIABLE (7 or less characters, CAPS): FREQ
  VNAME=FREQ
  IDENTIFY ROLE OF  FREQ (1  or  2  or  3  or  4  or  5  or  6  or  7):
  IROLEV=      4
 FREQ is an array with the number of rows equal to 20 .
  Each row corresponds to a load case.
  Do you want to reset the number of columns in FREQ?
  In this case, if you answer "no" FREQ would be dimensioned FREQ(20).
  Do you want to reset the number of columns in FREQ?  ANSWER=n
  PROVIDE A DEFINITION FOR  FREQ.  (LESS THAN 60 CHARACTERS!)
 FREQ = Fundamental frequency of unloaded plate
  DO YOU WANT TO INCLUDE A "HELP" PARAGRAPH? (y  or  n):   ANSWER=n


  CHOOSE AN ALLOWABLE FOR FREQ (ROLE 5 VARIABLE).

  CHOOSE: 0  or  1:   TYPE=      1
  PROVIDE A NAME FOR THE VARIABLE (7 or less characters, CAPS): MINCPS
  VNAME=MINCPS
  IDENTIFY ROLE OF  MINCPS (1  or  2  or  3  or  4  or  5  or  6  or  7):
  IROLEV=      5
 MINCPS is an array with the number of rows equal to 20 .
  Each row corresponds to a load case.
  PROVIDE A DEFINITION FOR  MINCPS.  (LESS THAN 60 CHARACTERS!)
 MINCPS = Minimum allowable value for the fundamental frequency
  DO YOU WANT TO INCLUDE A "HELP" PARAGRAPH? (y  or  n):   ANSWER=n


  CHOOSE FACTOR OF SAFETY FOR FREQ (ROLE 6 VARIABLE).
```

```
  CHOOSE: 0  or  1:   TYPE=      1
  PROVIDE A NAME FOR THE VARIABLE (7 or less characters, CAPS): FSFREQ
  VNAME=FSFREQ
  IDENTIFY ROLE OF  FSFREQ (1  or  2  or  3  or  4  or  5  or  6  or  7):
  IROLEV=      6
 FSFREQ is an array with the number of rows equal to 20 .
  Each row corresponds to a load case.
  PROVIDE A DEFINITION FOR  FSFREQ.  (LESS THAN 60 CHARACTERS!)
 FSFREQ = Factor of safety for FREQ
  DO YOU WANT TO INCLUDE A "HELP" PARAGRAPH? (y  or  n):   ANSWER=n
  PLEASE CHOOSE  1  OR  2 OR 3  FOR THIS CONSTRAINT (FREQ):
TYPE OF CONSTRAINT=      2
  ANY MORE RESPONSE VARIABLES (e.g.buckling) (ROLE 4 VARIABLES)?   ANSWER=y


  CHOOSE ANOTHER RESPONSE VARIABLE           (ROLE 4 VARIABLE).


  CHOOSE: 0  or  1:   TYPE=      1
  PROVIDE A NAME FOR THE VARIABLE (7 or less characters, CAPS): W
  VNAME=W
  IDENTIFY ROLE OF  W (1  or  2  or  3  or  4  or  5  or  6  or  7):
  IROLEV=      4
 W is an array with the number of rows equal to 20 .
  Each row corresponds to a load case.
  Do you want to reset the number of columns in W?
  In this case, if you answer "no" W would be dimensioned W(20).
  Do you want to reset the number of columns in W?  ANSWER=n
  PROVIDE A DEFINITION FOR  W.  (LESS THAN 60 CHARACTERS!)
 W = Normal deflection under uniform pressure
  DO YOU WANT TO INCLUDE A "HELP" PARAGRAPH? (y  or  n):   ANSWER=n


  CHOOSE AN ALLOWABLE FOR W (ROLE 5 VARIABLE).

  CHOOSE: 0  or  1:   TYPE=      1
  PROVIDE A NAME FOR THE VARIABLE (7 or less characters, CAPS): AW
  VNAME=AW
  IDENTIFY ROLE OF  AW (1  or  2  or  3  or  4  or  5  or  6  or  7):
  IROLEV=      5
 AW is an array with the number of rows equal to 20 .
  Each row corresponds to a load case.
  PROVIDE A DEFINITION FOR  AW.  (LESS THAN 60 CHARACTERS!)
 AW = Maximum allowable normal deflection under pressure
  DO YOU WANT TO INCLUDE A "HELP" PARAGRAPH? (y  or  n):   ANSWER=n


  CHOOSE FACTOR OF SAFETY FOR W (ROLE 6 VARIABLE).


  CHOOSE: 0  or  1:   TYPE=      1
  PROVIDE A NAME FOR THE VARIABLE (7 or less characters, CAPS): FW
  VNAME=FW
  IDENTIFY ROLE OF  FW (1  or  2  or  3  or  4  or  5  or  6  or  7):
  IROLEV=      6
 FW is an array with the number of rows equal to 20 .
```

```
  Each row corresponds to a load case.
  PROVIDE A DEFINITION FOR  FW.  (LESS THAN 60 CHARACTERS!)
 FW = Factor of safety for max deflection under pressure
  DO YOU WANT TO INCLUDE A "HELP" PARAGRAPH? (y  or  n):   ANSWER=n
  PLEASE CHOOSE  1  OR  2 OR 3  FOR THIS CONSTRAINT (W):
TYPE OF CONSTRAINT=      1
  ANY MORE RESPONSE VARIABLES (e.g.buckling) (ROLE 4 VARIABLES)?   ANSWER=n


  CHOOSE AN OBJECTIVE (e.g. minimum weight)  (ROLE 7 VARIABLE).

  CHOOSE: 0  or  1:   TYPE=      0
  TYPE=      0
  PROVIDE INTRODUCTORY EXPLANATORY TEXT
 LAST, AN OBJECTIVE MUST BE CHOSEN, SUCH AS MINIMUM WEIGHT
  LAST, AN OBJECTIVE MUST BE CHOSEN, SUCH AS MINIMUM WEIGHT
  ANY MORE LINES IN THIS PARAGRAPH? (y  or  <cr> or  n): y
  ANSWER=y
 OR MINIMUM COST.
  OR MINIMUM COST.
  ANY MORE LINES IN THIS PARAGRAPH? (y  or  <cr> or  n): n
  ANSWER=n
  CHOOSE: 0  or  1:   TYPE=      1
  PROVIDE A NAME FOR THE VARIABLE (7 or less characters, CAPS): WEIGHT
  VNAME=WEIGHT

  IDENTIFY ROLE OF  FREQ (1  or  2  or  3  or  4  or  5  or  6  or  7):
  1 = decision variable candidate(e.g.length, width, thickness)
  2 = fixed parameter (e.g. control integer, material property)
  3 = environmental factor (e.g. load, temperature)
  4 = response quantity (e.g. stress, buckling load, frequency)
  5 = allowable (e.g. maximum stress, minimum frequency)
  6 = factor of safety
  7 = objective (e.g. minimum weight, minimum cost)
  IDENTIFY ROLE OF  WEIGHT (1  or  2  or  3  or  4  or  5  or  6  or  7):
  IROLEV=      7
  PROVIDE A DEFINITION FOR  WEIGHT.  (LESS THAN 60 CHARACTERS!)
 WEIGHT = Weight of the plate
  DO YOU WANT TO INCLUDE A "HELP" PARAGRAPH? (y  or  n):   ANSWER=n


  DUMMY ENTRY WRITTEN AT END OF plate.PRO


  stoget.new has been created.
  begin.new has been created.
  struct.new has been created.
  behavior.new has been created.
  change.new has been created.
```

The purpose of GENTEXT is to generate a file of prompting phrases and helps called plate.PRO and five FORTRAN source libraries, BEGIN.NEW, STOGET.NEW, STRUCT.NEW, BEHAVIOR.NEW, and CHANGE.NEW. The purposes of these files are as follows:

==

TABLE 4   FILE OF PROMPTING PHRASES AND HELPS AND

```
          SOURCE CODE LIBRARIES GENERATED BY "GENTEXT"
==============================================================
 plate.PRO    = prompt file for input data for the problem
                class that you wish to set up for optimization.
                When BEGIN asks you for the name of the generic
                file, you should respond in this case with plate.

                The Prompt Numbers listed in TABLE 2 correspond
                to the prompts in this file.

 BEGIN.NEW    = source library for FORTRAN program which will
                be used to set up the starting design, material
                properties, and any other data you wish.

 STOGET.NEW   = source library for FORTRAN subroutines which
                are used to transfer labelled common blocks.
                These labelled common blocks are the data base.

 STRUCT.NEW   = source library for FORTRAN subroutines that
                perform the analysis for each iterate in the
                set of optimization iterations.  You may have
                to complete this routine (add dimension state-
                ments, subroutine calls, output statements,
                etc.).  The library, STRUCT.NEW, also contains
                a skeletal routine, SUB. TRANFR, that you can
                complete in order to translate data names from
                from those just established by you (TABLE 2) to
                other names used by the developer of previously
                written code that you may plan to incorporate
                into SUBROUTINE STRUCT and/or SUBROUTINES
                BEHX1, BEHX2, BEHX3,...BEHXn (described next).


 BEHAVIOR.NEW= a library of subroutine skeletons, BEHX1,BEHX2,
                BEHX3,...BEHXn, that, upon completion by you,
                will calculate behavior for a given design or
                design perturbation.  Skeletal subroutines for
                a user-written constraint condition, USRCON,
                and a skeletal routine for the objective func-
                tion, OBJECT, are also generated and are
                included in the BEHAVIOR.NEW library.

 CHANGE.NEW   = FORTRAN program that permits you to change
                certain program parameters without having to
                go back to BEGIN and run a case from scratch.
=============================================================

OTHER FILES CREATED BY THIS INTERACTIVE RUN:
  plate.INP     = A file that contains the input data that you
                have just provided.  This file becomes very
                useful if you make a mistake while typing
                input data. You can use it to avoid having
                to redo the input one datum at a time.
  plate.*       = several files that contain FORTRAN program
                fragments that are automatically concatenated
                to generate BEGIN.NEW, STOGET.NEW, STRUCT.NEW,
                BEHAVIOR.NEW and CHANGE.NEW
```

```
DESCRIPTION OF FILEs GENERATED BY THIS RUN:

plate.PRO    = Prompt file generated by this run.
plate.INP    = input data for GENTEXT generated by this run.
STOGET.NEW   = FORTRAN program for transfering common blocks
               to and from mass storage device.
BEGIN.NEW    = FORTRAN program for reading variables in an
               interactive mode.
STRUCT.NEW   = FORTRAN program for performing analysis.
BEHAVIOR.NEW = Library of skeletal subroutines for
               calculating response, user-written constraints,
               and the objective function.
CHANGE.NEW   = FORTRAN program that permits you to change
               certain program parameters without having to
               go back to BEGIN and run a case from scratch.
plate.etc   = various files with names such as NAME.PRO,

Next, if necessary, provide the algorithms called for in the
skeletal subroutines listed in the library BEHAVIOR.NEW. You
may find useful routines in the library UTIL.NEW.  In the
example, "plate", a linear interpolator, INTERP, is used in
SUBROUTINE BEHX2 for getting buckling loads v. aspect ratio.

Or, if necessary, complete the skeletal routines STRUCT and
TRANFR.  (You may find useful routines in UTIL.NEW).
If you are adding subroutine calls to SUBROUTINE STRUCT or
SUBROUTINE TRANFR, store the subroutines themselves in
the libraries called ADDCODEn.NEW, n = 1,2,3,...5. (Please
list one of the ADDCODEn.NEW libraries for instructions.)
Please see the example PANEL.CAS for how this is done.


After you have done this,  give the command GENPROGRAMS.
GENPROGRAMS will generate the absolute elements needed to
optimize whatever you have chosen as your objective (see
OBJECT routine in BEHAVIOR.NEW) in the presence of whatever
behavior or other factors (e.g. clearance) are quantified by
user-written subroutines collected in the libraries
ADDCODEn.NEW  and/or algorithms added to the skeletal
routines in the library BEHAVIOR.NEW .


  PLEASE PRINT OUT THE FILE plate.DEF IN ORDER TO OBTAIN A
  SUMMARY OF THE PURPOSES OF GENOPT, PROGRAM VARIABLE NAMES,
  DEFINITIONS, ROLES, AND OTHER USEFUL INFORMATION FOR WORK
  THAT YOU MUST DO NEXT.  YOU DO NOT HAVE TO COMMIT TO MEMORY
  THE INSTRUCTIONS JUST PRINTED. THEY ARE LISTED IN THE

                        plate.DEF

   FILE.
-------- end of the GENTEXT interactive session ---------
```

(The "plate" files that we now have in our working directory, .../genoptcase are as follows:

```
bush-> ls -al plate.*
-rw-r--r--  1 bush bush  2750 Mar  2 15:46 plate.CHA
-rw-r--r--  1 bush bush   703 Mar  2 15:46 plate.COM
-rw-r--r--  1 bush bush  5781 Mar  2 15:46 plate.CON
-rw-r--r--  1 bush bush 14074 Mar  2 15:46 plate.DAT
-rw-r--r--  1 bush bush 32268 Mar  2 15:46 plate.DEF
-rw-r--r--  1 bush bush 13908 Mar  2 10:08 plate.INP
-rw-r--r--  1 bush bush 15346 Mar  2 15:46 plate.NEW
-rw-r--r--  1 bush bush  3003 Mar  2 15:46 plate.PRO
-rw-r--r--  1 bush bush   933 Mar  2 15:46 plate.REA
-rw-r--r--  1 bush bush   147 Mar  2 15:46 plate.SET
-rw-r--r--  1 bush bush 10963 Mar  2 15:46 plate.SUB
-rw-r--r--  1 bush bush   933 Mar  2 15:46 plate.WRI
```

```
   TABLE 6 FROM ../genopt/doc/genopt.story:
       CONTENTS OF THE FILES CREATED BY "GENTEXT"
======================================================================
 FILE NAME                 DEFINITION OF FILE CONTENTS
----------------------------------------------------------------------
 plate.NEW        Part of BEGIN.NEW that contains calls to
                      SUBROUTINE DATUM and SUBROUTINE GETVAR.
                      This coding permits the interactive input
                      for the starting design. TABLE 9 shows part
                      of this file.
 plate.INP        See TABLE 4. Input data for GENTEXT.
 plate.COM        Labelled common blocks generated specifically
                      for the user-developed class of problems.
 plate.WRI        Part of subroutine for writing labelled common
                      blocks in SUBROUTINE STORCM (in Library STOGET).
 plate.REA        Part of subroutine for reading labelled common
                      blocks in SUBROUTINE GETCOM (in Library STOGET).
 plate.SET        Part of SUBROUTINE SETUPC in which new values
                      are installed in labelled common blocks from
                      the array VAR(I), which contains the latest
                      values of all candidates for decision variables.
                      NOTE: SUBROUTINE SETUPC is in the BEGIN library.
 plate.CON        Calls to subroutines, BEHX1, BEHX2, BEHX3,...,
                      which calculate behavior such as stresses
                      modal frequencies, buckling loads, etc. for
                      a fixed design configuration.  PLATE.CON also
                      contains calls to CON, which generate the value
                      of the behavioral constraints corresponding
                      to BEHX1, BEHX2, BEHX3,...  PLATE.CON generates
                      phrases that identify the meaning of each
                      behavioral constraint.
 plate.SUB        Skeletal subroutines, BEHX1, BEHX2, ..., and the
                      skeletal objective function, OBJECT.
 plate.DEF        List of user-established variable names,
                      definitions, and roles that these variables play
                      in the user-generated program.  Also, contains
                      list of files created by GENTEXT and the
                      functions of these files.
 plate.CHA        Part of SUBROUTINE NEWPAR (called in the CHANGE
                      processor) in which labelled common values are
                      updated.
 plate.DAT        Image of interactive input for user-developed
                      program, generated to save time in case you make
                      a mistake during input. This file is used by
```

```
                      the INSERT processor.
========================================================================
```

PART 4 INSPECT THE plate.DEF FILE

(Next, Inspect the plate.DEF file. A list of the plate.DEF file follows:)

```
-------------------- plate.DEF file --------------------
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: plate

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 "plate".

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 "plate" 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 [2b] Bushnell, D., "Minimum weight design of imperfect
C isogrid-stiffened ellipsoidal shells under uniform external
C pressure", AIAA paper 2009-2702, 50th AIAA Structures
C Meeting, Palm Springs, CA, May 4-7, 2009

C [2c] Bushnell, D. and Thornburgh, R. P., "Use of GENOPT and
C BIGBOSOR4 to optimize weld lands in axially compressed
C stiffened cylindrical shells and evaluation of the optimized
C designs by STAGS", AIAA Paper 2010-2927, AIAA 51st Structures
C Meeting, Orlando, Florida, April 2010

C [2d] Bushnell, D., "Use of GENOPT and BIGBOSOR4 to obtain
C optimum designs of an axially compressed cylindrical shell
C with a composite truss-core sandwich wall", AIAA Paper 2011-
C 1811, 52nd AIAA Structures Meeting, Denver, CO, April, 2011

C [2e] Bushnell, D., "Use of GENOPT and BIGBOSOR4 to obtain
C an optimum design of a deep submergence tank", unpublished
C report to the DOER company, Alameda, CA, June 30, 2009

C [2f] Bushnell, D., "Use of GENOPT and BIGBOSOR4 to obtain
C optimum designs of a double-walled inflatable cylindrical
C vacuum chamber", unpublished report for Michael Mayo,
C November, 2010

C [2g] Bushnell, D., "Use of GENOPT and BIGBOSOR4 to obtain
C optimum designs of a double-walled inflatable spherical
C vacuum chamber", unpublished report for Michael Mayo,
C December, 2010

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 imperf.isogrid-stiffened
C ellipsoidal shells...", written and placed in the file
C ..genopt/case/torisph/sdm50.report.pdf

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, Ref.[2b]

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

C [15] Bushnell, D., the case ..genopt/case/wavycyl, Ref.[2]

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

C [17] Bushnell, D., the case ..genopt/case/weldland, Ref.[2c]

C [18] Bushnell, D., the case ..genopt/case/trusscomp,Ref.[2d]

C [19] Bushnell, D., the case ..genopt/case/submarine,Ref.[2e]

C [20] Bushnell, D., the case ..genopt/case/sphere

C [21] Bushnell, D., the case ..genopt/case/balloon

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 = plate)
C        4.  GENOPT.HLP file    (type HELPG)
C==============================================================

C===============================================================
C   TABLE 2   GLOSSARY OF VARIABLES USED IN "plate"
C===============================================================
C  ARRAY  NUMBER OF         PROMPT
C    ?   (ROWS,COLS)  ROLE  NUMBER   NAME                 DEFINITION OF VARIABLE
C                         (plate.PRO)
C===============================================================
C    n   (   0,   0)    1      15   THICK    = thickness of the plate
C    n   (   0,   0)    1      20   LENGTH   = Length of the plate
C    n   (   0,   0)    1      25   WIDTH    = Width of the plate
C    n   (   0,   0)    2      30   E        = Young's modulus of the plate material
C    n   (   0,   0)    2      35   NU       = Poisson's ratio of the plate material
C    n   (   0,   0)    2      40   RHO      = Weight density (e.g. lb/in**3) of the
plate material
C    n   (   0,   0)    2      50   IAOBAXL  = Number of entries in the table of axial
buckling v. a/b in AOBAXL(IAOBAXL)
C    y   (  30,   0)    2      55   AOBAXL   = (plate length, LENGTH)/(plate width,
WIDTH)
C    y   (  30,   0)    2      60   KAXL     = buckling coefficient: uniform axial
compression
C    n   (   0,   0)    2      70   IAOBSHR  = Number of entries in the table of shear
buckling v. a/b in AOBSHR(IAOBSHR)
C    y   (  20,   0)    2      75   AOBSHR   = (plate length, LENGTH)/(plate width,
WIDTH)
C    y   (  20,   0)    2      80   KSHR     = buckling coefficient: uniform in-plane
shear
C    n   (   0,   0)    2      90   NCASES   = Number of load cases (number of
environments)  in Nx(NCASES)
C    y   (  20,   0)    3      95   Nx       = Axial tension/width of the plate (lb/in)
C    y   (  20,   0)    3     100   Ny       = Transverse tension/length of the plate
(lb/in)
C    y   (  20,   0)    3     105   Nxy      = In-plane shear/length applied to the plate
C    y   (  20,   0)    3     110   PRESS    = Uniform normal pressure on the plate
C    y   (  20,   0)    4     120   STRESS   = Maximum effective (von Mises) stress
C    y   (  20,   0)    5     125   MAXSTR   = Maximum effective (von Mises) stress
allowed
```

```
C    y   (  20,   0)    6     130   FSTRES   = Factor of safety for effective stress
C    y   (  20,   0)    4     135   BUCKLE   = Buckling load factor
C    y   (  20,   0)    5     140   MINBUC   = Minimum allowable buckling load factor
(use 1.0)
C    y   (  20,   0)    6     145   FBUCKL   = Factor of safety for buckling load factor
C    y   (  20,   0)    4     150   FREQ     = Fundamental frequency of unloaded plate
C    y   (  20,   0)    5     155   MINCPS   = Minimum allowable value for the
fundamental frequency
C    y   (  20,   0)    6     160   FSFREQ   = Factor of safety for FREQ
C    y   (  20,   0)    4     165   W        = Normal deflection under uniform pressure
C    y   (  20,   0)    5     170   AW       = Maximum allowable normal deflection under
pressure
C    y   (  20,   0)    6     175   FW       = Factor of safety for max deflection under
pressure
C    n   (   0,   0)    7     185   WEIGHT   = Weight of the plate
C
C=============================================================
C       TABLE 3   SEVEN ROLES THAT VARIABLES 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 ===========================================================
```

The purpose of GENTEXT is to generate a file of prompting phrases and helps called plate.PRO and five FORTRAN source libraries, BEGIN.NEW, STOGET.NEW, STRUCT.NEW, BEHAVIOR.NEW, and CHANGE.NEW. The purposes of these files are as follows:

```
=============================================================
TABLE 4   FILE OF PROMPTING PHRASES AND HELPS AND
          SOURCE CODE LIBRARIES GENERATED BY "GENTEXT"
=============================================================
 plate.PRO   = prompt file for input data for the problem
               class that you wish to set up for optimization.
               When BEGIN asks you for the name of the generic
               file, you should respond in this case with plate.

               The Prompt Numbers listed in TABLE 2 correspond
               to the prompts in this file.

 BEGIN.NEW   = source library for FORTRAN program which will
               be used to set up the starting design, material
```

properties, and any other data you wish.

STOGET.NEW = source library for FORTRAN subroutines which are used to transfer labelled common blocks. These labelled common blocks are the data base.

STRUCT.NEW = source library for FORTRAN subroutines that perform the analysis for each iterate in the set of optimization iterations. You may have to complete this routine (add dimension statements, subroutine calls, output statements, etc.). The library, STRUCT.NEW, also contains a skeletal routine, SUB. TRANFR, that you can complete in order to translate data names from from those just established by you (TABLE 2) to other names used by the developer of previously written code that you may plan to incorporate into SUBROUTINE STRUCT and/or SUBROUTINES BEHX1, BEHX2, BEHX3,...BEHXn (described next).

BEHAVIOR.NEW= a library of subroutine skeletons, BEHX1,BEHX2, BEHX3,...BEHXn, that, upon completion by you, will calculate behavior for a given design or design perturbation. Skeletal subroutines for a user-written constraint condition, USRCON, and a skeletal routine for the objective function, OBJECT, are also generated and are included in the BEHAVIOR.NEW library.

CHANGE.NEW = FORTRAN program that permits you to change certain program parameters without having to go back to BEGIN and run a case from scratch.

==

TABLE 5: CONTENTS OF SMALL FILES CREATED BY "GENTEXT"

FILE NAME	DEFINITION OF FILE CONTENTS
plate.PRO	Prompts and help paragraphs for interactive input to the user-developed optimization code.
plate.NEW	Part of BEGIN.NEW that contains calls to SUBROUTINE DATUM and SUBROUTINE GETVAR. This coding sets up the interactive input for the starting design in the user-generated design code.
plate.INP	Image of interactive input for user-developed program, generated to save time in case you make a mistake during input.
plate.COM	Labelled common blocks generated specifically for the user-developed class of problems.

```
plate.WRI            Part of subroutine for writing labelled common
                         blocks in SUBROUTINE STORCM (in Library STOGET).

plate.REA            Part of subroutine for reading labelled common
                         blocks in SUBROUTINE GETCOM (in Library STOGET).

plate.SET            Part of SUBROUTINE SETUPC in which new values
                         are installed in labelled common blocks from
                         the array VAR(I), which contains the latest
                         values of all candidates for decision variables.

plate.CON            Calls to subroutines, BEHX1, BEHX2, BEHX3,...,
                         which calculate behavior such as stresses
                         modal frequencies, buckling loads, etc.
                         Also, calls to CON, which generate the value
                         of the behavioral constraints corresponding
                         to BEHX1, BEHX2, BEHX3,...
                         Also, generates phrases that identify, in the
                         output of the user-generated program, the
                         exact meaning of each behavioral constraint.

plate.SUB            Skeletal subroutines, BEHX1, BEHX2, ..., and the
                         skeletal objective function, OBJECT.

plate.DEF            List of user-established variable names,
                         definitions, and roles that these variables play
                         in the user-generated program.  Also, contains
                         list of files created by GENTEXT and the
                         functions of these files.

plate.CHA            Part of SUBROUTINE NEWPAR (called in the CHANGE
                         processor) in which labelled common values are
                         updated.

plate.DAT            Image of interactive input for user-developed
                         program, generated to save time in case you make
                         a mistake during input. This file is used by
                         the INSERT processor.
============================================================
```

WHAT TO DO NEXT (THIS IS REALLY IMPORTANT!):
Next, if necessary, provide the algorithms called for in the skeletal subroutines listed in the library BEHAVIOR.NEW. You may find useful routines, such as a linear interpolator, in the library UTIL.NEW.

And/Or, if necessary, complete the skeletal routines STRUCT and TRANFR. (You may find useful routines in UTIL.NEW). If you are adding subroutine calls to SUBROUTINE STRUCT or SUBROUTINE TRANFR, store the subroutines themselves in the libraries called ADDCODEn.NEW, n = 1,2,3,...5. (Please list one of the ADDCODEn.NEW libraries for instructions.)

After you have done all this, give the command GENPROGRAMS. GENPROGRAMS will generate the absolute elements needed to optimize whatever you have chosen as your objective (see OBJECT routine in BEHAVIOR.NEW) in the presence of whatever behavior or other factors (e.g. clearance) are quantified by user-written subroutines collected in the libraries ADDCODEn.NEW and/or algorithms added to the skeletal routines in the library BEHAVIOR.NEW .

If an error occurs during GENPROGRAMS, check your FORTRAN coding. If you have to change something and rerun, make sure to save the old version under a different file name so that you can efficiently delete all outdated files with names *.NEW without losing a lot of good coding! The writer had fallen more than once into that trap during development of GENOPT.

If GENPROGRAMS runs without bombing, try test examples within the class of problems covered by your FORTRAN contributions to GENOPT before assigning specific design development tasks to individuals who may be more naive in the field covered by your FORTRAN contributions to GENOPT than you are!

Please see the cases under genopt/case for examples and more information.

USING GENOPT IN GENERAL AND WITH BIGBOSOR4

Please read the file, ..genopt/doc/getting.started.
Please also read the files:
...genopt/case/cylinder/howto.bosdec
...genopt/case/cylinder/howto.struct
...genopt/case/cylinder/howto.behavior
...genopt/case/torisph/howto.stags.pdf
...genopt/case/torisph/readme.equivellipse
...genopt/case/wavycyl/readme.wavycyl

The main things you must do are the following:

1. create a file called ..bosdec/sources/bosdec.src, the purpose of which is to create a BOSOR4 input file, *.ALL . in which "*" represents the users name for the specific case. The file, ..genopt/case/torisph/bosdec.equivellipse is a good example. Make sure to save bosdec.src by copying it into another file. Example: cp bosdec.src bosdec.equivellipse

2. Flesh out either or both the libraries, struct.new and/or behavior.new. In the case, ..genopt/case/torisph, only the library struct.new is fleshed out. The library behavior.new is not changed from that created automatically by GENOPT. In the case, genopt/case/cylinder, both struct.new and behavior.new are changed, struct.new in minor ways and behavior.new in major ways. Make sure to save struct.new and behavior.new. For example:

```
example: cp struct.new   struct.cylinder
         cp behavior.new behavior.cylinder
```

(You save copies of bosdec.src, struct.new, behavior.new because it usually takes quite a bit of effort to modify the versions automatically created by GENOPT in order to solve your generic class of problems.)
See the following files for examples of modified libraries:

```
genopt/case/torisph/struct.tori    (behavior.new not modified)
genopt/case/torisph/struct.ellipse (behavior.new not modified)
genopt/case/torisph/struct.equivellipse
                                   (behavior.new not modified)
genopt/case/cylinder/struct.cylinder
genopt/case/cylinder/behavior.cylinder
genopt/case/wavycyl/struct.wavycyl
genopt/case/wavycyl/behavior.wavycyl
genopt/case/plate/behavior.plate  (struct.new is not modified)
genopt/case/plate/behavior.plate  (struct.new is not modified)
genopt/case/sphere/behavior.plate (struct.new is not modified)
genopt/case/weldland/behavior.weldland  (tiny mod. struct.new)
genopt/case/trusscomp/behavior.trusscomp (tiny mod.struct.new)
genopt/case/submarine/behavior.submarine (tiny mod.struct.new)
genopt/case/span/behavior.span (tiny modification, struct.new)
genopt/case/balloon/behavior.balloon (tiny mod. of struct.new)
```

The "tiny modification" of struct.new consists of adding only three lines to struct.new: CALL OPNGEN, CALL RWDGEN, and CALL CLSGEN, added as described on p. 2 of Table 8 of the file ...genopt/case/trusscomp/trusscomp.vol1.pdf (see p. 53 of that file). The three added statements, CALL OPNGEN, CALL RWDGEN, and CALL CLSGEN, open, rewind, and close various files used by BIGBOSOR4. If you plan to optimize some other shell using GENOPT/BIGBOSOR4 you can "flesh out" struct.new in exactly the same way. To find the places in the "skeletal" version of struct.new that is automatically produced by GENTEXT, search for the string, "YOU MAY WANT" in order to find where you should insert the two lines, CALL OPNGEN and CALL RWDGEN. Search for the string, "NCONSX", in order to find where you should insert the line, CALL CLSGEN.

************** NOTE ****************** NOTE ******************
MAKE SURE ALWAYS TO SAVE COPIES OF struct.new AND behavior.new
THAT YOU HAVE PUT A LOT OF EFFORT INTO CREATING.
THE struct.new AND behavior.new FILES ARE DESTROYED BY
EXECUTION OF "gentext".
****

3. Execute the GENOPT script called GENPROGRAMS. This script "makes" the processors for the user-named generic case. The "makefile" called ..genopt/execute/usermake.linux is used. If GENPROGRAMS compiles everything successfully, which is not likely on your first try because you probably did a lot of FORTRAN coding to create bosdec.src, struct.new, behavior.new, GENPROGRAMS will end with a list like the following:

```
Here is a list of all your newly created executables:
-rwxr-xr-x  1 bush bush 71562 Oct  8 15:56 autochange.linux
-rwxr-xr-x  1 bush bush 139553 Oct  8 15:56 begin.linux
-rwxr-xr-x  1 bush bush 124383 Oct  8 15:56 change.linux
-rwxr-xr-x  1 bush bush 156054 Oct  8 15:56 chooseplot.linux
-rwxr-xr-x  1 bush bush 161231 Oct  8 15:56 decide.linux
-rwxr-xr-x  1 bush bush 104222 Oct  8 15:56 mainsetup.linux
-rwxr-xr-x  1 bush bush 1691559 Oct  8 15:56 optimize.linux
-rwxr-xr-x  1 bush bush 95653 Oct  8 15:56 store.linux
```

Next, type the command BEGIN to input data for a new specific case.

If GENPROGRAMS bombs due to fatal compilation errors, or even if GENPROGRAMS seems to finish successfully, it is best to inspect the file ..genoptcase/usermakelinux.log. If there are compilation errors, revise the appropriate source codes, bosdec.src and/or struct.new and/or behavior.new, and execute GENPROGRAMS again. Keep doing this until everything is okay.

4. Next, think up a good name for your specific case and run BEGIN, DECIDE, MAINSETUP, and OPTIMISE (several times) or SUPEROPT. (See the file ..genopt/doc/getting.started and the directories, genopt/case/cylinder and genopt/case/torisph for examples.) Even though you had a successful "make" via GENPROGRAMS in the previous step, something will doubtless not be satisfactory and you will have to or want to make further changes to one or more of the source files, bosdec.src, struct.new, behavior.new.

THE NEXT STEPS PERTAIN TO THE USE OF GENOPT WITH BIGBOSOR4

5. You must have the BIGBOSOR4 software in the directory, ..bosdec/sources. You need to have the following files there: addbosor4.src, b4util.src, opngen.src, prompter.src, gasp.F, gasp_linux.o, bio_linux.c, bio_linux.o, b4plot.src, as well as the bosdec.src file discussed above. (See "getting started".)

6. The "make" file, ..genopt/execute/usermake.linux, must include references to the BIGBOSOR4 sofware listed in Step 5. Please see the file ..genopt/execute/usermake.linux, which already exists. (You do not have to do anything about it!)

7. Suppose everything compiles correctly during the GENPROGRAMS execution, but when you try to run a specific case the run bombs. Suppose all of your contributed FORTRAN coding is in ..bosdec/sources/bosdec.src and in
      ..genoptcase/struct.new (..genoptcase/behavior.new did not need to be modified for your case, as is true for the generic case called "equivellipse" in ..genopt/case/torisph). It is very helpful to insert a "CALL EXIT" statement after one of the analyses performed in struct.new, then to execute GENPROGRAMS again to recompile the temporarily changed struct.new. The reason for doing this is explained in the file ..genopt/case/torisph/struct.equivellipse and also in the file ..genopt/doc/getting.started: you want to be able to make a BIGBOSOR4 run to be certain that:

a. ..bosdec/sources/bosdec.src created a valid BOSOR4 input file, and,
b. the BIGBOSOR4 run did not finish for some reason.

----------------- end of the plate.DEF file ------------------

----

PART 5 INSPECT THE plate.PRO FILE

(A list of the plate.PRO file follows. This file contains
 prompts and "help" paragraphs that will be seen by the end user.)

```
------ plate.PRO file (contains prompts/help paragraphs) --------
   5.0
       PROGRAM FOR OPTIMIZATION OF A RECTANGULAR PLATE
       SUBJECTED TO SEVERAL LOADING ENVIRONMENTS AND
       CONSTRAINTS ON STRESS, BUCKLING, DISPLACEMENT, AND FREQUENCY.


  10.0
       FIRST, PROVIDE ALL VARIABLES THAT CAN BE DECISION VARIABLES,
       THAT IS, VARIABLES THAT CAN CHANGE DURING OPTIMIZATION
       ITERATIONS (ROLE TYPE 1), AND FIXED VARIABLES (ROLE TYPE 2).

  15.1 thickness of the plate: THICK
  20.1 Length of the plate: LENGTH
  25.1 Width of the plate: WIDTH
  30.1 Young's modulus of the plate material: E
  35.1 Poisson's ratio of the plate material: NU
  40.1 Weight density (e.g. lb/in**3) of the plate material: RHO

  45.0
       THE FOLLOWING TABLE (FROM ROARK, 3RD EDITION, TABLE XVI,
       ENTRY NO. 1, PAGE 312), GIVES THE RELATIONSHIP OF PLATE
       (LENGTH/WIDTH) TO A COEFFICIENT FOR BUCKLING UNDER UNIFORM
       AXIAL COMPRESSION.

  50.1 Number IAOBAXL of rows in the array  AOBAXL: IAOBAXL
  55.1 (plate length, LENGTH)/(plate width, WIDTH): AOBAXL
  60.1 buckling coefficient: uniform axial compression: KAXL

  65.0
       THE FOLLOWING TABLE (FROM ROARK, 3RD EDITION, TABLE XVI,
       ENTRY NO. 10, PAGE 312), GIVES THE RELATIONSHIP OF PLATE
       (LENGTH/WIDTH) TO A COEFFICIENT FOR BUCKLING UNDER UNIFORM
       IN-PLANE SHEAR.

  70.1 Number IAOBSHR of rows in the array  AOBSHR: IAOBSHR
  75.1 (plate length, LENGTH)/(plate width, WIDTH): AOBSHR
  80.1 buckling coefficient: uniform in-plane shear: KSHR

  85.0
       NEXT, PROVIDE ALL ENVIRONMENTAL PARAMETERS (LOADS, TEMPERATURES)

  90.1 Number NCASES  of load cases (environments): NCASES
  95.1 Axial tension/width of the plate (lb/in): Nx
  95.2
       NOTE:  Nx must be negative for axial compression!

 100.1 Transverse tension/length of the plate (lb/in): Ny
 105.1 In-plane shear/length applied to the plate: Nxy
 110.1 Uniform normal pressure on the plate: PRESS

 115.0
       NEXT, PROVIDE RESPONSE PARAMETERS, ALLOWABLES, AND FACTORS OF
       SAFETY. THE ORDER IN WHICH YOU MUST PROVIDE THESE DATA IS:
```

```
       A,B,C; A,B,C; A,B,C; A,B,C; etc, in which
       A = RESPONSE,   B = ALLOWABLE,   C = FACTOR OF SAFETY.

 120.0 Maximum effective (von Mises) stress: STRESS
 125.1 Maximum effective (von Mises) stress allowed: MAXSTR
 130.1 Factor of safety for effective stress: FSTRES
 135.0 Buckling load factor: BUCKLE
 140.1 Minimum allowable buckling load factor (use 1.0): MINBUC
 145.1 Factor of safety for buckling load factor: FBUCKL
 150.0 Fundamental frequency of unloaded plate: FREQ
 155.1 Minimum allowable value for the fundamental frequency: MINCPS
 160.1 Factor of safety for FREQ: FSFREQ
 165.0 Normal deflection under uniform pressure: W
 170.1 Maximum allowable normal deflection under pressure: AW
 175.1 Factor of safety for max deflection under pressure: FW

 180.0
       LAST, AN OBJECTIVE MUST BE CHOSEN, SUCH AS MINIMUM WEIGHT
       OR MINIMUM COST.

 185.0 Weight of the plate: WEIGHT
 999.0 DUMMY ENTRY TO MARK END OF FILE
------------- end of the plate.PRO file, which contains ------------
------------- prompts and "help" paragraphs seen by ----------------
------------- the end user.                          ----------------


---------------------------------------------------------
```

PART 6 INSPECT THE "SKELETAL" behavior.new FILE

(The "skeletal" file, "behavior.new" follows. This is the version of behavior.new generated automatically by GENTEXT. The GENOPT user has to "flesh out" the subroutines in the behavior.new library.)

```
------------ behavior.new file (skeletal subroutines) ------------
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: plate

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 "plate".

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 "plate" 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 [2b] Bushnell, D., "Minimum weight design of imperfect
C isogrid-stiffened ellipsoidal shells under uniform external
C pressure", AIAA paper 2009-2702, 50th AIAA Structures
```

```
C Meeting, Palm Springs, CA, May 4-7, 2009

C [2c] Bushnell, D. and Thornburgh, R. P., "Use of GENOPT and
C BIGBOSOR4 to optimize weld lands in axially compressed
C stiffened cylindrical shells and evaluation of the optimized
C designs by STAGS", AIAA Paper 2010-2927, AIAA 51st Structures
C Meeting, Orlando, Florida, April 2010

C [2d] Bushnell, D., "Use of GENOPT and BIGBOSOR4 to obtain
C optimum designs of an axially compressed cylindrical shell
C with a composite truss-core sandwich wall", AIAA Paper 2011-
C 1811, 52nd AIAA Structures Meeting, Denver, CO, April, 2011

C [2e] Bushnell, D., "Use of GENOPT and BIGBOSOR4 to obtain
C an optimum design of a deep submergence tank", unpublished
C report to the DOER company, Alameda, CA, June 30, 2009

C [2f] Bushnell, D., "Use of GENOPT and BIGBOSOR4 to obtain
C optimum designs of a double-walled inflatable cylindrical
C vacuum chamber", unpublished report for Michael Mayo,
C November, 2010

C [2g] Bushnell, D., "Use of GENOPT and BIGBOSOR4 to obtain
C optimum designs of a double-walled inflatable spherical
C vacuum chamber", unpublished report for Michael Mayo,
C December, 2010

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 imperf.isogrid-stiffened
```

```
C ellipsoidal shells...", written and placed in the file
C ..genopt/case/torisph/sdm50.report.pdf

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, Ref.[2b]

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

C [15] Bushnell, D., the case ..genopt/case/wavycyl, Ref.[2]

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

C [17] Bushnell, D., the case ..genopt/case/weldland, Ref.[2c]

C [18] Bushnell, D., the case ..genopt/case/trusscomp,Ref.[2d]

C [19] Bushnell, D., the case ..genopt/case/submarine,Ref.[2e]

C [20] Bushnell, D., the case ..genopt/case/sphere

C [21] Bushnell, D., the case ..genopt/case/balloon

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 = plate)
C        4.  GENOPT.HLP file    (type HELPG)
C===============================================================

C================================================================
C   TABLE 2   GLOSSARY OF VARIABLES USED IN "plate"
C================================================================
C  ARRAY  NUMBER OF         PROMPT
C    ?   (ROWS,COLS)  ROLE  NUMBER   NAME                DEFINITION OF VARIABLE
C                         (plate.PRO)
C================================================================
C    n   (   0,   0)    1      15   THICK    = thickness of the plate
C    n   (   0,   0)    1      20   LENGTH   = Length of the plate
C    n   (   0,   0)    1      25   WIDTH    = Width of the plate
C    n   (   0,   0)    2      30   E        = Young's modulus of the plate mate
C    n   (   0,   0)    2      35   NU       = Poisson's ratio of the plate mate
C    n   (   0,   0)    2      40   RHO      = Weight density (e.g. lb/in**3) of
C    n   (   0,   0)    2      50   IAOBAXL  = Number of entries in the table of
C    y   (  30,   0)    2      55   AOBAXL   = (plate length, LENGTH)/(plate wid
C    y   (  30,   0)    2      60   KAXL     = buckling coefficient: uniform axi
C    n   (   0,   0)    2      70   IAOBSHR  = Number of entries in the table of
C    y   (  20,   0)    2      75   AOBSHR   = (plate length, LENGTH)/(plate wid
C    y   (  20,   0)    2      80   KSHR     = buckling coefficient: uniform in-
C    n   (   0,   0)    2      90   NCASES   = Number of load cases (number of e
C    y   (  20,   0)    3      95   Nx       = Axial tension/width of the plate
C    y   (  20,   0)    3     100   Ny       = Transverse tension/length of the
C    y   (  20,   0)    3     105   Nxy      = In-plane shear/length applied to
C    y   (  20,   0)    3     110   PRESS    = Uniform normal pressure on the pl
C    y   (  20,   0)    4     120   STRESS   = Maximum effective (von Mises) str
C    y   (  20,   0)    5     125   MAXSTR   = Maximum effective (von Mises) str
C    y   (  20,   0)    6     130   FSTRES   = Factor of safety for effective st
C    y   (  20,   0)    4     135   BUCKLE   = Buckling load factor
C    y   (  20,   0)    5     140   MINBUC   = Minimum allowable buckling load f
C    y   (  20,   0)    6     145   FBUCKL   = Factor of safety for buckling loa
C    y   (  20,   0)    4     150   FREQ     = Fundamental frequency of unloaded
C    y   (  20,   0)    5     155   MINCPS   = Minimum allowable value for the f
C    y   (  20,   0)    6     160   FSFREQ   = Factor of safety for FREQ
C    y   (  20,   0)    4     165   W        = Normal deflection under uniform p
C    y   (  20,   0)    5     170   AW       = Maximum allowable normal deflecti
C    y   (  20,   0)    6     175   FW       = Factor of safety for max deflecti
C    n   (   0,   0)    7     185   WEIGHT   = Weight of the plate
C
C=DECK      BEHX1
      SUBROUTINE BEHX1
     1 (IFILE,NPRINX,IMODX,IFAST,ILOADX,PHRASE)
C
C   PURPOSE: OBTAIN Maximum effective (von Mises) stress
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     STRESS(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 = Maximum effective (von Mises) stress
C
C   OUTPUT:
C
C     STRESS(ILOADX)
C
       CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      COMMON/FV07/AOBAXL(30),IAOBAXL
      REAL AOBAXL
      COMMON/FV08/KAXL(30)
      REAL KAXL
      COMMON/FV09/AOBSHR(20),IAOBSHR
      REAL AOBSHR
      COMMON/FV10/KSHR(20)
      REAL KSHR
      COMMON/FV11/Nx(20)
      REAL Nx
      COMMON/FV17/STRESS(20),MAXSTR(20),FSTRES(20)
      REAL STRESS,MAXSTR,FSTRES
      COMMON/FV20/BUCKLE(20),MINBUC(20),FBUCKL(20)
      REAL BUCKLE,MINBUC,FBUCKL
      COMMON/FV23/FREQ(20),MINCPS(20),FSFREQ(20)
      REAL FREQ,MINCPS,FSFREQ
      COMMON/FV26/W(20),AW(20),FW(20)
      REAL W,AW,FW
      COMMON/FV01/THICK,LENGTH,WIDTH,E,NU,RHO,WEIGHT
      REAL THICK,LENGTH,WIDTH,E,NU,RHO,WEIGHT
      COMMON/FV12/Ny(20),Nxy(20),PRESS(20)
      REAL Ny,Nxy,PRESS
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 Buckling load factor
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     BUCKLE(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 = Buckling load factor
C
C   OUTPUT:
C
C     BUCKLE(ILOADX)
C
       CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      COMMON/FV07/AOBAXL(30),IAOBAXL
      REAL AOBAXL
      COMMON/FV08/KAXL(30)
      REAL KAXL
      COMMON/FV09/AOBSHR(20),IAOBSHR
      REAL AOBSHR
      COMMON/FV10/KSHR(20)
      REAL KSHR
      COMMON/FV11/Nx(20)
      REAL Nx
      COMMON/FV17/STRESS(20),MAXSTR(20),FSTRES(20)
      REAL STRESS,MAXSTR,FSTRES
      COMMON/FV20/BUCKLE(20),MINBUC(20),FBUCKL(20)
      REAL BUCKLE,MINBUC,FBUCKL
```

```
      COMMON/FV23/FREQ(20),MINCPS(20),FSFREQ(20)
      REAL FREQ,MINCPS,FSFREQ
      COMMON/FV26/W(20),AW(20),FW(20)
      REAL W,AW,FW
      COMMON/FV01/THICK,LENGTH,WIDTH,E,NU,RHO,WEIGHT
      REAL THICK,LENGTH,WIDTH,E,NU,RHO,WEIGHT
      COMMON/FV12/Ny(20),Nxy(20),PRESS(20)
      REAL Ny,Nxy,PRESS
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,PHRASE)
C
C   PURPOSE: OBTAIN Fundamental frequency of unloaded plate
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     FREQ(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 = Fundamental frequency of unloaded plate
C
C   OUTPUT:
C
C     FREQ(ILOADX)
C
```

```
        CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      COMMON/FV07/AOBAXL(30),IAOBAXL
      REAL AOBAXL
      COMMON/FV08/KAXL(30)
      REAL KAXL
      COMMON/FV09/AOBSHR(20),IAOBSHR
      REAL AOBSHR
      COMMON/FV10/KSHR(20)
      REAL KSHR
      COMMON/FV11/Nx(20)
      REAL Nx
      COMMON/FV17/STRESS(20),MAXSTR(20),FSTRES(20)
      REAL STRESS,MAXSTR,FSTRES
      COMMON/FV20/BUCKLE(20),MINBUC(20),FBUCKL(20)
      REAL BUCKLE,MINBUC,FBUCKL
      COMMON/FV23/FREQ(20),MINCPS(20),FSFREQ(20)
      REAL FREQ,MINCPS,FSFREQ
      COMMON/FV26/W(20),AW(20),FW(20)
      REAL W,AW,FW
      COMMON/FV01/THICK,LENGTH,WIDTH,E,NU,RHO,WEIGHT
      REAL THICK,LENGTH,WIDTH,E,NU,RHO,WEIGHT
      COMMON/FV12/Ny(20),Nxy(20),PRESS(20)
      REAL Ny,Nxy,PRESS
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,PHRASE)
C
C   PURPOSE: OBTAIN Normal deflection under uniform pressure
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     W(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 deflection under uniform pressure
C
C    OUTPUT:
C
C      W(ILOADX)
C
        CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      COMMON/FV07/AOBAXL(30),IAOBAXL
      REAL AOBAXL
      COMMON/FV08/KAXL(30)
      REAL KAXL
      COMMON/FV09/AOBSHR(20),IAOBSHR
      REAL AOBSHR
      COMMON/FV10/KSHR(20)
      REAL KSHR
      COMMON/FV11/Nx(20)
      REAL Nx
      COMMON/FV17/STRESS(20),MAXSTR(20),FSTRES(20)
      REAL STRESS,MAXSTR,FSTRES
      COMMON/FV20/BUCKLE(20),MINBUC(20),FBUCKL(20)
      REAL BUCKLE,MINBUC,FBUCKL
      COMMON/FV23/FREQ(20),MINCPS(20),FSFREQ(20)
      REAL FREQ,MINCPS,FSFREQ
      COMMON/FV26/W(20),AW(20),FW(20)
      REAL W,AW,FW
      COMMON/FV01/THICK,LENGTH,WIDTH,E,NU,RHO,WEIGHT
      REAL THICK,LENGTH,WIDTH,E,NU,RHO,WEIGHT
      COMMON/FV12/Ny(20),Nxy(20),PRESS(20)
      REAL Ny,Nxy,PRESS
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/FV07/AOBAXL(30),IAOBAXL
      REAL AOBAXL
      COMMON/FV08/KAXL(30)
      REAL KAXL
      COMMON/FV09/AOBSHR(20),IAOBSHR
      REAL AOBSHR
      COMMON/FV10/KSHR(20)
      REAL KSHR
      COMMON/FV11/Nx(20)
      REAL Nx
      COMMON/FV17/STRESS(20),MAXSTR(20),FSTRES(20)
      REAL STRESS,MAXSTR,FSTRES
      COMMON/FV20/BUCKLE(20),MINBUC(20),FBUCKL(20)
      REAL BUCKLE,MINBUC,FBUCKL
      COMMON/FV23/FREQ(20),MINCPS(20),FSFREQ(20)
      REAL FREQ,MINCPS,FSFREQ
      COMMON/FV26/W(20),AW(20),FW(20)
      REAL W,AW,FW
      COMMON/FV01/THICK,LENGTH,WIDTH,E,NU,RHO,WEIGHT
      REAL THICK,LENGTH,WIDTH,E,NU,RHO,WEIGHT
      COMMON/FV12/Ny(20),Nxy(20),PRESS(20)
      REAL Ny,Nxy,PRESS
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 plate
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 plate
C
       CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      COMMON/FV07/AOBAXL(30),IAOBAXL
      REAL AOBAXL
      COMMON/FV08/KAXL(30)
      REAL KAXL
      COMMON/FV09/AOBSHR(20),IAOBSHR
      REAL AOBSHR
      COMMON/FV10/KSHR(20)
      REAL KSHR
      COMMON/FV11/Nx(20)
      REAL Nx
      COMMON/FV17/STRESS(20),MAXSTR(20),FSTRES(20)
      REAL STRESS,MAXSTR,FSTRES
      COMMON/FV20/BUCKLE(20),MINBUC(20),FBUCKL(20)
      REAL BUCKLE,MINBUC,FBUCKL
      COMMON/FV23/FREQ(20),MINCPS(20),FSFREQ(20)
      REAL FREQ,MINCPS,FSFREQ
      COMMON/FV26/W(20),AW(20),FW(20)
      REAL W,AW,FW
      COMMON/FV01/THICK,LENGTH,WIDTH,E,NU,RHO,WEIGHT
      REAL THICK,LENGTH,WIDTH,E,NU,RHO,WEIGHT
      COMMON/FV12/Ny(20),Nxy(20),PRESS(20)
      REAL Ny,Nxy,PRESS
C
C
C  INSERT SUBROUTINE STATEMENTS HERE.
C
C
      OBJGEN =WEIGHT
```

```
C
C
      RETURN
      END
C
C
C
------------ end of behavior.new (skeletal subroutines) -----------
```

PART 7 INSPECT THE struct.new FILE

(The "skeletal" file, "struct.new" follows. This is the version of struct.new generated automatically by GENTEXT. The GENOPT user may have to "flesh out" the subroutines in the behavior.new library. In this particular generic case the struct.new library generated automatically by GENTEXT remains as is.)

```
------------ struct.new (skeletal subroutines) ----------------
C=DECK      STRUCT
      SUBROUTINE STRUCT(IMODX,CONSTX,OBJGEN,CONMAX,NCONSX,IPOINC,
     1 PCWORD,CPLOTX,ILOADX,ISTARX,NUSERC,IBEHV,IDV,IFAST,JJJ1)
C
C  PURPOSE IS TO PERFORM THE ANALYSIS FOR A GIVEN DESIGN AND LOADING.
C  CONSTRAINT CONDITIONS ARE ALSO GENERATED.
C
C  Common blocks already present in the struct.tmpl file, that is,
C  in the "skeletal" file possibly to be augmented by the user:
      COMMON/PRMFIL/IFILEX,IFILE2,IOUT,IPRM(5)
      COMMON/PRMOUT/IFILE3,IFILE4,IFILE8,IFILE9,IFIL11
      COMMON/INDAT/INFILE
      COMMON/LWRUPR/VLBX(50),VUBX(50),CLINKX(50,45),VLINKX(50),VBVX(99)
      COMMON/NUMPAR/IPARX,IVARX,IALLOW,ICONSX,NDECX,NLINKX,NESCAP,ITYPEX
      COMMON/PARAMS/PARX(99),VARX(50),ALLOWX(99),CONSXX(99),DECX(50),
     1              ESCX(50)
      COMMON/WORDS1/WORDPX(99),WORDVX(50),WORDAX(99),WORDCC(99),
     1              WORDDX(50)
      COMMON/WORDS2/WORDLX(50),WORDEX(50),WORDIQ(45)
      COMMON/OPTVAR/IDVX(50),ILVX(50),IDLINK(50,45),IEVX(50),JTERMS(45)
      COMMON/NUMPR2/ILARX,ICARX,IOARX,IFLATX,NCASES,NPRINX
      COMMON/PARAM2/FLARX(50),CARX(99),OARX(50),FSAFEX(99),CPWRX(50,45)
      COMMON/PARAM3/CINEQX(45,45),DPWREQ(45,45)
      COMMON/PARAM4/IDINEQ(45,45),NINEQX,JINEQX(45),IEQTYP(45)
      COMMON/WORDS3/WORDFX(50),WORDBX(99),WORDOB(50),WORDSX(99)
      COMMON/WORDS4/WORDMX(99)
      COMMON/PWORD/PHRASE
      COMMON/PWORD2/IBLANK
      COMMON/ISKIPX/ISKIP(30)
      DIMENSION IBEHV(99)
C
C===========================================================================
C  Start of first part of STRUCT written by "GENTEXT"
C  INSERT ADDITIONAL COMMON BLOCKS HERE: (THESE ARE "GENTEXT" VARIABLES)
      COMMON/FV07/AOBAXL(30),IAOBAXL
```

```
      REAL AOBAXL
      COMMON/FV08/KAXL(30)
      REAL KAXL
      COMMON/FV09/AOBSHR(20),IAOBSHR
      REAL AOBSHR
      COMMON/FV10/KSHR(20)
      REAL KSHR
      COMMON/FV11/Nx(20)
      REAL Nx
      COMMON/FV17/STRESS(20),MAXSTR(20),FSTRES(20)
      REAL STRESS,MAXSTR,FSTRES
      COMMON/FV20/BUCKLE(20),MINBUC(20),FBUCKL(20)
      REAL BUCKLE,MINBUC,FBUCKL
      COMMON/FV23/FREQ(20),MINCPS(20),FSFREQ(20)
      REAL FREQ,MINCPS,FSFREQ
      COMMON/FV26/W(20),AW(20),FW(20)
      REAL W,AW,FW
      COMMON/FV01/THICK,LENGTH,WIDTH,E,NU,RHO,WEIGHT
      REAL THICK,LENGTH,WIDTH,E,NU,RHO,WEIGHT
      COMMON/FV12/Ny(20),Nxy(20),PRESS(20)
      REAL Ny,Nxy,PRESS
C
C
      CHARACTER*80 PHRASE,CODPHR,PCWORD
      CHARACTER*80 WORDPX,WORDVX,WORDAX,WORDCX,WORDDX,WORDLX,WORDEX
      CHARACTER*80 WORDFX,WORDBX,WORDOB,WORDSX,WORDMX,WORDCC,WORDIQ
c     CHARACTER*4 ANSOUT,CHARAC,ANSWER
      CHARACTER*2 CIX
      character*2 CJX
      CHARACTER*13 CODNAM
c     DIMENSION ISUBX(100)
c     LOGICAL ANSL1
C
      DIMENSION CONSTX(*),IPOINC(*),PCWORD(*),CPLOTX(*)
C  End of first part of STRUCT written by "GENTEXT"
C=======================================================================
C
C  INSERT ADDITIONAL DIMENSION AND/OR LABELLED COMMON BLOCKS HERE,
C  IF NECESSARY. THESE WOULD BE STATEMENTS THAT ARE CONSISTENT WITH
C  SUBROUTINES THAT YOU OR OTHERS MAY HAVE WRITTEN THAT ARE REQUIRED
C  FOR WHATEVER ANALYSIS YOU ARE PERSUING.  MAKE SURE THAT YOU DO NOT
C  INTRODUCE NAME CONFLICTS WITH THE "GENTEXT" LABELLED COMMON BLOCKS
C  LISTED ABOVE.
C
C  Please note that you do not have to modify STRUCT.NEW if you would
C  rather provide all of your algorithms via the BEHAVIOR.NEW library.
C  (See instructions in BEHAVIOR.NEW).
C
C  If you are using a lot of software previously written either by
C  yourself or others, or if there are a lot of behavioral constraints
C  that are best generated by looping over array indices (such as
C  occurs, for example, with stress constraints in laminates of
C  composite materials), then it may be best to insert your common
C  blocks and dimension statements here, your subroutine calls
C  below (where indicated), and your subroutines in any of the libraries
C  called ADDCODEn.NEW, n = 1,2,...,5.  Please note that you
C  may also have to add statements to SUBROUTINE TRANFR, the
C  purpose of which is described below (in TRANFR).
```

```
C
C  The several test cases provided with GENOPT demonstrate different
C  methods:
C
C  PLATE  : leave STRUCT.NEW unchanged and modify BEHAVIOR.NEW
C  SPHERE : leave STRUCT.NEW unchanged and modify BEHAVIOR.NEW
C  TORISPH: leave BEHAVIOR.NEW unchanged except possibly for the objective
C           function (SUBROUTINE OBJECT), modify STRUCT.NEW,
C           possibly add a subroutine library called ADDCODE1.NEW, and
C           possibly augment the usermake.linux file to collect object
C           libraries from other directories. In the "TORISPH" case
C           BEHAVIOR.NEW remains unchanged, no ADDCODE1.NEW library is
C           added, and usermake.linux is not changed.  Instead, the
C           BIGBOSOR4 code is added and SUBROUTINE BOSDEC is written
C           by the genopt user. The BIGBOSOR4 code and SUBROUTINE
C           BOSDEC must be stored in /home/progs/bosdec/sources, as
C           follows:
C    BIGBOSOR4 code:
C     -rw-r--r--  1 bush bush 579671 Feb 29 07:19 addbosor4.src
C     -rw-r--r--  1 bush bush  83175 Feb 22 09:13 b4plot.src
C     -rw-r--r--  1 bush bush  89671 Feb 28 16:20 b4util.src
C     -rw-r--r--  1 bush bush  22723 Feb 10 14:27 bio.c
C     -rw-r--r--  1 bush bush  31175 Feb 10 14:27 bio_linux.c
C     -rw-r--r--  1 bush bush  37152 Feb 10 14:27 bio_linux.o
C     -rw-r--r--  1 bush bush  15650 Feb 10 14:26 gasp.F
C     -rw-r--r--  1 bush bush  18364 Feb 10 14:26 gasp_linux.o
C     -rw-r--r--  1 bush bush   6310 Feb 13 10:12 opngen.src
C     -rw-r--r--  1 bush bush  22440 Feb 10 14:25 prompter.src
C     -rw-r--r--  1 bush bush  13426 Feb 22 09:14 resetup.src
C    BOSDEC.src code:
C     -rw-r--r--  1 bush bush  33851 Mar  1 08:34 bosdec.src
C
C  WAVYCYL: both BEHAVIOR.NEW and STRUCT.NEW are both changed. Otherwise
C           the activity is the same as that described for TORISPH,
C           except, of course, that struct.new is different from
C           that used in connection with TORISPH.
C
C  CYLINDER:same as the description for WAVYCYL.
C
C
C  INSERT YOUR ADDITIONAL COMMON BLOCKS FOR THIS GENERIC CASE HERE:
C
C
C  THE FOLLOWING CODE WAS WRITTEN BY "GENTEXT":
C
C=================================================================
C  Start the second portion of STRUCT written by "GENTEXT":
C
      ICARX   = ISTARX
      INUMTT = 0
      ICONSX = 0
      KCONX   = 0
      IF (IMODX.EQ.0) THEN
         CALL MOVERX(0.,0,CONSTX,1,99)
         CALL MOVERX(0, 0,IPOINC,1,1500)
      ENDIF
C
      IF (ILOADX.EQ.1) THEN
```

```
C
C  ESTABLISH FIRST ANY CONSTRAINTS THAT ARE INEQUALITY RELATIONSHIPS
C  AMONG THE VARIABLES IN THE ARRAY VARX(*) (THAT IS, VARIABLES THAT
C  ARE EITHER DECISION VARIABLES, LINKED VARIABLES, ESCAPE VARIABLES,
C  OR CANDIDATES FOR ANY OF THESE TYPES OF VARIABLES.
C
         IF (NINEQX.GT.0)
     1        CALL VARCON(WORDIQ,WORDMX,CINEQX,DPWREQ,IDINEQ,
     1        NINEQX,JINEQX,IEQTYP,INUMTT,IMODX,CONMAX,IPOINC,
     1        ICONSX,CONSTX,VARX,PCWORD,CPLOTX,ICARX)
C
C  NEXT, ESTABLISH USER-WRITTEN CONSTRAINTS. AT PRESENT, THE PROGRAM
C  ALLOWS ONLY ONE USER-WRITTEN CONSTRAINT. HOWEVER, THE USER CAN
C  EASILY EXPAND THIS CAPABILITY SIMPLY BY ADDING SUBROUTINES THAT
C  ARE ANALOGOUS TO USRCON (WITH NAMES SUCH AS USRCN2, USRCN3, ETC.
C  TO THE  BEHAVIOR.NEW LIBRARY, AND ADD CALLS TO THESE ADDITIONAL
C  SUBROUTINES FOLLOWING THE CALL TO USRCON IMMEDIATELY BELOW.
C
         CALL USRCON(INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
     1            WORDMX,PCWORD,CPLOTX,ICARX,IFILE8)
C
         NUSERC = ICARX - NINEQX
      ENDIF
C
      IF (NPRINX.GT.0) THEN
         WRITE(IFILE8,'(1X,A,I2,A)')
     1 ' BEHAVIOR FOR ',ILOADX,' ENVIRONMENT (LOAD SET)'
         WRITE(IFILE8,'(A)')'  '
         WRITE(IFILE8,'(A)')
     1 ' CONSTRAINT  BEHAVIOR              DEFINITION'
         WRITE(IFILE8,'(A)')
     1 '   NUMBER     VALUE'
      ENDIF
C
      CALL CONVR2(ILOADX,CIX)
      IF (NPRINX.GT.0) THEN
         WRITE(IFILE8,'(1X,A)')'  '
         WRITE(IFILE8,'(1X,A,I2)')
     1 ' BEHAVIOR FOR LOAD SET NUMBER, ILOADX=',ILOADX
      ENDIF
C
C  End of the second portion of STRUCT written by "GENTEXT"
C==========================================================================
C
C  USER: YOU MAY WANT TO INSERT SUBROUTINE CALLS FROM SOFTWARE DEVELOPED
C        ELSEWHERE FOR ANY CALCULATIONS PERTAINING TO THIS LOAD SET.
C
C==========================================================================
C  Start of the final portion of STRUCT written by "GENTEXT"
C
C  INSERT THE PROGRAM FILE HERE:
C
C  Behavior and constraints generated next for STRESS:
C  STRESS = Maximum effective (von Mises) stress
C
      PHRASE =
     1 'Maximum effective (von Mises) stress'
      CALL BLANKX(PHRASE,IENDP4)
```

```
      JXX = 0
      JXX = JXX + 1
      STRESS(ILOADX) = 0.0
      IF (IBEHV(JXX).EQ.0) CALL BEHX1
     1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX  ,
     1 'Maximum effective (von Mises) stress')
      IF (STRESS(ILOADX  ).EQ.0.)  STRESS(ILOADX  ) = 1.E-10
      IF (MAXSTR(ILOADX  ).EQ.0.)  MAXSTR(ILOADX  ) = 1.0
      IF (FSTRES(ILOADX  ).EQ.0.)  FSTRES(ILOADX  ) = 1.0
      KCONX = KCONX + 1
      CARX(KCONX) =STRESS(ILOADX  )
      WORDCX= '(STRESS('//CIX//')/MAXSTR('//CIX//
     1  ')) X FSTRES('//CIX//')'
      CALL CONX(STRESS(ILOADX  ),MAXSTR(ILOADX  ),FSTRES(ILOADX  )
     1,'Maximum effective (von Mises) stress',
     1 'Maximum effective (von Mises) stress allowed',
     1 'Factor of safety for effective stress',
     1 1,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
     1 WORDMX,PCWORD,CPLOTX,ICARX)
      IF (IMODX.EQ.0) THEN
         CODPHR =
     1 '  Maximum effective (von Mises) stress: '
         IENDP4 =40
         CODNAM ='STRESS('//CIX//')'
         MLET4 =6 + 4
         WORDBX(KCONX)= CODPHR(1:IENDP4)//CODNAM(1:MLET4)
         IF (NPRINX.GT.0) WRITE(IFILE8,'(I5,6X,G14.7,A,A)')
     1    KCONX,CARX(KCONX),CODPHR(1:IENDP4),CODNAM(1:MLET4)
      ENDIF
  130 CONTINUE
  131 CONTINUE
C
C  Behavior and constraints generated next for BUCKLE:
C  BUCKLE = Buckling load factor
C
      PHRASE =
     1 'Buckling load factor'
      CALL BLANKX(PHRASE,IENDP4)
      JXX = JXX + 1
      BUCKLE(ILOADX) = 0.0
      IF (IBEHV(JXX).EQ.0) CALL BEHX2
     1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX  ,
     1 'Buckling load factor')
      IF (BUCKLE(ILOADX  ).EQ.0.)  BUCKLE(ILOADX  ) = 1.E+10
      IF (MINBUC(ILOADX  ).EQ.0.)  MINBUC(ILOADX  ) = 1.0
      IF (FBUCKL(ILOADX  ).EQ.0.)  FBUCKL(ILOADX  ) = 1.0
      KCONX = KCONX + 1
      CARX(KCONX) =BUCKLE(ILOADX  )
      WORDCX= '(BUCKLE('//CIX//')/MINBUC('//CIX//
     1  ')) / FBUCKL('//CIX//')'
      CALL CONX(BUCKLE(ILOADX  ),MINBUC(ILOADX  ),FBUCKL(ILOADX  )
     1,'Buckling load factor',
     1 'Minimum allowable buckling load factor (use 1.0)',
     1 'Factor of safety for buckling load factor',
     1 2,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
     1 WORDMX,PCWORD,CPLOTX,ICARX)
      IF (IMODX.EQ.0) THEN
         CODPHR =
```

```
     1 '  Buckling load factor: '
         IENDP4 =24
         CODNAM ='BUCKLE('//CIX//')'
         MLET4 =6 + 4
         WORDBX(KCONX)= CODPHR(1:IENDP4)//CODNAM(1:MLET4)
         IF (NPRINX.GT.0) WRITE(IFILE8,'(I5,6X,G14.7,A,A)')
     1    KCONX,CARX(KCONX),CODPHR(1:IENDP4),CODNAM(1:MLET4)
      ENDIF
  145 CONTINUE
  146 CONTINUE
C
C  Behavior and constraints generated next for FREQ:
C  FREQ = Fundamental frequency of unloaded plate
C
      PHRASE =
     1 'Fundamental frequency of unloaded plate'
      CALL BLANKX(PHRASE,IENDP4)
      JXX = JXX + 1
      FREQ(ILOADX) = 0.0
      IF (IBEHV(JXX).EQ.0) CALL BEHX3
     1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX  ,
     1 'Fundamental frequency of unloaded plate')
      IF (FREQ(ILOADX  ).EQ.0.)  FREQ(ILOADX  ) = 1.E+10
      IF (MINCPS(ILOADX  ).EQ.0.)  MINCPS(ILOADX  ) = 1.0
      IF (FSFREQ(ILOADX  ).EQ.0.)  FSFREQ(ILOADX  ) = 1.0
      KCONX = KCONX + 1
      CARX(KCONX) =FREQ(ILOADX  )
      WORDCX= '(FREQ('//CIX//')/MINCPS('//CIX//
     1  ')) / FSFREQ('//CIX//')'
      CALL CONX(FREQ(ILOADX  ),MINCPS(ILOADX  ),FSFREQ(ILOADX  )
     1,'Fundamental frequency of unloaded plate',
     1 'Minimum allowable value for the fundamental frequency',
     1 'Factor of safety for FREQ',
     1 2,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
     1 WORDMX,PCWORD,CPLOTX,ICARX)
      IF (IMODX.EQ.0) THEN
         CODPHR =
     1 '  Fundamental frequency of unloaded plate: '
         IENDP4 =43
         CODNAM ='FREQ('//CIX//')'
         MLET4 =4 + 4
         WORDBX(KCONX)= CODPHR(1:IENDP4)//CODNAM(1:MLET4)
         IF (NPRINX.GT.0) WRITE(IFILE8,'(I5,6X,G14.7,A,A)')
     1    KCONX,CARX(KCONX),CODPHR(1:IENDP4),CODNAM(1:MLET4)
      ENDIF
  160 CONTINUE
  161 CONTINUE
C
C  Behavior and constraints generated next for W:
C  W = Normal deflection under uniform pressure
C
      PHRASE =
     1 'Normal deflection under uniform pressure'
      CALL BLANKX(PHRASE,IENDP4)
      JXX = JXX + 1
      W(ILOADX) = 0.0
      IF (IBEHV(JXX).EQ.0) CALL BEHX4
     1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX  ,
```

```
     1 'Normal deflection under uniform pressure')
      IF (W(ILOADX  ).EQ.0.)  W(ILOADX  ) = 1.E-10
      IF (AW(ILOADX  ).EQ.0.)  AW(ILOADX  ) = 1.0
      IF (FW(ILOADX  ).EQ.0.)  FW(ILOADX  ) = 1.0
      KCONX = KCONX + 1
      CARX(KCONX) =W(ILOADX  )
      WORDCX= '(W('//CIX//')/AW('//CIX//
     1  ')) X FW('//CIX//')'
      CALL CONX(W(ILOADX  ),AW(ILOADX  ),FW(ILOADX  )
     1,'Normal deflection under uniform pressure',
     1 'Maximum allowable normal deflection under pressure',
     1 'Factor of safety for max deflection under pressure',
     1 1,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
     1 WORDMX,PCWORD,CPLOTX,ICARX)
      IF (IMODX.EQ.0) THEN
         CODPHR =
     1 '  Normal deflection under uniform pressure: '
         IENDP4 =44
         CODNAM ='W('//CIX//')'
         MLET4 =1 + 4
         WORDBX(KCONX)= CODPHR(1:IENDP4)//CODNAM(1:MLET4)
         IF (NPRINX.GT.0) WRITE(IFILE8,'(I5,6X,G14.7,A,A)')
     1    KCONX,CARX(KCONX),CODPHR(1:IENDP4),CODNAM(1:MLET4)
      ENDIF
  175 CONTINUE
  176 CONTINUE
C
C  NEXT, EVALUATE THE OBJECTIVE, OBJGEN:
      IF (ILOADX.EQ.1) THEN
         PHRASE ='Weight of the plate'
         CALL BLANKX(PHRASE,IENDP4)
         CALL OBJECT(IFILE8,NPRINX,IMODX,OBJGEN,
     1    'Weight of the plate')
      ENDIF
      NCONSX = ICONSX
C
C
      RETURN
      END
C
C
C
C
C
C  End of the final portion of STRUCT written by "GENTEXT"
C=====================================================================
C
C=DECK      TRANFR
      SUBROUTINE TRANFR(ARG1,ARG2,ARG3,ARG4,ARG5)
C
C  USER:  DO NOT FORGET TO MODIFY THE ARGUMENT LIST OF TRANFR AS
C         APPROPRIATE FOR YOUR CASE!
C
C  PURPOSE IS TO TRANSFER DATA FROM THE LABELLED COMMON BLOCKS
C  SET UP BY THE GENOPT CODE TO LABELLED COMMON OR ARGUMENTS IN
C  THE SUBROUTINE ARGUMENT LIST THAT MATCH PREVIOUSLY WRITTEN CODE
C  BY YOURSELF OR OTHER PROGRAM DEVELOPERS.  THE USER SHOULD ESTABLISH
C  THE ARGUMENT LIST AND/OR LABELLED COMMON BLOCKS THAT MATCH VARIABLES
```

```
C  IN THE PREVIOUSLY WRITTEN CODE.  FOR AN EXAMPLE, SEE THE DISCUSSION
C  OF THE CASE CALLED "PANEL".
C
C=========================================================================
C  Start of part of TRANFR written by "GENTEXT"
C  INSERT ADDITIONAL COMMON BLOCKS HERE: (THESE ARE "GENTEXT" VARIABLES)
      COMMON/FV07/AOBAXL(30),IAOBAXL
      REAL AOBAXL
      COMMON/FV08/KAXL(30)
      REAL KAXL
      COMMON/FV09/AOBSHR(20),IAOBSHR
      REAL AOBSHR
      COMMON/FV10/KSHR(20)
      REAL KSHR
      COMMON/FV11/Nx(20)
      REAL Nx
      COMMON/FV17/STRESS(20),MAXSTR(20),FSTRES(20)
      REAL STRESS,MAXSTR,FSTRES
      COMMON/FV20/BUCKLE(20),MINBUC(20),FBUCKL(20)
      REAL BUCKLE,MINBUC,FBUCKL
      COMMON/FV23/FREQ(20),MINCPS(20),FSFREQ(20)
      REAL FREQ,MINCPS,FSFREQ
      COMMON/FV26/W(20),AW(20),FW(20)
      REAL W,AW,FW
      COMMON/FV01/THICK,LENGTH,WIDTH,E,NU,RHO,WEIGHT
      REAL THICK,LENGTH,WIDTH,E,NU,RHO,WEIGHT
      COMMON/FV12/Ny(20),Nxy(20),PRESS(20)
      REAL Ny,Nxy,PRESS
C
C
C  End of part of TRANFR written by "GENTEXT"
C=========================================================================
C  INSERT ADDITIONAL DIMENSION AND/OR LABELLED COMMON BLOCKS HERE,
C  IF NECESSARY. THESE WOULD BE STATEMENTS THAT ARE CONSISTENT WITH
C  SUBROUTINES THAT YOU OR OTHERS MAY HAVE WRITTEN THAT ARE REQUIRED
C  FOR WHATEVER ANALYSIS YOU ARE NOW PERSUING.  MAKE SURE THERE ARE
C  NO NAME CONFLICTS WITH THE "GENTEXT" LABELLED COMMON BLOCKS.
C
C
C  INSERT APPROPRIATE FORTRAN STATEMENTS HERE (DON'T FORGET TO CORRECT
C  THE ARGUMENT LIST OF SUBROUTINE TRANFR!)
C  PROGRAM FILE:
C
C
      RETURN
      END
C
C
C
------------ end of struct.new (skeletal subroutines) -----------
```

PART 8 INSPECT THE "FLESHED OUT" behavior.plate FILE

(Next, the GENOPT user "fleshes out" the behavior.new library. The "fleshed out" behavior.new library, called "behavior.plate" in this directory, follows. NOTE: The FORTRAN statements added by the GENOPT user to “flesh out” the skeletal behavior.new file listed in PART 6 are listed here in **bold face.**)

```
------------ behavior.plate ("fleshed out" subroutines) ------------
C=DECK      BEHAVIOR.NEW

 (lines skipped to save space. These skipped lines are the
  same as those displayed for the skeletal file, behavior.new.)

C
C=DECK      BEHX1
      SUBROUTINE BEHX1
     1 (IFILE,NPRINX,IMODX,IFAST,ILOADX,PHRASE)
C
C   PURPOSE: OBTAIN Maximum effective (von Mises) stress
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     STRESS(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    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 = Maximum effective (von Mises) stress
C
C   OUTPUT:
C
C     STRESS(ILOADX)
C
       CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      COMMON/FV07/AOBAXL(30),IAOBAX
      REAL AOBAXL
      COMMON/FV08/KAXL(30)
      REAL KAXL
      COMMON/FV09/AOBSHR(20),IAOBSH
      REAL AOBSHR
      COMMON/FV10/KSHR(20)
      REAL KSHR
```

```
      COMMON/FV11/Nx(20)
      REAL Nx
      COMMON/FV17/STRESS(20),MAXSTR(20),FSTRES(20)
      REAL STRESS,MAXSTR,FSTRES
      COMMON/FV20/BUCKLE(20),MINBUC(20),FBUCKL(20)
      REAL BUCKLE,MINBUC,FBUCKL
      COMMON/FV23/FREQ(20),MINCPS(20),FSFREQ(20)
      REAL FREQ,MINCPS,FSFREQ
      COMMON/FV26/W(20),AW(20),FW(20)
      REAL W,AW,FW
      COMMON/FV01/THICK,LENGTH,WIDTH,E,NU,RHO,WEIGHT
      REAL THICK,LENGTH,WIDTH,E,NU,RHO,WEIGHT
      COMMON/FV12/Ny(20),Nxy(20),PRESS(20)
      REAL Ny,Nxy,PRESS
C
C
C  INSERT SUBROUTINE STATEMENTS HERE.
C
C
      SA1 = Nx(ILOADX)/THICK
      SB1 = Ny(ILOADX)/THICK
      SAB = Nxy(ILOADX)/THICK
      ALPHA = WIDTH/LENGTH
C
C  MAX. EFFECTIVE STRESS UNDER UNIFORM PRESSURE (LINEAR THEORY)
C  (TAKEN FROM ROARK, 3RD EDITION, 1954, TABLE X, FORMULA 36, P. 203
C   VALID FOR POISSON RATIO NU = 0.3)
C
      SA2 = (PRESS(ILOADX)*WIDTH**2/THICK**2)*(0.225 +0.382*ALPHA**2
     1                                       -0.320*ALPHA**3)
      SB2 = 0.75*PRESS(ILOADX)*WIDTH**2/
     1               (THICK**2*(1. +1.61*ALPHA**3))
C
C  EFFECTIVE STRESS AT SURFACE WHERE PRESSURE IS APPLIED:
C
      SATOP = SA1 - SA2
      SBTOP = SB1 - SB2
      SEFTOP= SQRT(SATOP**2 +SBTOP**2 - SATOP*SBTOP +3.*SAB**2)
C
C  EFFECTIVE STRESS AT OPPOSITE SURFACE:
C
      SABOT = SA1 + SA2
      SBBOT = SB1 + SB2
      SEFBOT= SQRT(SABOT**2 +SBBOT**2 - SABOT*SBBOT +3.*SAB**2)
C
      STRESS(ILOADX) = MAX(SEFTOP,SEFBOT)
C
C
      RETURN
      END
C
C
C
C
C=DECK      BEHX2
      SUBROUTINE BEHX2
     1 (IFILE,NPRINX,IMODX,IFAST,ILOADX,PHRASE)
C
```

```
C   PURPOSE: OBTAIN Buckling load factor
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     BUCKLE(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    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 = Buckling load factor
C
C   OUTPUT:
C
C     BUCKLE(ILOADX)
C
       CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      COMMON/FV07/AOBAXL(30),IAOBAX
      REAL AOBAXL
      COMMON/FV08/KAXL(30)
      REAL KAXL
      COMMON/FV09/AOBSHR(20),IAOBSH
      REAL AOBSHR
      COMMON/FV10/KSHR(20)
      REAL KSHR
      COMMON/FV11/Nx(20)
      REAL Nx
      COMMON/FV17/STRESS(20),MAXSTR(20),FSTRES(20)
      REAL STRESS,MAXSTR,FSTRES
      COMMON/FV20/BUCKLE(20),MINBUC(20),FBUCKL(20)
      REAL BUCKLE,MINBUC,FBUCKL
      COMMON/FV23/FREQ(20),MINCPS(20),FSFREQ(20)
      REAL FREQ,MINCPS,FSFREQ
      COMMON/FV26/W(20),AW(20),FW(20)
      REAL W,AW,FW
      COMMON/FV01/THICK,LENGTH,WIDTH,E,NU,RHO,WEIGHT
      REAL THICK,LENGTH,WIDTH,E,NU,RHO,WEIGHT
      COMMON/FV12/Ny(20),Nxy(20),PRESS(20)
      REAL Ny,Nxy,PRESS
C
C
C  INSERT SUBROUTINE STATEMENTS HERE.
```

```
C
C
C
      BUCKLE(ILOADX) = 0.
C
      GO TO (10,20,30), ILOADX
C
   10 CONTINUE
C
C  BUCKLING LOAD FACTOR UNDER UNIFORM AXIAL COMPRESSION, Nx:
C  TAKEN FROM ROARK, 3RD EDITION, 1954, TABLE XVI, FORMULA 1, P. 312
C
      CALL INTERP(IFILE,IAOBAX,AOBAXL,KAXL,LENGTH/WIDTH,COEFAX)
C     WRITE(6,*)' COEFAX,LENGTH/WIDTH =',COEFAX,LENGTH/WIDTH
      IF (Nx(ILOADX).LT.0.)
     1 BUCKLE(ILOADX) =(COEFAX*(E*THICK/(1.-NU**2))*(THICK/WIDTH)**2)/
     1 ABS(Nx(ILOADX))
      GO TO 50
C
   20 CONTINUE
C
C  BUCKLING UNDER UNIFORM IN-PLANE SHEAR, Nxy:
C  AGAIN, TAKEN FROM ROARK, 3RD EDITION, TABLE XVI, FORMULA 10, P. 313
C
      CALL INTERP(IFILE,IAOBSH,AOBSHR,KSHR,LENGTH/WIDTH,COEFSH)
C     WRITE(6,*)' COEFSH,LENGTH/WIDTH =',COEFSH,LENGTH/WIDTH
      IF (Nxy(ILOADX).NE.0.)
     1 BUCKLE(ILOADX)=(COEFSH*(E*THICK/(1.-NU**2))*(THICK/WIDTH)**2)/
     1 ABS(Nxy(ILOADX))
      GO TO 50
C
   30 CONTINUE
C
C  THERE IS NO BUCKLING UNDER UNIFORM NORMAL PRESSURE.
C
   50 CONTINUE
C
C
      RETURN
      END
C
C
C
C
C=DECK      BEHX3
      SUBROUTINE BEHX3
     1 (IFILE,NPRINX,IMODX,IFAST,ILOADX,PHRASE)
C
C   PURPOSE: OBTAIN Fundamental frequency of unloaded plate
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     FREQ(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    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 = Fundamental frequency of unloaded plate
C
C   OUTPUT:
C
C     FREQ(ILOADX)
C
       CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      COMMON/FV07/AOBAXL(30),IAOBAX
      REAL AOBAXL
      COMMON/FV08/KAXL(30)
      REAL KAXL
      COMMON/FV09/AOBSHR(20),IAOBSH
      REAL AOBSHR
      COMMON/FV10/KSHR(20)
      REAL KSHR
      COMMON/FV11/Nx(20)
      REAL Nx
      COMMON/FV17/STRESS(20),MAXSTR(20),FSTRES(20)
      REAL STRESS,MAXSTR,FSTRES
      COMMON/FV20/BUCKLE(20),MINBUC(20),FBUCKL(20)
      REAL BUCKLE,MINBUC,FBUCKL
      COMMON/FV23/FREQ(20),MINCPS(20),FSFREQ(20)
      REAL FREQ,MINCPS,FSFREQ
      COMMON/FV26/W(20),AW(20),FW(20)
      REAL W,AW,FW
      COMMON/FV01/THICK,LENGTH,WIDTH,E,NU,RHO,WEIGHT
      REAL THICK,LENGTH,WIDTH,E,NU,RHO,WEIGHT
      COMMON/FV12/Ny(20),Nxy(20),PRESS(20)
      REAL Ny,Nxy,PRESS
C
C
C  INSERT SUBROUTINE STATEMENTS HERE.
C
C
C
      FREQ(ILOADX) = 0.
C
      GO TO (50,50,30),ILOADX
C
C  FIND FUNDAMENTAL NATURAL FREQUENCY FOR UNLOADED PLATE.
C  THE VALUE ESTABLISHED FOR ILOADX = 3 IGNORES THE EFFECT OF ANY TENSION
C  THAT MAY DEVELOP IN THE PLATE WHEN IT IS LOADED BY NORMAL
```

```
C  PRESSURE.  NO FREQUENCIES ARE CALCULATED FOR THE PLATE AS
C  LOADED AXIALLY (LOAD CASE 1) OR IN UNIFORM IN-PLANE SHEAR
C  (LOAD CASE 2).
C
   30 CONTINUE
      D = E*THICK**3/(12.*(1.-NU**2))
      DMASS = RHO/386.4
      PI = 3.1415927
C
C  CALCULATE FUNDAMENTAL FREQUENCY IN CPS, NOT RADIANS/SECOND:
C  TAKEN FROM LEISSA, "VIBRATION OF PLATES", NASA SP-160, P. 44,
C  EQ. 4.20:
C
      FREQ(ILOADX)
     1       = SQRT(D/DMASS)*((1./LENGTH)**2 + (1./WIDTH)**2)*.5*PI
C
   50 CONTINUE
C
C
      RETURN
      END
C
C
C
C
C=DECK      BEHX4
      SUBROUTINE BEHX4
     1 (IFILE,NPRINX,IMODX,IFAST,ILOADX,PHRASE)
C
C   PURPOSE: OBTAIN Normal deflection under uniform pressure
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     W(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    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 deflection under uniform pressure
C
C   OUTPUT:
C
```

```
C     W(ILOADX)
C
       CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      COMMON/FV07/AOBAXL(30),IAOBAX
      REAL AOBAXL
      COMMON/FV08/KAXL(30)
      REAL KAXL
      COMMON/FV09/AOBSHR(20),IAOBSH
      REAL AOBSHR
      COMMON/FV10/KSHR(20)
      REAL KSHR
      COMMON/FV11/Nx(20)
      REAL Nx
      COMMON/FV17/STRESS(20),MAXSTR(20),FSTRES(20)
      REAL STRESS,MAXSTR,FSTRES
      COMMON/FV20/BUCKLE(20),MINBUC(20),FBUCKL(20)
      REAL BUCKLE,MINBUC,FBUCKL
      COMMON/FV23/FREQ(20),MINCPS(20),FSFREQ(20)
      REAL FREQ,MINCPS,FSFREQ
      COMMON/FV26/W(20),AW(20),FW(20)
      REAL W,AW,FW
      COMMON/FV01/THICK,LENGTH,WIDTH,E,NU,RHO,WEIGHT
      REAL THICK,LENGTH,WIDTH,E,NU,RHO,WEIGHT
      COMMON/FV12/Ny(20),Nxy(20),PRESS(20)
      REAL Ny,Nxy,PRESS
C
C
C  INSERT SUBROUTINE STATEMENTS HERE.
C
C
C
C  AS WITH THE VIBRATION OF THE UNLOADED PLATE, HERE WE ARE
C  NOT CONCERNED WITH THE NORMAL DEFLECTION UNDER AXIAL COMPRESSION
C  (Nx, ILOADX=1) OR UNDER UNIFORM IN-PLANE SHEAR (Nxy, ILOADX=2).
C
       W(ILOADX) = 0
       GO TO (50,50,30),ILOADX
C
   30 CONTINUE
C
C  TAKEN FROM ROARK, 3RD EDITION, TABLE X, FORMULA 36, P. 203, 1954:
C
      W(ILOADX) = .1422*PRESS(ILOADX)*WIDTH**4/
     1          (E*THICK**3*(1. +2.21*(WIDTH/LENGTH)**3))
C
   50 CONTINUE
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)
```

```
 (lines skipped to save space. SUBROUTINE USRCON is not
  changed by the GENOPT user. It is listed above in behavior.new)
C
      RETURN
      END
C
C
C
C=DECK      USRLNK
      SUBROUTINE USRLNK(VARI,I,VARIAB)

 (lines skipped to save space. SUBROUTINE USRLNK is not
  changed by the GENOPT user. It is listed above in behavior.new)

      RETURN
      END
C
C=DECK      OBJECT
      SUBROUTINE OBJECT(IFILE,NPRINX,IMODX,OBJGEN,PHRASE)
C   PURPOSE:Weight of the plate
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   DEFINITION OF PHRASE:
C     PHRASE = Weight of the plate
C
       CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      COMMON/FV07/AOBAXL(30),IAOBAX
      REAL AOBAXL
      COMMON/FV08/KAXL(30)
      REAL KAXL
      COMMON/FV09/AOBSHR(20),IAOBSH
      REAL AOBSHR
      COMMON/FV10/KSHR(20)
      REAL KSHR
      COMMON/FV11/Nx(20)
      REAL Nx
      COMMON/FV17/STRESS(20),MAXSTR(20),FSTRES(20)
      REAL STRESS,MAXSTR,FSTRES
      COMMON/FV20/BUCKLE(20),MINBUC(20),FBUCKL(20)
      REAL BUCKLE,MINBUC,FBUCKL
      COMMON/FV23/FREQ(20),MINCPS(20),FSFREQ(20)
      REAL FREQ,MINCPS,FSFREQ
      COMMON/FV26/W(20),AW(20),FW(20)
      REAL W,AW,FW
      COMMON/FV01/THICK,LENGTH,WIDTH,E,NU,RHO,WEIGHT
      REAL THICK,LENGTH,WIDTH,E,NU,RHO,WEIGHT
      COMMON/FV12/Ny(20),Nxy(20),PRESS(20)
      REAL Ny,Nxy,PRESS
```

```
C
C
C  INSERT SUBROUTINE STATEMENTS HERE.
C
C  CALCULATE THE WEIGHT OF THE RECTANGULAR PLATE...
C
      WEIGHT = RHO*LENGTH*WIDTH*THICK
C
      OBJGEN =WEIGHT
C
C
      RETURN
      END
C
C
C
---------- end of behavior.plate ("fleshed out" subroutines) ----------

(Next, the GENOPT user must copy behavior.plate into behavior.new,
 then compile the set of GENOPT programs to be used by the end user.)
```

bush-> cp behavior.plate behavior.new

```
cp: overwrite `behavior.new'? y


----------------------------------------------------------
```

PART 9 COMPILE THE GENOPT PROGRAMS FOR THE GENERIC CASE, "plate"

bush-> genprograms

```
B (background) or F (foreground): f

 (A lot of stuff zips by on your screen. At the end of it
  you will see the following lines.)

------------ lines on screen at the end of ------------
------------ the execution of genprograms  ------------
Congratulations!  Your code compiled successfully.  You should
now check to make sure that you get correct results from a
simple test case with a known answer before attempting a more
complicated case.

Here is a list of all your newly created executables:
-rwxr-xr-x  1 bush bush 75811 Mar  1 15:02 autochange.linux
-rwxr-xr-x  1 bush bush 145829 Mar  1 15:02 begin.linux
-rwxr-xr-x  1 bush bush 121207 Mar  1 15:02 change.linux
-rwxr-xr-x  1 bush bush 147349 Mar  1 15:02 chooseplot.linux
-rwxr-xr-x  1 bush bush 152718 Mar  1 15:02 decide.linux
-rwxr-xr-x  1 bush bush 98008 Mar  1 15:02 mainsetup.linux
-rwxr-xr-x  1 bush bush 526334 Mar  1 15:02 optimize.linux
-rwxr-xr-x  1 bush bush 111738 Mar  1 15:02 store.linux

Next, type the command BEGIN to input data for a new case.
--- end of lines on screen at the end of genprograms -------
```

```
(The GENOPT user's tasks are done. There now exists a
 generic capability to optimize isotropic plates under
 certain kinds of loading.

 Next, the end user begins his tasks: executing BEGIN,
 DECIDE, MAINSETUP, OPTIMIZE or SUPEROPT, etc. etc.)
```

PART 10 THE END USER RUNS "BEGIN" FOR THE SPECIFIC CASE, "plate1"

bush-> begin

```
GENOPT = /home/progs/genopt

 THE NAME OF THE PROMPT FILE ASKED FOR NEXT
 IS THE NAME OF THE CLASS OF PROBLEMS THAT THE GENOPT-USER
 HAS CHOSEN, NOT THE NAME OF THE PARTICULAR CASE BEING
 STUDIED HERE. IT IS THE "NAME" PART OF "NAME".PRO.

  ENTER THE GENERIC CASE NAME: plate

 FROM HERE ON, WHENEVER THE CASE NAME IS REQUESTED,
 YOU PROVIDE THE NAME OF THE PARTICULAR INSTANCE IN THE CLASS
 OF PROBLEMS THAT YOU ARE NOW STUDYING.  THIS NAME MUST BE
 DIFFERENT FROM THE NAME YOU HAVE JUST PROVIDED ABOVE.

  ENTER THE SPECIFIC CASE NAME: plate1


 ****************   BEGIN   ******************

 Purpose of BEGIN is to permit you to provide a starting design
 in an interactive mode. You give starting dimensions, material
 properties, allowables. The interactive session is stored on
 a file called plate1.BEG, in which plate1 is a name that you
 have chosen for the specific case. (The name, plate1 must
 remain the same as you use BEGIN, DECIDE, MAINSETUP, OPTIMIZE,
 and CHANGE.)  In future runs of the same or a
 slightly modified case, you will find it convenient to use the
 file plate1.BEG as input.  Rather than answer all the questions
 interactively, you can use plate1.BEG or an edited version of
 plate1.BEG as input to BEGIN.  BEGIN also generates an output
 file called plate1.OPB. OPB lists a summary of the case, and if
 you choose the tutorial option, the questions, helps, and your
 answers for each input datum.

 *************************************************

 Are you correcting, adding to, or using an existing file?=y
 y
 Do you want a tutorial session and tutorial output?= N
 Now you start to provide input data. You will be prompted by
 short questions. If you need help, just type H as an
 answer to the prompt instead of the datum called for. In most
```

instances you will then be given more information on the datum you must provide. It may be a good idea to run the tutorial option if you are a new user of this program.

```
PROGRAM FOR OPTIMIZATION OF A RECTANGULAR PLATE
SUBJECTED TO SEVERAL LOADING ENVIRONMENTS AND
CONSTRAINTS ON STRESS, BUCKLING, DISPLACEMENT, AND FREQUENCY.



FIRST, PROVIDE ALL VARIABLES THAT CAN BE DECISION VARIABLES,
THAT IS, VARIABLES THAT CAN CHANGE DURING OPTIMIZATION
ITERATIONS (ROLE TYPE 1), AND FIXED VARIABLES (ROLE TYPE 2).

thickness of the plate: THICK=  0.1000000
Length of the plate: LENGTH=        10
Width of the plate: WIDTH=   6.666700
Young's modulus of the plate material: E=  0.1000000E+08
Poisson's ratio of the plate material: NU=  0.3000000
Weight density (e.g. lb/in**3) of the plate material: RHO=  0.1000000

THE FOLLOWING TABLE (FROM ROARK, 3RD EDITION, TABLE XVI,
ENTRY NO. 1, PAGE 312), GIVES THE RELATIONSHIP OF PLATE
(LENGTH/WIDTH) TO A COEFFICIENT FOR BUCKLING UNDER UNIFORM
AXIAL COMPRESSION.


DEFINITION OF THE ROW INDEX OF THE ARRAY, AOBAXL =
 Number of entries in the table of axial buckling v. a/b

Number IAOBAXL of rows in the array  AOBAXL: IAOBAXL=        16
(plate length, LENGTH)/(plate width, WIDTH): AOBAXL(  1)=  0.2000000
(plate length, LENGTH)/(plate width, WIDTH): AOBAXL(  2)=  0.3000000
(plate length, LENGTH)/(plate width, WIDTH): AOBAXL(  3)=  0.4000000
(plate length, LENGTH)/(plate width, WIDTH): AOBAXL(  4)=  0.6000000
(plate length, LENGTH)/(plate width, WIDTH): AOBAXL(  5)=  0.8000000
(plate length, LENGTH)/(plate width, WIDTH): AOBAXL(  6)=   1.000000
(plate length, LENGTH)/(plate width, WIDTH): AOBAXL(  7)=   1.200000
(plate length, LENGTH)/(plate width, WIDTH): AOBAXL(  8)=   1.400000
(plate length, LENGTH)/(plate width, WIDTH): AOBAXL(  9)=   1.600000
(plate length, LENGTH)/(plate width, WIDTH): AOBAXL( 10)=   1.800000
(plate length, LENGTH)/(plate width, WIDTH): AOBAXL( 11)=   2.000000
(plate length, LENGTH)/(plate width, WIDTH): AOBAXL( 12)=   2.200000
(plate length, LENGTH)/(plate width, WIDTH): AOBAXL( 13)=   2.400000
(plate length, LENGTH)/(plate width, WIDTH): AOBAXL( 14)=   2.700000
(plate length, LENGTH)/(plate width, WIDTH): AOBAXL( 15)=   3.000000
(plate length, LENGTH)/(plate width, WIDTH): AOBAXL( 16)=   30.00000

DEFINITION OF THE ROW INDEX OF THE ARRAY, KAXL =
 Number of entries in the table of axial buckling v. a/b

buckling coefficient: uniform axial compression: KAXL(  1)=   22.20000
buckling coefficient: uniform axial compression: KAXL(  2)=   10.90000
buckling coefficient: uniform axial compression: KAXL(  3)=   6.920000
buckling coefficient: uniform axial compression: KAXL(  4)=   4.230000
buckling coefficient: uniform axial compression: KAXL(  5)=   3.450000
buckling coefficient: uniform axial compression: KAXL(  6)=   3.290000
```

```
buckling coefficient: uniform axial compression: KAXL(  7)=   3.400000
buckling coefficient: uniform axial compression: KAXL(  8)=   3.680000
buckling coefficient: uniform axial compression: KAXL(  9)=   3.450000
buckling coefficient: uniform axial compression: KAXL( 10)=   3.320000
buckling coefficient: uniform axial compression: KAXL( 11)=   3.290000
buckling coefficient: uniform axial compression: KAXL( 12)=   3.320000
buckling coefficient: uniform axial compression: KAXL( 13)=   3.400000
buckling coefficient: uniform axial compression: KAXL( 14)=   3.320000
buckling coefficient: uniform axial compression: KAXL( 15)=   3.290000
buckling coefficient: uniform axial compression: KAXL( 16)=   3.290000

THE FOLLOWING TABLE (FROM ROARK, 3RD EDITION, TABLE XVI,
ENTRY NO. 10, PAGE 312), GIVES THE RELATIONSHIP OF PLATE
(LENGTH/WIDTH) TO A COEFFICIENT FOR BUCKLING UNDER UNIFORM
IN-PLANE SHEAR.


DEFINITION OF THE ROW INDEX OF THE ARRAY, AOBSHR =
 Number of entries in the table of shear buckling v. a/b

Number IAOBSHR of rows in the array  AOBSHR: IAOBSHR=        13
(plate length, LENGTH)/(plate width, WIDTH): AOBSHR(  1)=  0.6666700
(plate length, LENGTH)/(plate width, WIDTH): AOBSHR(  2)=  0.7143000
(plate length, LENGTH)/(plate width, WIDTH): AOBSHR(  3)=  0.8333300
(plate length, LENGTH)/(plate width, WIDTH): AOBSHR(  4)=   1.000000
(plate length, LENGTH)/(plate width, WIDTH): AOBSHR(  5)=   1.200000
(plate length, LENGTH)/(plate width, WIDTH): AOBSHR(  6)=   1.400000
(plate length, LENGTH)/(plate width, WIDTH): AOBSHR(  7)=   1.500000
(plate length, LENGTH)/(plate width, WIDTH): AOBSHR(  8)=   1.600000
(plate length, LENGTH)/(plate width, WIDTH): AOBSHR(  9)=   1.800000
(plate length, LENGTH)/(plate width, WIDTH): AOBSHR( 10)=   2.000000
(plate length, LENGTH)/(plate width, WIDTH): AOBSHR( 11)=   2.500000
(plate length, LENGTH)/(plate width, WIDTH): AOBSHR( 12)=   3.000000
(plate length, LENGTH)/(plate width, WIDTH): AOBSHR( 13)=   30.00000

DEFINITION OF THE ROW INDEX OF THE ARRAY, KSHR =
 Number of entries in the table of shear buckling v. a/b

buckling coefficient: uniform in-plane shear: KSHR(  1)=   5.840000
buckling coefficient: uniform in-plane shear: KSHR(  2)=   6.000000
buckling coefficient: uniform in-plane shear: KSHR(  3)=   6.580000
buckling coefficient: uniform in-plane shear: KSHR(  4)=   7.750000
buckling coefficient: uniform in-plane shear: KSHR(  5)=   6.580000
buckling coefficient: uniform in-plane shear: KSHR(  6)=   6.000000
buckling coefficient: uniform in-plane shear: KSHR(  7)=   5.840000
buckling coefficient: uniform in-plane shear: KSHR(  8)=   5.760000
buckling coefficient: uniform in-plane shear: KSHR(  9)=   5.590000
buckling coefficient: uniform in-plane shear: KSHR( 10)=   5.430000
buckling coefficient: uniform in-plane shear: KSHR( 11)=   5.180000
buckling coefficient: uniform in-plane shear: KSHR( 12)=   5.020000
buckling coefficient: uniform in-plane shear: KSHR( 13)=   4.400000

NEXT, PROVIDE ALL ENVIRONMENTAL PARAMETERS (LOADS, TEMPERATURES)

Number NCASES  of load cases (environments): NCASES=         3
Axial tension/width of the plate (lb/in): Nx(  1)=     -1000
Axial tension/width of the plate (lb/in): Nx(  2)=         0
Axial tension/width of the plate (lb/in): Nx(  3)=   0.000000
```

```
Transverse tension/length of the plate (lb/in): Ny(  1)=         0
Transverse tension/length of the plate (lb/in): Ny(  2)=         0
Transverse tension/length of the plate (lb/in): Ny(  3)=         0
In-plane shear/length applied to the plate: Nxy(  1)=         0
In-plane shear/length applied to the plate: Nxy(  2)=      1500
In-plane shear/length applied to the plate: Nxy(  3)=         0
Uniform normal pressure on the plate: PRESS(  1)=         0
Uniform normal pressure on the plate: PRESS(  2)=         0
Uniform normal pressure on the plate: PRESS(  3)=        12

NEXT, PROVIDE RESPONSE PARAMETERS, ALLOWABLES, AND FACTORS OF
SAFETY. THE ORDER IN WHICH YOU MUST PROVIDE THESE DATA IS:
A,B,C; A,B,C; A,B,C; A,B,C; etc, in which
A = RESPONSE,   B = ALLOWABLE,   C = FACTOR OF SAFETY.


DEFINITION OF THE ROW INDEX OF THE ARRAY, STRESS =
 Number of load cases (number of environments)

Maximum effective (von Mises) stress: STRESS

Maximum effective (von Mises) stress: STRESS
Maximum effective (von Mises) stress: STRESS

DEFINITION OF THE ROW INDEX OF THE ARRAY, MAXSTR =
 Number of load cases (number of environments)

Maximum effective (von Mises) stress allowed: MAXSTR(  1)=     30000
Maximum effective (von Mises) stress allowed: MAXSTR(  2)=     30000
Maximum effective (von Mises) stress allowed: MAXSTR(  3)=     30000

DEFINITION OF THE ROW INDEX OF THE ARRAY, FSTRES =
 Number of load cases (number of environments)

Factor of safety for effective stress: FSTRES(  1)=   1.100000
Factor of safety for effective stress: FSTRES(  2)=   1.100000
Factor of safety for effective stress: FSTRES(  3)=   1.000000
DEFINITION OF THE ROW INDEX OF THE ARRAY, BUCKLE =
 Number of load cases (number of environments)

Buckling load factor: BUCKLE

Buckling load factor: BUCKLE
Buckling load factor: BUCKLE

DEFINITION OF THE ROW INDEX OF THE ARRAY, MINBUC =
 Number of load cases (number of environments)

Minimum allowable buckling load factor (use 1.0): MINBUC(  1)=         1
Minimum allowable buckling load factor (use 1.0): MINBUC(  2)=         1
Minimum allowable buckling load factor (use 1.0): MINBUC(  3)=         1

DEFINITION OF THE ROW INDEX OF THE ARRAY, FBUCKL =
 Number of load cases (number of environments)

Factor of safety for buckling load factor: FBUCKL(  1)=   1.200000
Factor of safety for buckling load factor: FBUCKL(  2)=   1.200000
Factor of safety for buckling load factor: FBUCKL(  3)=         1
```

```
DEFINITION OF THE ROW INDEX OF THE ARRAY, FREQ =
 Number of load cases (number of environments)

Fundamental frequency of unloaded plate: FREQ

Fundamental frequency of unloaded plate: FREQ
Fundamental frequency of unloaded plate: FREQ

DEFINITION OF THE ROW INDEX OF THE ARRAY, MINCPS =
 Number of load cases (number of environments)

Minimum allowable value for the fundamental frequency: MINCPS(  1)=        0
Minimum allowable value for the fundamental frequency: MINCPS(  2)=        0
Minimum allowable value for the fundamental frequency: MINCPS(  3)=      130

DEFINITION OF THE ROW INDEX OF THE ARRAY, FSFREQ =
 Number of load cases (number of environments)

Factor of safety for FREQ: FSFREQ(  1)=   1.000000
Factor of safety for FREQ: FSFREQ(  2)=   1.000000
Factor of safety for FREQ: FSFREQ(  3)=   1.000000
DEFINITION OF THE ROW INDEX OF THE ARRAY, W =
 Number of load cases (number of environments)

Normal deflection under uniform pressure: W

Normal deflection under uniform pressure: W
Normal deflection under uniform pressure: W

DEFINITION OF THE ROW INDEX OF THE ARRAY, AW =
 Number of load cases (number of environments)

Maximum allowable normal deflection under pressure: AW(  1)=        0
Maximum allowable normal deflection under pressure: AW(  2)=        0
Maximum allowable normal deflection under pressure: AW(  3)=  0.1000000

DEFINITION OF THE ROW INDEX OF THE ARRAY, FW =
 Number of load cases (number of environments)

Factor of safety for max deflection under pressure: FW(  1)=        0
Factor of safety for max deflection under pressure: FW(  2)=        0
Factor of safety for max deflection under pressure: FW(  3)=        1
LAST, AN OBJECTIVE MUST BE CHOSEN, SUCH AS MINIMUM WEIGHT
OR MINIMUM COST.

Weight of the plate: WEIGHT

  3 decision variable candidates have now been identified.
 50 decision variable candidates are permitted.
 47 additional decision variable candidates are allowed.

 61 fixed parameters have now been identified.
 99 fixed parameters are permitted.
 38 additional fixed parameters are allowed.

 12 environmental parameters have now been identified.
 50 environmental parameters are permitted.
 38 additional environmental parameters are allowed.
```

```
  12 allowables have now been identified.
  99 allowables are permitted.
  87 additional allowables are permitted.

  12 factors of safety have now been identified.
  99 factors of safety are permitted.
  87 additional factors of safety are allowed.

 DESCRIPTION OF FILES GENERATED BY THIS CASE:

 plate1.NAM = This file contains only the name of the case.
 plate1.BEG = Summary of interactive session you have just
          completed. This file can be edited and used for
          future runs of BEGIN.

 plate1.CBL = Contains the plate1 data base.

 plate1.OPB = Output from BEGIN. Please list this file and
          inspect it and the plate1.BEG file carefully before
          proceeding.

 For further information about files generated during operation
 of GENOPT give the command HELPG FILES.

 Next, give the command   DECIDE or CHANGE   .
---------- end of the interactive "BEGIN" session ---------


(The interactive input data for BEGIN is stored on a
 file called "plate1.BEG". The plate1.BEG file follows)

--------- plate1.BEG file (input for BEGIN) ---------------
      N         $ Do you want a tutorial session and tutorial output?
  0.1000000     $ thickness of the plate: THICK
        10      $ Length of the plate: LENGTH
   6.666700     $ Width of the plate: WIDTH
  0.1000000E+08 $ Young's modulus of the plate material: E
  0.3000000     $ Poisson's ratio of the plate material: NU
  0.1000000     $ Weight density (e.g. lb/in**3) of the plate material: RHO
        16      $ Number IAOBAX of rows in the array  AOBAXL: IAOBAX
  0.2000000     $ (plate length, LENGTH)/(plate width, WIDTH): AOBAXL( 1)
  0.3000000     $ (plate length, LENGTH)/(plate width, WIDTH): AOBAXL( 2)
  0.4000000     $ (plate length, LENGTH)/(plate width, WIDTH): AOBAXL( 3)
  0.6000000     $ (plate length, LENGTH)/(plate width, WIDTH): AOBAXL( 4)
  0.8000000     $ (plate length, LENGTH)/(plate width, WIDTH): AOBAXL( 5)
   1.000000     $ (plate length, LENGTH)/(plate width, WIDTH): AOBAXL( 6)
   1.200000     $ (plate length, LENGTH)/(plate width, WIDTH): AOBAXL( 7)
   1.400000     $ (plate length, LENGTH)/(plate width, WIDTH): AOBAXL( 8)
   1.600000     $ (plate length, LENGTH)/(plate width, WIDTH): AOBAXL( 9)
   1.800000     $ (plate length, LENGTH)/(plate width, WIDTH): AOBAXL(10)
   2.000000     $ (plate length, LENGTH)/(plate width, WIDTH): AOBAXL(11)
   2.200000     $ (plate length, LENGTH)/(plate width, WIDTH): AOBAXL(12)
   2.400000     $ (plate length, LENGTH)/(plate width, WIDTH): AOBAXL(13)
   2.700000     $ (plate length, LENGTH)/(plate width, WIDTH): AOBAXL(14)
   3.000000     $ (plate length, LENGTH)/(plate width, WIDTH): AOBAXL(15)
   30.00000     $ (plate length, LENGTH)/(plate width, WIDTH): AOBAXL(16)
   22.20000     $ Coefficient for buckling under uniform axial compression: KAXL( 1)
```

```
 10.90000     $ Coefficient for buckling under uniform axial compression: KAXL( 2)
 6.920000     $ Coefficient for buckling under uniform axial compression: KAXL( 3)
 4.230000     $ Coefficient for buckling under uniform axial compression: KAXL( 4)
 3.450000     $ Coefficient for buckling under uniform axial compression: KAXL( 5)
 3.290000     $ Coefficient for buckling under uniform axial compression: KAXL( 6)
 3.400000     $ Coefficient for buckling under uniform axial compression: KAXL( 7)
 3.680000     $ Coefficient for buckling under uniform axial compression: KAXL( 8)
 3.450000     $ Coefficient for buckling under uniform axial compression: KAXL( 9)
 3.320000     $ Coefficient for buckling under uniform axial compression: KAXL(10)
 3.290000     $ Coefficient for buckling under uniform axial compression: KAXL(11)
 3.320000     $ Coefficient for buckling under uniform axial compression: KAXL(12)
 3.400000     $ Coefficient for buckling under uniform axial compression: KAXL(13)
 3.320000     $ Coefficient for buckling under uniform axial compression: KAXL(14)
 3.290000     $ Coefficient for buckling under uniform axial compression: KAXL(15)
 3.290000     $ Coefficient for buckling under uniform axial compression: KAXL(16)
      13      $ Number IAOBSH of rows in the array  AOBSHR: IAOBSH
0.6666700     $ (plate length, LENGTH)/(plate width, WIDTH): AOBSHR( 1)
0.7143000     $ (plate length, LENGTH)/(plate width, WIDTH): AOBSHR( 2)
0.8333300     $ (plate length, LENGTH)/(plate width, WIDTH): AOBSHR( 3)
 1.000000     $ (plate length, LENGTH)/(plate width, WIDTH): AOBSHR( 4)
 1.200000     $ (plate length, LENGTH)/(plate width, WIDTH): AOBSHR( 5)
 1.400000     $ (plate length, LENGTH)/(plate width, WIDTH): AOBSHR( 6)
 1.500000     $ (plate length, LENGTH)/(plate width, WIDTH): AOBSHR( 7)
 1.600000     $ (plate length, LENGTH)/(plate width, WIDTH): AOBSHR( 8)
 1.800000     $ (plate length, LENGTH)/(plate width, WIDTH): AOBSHR( 9)
 2.000000     $ (plate length, LENGTH)/(plate width, WIDTH): AOBSHR(10)
 2.500000     $ (plate length, LENGTH)/(plate width, WIDTH): AOBSHR(11)
 3.000000     $ (plate length, LENGTH)/(plate width, WIDTH): AOBSHR(12)
 30.00000     $ (plate length, LENGTH)/(plate width, WIDTH): AOBSHR(13)
 5.840000     $ Coefficient for buckling under uniform in-plane shear: KSHR( 1)
 6.000000     $ Coefficient for buckling under uniform in-plane shear: KSHR( 2)
 6.580000     $ Coefficient for buckling under uniform in-plane shear: KSHR( 3)
 7.750000     $ Coefficient for buckling under uniform in-plane shear: KSHR( 4)
 6.580000     $ Coefficient for buckling under uniform in-plane shear: KSHR( 5)
 6.000000     $ Coefficient for buckling under uniform in-plane shear: KSHR( 6)
 5.840000     $ Coefficient for buckling under uniform in-plane shear: KSHR( 7)
 5.760000     $ Coefficient for buckling under uniform in-plane shear: KSHR( 8)
 5.590000     $ Coefficient for buckling under uniform in-plane shear: KSHR( 9)
 5.430000     $ Coefficient for buckling under uniform in-plane shear: KSHR(10)
 5.180000     $ Coefficient for buckling under uniform in-plane shear: KSHR(11)
 5.020000     $ Coefficient for buckling under uniform in-plane shear: KSHR(12)
 4.400000     $ Coefficient for buckling under uniform in-plane shear: KSHR(13)
       3      $ Number NCASES of load cases (environments): NCASES
   -1000      $ Axial tension per unit width of the plate (lb/in): Nx( 1)
       0      $ Axial tension per unit width of the plate (lb/in): Nx( 2)
0.0000000E+00 $ Axial tension per unit width of the plate (lb/in): Nx( 3)
       0      $ Transverse tension per unit length of the plate (lb/in): Ny( 1)
       0      $ Transverse tension per unit length of the plate (lb/in): Ny( 2)
       0      $ Transverse tension per unit length of the plate (lb/in): Ny( 3)
       0      $ In-plane shear per unit edge length applied to the plate.: Nxy( 1)
    1500      $ In-plane shear per unit edge length applied to the plate.: Nxy( 2)
       0      $ In-plane shear per unit edge length applied to the plate.: Nxy( 3)
       0      $ Uniform normal pressure on the plate: PRESS( 1)
       0      $ Uniform normal pressure on the plate: PRESS( 2)
      12      $ Uniform normal pressure on the plate: PRESS( 3)
   30000      $ Maximum effective (von Mises) stress allowed: MAXSTR( 1)
   30000      $ Maximum effective (von Mises) stress allowed: MAXSTR( 2)
   30000      $ Maximum effective (von Mises) stress allowed: MAXSTR( 3)
```

```
   1.100000     $ Factor of safety for effective stress: FSTRES( 1)
   1.100000     $ Factor of safety for effective stress: FSTRES( 2)
   1.000000     $ Factor of safety for effective stress: FSTRES( 3)
         1      $ Minimum allowable buckling load factor (use 1.0): MINBUC( 1)
         1      $ Minimum allowable buckling load factor (use 1.0): MINBUC( 2)
         1      $ Minimum allowable buckling load factor (use 1.0): MINBUC( 3)
   1.200000     $ Factor of safety for buckling load factor: FBUCKL( 1)
   1.200000     $ Factor of safety for buckling load factor: FBUCKL( 2)
         1      $ Factor of safety for buckling load factor: FBUCKL( 3)
         0      $ Minimum allowable value for the fundamental frequency: MINCPS( 1)
         0      $ Minimum allowable value for the fundamental frequency: MINCPS( 2)
       130      $ Minimum allowable value for the fundamental frequency: MINCPS( 3)
   1.000000     $ Factor of safety for FREQ: FSFREQ( 1)
   1.000000     $ Factor of safety for FREQ: FSFREQ( 2)
   1.000000     $ Factor of safety for FREQ: FSFREQ( 3)
         0      $ Maximum allowable normal deflection under pressure: AW( 1)
         0      $ Maximum allowable normal deflection under pressure: AW( 2)
  0.1000000     $ Maximum allowable normal deflection under pressure: AW( 3)
         0      $ Factor of safety for max deflection under pressure: FW( 1)
         0      $ Factor of safety for max deflection under pressure: FW( 2)
         1      $ Factor of safety for max deflection under pressure: FW( 3)
-------------- end of plate1.BEG file (input for BEGIN -----------------
```

PART 11 THE END USER RUNS "DECIDE" FOR THE SPECIFIC CASE, "plate1"

(Next, execute DECIDE. Choose decision variables, lower and upper bounds,
and equality constraints, if any, and inequality constraints, if any.)

bush-> decide

```
  ENTER THE SPECIFIC CASE NAME: plate1


 *****************   DECIDE   *******************
 The purpose of DECIDE is to permit you to choose decision
 variables, linked variables, inequality constraints based on
 dimensions, not behavior, and escape variables for the
 optimization run or runs to follow. The results of the
 interactive session are saved in a file called plate1.DEC, in
 which plate1 is your name for the case. You may find this file
 useful for future runs of DECIDE in which you want to avoid
 answering many questions interactively. DECIDE also generates
 a file called plate1.OPD.  plate1.OPD contains a summary of
 optimization parameters. If you choose the tutorial option,
 plate1.OPD contains a complete list of the interactive session,
 including prompting questions, all "help" paragraphs, your
 responses to the prompting questions, and evolving lists
 of optimization parameters as they are chosen by you.
 ****************************************************

 Are you correcting, adding to, or using an existing file?=n
 n
 Do you want a tutorial session and tutorial output?= n
```

```
PARAMETERS FROM WHICH A DECISION VARIABLE MUST NOW BE CHOSEN
VAR.   CURRENT
NO.     VALUE            DEFINITION
 1    1.000E-01  thickness of the plate: THICK
 2    1.000E+01  Length of the plate: LENGTH
 3    6.667E+00  Width of the plate: WIDTH
Choose a decision variable (1,2,3,...)=         1
Lower bound of variable no.(  1)=  0.3000000E-01
Upper bound of variable no.(  1)=   1.000000

You will next be asked if this particular decision variable
should have its movement during optimization cycles especially
restricted. It is sometimes necessary to ensure that certain
of the decision variables don't change too much in successive
optimization cycles. This is often true for decision variables
such as angles that may have widely separated lower and upper
bounds (such as lower bound = 5.0 degrees and upper bound =
90 degrees) and for which the behavior is quite sensitive to
changes in this decision variable. In most cases you will
probably answer "no" (n or N) to the following prompting
question. However, if this decision variable is especially
critical, you may well wish to answer "yes" (y or Y). If you
answer y or Y you will then be asked to provide a maximum
move for this especially critical decision variable. For
example, for an angle in degrees you might well specify
that the maximum move be 5.0 degrees in one optimization cycle.

Do you want especially to restrict variable no.(  1)= n
Any more decision variables (Y or N) ?= y
           DECISION VARIABLES CHOSEN SO FAR
VAR.   CURRENT
NO.     VALUE            DEFINITION
 1    1.000E-01  thickness of the plate: THICK
PARAMETERS FROM WHICH A DECISION VARIABLE MUST NOW BE CHOSEN
VAR.   CURRENT
NO.     VALUE            DEFINITION
 2    1.000E+01  Length of the plate: LENGTH
 3    6.667E+00  Width of the plate: WIDTH
Choose a decision variable (1,2,3,...)=         2
Lower bound of variable no.(  2)=   5.000000
Upper bound of variable no.(  2)=   100.0000

You will next be asked if this particular decision variable
should have its movement during optimization cycles especially
restricted. It is sometimes necessary to ensure that certain
of the decision variables don't change too much in successive
optimization cycles. This is often true for decision variables
such as angles that may have widely separated lower and upper
bounds (such as lower bound = 5.0 degrees and upper bound =
90 degrees) and for which the behavior is quite sensitive to
changes in this decision variable. In most cases you will
probably answer "no" (n or N) to the following prompting
question. However, if this decision variable is especially
critical, you may well wish to answer "yes" (y or Y). If you
answer y or Y you will then be asked to provide a maximum
move for this especially critical decision variable. For
example, for an angle in degrees you might well specify
that the maximum move be 5.0 degrees in one optimization cycle.
```

```
Do you want especially to restrict variable no.(  2)= n
Any more decision variables (Y or N) ?= y
           DECISION VARIABLES CHOSEN SO FAR
VAR.   CURRENT
NO.     VALUE            DEFINITION
 1    1.000E-01  thickness of the plate: THICK
 2    1.000E+01  Length of the plate: LENGTH
PARAMETERS FROM WHICH A DECISION VARIABLE MUST NOW BE CHOSEN
VAR.   CURRENT
NO.     VALUE            DEFINITION
 3    6.667E+00  Width of the plate: WIDTH
Choose a decision variable (1,2,3,...)=         3
Lower bound of variable no.(  3)=   5.000000
Upper bound of variable no.(  3)=   10.00000

You will next be asked if this particular decision variable
should have its movement during optimization cycles especially
restricted. It is sometimes necessary to ensure that certain
of the decision variables don't change too much in successive
optimization cycles. This is often true for decision variables
such as angles that may have widely separated lower and upper
bounds (such as lower bound = 5.0 degrees and upper bound =
90 degrees) and for which the behavior is quite sensitive to
changes in this decision variable. In most cases you will
probably answer "no" (n or N) to the following prompting
question. However, if this decision variable is especially
critical, you may well wish to answer "yes" (y or Y). If you
answer y or Y you will then be asked to provide a maximum
move for this especially critical decision variable. For
example, for an angle in degrees you might well specify
that the maximum move be 5.0 degrees in one optimization cycle.

Do you want especially to restrict variable no.(  3)= n
Any more decision variables (Y or N) ?= n

  3 decision variables have now been identified.
 40 decision variables are permitted.
 37 additional decision variables are allowed.


Next, choose linked variables.

A linked variable is a variable that is not a decision
variable, but is expressed in terms of decision variables,
thus:

 (linked variable)  =  C1*(decision variable no. j1)**D1
                      +C2*(decision variable no. j2)**D2
                      +C3*(decision variable no. j3)**D3
                      +etc (up to max. of 5 terms)
                      +C0

 in which C1, C2,..;  and C0, and D1, D2... are constants.
 For example,
 suppose you have a problem in which the area of a plate
 is fixed, but either the length or width may be a decision
 variable. Suppose we arbitrarily choose the length to be
```

```
 the decision variable, then we know that

              WIDTH = AREA/LENGTH

 From the general expression above, C0 = 0.0, C1 = AREA,
 and D1 = -1.  Decision variable no. j1 is LENGTH.

 NOTE: YOU MAY USE THE SIMPLE POLYNOMIAL LINKING JUST EXPLAINED
       OR YOU MAY DEFINE YOUR OWN LINKING EXPRESSION IN
       SUBROUTINE USRLNK OF THE BEHAVIOR.F FILE.

Any linked variables (Y or N) ?= n

Next, establish inequality relations among variables
of the two forms:

 1.0 > f(v1,v2,v3,...)    or   1.0 < f(v1,v2,v3,...)

in which the expression f(v1,v2,v3,...) has the form:

  f(v1,v2,v3,...) = C0 +C1*v1**D1 +C2*v2**D2 +C3*v3**D3 + ...
                       +etc (up to max. of 15 terms).

The variables, v1, v2, v3,..., can be any of the variables
that are decision variables or potential candidates for
decision variables or linked variables.

Any inequality relations among variables? (type H)= y
Want to see an example of how to calculate C0, C1, D1,..?= y

For example, suppose that you want to impose the condition
that the area of a rectangular plate be greater than 50.
Let v1 = length of plate;  v2 = width of plate. You want
     v1*v2 > 50                              (1)
The expression (1) must be expressed in the form
       1.0 > C0 + C1*v1**D1 + C2*v2**D2      (2)
If we: (a) divide both sides of (1) by v2 (assume v2 is positive!);
       (b) subtract v1 from both sides of (1); and
       (c) add 1.0 to both sides of (1), we obtain
             1.0 > 1.0 -v1 +50/v2            (3)
From (3) we easily deduce that in (2): C0 = 1.0; C1 = -1.0;
D1 = 1.0; C2 = 50; and D2 = -1.0.  Eq.(3) represents the
first type of expression given above. The actual value of
the constraint, CONSTR, used later by the optimizer would be
           CONSTR = 2. - (1.0 -v1 +50/v2).
This constraint is critical if its value is less than 1.0.

Identify the type of inequality expression (1 or 2)=         2

Now start building the expression: C0 +C1*v1**D1 +C2*v2**D2 +...
You are allowed up to 45 terms in the expression, including C0.

Give a value to the constant, C0=   1.000000
LIST OF VARIABLES FROM WHICH YOU MUST NOW CHOOSE ANY ONE
VAR.   CURRENT
NO.     VALUE            DEFINITION
 1    1.000E-01  thickness of the plate: THICK
 2    1.000E+01  Length of the plate: LENGTH
```

```
 3    6.667E+00  Width of the plate: WIDTH
Choose a variable from the list above (1, 2, 3,...)=        2
Choose a value for the coefficient, C1=  -1.000000
Choose a value for the power, D1=   1.000000
1 <1-1.00*V(2) + ...
Any more terms in the expression: C0 +C1*v1**D1 +C2*v2**D2 +...= y
LIST OF VARIABLES FROM WHICH YOU MUST NOW CHOOSE ANY ONE
VAR.   CURRENT
NO.     VALUE            DEFINITION
 1    1.000E-01  thickness of the plate: THICK
 2    1.000E+01  Length of the plate: LENGTH
 3    6.667E+00  Width of the plate: WIDTH
Choose a variable from the list above (1, 2, 3,...)=        3
Choose a value for the coefficient, Cn=   100.0000
Choose a value for the power, Dn=  -1.000000
 1 <
    1-1.00*V(2)
    +100.0*V(3)**-1
Any more terms in the expression: C0 +C1*v1**D1 +C2*v2**D2 +...= n
Are there any more inequality expressions?= y
Identify the type of inequality expression (1 or 2)=        1
Now start building the expression: C0 +C1*v1**D1 +C2*v2**D2 +...
You are allowed up to 45 terms in the expression, including C0.

Give a value to the constant, C0=   1.000000
LIST OF VARIABLES FROM WHICH YOU MUST NOW CHOOSE ANY ONE
VAR.   CURRENT
NO.     VALUE            DEFINITION
 1    1.000E-01  thickness of the plate: THICK
 2    1.000E+01  Length of the plate: LENGTH
 3    6.667E+00  Width of the plate: WIDTH
Choose a variable from the list above (1, 2, 3,...)=        2
Choose a value for the coefficient, C1=  -1.000000
Choose a value for the power, D1=   1.000000
1 >1-1.00*V(2) + ...
Any more terms in the expression: C0 +C1*v1**D1 +C2*v2**D2 +...= y
LIST OF VARIABLES FROM WHICH YOU MUST NOW CHOOSE ANY ONE
VAR.   CURRENT
NO.     VALUE            DEFINITION
 1    1.000E-01  thickness of the plate: THICK
 2    1.000E+01  Length of the plate: LENGTH
 3    6.667E+00  Width of the plate: WIDTH
Choose a variable from the list above (1, 2, 3,...)=        3
Choose a value for the coefficient, Cn=   50.00000
Choose a value for the power, Dn=  -1.000000
 1 >
    1-1.00*V(2)
    +50.0*V(3)**-1
Any more terms in the expression: C0 +C1*v1**D1 +C2*v2**D2 +...= n
Are there any more inequality expressions?= y
Identify the type of inequality expression (1 or 2)=        1

Now start building the expression: C0 +C1*v1**D1 +C2*v2**D2 +...
You are allowed up to 45 terms in the expression, including C0.

Give a value to the constant, C0=   1.000000
LIST OF VARIABLES FROM WHICH YOU MUST NOW CHOOSE ANY ONE
VAR.   CURRENT
```

```
 NO.     VALUE            DEFINITION
  1    1.000E-01  thickness of the plate: THICK
  2    1.000E+01  Length of the plate: LENGTH
  3    6.667E+00  Width of the plate: WIDTH
 Choose a variable from the list above (1, 2, 3,...)=        2
 Choose a value for the coefficient, C1=  -1.000000
 Choose a value for the power, D1=   1.000000
 1 >1-1.00*V(2) + ...
 Any more terms in the expression: C0 +C1*v1**D1 +C2*v2**D2 +...= y
 LIST OF VARIABLES FROM WHICH YOU MUST NOW CHOOSE ANY ONE
 VAR.   CURRENT
 NO.     VALUE            DEFINITION
  1    1.000E-01  thickness of the plate: THICK
  2    1.000E+01  Length of the plate: LENGTH
  3    6.667E+00  Width of the plate: WIDTH
 Choose a variable from the list above (1, 2, 3,...)=        3
 Choose a value for the coefficient, Cn=   1.000000
 Choose a value for the power, Dn=   1.000000
  1 >
    1-1.00*V(2)
    +1.00*V(3)
 Any more terms in the expression: C0 +C1*v1**D1 +C2*v2**D2 +...= n
 Are there any more inequality expressions?= n
 Any escape variables (Y or N) ?= y
 Want to have escape variables chosen by default?= y
          ESCAPE VARIABLES FOR THE OPTIMIZATION PROBLEM
 VAR.   CURRENT
 NO.     VALUE            DEFINITION
  1    1.000E-01  thickness of the plate: THICK

 DESCRIPTION OF FILES GENERATED/USED BY THIS CASE:

 plate1.DEC = Summary of interactive session you have just
          completed. This file can be edited and used for
          future runs of DECIDE.

 plate1.CBL = Contains the plate1 data base.

 plate1.OPD = Output from DECIDE. Please list this file and
          inspect it and the plate1.DEC file carefully before
          proceeding.

 For further information about files generated during operation
 of GENOPT give the command HELPG FILES.

 Next, give the command   MAINSETUP   .
--------- end of the DECIDE interactive session ------------


(The interactive input for DECIDE has been saved on a
 file called plate1.DEC. A list of this file follows:)

----------- the plate1.DEC file (input for DECIDE) ------------
      n         $ Do you want a tutorial session and tutorial output?
         1      $ Choose a decision variable (1,2,3,...)
  0.3000000E-01 $ Lower bound of variable no.(  1)
   1.000000     $ Upper bound of variable no.(  1)
      n         $ Do you want especially to restrict variable no.(  1)
```

```
      y         $ Any more decision variables (Y or N) ?
         2      $ Choose a decision variable (1,2,3,...)
   5.000000     $ Lower bound of variable no.(  2)
   100.0000     $ Upper bound of variable no.(  2)
      n         $ Do you want especially to restrict variable no.(  2)
      y         $ Any more decision variables (Y or N) ?
         3      $ Choose a decision variable (1,2,3,...)
   5.000000     $ Lower bound of variable no.(  3)
   10.00000     $ Upper bound of variable no.(  3)
      n         $ Do you want especially to restrict variable no.(  3)
      n         $ Any more decision variables (Y or N) ?
      n         $ Any linked variables (Y or N) ?
      y         $ Any inequality relations among variables? (type H)
      y         $ Want to see an example of how to calculate C0, C1, D1,..?
         2      $ Identify the type of inequality expression (1 or 2)
   1.000000     $ Give a value to the constant, C0
         2      $ Choose a variable from the list above (1, 2, 3,...)
  -1.000000     $ Choose a value for the coefficient, C1
   1.000000     $ Choose a value for the power, D1
      y         $ Any more terms in the expression: C0 +C1*v1**D1 +C2*v2**D2 +...
         3      $ Choose a variable from the list above (1, 2, 3,...)
   100.0000     $ Choose a value for the coefficient, Cn
  -1.000000     $ Choose a value for the power, Dn
      n         $ Any more terms in the expression: C0 +C1*v1**D1 +C2*v2**D2 +...
      y         $ Are there any more inequality expressions?
         1      $ Identify the type of inequality expression (1 or 2)
   1.000000     $ Give a value to the constant, C0
         2      $ Choose a variable from the list above (1, 2, 3,...)
  -1.000000     $ Choose a value for the coefficient, C1
   1.000000     $ Choose a value for the power, D1
      y         $ Any more terms in the expression: C0 +C1*v1**D1 +C2*v2**D2 +...
         3      $ Choose a variable from the list above (1, 2, 3,...)
   50.00000     $ Choose a value for the coefficient, Cn
  -1.000000     $ Choose a value for the power, Dn
      n         $ Any more terms in the expression: C0 +C1*v1**D1 +C2*v2**D2 +...
      y         $ Are there any more inequality expressions?
         1      $ Identify the type of inequality expression (1 or 2)
   1.000000     $ Give a value to the constant, C0
         2      $ Choose a variable from the list above (1, 2, 3,...)
  -1.000000     $ Choose a value for the coefficient, C1
   1.000000     $ Choose a value for the power, D1
      y         $ Any more terms in the expression: C0 +C1*v1**D1 +C2*v2**D2 +...
         3      $ Choose a variable from the list above (1, 2, 3,...)
   1.000000     $ Choose a value for the coefficient, Cn
   1.000000     $ Choose a value for the power, Dn
      n         $ Any more terms in the expression: C0 +C1*v1**D1 +C2*v2**D2 +...
      n         $ Are there any more inequality expressions?
      y         $ Any escape variables (Y or N) ?
      y         $ Want to have escape variables chosen by default?
---------------- end of the plate1.DEC file -----------------------


---------------------------------------------------------
```

PART 12 THE END USER RUNS "MAINSETUP" FOR THE SPECIFIC CASE, "plate1"

(Next, provide input for MAINSETUP.)

bush-> mainsetup

```
  ENTER THE SPECIFIC CASE NAME: plate1
 Are you correcting, adding to, or using an existing file?=n
 Do you want a tutorial session and tutorial output?= n
 n

 *********************** MAINSETUP *************************
 The purpose of MAINSETUP is to permit you to choose an
 analysis mode for the case plate1.  The control integer is
 called ITYPE.  If ITYPE = 1, then design iterations are to
 be performed and the objective function minimized.  If
 ITYPE = 2 a simple analysis for a given design is to be
 performed (no optimization).  Results will be stored on the
 file called plate1.OPM.  Please inspect this file before
 doing more design iterations.
 ************************************************************


 ***********************************************************
 Next, choose behaviors NOT to process for load set,  1 of   3
 ***********************************************************

 You must provide input data, IBEHAV, for EACH LOAD SET.
 Provide IBEHAV for the next load set.

 Because of excessive time required for computer runs, you
 may not want to run all the analyses listed next.

 Please indicate, by choosing numbers at the left-hand margin,
 which of the analyses you do NOT want to run. Choose one at
 a time.

 If you want to run ALL the analyses, just hit zero (0).

      LIST FROM WHICH AN INDEX MUST NOW BE CHOSEN
 BEHAVIOR
  NUMBER                  DEFINITION
    1  Maximum effective (von Mises) stress: STRESS(1 )
    2  Buckling load factor: BUCKLE(1 )
    3  Fundamental frequency of unloaded plate: FREQ(1 )
    4  Normal deflection under uniform pressure: W(1 )
 Choose an analysis you DON'T want (1, 2,..), IBEHAV=        0

 ***********************************************************
 Next, choose behaviors NOT to process for load set,  2 of   3
 ***********************************************************

 You must provide input data, IBEHAV, for EACH LOAD SET.
 Provide IBEHAV for the next load set.

 Because of excessive time required for computer runs, you
 may not want to run all the analyses listed next.

 Please indicate, by choosing numbers at the left-hand margin,
 which of the analyses you do NOT want to run. Choose one at
```

```
a time.

If you want to run ALL the analyses, just hit zero (0).

     LIST FROM WHICH AN INDEX MUST NOW BE CHOSEN
BEHAVIOR
 NUMBER                  DEFINITION
   1  Maximum effective (von Mises) stress: STRESS(2 )
   2  Buckling load factor: BUCKLE(2 )
   3  Fundamental frequency of unloaded plate: FREQ(2 )
   4  Normal deflection under uniform pressure: W(2 )
Choose an analysis you DON'T want (1, 2,..), IBEHAV=         0

***********************************************************
Next, choose behaviors NOT to process for load set,  3 of   3
***********************************************************

You must provide input data, IBEHAV, for EACH LOAD SET.
Provide IBEHAV for the next load set.

Because of excessive time required for computer runs, you
may not want to run all the analyses listed next.

Please indicate, by choosing numbers at the left-hand margin,
which of the analyses you do NOT want to run. Choose one at
a time.

If you want to run ALL the analyses, just hit zero (0).

     LIST FROM WHICH AN INDEX MUST NOW BE CHOSEN
BEHAVIOR
 NUMBER                  DEFINITION
   1  Maximum effective (von Mises) stress: STRESS(3 )
   2  Buckling load factor: BUCKLE(3 )
   3  Fundamental frequency of unloaded plate: FREQ(3 )
   4  Normal deflection under uniform pressure: W(3 )
Choose an analysis you DON'T want (1, 2,..), IBEHAV=         0
NPRINT= output index (0=GOOD, 1=ok, 2=debug, 3=too much)=         0
Choose type of analysis (1=opt., 2=fixed, 3=sensit.) ITYPE=         1

Next you will be asked for the number of design iterations.
This is the number of iterations corresponding to a single
execution of "OPTIMIZE", not the total number of iterations
to be processed for your entire case. It is almost always
best to use a small number like 5 iterations. The best
optimization strategy is explained in connection with Fig. 83
on p. 582 of the long 1987 PANDA2 paper, "PANDA2 - Program
for minimum weight design of stiffened, composite, locally
buckled panels. Computers & Structures, Vol. 25, No. 4,
pp. 469 - 605, 1987. You should get an optimum design by
several executions of "OPTIMIZE" with 5 iterations in each
execution. Better yet, use SUPEROPT. With many executions of
"OPTIMIZE" and few design iterations with each execution you
obtain the most efficient convergence to an optimum design.
When you execute SUPEROPT you get more "starting" designs
per SUPEROPT run when you use a small number like 5 for the
number of iterations, therefore a more complete exploration of
design space in the search for the best "global" optimum design.
```

```
The developer of GENOPT almost always uses 5 iterations.

How many design iterations in this run (3 to 25)?=         5
Take "shortcuts" for perturbed designs (Y or N)?= n
Choose 1 or 2 or 3 or 4 or 5 for IDESIGN=         2

Next, choose a control for move limits to be used during
optimization cycles. By "move limits" we are referring to
the size of the boxes that appear in Fig. 2 of the paper,
"GENOPT - a program that writes user-friendly optimization code",
Int. J. Solids and Structures, Vol. 26, pp 1173- 1210, 1990.
You are given five choices: IMOVE = 1 or 2 or 3 or 4 or 5:

IMOVE = 1 means SMOVE = 0.10
IMOVE = 2 means SMOVE = 0.50
IMOVE = 3 means SMOVE = 0.01
IMOVE = 4 means SMOVE = 0.02
IMOVE = 5 means SMOVE = 0.05

Small SMOVE (initial move limit) keeps the boxes small and
leads to the requirement for many "OPTIMIZE" commands to
obtain an optimum design; the "conservative" approach may
be boring, but it may be the most reliable.  "Liberal" move
limits allow bigger boxes, generally leading to the need for
fewer "OPTIMIZEs".  However, the decision variables may
jump around a lot and have difficulty converging to those
corresponding to an optimum design.

THE BEST CHOICE INITIALLY IS TO USE IMOVE = 1

For early optimization cycles you can choose "liberal" move
limits, changing to more "conservative" move limits after
several "OPTIMIZEs".

In practical problems (such as realistic design problems
as opposed to mathematical "toy" problems) it is best to
choose "conservative" move limits.

Choose 1 or 2 or 3 or 4 or 5 for move limits, IMOVE=         1
Do you want default (RATIO=10) for initial move limit jump?= y
Do you want the default perturbation (dx/x = 0.05)?= y
Do you want to have dx/x modified by GENOPT?= n
Do you want to reset total iterations to zero (Type H)?= n
Choose IAUTOF= 1 or 2 or 3 or 4 or 5 or 6 to change X(i)=         1

DESCRIPTION OF FILES USED AND GENERATED IN THIS RUN:

plate1.NAM = This file contains only the name of the case.
plate1.CBL = Labelled common blocks for analysis.
           (This is an unformatted sequential file.)
plate1.OPT = This file contains the input data for MAINSETUP
            as well as OPTIMIZE. The batch command OPTIMIZE
            can be given over and over again without having
            to return to MAINSETUP because plate1.OPT exists.
URPROMPT.DAT= Prompt file for interactive input.

For further information about files used and generated
during operation of this program, give the command
```

```
 HELPG FILES.

 Menu of commands: OPTIMIZE, SUPEROPT

 IN ORDER TO AVOID FALSE CONVERGENCE OF THE DESIGN, BE SURE TO
 RUN  "OPTIMIZE"  MANY TIMES DURING AN OPTIMIZATION AND/OR USE
 THE "GLOBAL" OPTIMIZING SCRIPT, "SUPEROPT".
 **** NOTE: It is almost always best to set the number of ****
 **** iterations per execution of "OPTIMIZE" equal to 5   ****
 **** in response to the following prompt in "MAINSETUP": ****
 **** "How many design iterations in this run (3 to 25)?" ****
 **** Hence, the *.OPT file should almost always have the ****
 **** following line in it:                               ****
 *** "5   $ How many design iterations in this run (3 to 25)?"
--------------- end of MAINSETUP interactive session -----------


(The input data for MAINSETUP are saved on the file, plate.OPT,
 as follows:)

-------- the plate.OPT file (input for MAINSETUP) -------
      n         $ Do you want a tutorial session and tutorial output?
         0      $ Choose an analysis you DON'T want (1, 2,..), IBEHAV
         0      $ Choose an analysis you DON'T want (1, 2,..), IBEHAV
         0      $ Choose an analysis you DON'T want (1, 2,..), IBEHAV
         0      $ NPRINT= output index (0=GOOD, 1=ok, 2=debug, 3=too much)
         1      $ Choose type of analysis (1=opt., 2=fixed, 3=sensit.) ITYPE
         5      $ How many design iterations in this run (3 to 25)?
      n         $ Take "shortcuts" for perturbed designs (Y or N)?
         2      $ Choose 1 or 2 or 3 or 4 or 5 for IDESIGN
         1      $ Choose 1 or 2 or 3 or 4 or 5 for move limits, IMOVE
      y         $ Do you want default (RATIO=10) for initial move limit jump?
      y         $ Do you want the default perturbation (dx/x = 0.05)?
      n         $ Do you want to have dx/x modified by GENOPT?
      n         $ Do you want to reset total iterations to zero (Type H)?
         1      $ Choose IAUTOF= 1 or 2 or 3 or 4 or 5 or 6 to change X(i)
------------- end of the plate1.OPT file -------------------------


---------------------------------------------------------
```

PART 13 THE END USER RUNS "OPTIMIZE" FOR THE SPECIFIC CASE, "plate1",
IN THE "OPTIMIZATION MODE" (ITYPE = 1)

```
(Next, perform the optimization. In this simple "toy" case,
 we decide not to use SUPEROPT, but simply to execute OPTIMIZE
 three times in succession, as follows:)
```

bush-> optimize

```
Enter specific case name: plate1
B (background) or F (foreground): f

Running GENOPT: optimize, case: plate1

Executing optimize
```

```
Normal termination: optimize
Executing store
Normal termination: store

Job finished.
Inspect the output files plate1.OPM plate1.OPP

Menu: chooseplot, change, mainsetup, optimize, helpg
```

bush-> optimize

```
Enter specific case name: plate1
B (background) or F (foreground): f

Running GENOPT: optimize, case: plate1

Executing optimize
Normal termination: optimize
Executing store
Normal termination: store

Job finished.
Inspect the output files plate1.OPM plate1.OPP

Menu: chooseplot, change, mainsetup, optimize, helpg
```

bush-> optimize

```
Enter specific case name: plate1
B (background) or F (foreground): f

Running GENOPT: optimize, case: plate1

Executing optimize
Normal termination: optimize
Executing store
Normal termination: store

Job finished.
Inspect the output files plate1.OPM plate1.OPP

Menu: chooseplot, change, mainsetup, optimize, helpg


(The above series of three executions of OPTIMIZE has
 produced a file called plate1.OPP. Near the end of
 this file the following list appears:)

---- abridged plate1.OPP file (output from OPTIMIZE ----
---- when OPTIMIZE is executed with ITYPE = 1, that ----
---- is, OPTIMIZE is executed in the optimization   ----
---- mode.                                          ----
 ==============================================================
 ITERATION                                        NUMBER OF
  NUMBER      OBJECTIVE     THE DESIGN IS...   CRITICAL MARGINS
 --------------------------------------------------------------
 ------------------------------------------------------OPTIMIZE
```

```
    1         6.6667E-01      NOT FEASIBLE          5
    2         5.9400E-01         FEASIBLE           1
    3         4.6254E-01      NOT FEASIBLE          1
    4         5.1963E-01         FEASIBLE           2
    5         4.9661E-01     ALMOST FEASIBLE        4
    6         4.8260E-01     ALMOST FEASIBLE        5
 -----------------------------------------------------OPTIMIZE
    7         4.8260E-01     ALMOST FEASIBLE        5
    8         4.7454E-01     ALMOST FEASIBLE        4
    9         4.7617E-01         FEASIBLE           4
 -----------------------------------------------------OPTIMIZE
   10         4.7617E-01         FEASIBLE           4
 ==================================================================================
  VALUES OF DESIGN VARIABLES CORRESPONDING TO BEST FEASIBLE DESIGN
 VAR.   CURRENT
 NO.     VALUE            DEFINITION
  1    9.523E-02  thickness of the plate: THICK
  2    1.000E+01  Length of the plate: LENGTH
  3    5.000E+00  Width of the plate: WIDTH
 MARGINS CORRESPONDING TO THE DESIGN (F.S.= FACTOR OF SAFETY)
 MAR.   CURRENT
 NO.     VALUE            DEFINITION
  1    1.000E+01  1-1.00*V(2)+100.0*V(3)**-1 -1.
  2   -2.384E-05  1.- (1-1.00*V(2)+50.0*V(3)**-1)
  3    5.000E+00  1.- (1-1.00*V(2)+1.00*V(3))
  4    6.150E-01  1-(STRESS(1 )/MAXSTR(1 )) X FSTRES(1 ); F.S.=  1.10
  5    4.090E-02  (BUCKLE(1 )/MINBUC(1 )) / FBUCKL(1 )-1; F.S.=  1.20
  6    1.000E+10  (FREQ(1 )/MINCPS(1 )) / FSFREQ(1 )-1; F.S.=  1.00
  7    1.000E+00  1-(W(1 )/AW(1 )) X FW(1 ); F.S.=  1.00
  8   -3.028E-04  1-(STRESS(2 )/MAXSTR(2 )) X FSTRES(2 ); F.S.=  1.10
  9    1.453E-01  (BUCKLE(2 )/MINBUC(2 )) / FBUCKL(2 )-1; F.S.=  1.20
 10    1.000E+10  (FREQ(2 )/MINCPS(2 )) / FSFREQ(2 )-1; F.S.=  1.00
 11    1.000E+00  1-(W(2 )/AW(2 )) X FW(2 ); F.S.=  1.00
 12    4.028E-01  1-(STRESS(3 )/MAXSTR(3 )) X FSTRES(3 ); F.S.=  1.00
 13    1.000E+10  (BUCKLE(3 )/MINBUC(3 )) / FBUCKL(3 )-1; F.S.=  1.00
 14    5.619E-02  (FREQ(3 )/MINCPS(3 )) / FSFREQ(3 )-1; F.S.=  1.00
 15    3.250E-02  1-(W(3 )/AW(3 )) X FW(3 ); F.S.=  1.00

 ***************************************************
 ******************** DESIGN OBJECTIVE *******************
 *************                            ***********
  CORRESPONDING VALUE OF THE OBJECTIVE FUNCTION:
 VAR.   CURRENT
 NO.     VALUE            DEFINITION
  1    4.762E-01  Weight of the plate: WEIGHT

 *************                         ************
 ******************** DESIGN OBJECTIVE *******************
 *****************************************************
----------- end of abridged plate1.OPP file ------------------

---------------------------------------------------------
```

PART 14 THE END USER RUNS "CHOOSEPLOT" FOR THE SPECIFIC CASE, "plate1"

(Next, execute CHOOSEPLOT to choose what to plot
versus design iterations.)

bush-> chooseplot

```
  ENTER THE SPECIFIC CASE NAME: plate1


 **************** CHOOSEPLOT  ******************
 The purpose of CHOOSEPLOT is to permit you to choose:

    1. Which design parameters (Role 1 variables) are to be
       plotted v. design iterations;

    2. Which design margins are to be plotted v. design
       iterations.

 The results of the interactive session are saved in a file
 called plate1.CPL, in which plate1 is your name for the case.
 You may find this file useful for future runs of CHOOSEPLOT
 in which you want to avoid answering many questions
 interactively. CHOOSEPLOT also generates the four files:
 plate1.OPL, plate1.PL3, plate1.PL4, and plate1.PL5 , which are
 described briefly at the end of this run.
 If you choose the tutorial option, plate1.OPL contains a
 complete list of the interactive session, including prompting
 questions, all "help" paragraphs, your responses to the
 prompting questions, and evolving lists of which parameters
 are to be plotted as they are chosen by you.  In addition to
 the parameters chosen here for plotting, CHOOSEPLOT auto-
 matically sets up a file of the objective for plotting.

 Are you correcting, adding to, or using an existing file?=n
 n
 Do you want a tutorial session and tutorial output?= n
 Any design variables to be plotted v. iterations (Y or N)?= y
     LIST FROM WHICH A VARIABLE TO BE PLOTTED MUST NOW BE CHOSEN
 VAR.   CURRENT
 NO.     VALUE            DEFINITION
  1    9.523E-02  thickness of the plate: THICK
  2    1.000E+01  Length of the plate: LENGTH
  3    5.000E+00  Width of the plate: WIDTH
 Choose a variable to be plotted v. iterations (1,2,3,..)=         2
 Any more design variables to be plotted (Y or N) ?= y
       DESIGN VARIABLES CHOSEN SOFAR FOR PLOTTING v. ITERATIONS
 VAR.   CURRENT
 NO.     VALUE            DEFINITION
  2    1.000E+01  Length of the plate: LENGTH
     LIST FROM WHICH A VARIABLE TO BE PLOTTED MUST NOW BE CHOSEN
 VAR.   CURRENT
 NO.     VALUE            DEFINITION
  1    9.523E-02  thickness of the plate: THICK
  3    5.000E+00  Width of the plate: WIDTH
 Choose a variable to be plotted v. iterations (1,2,3,..)=         3
 Any more design variables to be plotted (Y or N) ?= n
 Any design margins to be plotted v. iterations (Y or N)?= y
     LIST FROM WHICH A MARGIN TO BE PLOTTED MUST NOW BE CHOSEN
 VAR.   CURRENT
```

```
NO.     VALUE            DEFINITION
 1    1.000E+01  1-1.00*V(2)+100.0*V(3)**-1 -1.
 2   -2.384E-05  1.- (1-1.00*V(2)+50.0*V(3)**-1)
 3    5.000E+00  1.- (1-1.00*V(2)+1.00*V(3))
 4    6.150E-01  1-(STRESS(1 )/MAXSTR(1 )) X FSTRES(1 ); F.S.=  1.10
 5    4.090E-02  (BUCKLE(1 )/MINBUC(1 )) / FBUCKL(1 )-1; F.S.=  1.20
 7    1.000E+00  1-(W(1 )/AW(1 )) X FW(1 ); F.S.=  1.00
 8   -3.028E-04  1-(STRESS(2 )/MAXSTR(2 )) X FSTRES(2 ); F.S.=  1.10
 9    1.453E-01  (BUCKLE(2 )/MINBUC(2 )) / FBUCKL(2 )-1; F.S.=  1.20
11    1.000E+00  1-(W(2 )/AW(2 )) X FW(2 ); F.S.=  1.00
12    4.028E-01  1-(STRESS(3 )/MAXSTR(3 )) X FSTRES(3 ); F.S.=  1.00
14    5.619E-02  (FREQ(3 )/MINCPS(3 )) / FSFREQ(3 )-1; F.S.=  1.00
15    3.250E-02  1-(W(3 )/AW(3 )) X FW(3 ); F.S.=  1.00
Choose a margin to be plotted v. iterations (1,2,3,..)=         1
Any more margins to be plotted (Y or N) ?= y
    DESIGN MARGINS CHOSEN SO FAR FOR PLOTTING v. ITERATIONS
MAR.   CURRENT
NO.     VALUE            DEFINITION
 1    1.000E+01  1-1.00*V(2)+100.0*V(3)**-1 -1.
    LIST FROM WHICH A MARGIN TO BE PLOTTED MUST NOW BE CHOSEN
VAR.   CURRENT
NO.     VALUE            DEFINITION
 2   -2.384E-05  1.- (1-1.00*V(2)+50.0*V(3)**-1)
 3    5.000E+00  1.- (1-1.00*V(2)+1.00*V(3))
 4    6.150E-01  1-(STRESS(1 )/MAXSTR(1 )) X FSTRES(1 ); F.S.=  1.10
 5    4.090E-02  (BUCKLE(1 )/MINBUC(1 )) / FBUCKL(1 )-1; F.S.=  1.20
 7    1.000E+00  1-(W(1 )/AW(1 )) X FW(1 ); F.S.=  1.00
 8   -3.028E-04  1-(STRESS(2 )/MAXSTR(2 )) X FSTRES(2 ); F.S.=  1.10
 9    1.453E-01  (BUCKLE(2 )/MINBUC(2 )) / FBUCKL(2 )-1; F.S.=  1.20
11    1.000E+00  1-(W(2 )/AW(2 )) X FW(2 ); F.S.=  1.00
12    4.028E-01  1-(STRESS(3 )/MAXSTR(3 )) X FSTRES(3 ); F.S.=  1.00
14    5.619E-02  (FREQ(3 )/MINCPS(3 )) / FSFREQ(3 )-1; F.S.=  1.00
15    3.250E-02  1-(W(3 )/AW(3 )) X FW(3 ); F.S.=  1.00
Choose a margin to be plotted v. iterations (1,2,3,..)=         2
Any more margins to be plotted (Y or N) ?= y
    DESIGN MARGINS CHOSEN SO FAR FOR PLOTTING v. ITERATIONS
MAR.   CURRENT
NO.     VALUE            DEFINITION
 1    1.000E+01  1-1.00*V(2)+100.0*V(3)**-1 -1.
 2   -2.384E-05  1.- (1-1.00*V(2)+50.0*V(3)**-1)
    LIST FROM WHICH A MARGIN TO BE PLOTTED MUST NOW BE CHOSEN
VAR.   CURRENT
NO.     VALUE            DEFINITION
 3    5.000E+00  1.- (1-1.00*V(2)+1.00*V(3))
 4    6.150E-01  1-(STRESS(1 )/MAXSTR(1 )) X FSTRES(1 ); F.S.=  1.10
 5    4.090E-02  (BUCKLE(1 )/MINBUC(1 )) / FBUCKL(1 )-1; F.S.=  1.20
 7    1.000E+00  1-(W(1 )/AW(1 )) X FW(1 ); F.S.=  1.00
 8   -3.028E-04  1-(STRESS(2 )/MAXSTR(2 )) X FSTRES(2 ); F.S.=  1.10
 9    1.453E-01  (BUCKLE(2 )/MINBUC(2 )) / FBUCKL(2 )-1; F.S.=  1.20
11    1.000E+00  1-(W(2 )/AW(2 )) X FW(2 ); F.S.=  1.00
12    4.028E-01  1-(STRESS(3 )/MAXSTR(3 )) X FSTRES(3 ); F.S.=  1.00
14    5.619E-02  (FREQ(3 )/MINCPS(3 )) / FSFREQ(3 )-1; F.S.=  1.00
15    3.250E-02  1-(W(3 )/AW(3 )) X FW(3 ); F.S.=  1.00
Choose a margin to be plotted v. iterations (1,2,3,..)=         3
Any more margins to be plotted (Y or N) ?= y
    DESIGN MARGINS CHOSEN SO FAR FOR PLOTTING v. ITERATIONS
MAR.   CURRENT
NO.     VALUE            DEFINITION
```

```
 1    1.000E+01  1-1.00*V(2)+100.0*V(3)**-1 -1.
 2   -2.384E-05  1.- (1-1.00*V(2)+50.0*V(3)**-1)
 3    5.000E+00  1.- (1-1.00*V(2)+1.00*V(3))
    LIST FROM WHICH A MARGIN TO BE PLOTTED MUST NOW BE CHOSEN
VAR.   CURRENT
NO.      VALUE            DEFINITION
 4    6.150E-01  1-(STRESS(1 )/MAXSTR(1 )) X FSTRES(1 ); F.S.=  1.10
 5    4.090E-02  (BUCKLE(1 )/MINBUC(1 )) / FBUCKL(1 )-1; F.S.=  1.20
 7    1.000E+00  1-(W(1 )/AW(1 )) X FW(1 ); F.S.=  1.00
 8   -3.028E-04  1-(STRESS(2 )/MAXSTR(2 )) X FSTRES(2 ); F.S.=  1.10
 9    1.453E-01  (BUCKLE(2 )/MINBUC(2 )) / FBUCKL(2 )-1; F.S.=  1.20
11    1.000E+00  1-(W(2 )/AW(2 )) X FW(2 ); F.S.=  1.00
12    4.028E-01  1-(STRESS(3 )/MAXSTR(3 )) X FSTRES(3 ); F.S.=  1.00
14    5.619E-02  (FREQ(3 )/MINCPS(3 )) / FSFREQ(3 )-1; F.S.=  1.00
15    3.250E-02  1-(W(3 )/AW(3 )) X FW(3 ); F.S.=  1.00
Choose a margin to be plotted v. iterations (1,2,3,..)=         4
Any more margins to be plotted (Y or N) ?= y
    DESIGN MARGINS CHOSEN SO FAR FOR PLOTTING v. ITERATIONS
MAR.   CURRENT
NO.      VALUE            DEFINITION
 1    1.000E+01  1-1.00*V(2)+100.0*V(3)**-1 -1.
 2   -2.384E-05  1.- (1-1.00*V(2)+50.0*V(3)**-1)
 3    5.000E+00  1.- (1-1.00*V(2)+1.00*V(3))
 4    6.150E-01  1-(STRESS(1 )/MAXSTR(1 )) X FSTRES(1 ); F.S.=  1.10
    LIST FROM WHICH A MARGIN TO BE PLOTTED MUST NOW BE CHOSEN
VAR.   CURRENT
NO.      VALUE            DEFINITION
 5    4.090E-02  (BUCKLE(1 )/MINBUC(1 )) / FBUCKL(1 )-1; F.S.=  1.20
 7    1.000E+00  1-(W(1 )/AW(1 )) X FW(1 ); F.S.=  1.00
 8   -3.028E-04  1-(STRESS(2 )/MAXSTR(2 )) X FSTRES(2 ); F.S.=  1.10
 9    1.453E-01  (BUCKLE(2 )/MINBUC(2 )) / FBUCKL(2 )-1; F.S.=  1.20
11    1.000E+00  1-(W(2 )/AW(2 )) X FW(2 ); F.S.=  1.00
12    4.028E-01  1-(STRESS(3 )/MAXSTR(3 )) X FSTRES(3 ); F.S.=  1.00
14    5.619E-02  (FREQ(3 )/MINCPS(3 )) / FSFREQ(3 )-1; F.S.=  1.00
15    3.250E-02  1-(W(3 )/AW(3 )) X FW(3 ); F.S.=  1.00
Choose a margin to be plotted v. iterations (1,2,3,..)=         5
Any more margins to be plotted (Y or N) ?= y
    DESIGN MARGINS CHOSEN SO FAR FOR PLOTTING v. ITERATIONS
MAR.   CURRENT
NO.      VALUE            DEFINITION
 1    1.000E+01  1-1.00*V(2)+100.0*V(3)**-1 -1.
 2   -2.384E-05  1.- (1-1.00*V(2)+50.0*V(3)**-1)
 3    5.000E+00  1.- (1-1.00*V(2)+1.00*V(3))
 4    6.150E-01  1-(STRESS(1 )/MAXSTR(1 )) X FSTRES(1 ); F.S.=  1.10
 5    4.090E-02  (BUCKLE(1 )/MINBUC(1 )) / FBUCKL(1 )-1; F.S.=  1.20
    LIST FROM WHICH A MARGIN TO BE PLOTTED MUST NOW BE CHOSEN
VAR.   CURRENT
NO.      VALUE            DEFINITION
 7    1.000E+00  1-(W(1 )/AW(1 )) X FW(1 ); F.S.=  1.00
 8   -3.028E-04  1-(STRESS(2 )/MAXSTR(2 )) X FSTRES(2 ); F.S.=  1.10
 9    1.453E-01  (BUCKLE(2 )/MINBUC(2 )) / FBUCKL(2 )-1; F.S.=  1.20
11    1.000E+00  1-(W(2 )/AW(2 )) X FW(2 ); F.S.=  1.00
12    4.028E-01  1-(STRESS(3 )/MAXSTR(3 )) X FSTRES(3 ); F.S.=  1.00
14    5.619E-02  (FREQ(3 )/MINCPS(3 )) / FSFREQ(3 )-1; F.S.=  1.00
15    3.250E-02  1-(W(3 )/AW(3 )) X FW(3 ); F.S.=  1.00
Choose a margin to be plotted v. iterations (1,2,3,..)=         7
Any more margins to be plotted (Y or N) ?= y
    DESIGN MARGINS CHOSEN SO FAR FOR PLOTTING v. ITERATIONS
```

```
MAR.   CURRENT
NO.     VALUE            DEFINITION
 1    1.000E+01  1-1.00*V(2)+100.0*V(3)**-1 -1.
 2   -2.384E-05  1.- (1-1.00*V(2)+50.0*V(3)**-1)
 3    5.000E+00  1.- (1-1.00*V(2)+1.00*V(3))
 4    6.150E-01  1-(STRESS(1 )/MAXSTR(1 )) X FSTRES(1 ); F.S.=  1.10
 5    4.090E-02  (BUCKLE(1 )/MINBUC(1 )) / FBUCKL(1 )-1; F.S.=  1.20
 7    1.000E+00  1-(W(1 )/AW(1 )) X FW(1 ); F.S.=  1.00
    LIST FROM WHICH A MARGIN TO BE PLOTTED MUST NOW BE CHOSEN
VAR.   CURRENT
NO.     VALUE            DEFINITION
 8   -3.028E-04  1-(STRESS(2 )/MAXSTR(2 )) X FSTRES(2 ); F.S.=  1.10
 9    1.453E-01  (BUCKLE(2 )/MINBUC(2 )) / FBUCKL(2 )-1; F.S.=  1.20
11    1.000E+00  1-(W(2 )/AW(2 )) X FW(2 ); F.S.=  1.00
12    4.028E-01  1-(STRESS(3 )/MAXSTR(3 )) X FSTRES(3 ); F.S.=  1.00
14    5.619E-02  (FREQ(3 )/MINCPS(3 )) / FSFREQ(3 )-1; F.S.=  1.00
15    3.250E-02  1-(W(3 )/AW(3 )) X FW(3 ); F.S.=  1.00
Choose a margin to be plotted v. iterations (1,2,3,..)=         8
Any more margins to be plotted (Y or N) ?= y
    DESIGN MARGINS CHOSEN SO FAR FOR PLOTTING v. ITERATIONS
MAR.   CURRENT
NO.     VALUE            DEFINITION
 1    1.000E+01  1-1.00*V(2)+100.0*V(3)**-1 -1.
 2   -2.384E-05  1.- (1-1.00*V(2)+50.0*V(3)**-1)
 3    5.000E+00  1.- (1-1.00*V(2)+1.00*V(3))
 4    6.150E-01  1-(STRESS(1 )/MAXSTR(1 )) X FSTRES(1 ); F.S.=  1.10
 5    4.090E-02  (BUCKLE(1 )/MINBUC(1 )) / FBUCKL(1 )-1; F.S.=  1.20
 7    1.000E+00  1-(W(1 )/AW(1 )) X FW(1 ); F.S.=  1.00
 8   -3.028E-04  1-(STRESS(2 )/MAXSTR(2 )) X FSTRES(2 ); F.S.=  1.10
    LIST FROM WHICH A MARGIN TO BE PLOTTED MUST NOW BE CHOSEN
VAR.   CURRENT
NO.     VALUE            DEFINITION
 9    1.453E-01  (BUCKLE(2 )/MINBUC(2 )) / FBUCKL(2 )-1; F.S.=  1.20
11    1.000E+00  1-(W(2 )/AW(2 )) X FW(2 ); F.S.=  1.00
12    4.028E-01  1-(STRESS(3 )/MAXSTR(3 )) X FSTRES(3 ); F.S.=  1.00
14    5.619E-02  (FREQ(3 )/MINCPS(3 )) / FSFREQ(3 )-1; F.S.=  1.00
15    3.250E-02  1-(W(3 )/AW(3 )) X FW(3 ); F.S.=  1.00
Choose a margin to be plotted v. iterations (1,2,3,..)=         9
Any more margins to be plotted (Y or N) ?= y
    DESIGN MARGINS CHOSEN SO FAR FOR PLOTTING v. ITERATIONS
MAR.   CURRENT
NO.     VALUE            DEFINITION
 1    1.000E+01  1-1.00*V(2)+100.0*V(3)**-1 -1.
 2   -2.384E-05  1.- (1-1.00*V(2)+50.0*V(3)**-1)
 3    5.000E+00  1.- (1-1.00*V(2)+1.00*V(3))
 4    6.150E-01  1-(STRESS(1 )/MAXSTR(1 )) X FSTRES(1 ); F.S.=  1.10
 5    4.090E-02  (BUCKLE(1 )/MINBUC(1 )) / FBUCKL(1 )-1; F.S.=  1.20
 7    1.000E+00  1-(W(1 )/AW(1 )) X FW(1 ); F.S.=  1.00
 8   -3.028E-04  1-(STRESS(2 )/MAXSTR(2 )) X FSTRES(2 ); F.S.=  1.10
 9    1.453E-01  (BUCKLE(2 )/MINBUC(2 )) / FBUCKL(2 )-1; F.S.=  1.20
    LIST FROM WHICH A MARGIN TO BE PLOTTED MUST NOW BE CHOSEN
VAR.   CURRENT
NO.     VALUE            DEFINITION
11    1.000E+00  1-(W(2 )/AW(2 )) X FW(2 ); F.S.=  1.00
12    4.028E-01  1-(STRESS(3 )/MAXSTR(3 )) X FSTRES(3 ); F.S.=  1.00
14    5.619E-02  (FREQ(3 )/MINCPS(3 )) / FSFREQ(3 )-1; F.S.=  1.00
15    3.250E-02  1-(W(3 )/AW(3 )) X FW(3 ); F.S.=  1.00
Choose a margin to be plotted v. iterations (1,2,3,..)=        11
```

```
Any more margins to be plotted (Y or N) ?= y
    DESIGN MARGINS CHOSEN SO FAR FOR PLOTTING v. ITERATIONS
MAR.   CURRENT
NO.     VALUE            DEFINITION
 1    1.000E+01  1-1.00*V(2)+100.0*V(3)**-1 -1.
 2   -2.384E-05  1.- (1-1.00*V(2)+50.0*V(3)**-1)
 3    5.000E+00  1.- (1-1.00*V(2)+1.00*V(3))
 4    6.150E-01  1-(STRESS(1 )/MAXSTR(1 )) X FSTRES(1 ); F.S.=  1.10
 5    4.090E-02  (BUCKLE(1 )/MINBUC(1 )) / FBUCKL(1 )-1; F.S.=  1.20
 7    1.000E+00  1-(W(1 )/AW(1 )) X FW(1 ); F.S.=  1.00
 8   -3.028E-04  1-(STRESS(2 )/MAXSTR(2 )) X FSTRES(2 ); F.S.=  1.10
 9    1.453E-01  (BUCKLE(2 )/MINBUC(2 )) / FBUCKL(2 )-1; F.S.=  1.20
11    1.000E+00  1-(W(2 )/AW(2 )) X FW(2 ); F.S.=  1.00
    LIST FROM WHICH A MARGIN TO BE PLOTTED MUST NOW BE CHOSEN
VAR.   CURRENT
NO.     VALUE            DEFINITION
12    4.028E-01  1-(STRESS(3 )/MAXSTR(3 )) X FSTRES(3 ); F.S.=  1.00
14    5.619E-02  (FREQ(3 )/MINCPS(3 )) / FSFREQ(3 )-1; F.S.=  1.00
15    3.250E-02  1-(W(3 )/AW(3 )) X FW(3 ); F.S.=  1.00
Choose a margin to be plotted v. iterations (1,2,3,..)=       12
Any more margins to be plotted (Y or N) ?= y
    DESIGN MARGINS CHOSEN SO FAR FOR PLOTTING v. ITERATIONS
MAR.   CURRENT
NO.     VALUE            DEFINITION
 1    1.000E+01  1-1.00*V(2)+100.0*V(3)**-1 -1.
 2   -2.384E-05  1.- (1-1.00*V(2)+50.0*V(3)**-1)
 3    5.000E+00  1.- (1-1.00*V(2)+1.00*V(3))
 4    6.150E-01  1-(STRESS(1 )/MAXSTR(1 )) X FSTRES(1 ); F.S.=  1.10
 5    4.090E-02  (BUCKLE(1 )/MINBUC(1 )) / FBUCKL(1 )-1; F.S.=  1.20
 7    1.000E+00  1-(W(1 )/AW(1 )) X FW(1 ); F.S.=  1.00
 8   -3.028E-04  1-(STRESS(2 )/MAXSTR(2 )) X FSTRES(2 ); F.S.=  1.10
 9    1.453E-01  (BUCKLE(2 )/MINBUC(2 )) / FBUCKL(2 )-1; F.S.=  1.20
11    1.000E+00  1-(W(2 )/AW(2 )) X FW(2 ); F.S.=  1.00
12    4.028E-01  1-(STRESS(3 )/MAXSTR(3 )) X FSTRES(3 ); F.S.=  1.00
    LIST FROM WHICH A MARGIN TO BE PLOTTED MUST NOW BE CHOSEN
VAR.   CURRENT
NO.     VALUE            DEFINITION
14    5.619E-02  (FREQ(3 )/MINCPS(3 )) / FSFREQ(3 )-1; F.S.=  1.00
15    3.250E-02  1-(W(3 )/AW(3 )) X FW(3 ); F.S.=  1.00
Choose a margin to be plotted v. iterations (1,2,3,..)=       14
Any more margins to be plotted (Y or N) ?= y
Choose a margin to be plotted v. iterations (1,2,3,..)=       14
Any more margins to be plotted (Y or N) ?= y
    DESIGN MARGINS CHOSEN SO FAR FOR PLOTTING v. ITERATIONS
MAR.   CURRENT
NO.     VALUE            DEFINITION
 1    1.000E+01  1-1.00*V(2)+100.0*V(3)**-1 -1.
 2   -2.384E-05  1.- (1-1.00*V(2)+50.0*V(3)**-1)
 3    5.000E+00  1.- (1-1.00*V(2)+1.00*V(3))
 4    6.150E-01  1-(STRESS(1 )/MAXSTR(1 )) X FSTRES(1 ); F.S.=  1.10
 5    4.090E-02  (BUCKLE(1 )/MINBUC(1 )) / FBUCKL(1 )-1; F.S.=  1.20
 7    1.000E+00  1-(W(1 )/AW(1 )) X FW(1 ); F.S.=  1.00
 8   -3.028E-04  1-(STRESS(2 )/MAXSTR(2 )) X FSTRES(2 ); F.S.=  1.10
 9    1.453E-01  (BUCKLE(2 )/MINBUC(2 )) / FBUCKL(2 )-1; F.S.=  1.20
11    1.000E+00  1-(W(2 )/AW(2 )) X FW(2 ); F.S.=  1.00
12    4.028E-01  1-(STRESS(3 )/MAXSTR(3 )) X FSTRES(3 ); F.S.=  1.00
14    5.619E-02  (FREQ(3 )/MINCPS(3 )) / FSFREQ(3 )-1; F.S.=  1.00
    LIST FROM WHICH A MARGIN TO BE PLOTTED MUST NOW BE CHOSEN
```

```
 VAR.   CURRENT
 NO.     VALUE            DEFINITION
 15    3.250E-02  1-(W(3 )/AW(3 )) X FW(3 ); F.S.=  1.00
 Choose a margin to be plotted v. iterations (1,2,3,..)=        15
 Any more margins to be plotted (Y or N) ?= n
 Give maximum value (positive) to be included in plot frame.=         1


 DESCRIPTION OF FILES GENERATED BY THIS CASE:

 plate1.CPL = Summary of interactive session you have just
          completed. This file can be edited and used for
          future runs of CHOOSEPLOT.
 plate1.CBL = Contains the plate1 data base.
 plate1.OPL = Output from CHOOSEPLOT. Please list this file and
          inspect it and the plate1.CPL file carefully before
          proceeding.
 plate1.PL3 = File for margin plots via DIPLOT
 plate1.PL4 = File for design parameter plots via DIPLOT
 plate1.PL5 = File for objective plot via DIPLOT

 For further information about files generated during operation
 of GENOPT give the command HELPG FILES.

 NEXT, GIVE THE COMMAND:  DIPLOT   .
-------------- end of interactive CHOOSEPLOT session ----------------


(The interactive input data for CHOOSEPLOT has been saved on the file,
 plate1.CPL. A list of plate1.CPL follows.)

----------- the plate1.CPL file (input for CHOOSEPLOT --------------
      n          $ Do you want a tutorial session and tutorial output?
      y          $ Any design variables to be plotted v. iterations (Y or N)?
         2       $ Choose a variable to be plotted v. iterations (1,2,3,..)
      y          $ Any more design variables to be plotted (Y or N) ?
         3       $ Choose a variable to be plotted v. iterations (1,2,3,..)
      n          $ Any more design variables to be plotted (Y or N) ?
      y          $ Any design margins to be plotted v. iterations (Y or N)?
         1       $ Choose a margin to be plotted v. iterations (1,2,3,..)
      y          $ Any more margins to be plotted (Y or N) ?
         2       $ Choose a margin to be plotted v. iterations (1,2,3,..)
      y          $ Any more margins to be plotted (Y or N) ?
         3       $ Choose a margin to be plotted v. iterations (1,2,3,..)
      y          $ Any more margins to be plotted (Y or N) ?
         4       $ Choose a margin to be plotted v. iterations (1,2,3,..)
      y          $ Any more margins to be plotted (Y or N) ?
         5       $ Choose a margin to be plotted v. iterations (1,2,3,..)
      y          $ Any more margins to be plotted (Y or N) ?
         7       $ Choose a margin to be plotted v. iterations (1,2,3,..)
      y          $ Any more margins to be plotted (Y or N) ?
         8       $ Choose a margin to be plotted v. iterations (1,2,3,..)
      y          $ Any more margins to be plotted (Y or N) ?
         9       $ Choose a margin to be plotted v. iterations (1,2,3,..)
      y          $ Any more margins to be plotted (Y or N) ?
        11       $ Choose a margin to be plotted v. iterations (1,2,3,..)
      y          $ Any more margins to be plotted (Y or N) ?
        12       $ Choose a margin to be plotted v. iterations (1,2,3,..)
```

```
      y         $ Any more margins to be plotted (Y or N) ?
        14      $ Choose a margin to be plotted v. iterations (1,2,3,..)
      y         $ Any more margins to be plotted (Y or N) ?
        15      $ Choose a margin to be plotted v. iterations (1,2,3,..)
      n         $ Any more margins to be plotted (Y or N) ?
         1      $ Give maximum value (positive) to be included in plot frame.
----------- end of the plate1.CPL file ----------------------------
```

PART 15 THE END USER RUNS "DIPLOT" FOR THE SPECIFIC CASE, "plate1"

(Next, execute DIPLOT in order to obtain the files,
plate1.i.ps, i = 3, 4, 5, ... 10, ... In this particular
case we will have three files, as follows:
plate1.3.ps = design margins versus design iterations
plate1.4.ps = decision variables versus design iterations
plate1.5.ps = objective versus design iterations.)

bush-> diplot

```
Enter the specific case name: plate1
Print the plot file on the printer called: <lp> (y or n)? n
plate1.PL6: No such file or directory.
plate1.PL7: No such file or directory.
plate1.PL8: No such file or directory.
plate1.PL9: No such file or directory.
plate1.PL10: No such file or directory.
The PostScript files, plate1.3.ps through plate1.10.ps, contain the
graphics for your plot.  They can be printed on any PostScript
printer or viewed on the console with a PostScript previewing
software program.
```

(Next, view the three files, plate1.3.ps, plate1.4.ps, plate1.5.ps,
on your screen. We use the utility called "ghost view", abbreviated
"gv".)

```
bush-> gv plate1.3.ps (generate plate1.3.png file by taking screen snapshot)
bush-> gv plate1.4.ps (generate plate1.4.png file by taking screen snapshot)
bush-> gv plate1.5.ps (generate plate1.5.png file by taking screen snapshot)
```

The three plots, plate1.3.png, plate1.4.png, and plate1.5.png follow on the next three pages.

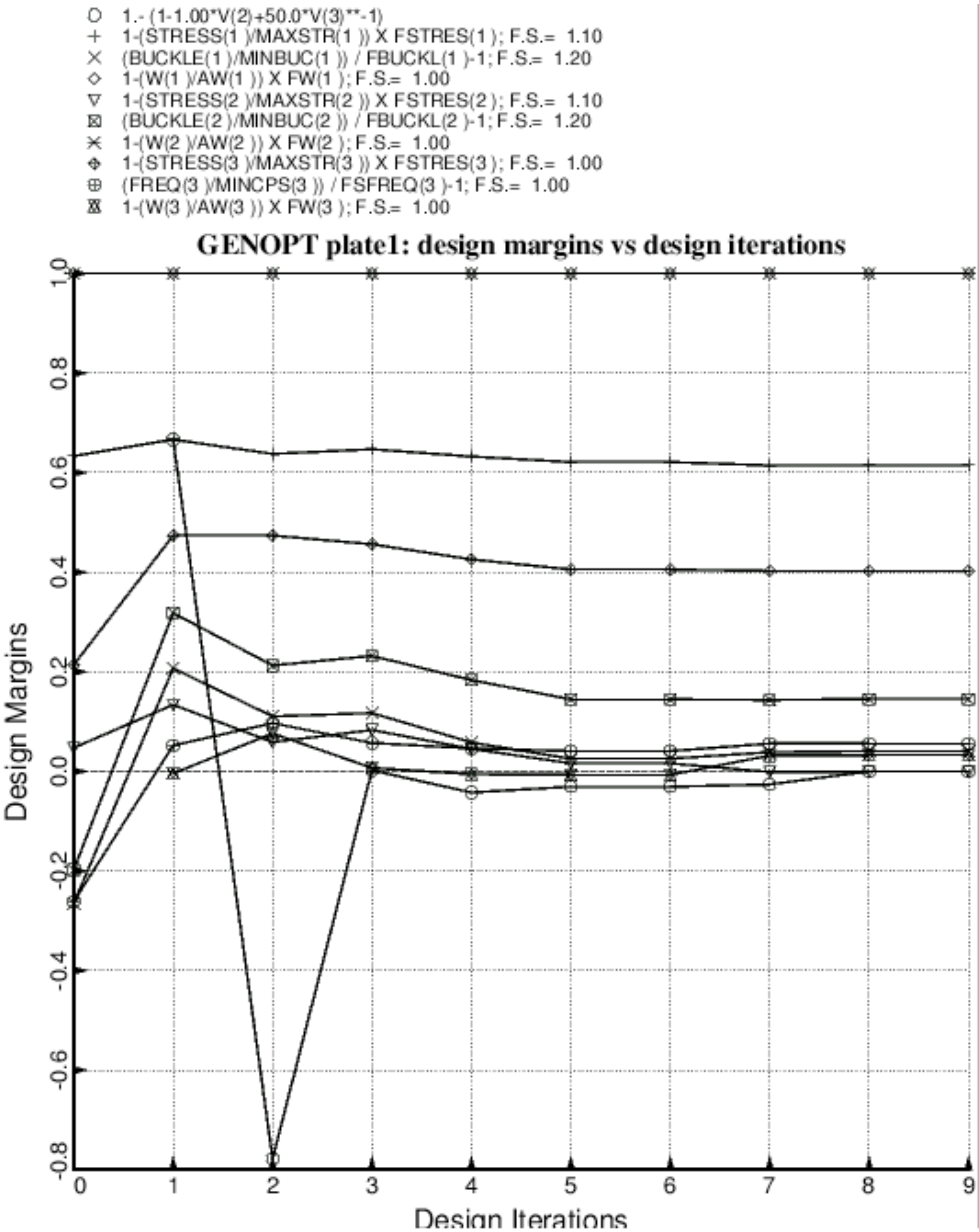

plate1.3.ps = design margins versus design iterations corresponding to the three successive executions of OPTIMIZE as listed in PART 13 and the choice of margins in CHOOSEPLOT as listed in PART 14

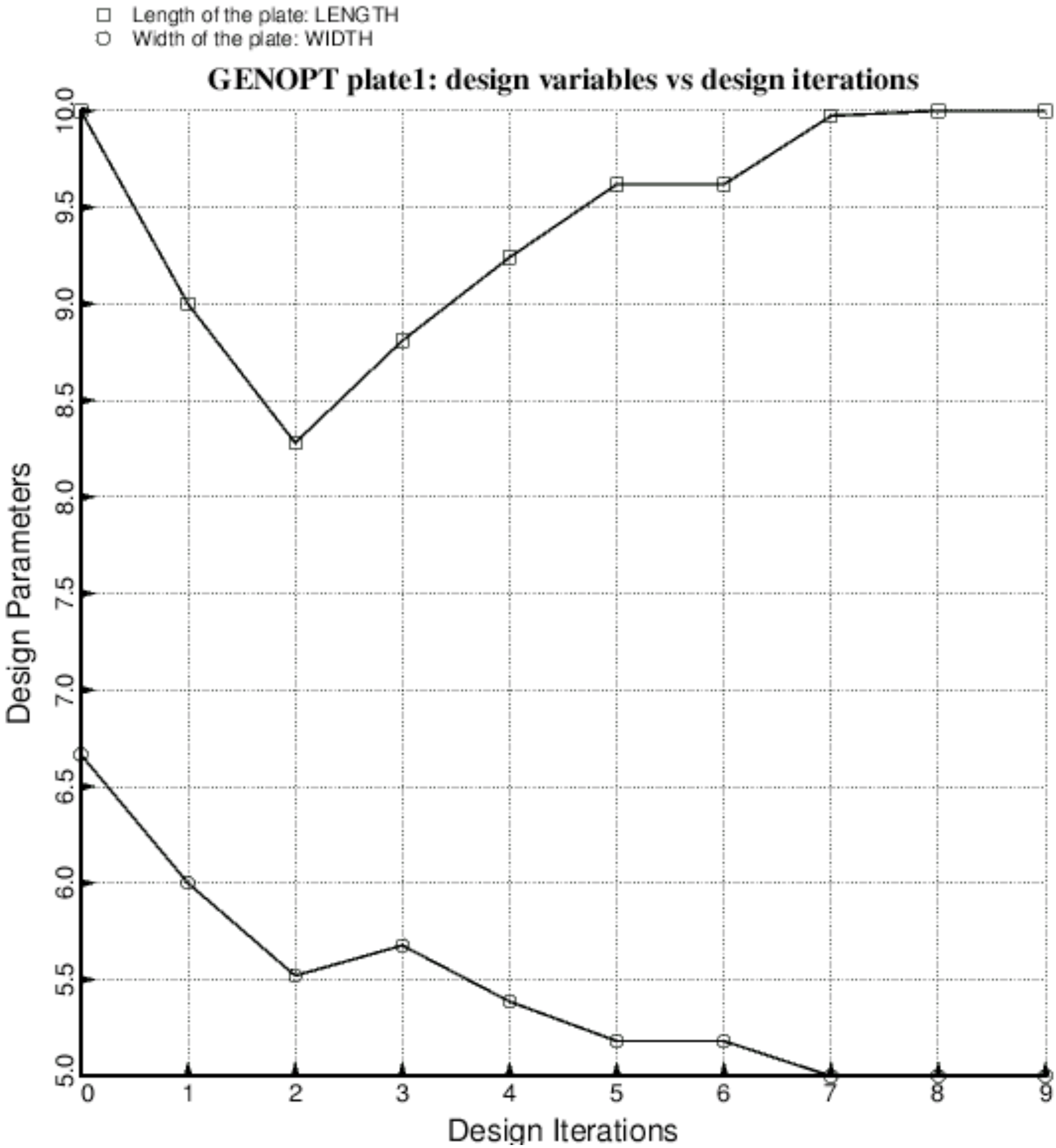

plate1.4.ps = decision variables versus design iterations corresponding to the three successive executions of OPTIMIZE as listed in PART 13 and the choice of decision variables in CHOOSEPLOT as listed in PART 14

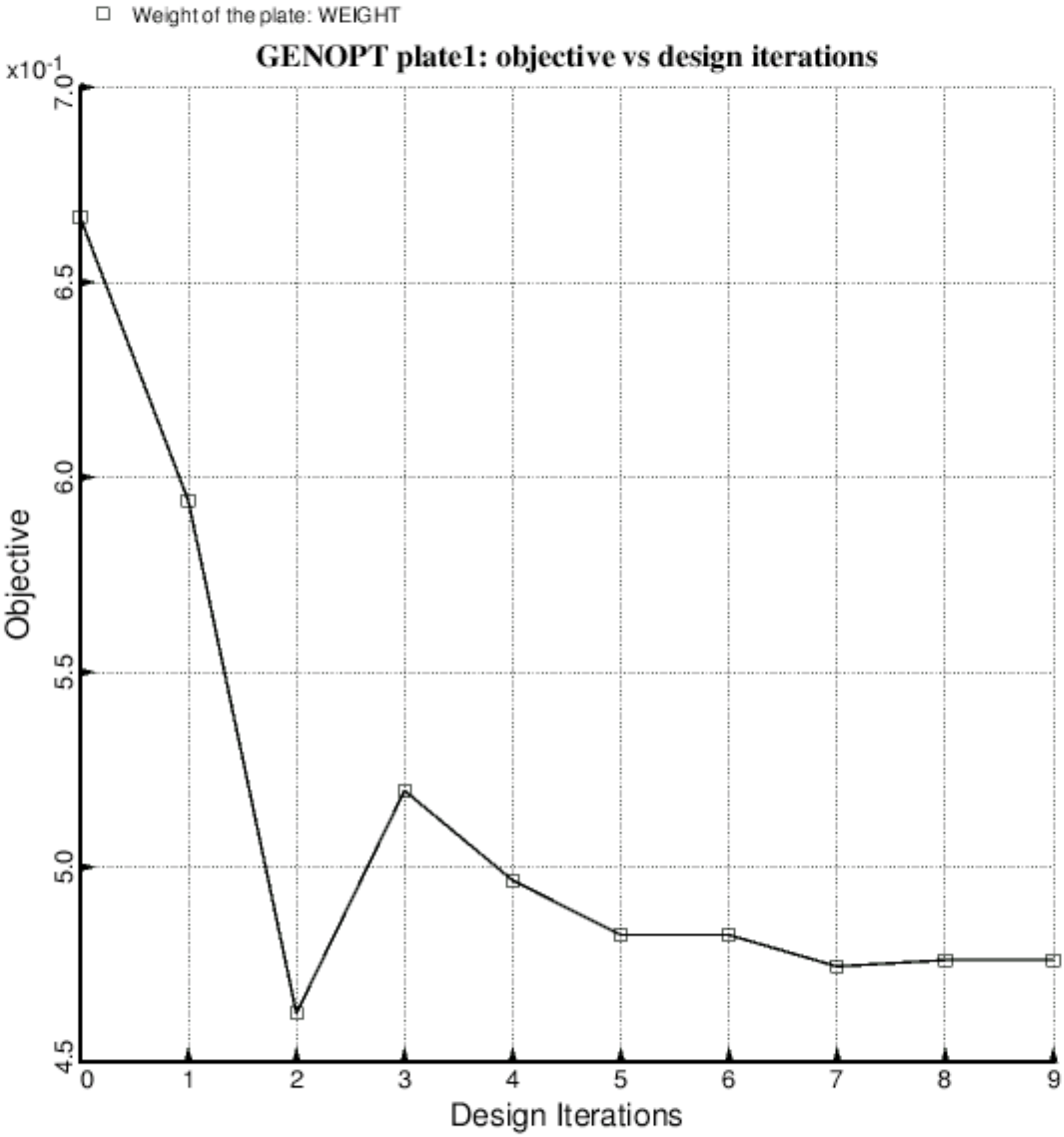

plate1.5.ps = design objective versus design iterations corresponding to the three successive executions of OPTIMIZE as listed in PART 13

PART 16 THE END USER GENERATES RESULTS FOR THE OPTIMIZED DESIGN FOR THE SPECIFIC CASE, "plate1" (ITYPE = 2 IN plate1.OPT)

(Next, we wish to obtain results for the optimized design, which just happens, in this case, to correspond to the last design found, that is, the design at Iteration No. 9. In order to do this, we edit the input file, plate1.OPT, for MAINSETUP, then execute MAINSETUP followed by OPTIMIZE. The new input file for MAINSETUP follows:)

```
-------------- the new plate1.OPT file (input for MAINSETUP) ---------
      n         $ Do you want a tutorial session and tutorial output?
         0      $ Choose an analysis you DON'T want (1, 2,..), IBEHAV
         0      $ Choose an analysis you DON'T want (1, 2,..), IBEHAV
         0      $ Choose an analysis you DON'T want (1, 2,..), IBEHAV
         2      $ NPRINT= output index (0=GOOD, 1=ok, 2=debug, 3=too much)
         2      $ Choose type of analysis (1=opt., 2=fixed, 3=sensit.) ITYPE
         5      $ How many design iterations in this run (3 to 25)?
      n         $ Take "shortcuts" for perturbed designs (Y or N)?
         2      $ Choose 1 or 2 or 3 or 4 or 5 for IDESIGN
         1      $ Choose 1 or 2 or 3 or 4 or 5 for move limits, IMOVE
      y         $ Do you want default (RATIO=10) for initial move limit jump?
      y         $ Do you want the default perturbation (dx/x = 0.05)?
      n         $ Do you want to have dx/x modified by GENOPT?
      n         $ Do you want to reset total iterations to zero (Type H)?
         1      $ Choose IAUTOF= 1 or 2 or 3 or 4 or 5 or 6 to change X(i)
--------- end of the new plate1.OPT file, analysis of fixed design --------
```

(Next, execute MAINSETUP followed by OPTIMIZE.)

bush-> mainsetup

```
  ENTER THE SPECIFIC CASE NAME: plate1
 Are you correcting, adding to, or using an existing file?=y
```

(The interactive MAINSETUP session zips by on your screen)

(Perform the analysis of the fixed, optimized design (ITYPE = 2))

bush-> optimize

```
Enter specific case name: plate1
B (background) or F (foreground): f

Running GENOPT: optimize, case: plate1

Executing optimize
Normal termination: optimize
Executing store
Normal termination: store

Job finished.

Menu: chooseplot, change, mainsetup, optimize, helpg
```

PART 17 THE END USER INSPECTS RESULTS FOR THE OPTIMIZED DESIGN FOR THE SPECIFIC CASE, "plate1")

(Inspect the plate1.OPM file. In this particular case the following plate1.OPM file corresponds to the optimized design. It is the design that exists at Iteration No. 9 in this particular case. However, note that most of the time the last design processed during optimization cycles is not necessarily the best design. This is especially true when the "global" optimizer, SUPEROPT, is used.)

```
---------- plate1.OPM file (ITYPE=2) ---------------
      n         $ Do you want a tutorial session and tutorial output?
         0      $ Choose an analysis you DON'T want (1, 2,..), IBEHAV
         0      $ Choose an analysis you DON'T want (1, 2,..), IBEHAV
         0      $ Choose an analysis you DON'T want (1, 2,..), IBEHAV
         2      $ NPRINT= output index (0=GOOD, 1=ok, 2=debug, 3=too much)
         2      $ Choose type of analysis (1=opt., 2=fixed, 3=sensit.) ITYPE
         5      $ How many design iterations in this run (3 to 25)?
      n         $ Take "shortcuts" for perturbed designs (Y or N)?
         2      $ Choose 1 or 2 or 3 or 4 or 5 for IDESIGN
         1      $ Choose 1 or 2 or 3 or 4 or 5 for move limits, IMOVE
      y         $ Do you want default (RATIO=10) for initial move limit jump?
      y         $ Do you want the default perturbation (dx/x = 0.05)?
      n         $ Do you want to have dx/x modified by GENOPT?
      n         $ Do you want to reset total iterations to zero (Type H)?
         1      $ Choose IAUTOF= 1 or 2 or 3 or 4 or 5 or 6 to change X(i)

 ************ END OF THE  plate1.OPT  FILE ***********
 ************ AUGUST, 2010 VERSION OF GENOPT ****************
 ******** BEGINNING OF THE  plate1.OPM  FILE *********


 ********************** MAIN PROCESSOR *********************
 The purpose of the mainprocessor, OPTIMIZE, is to perform,
 in a batch mode, the work specified by MAINSETUP for the case
 called plate1.  Results are stored in the file  plate1.OPM.
 Please inspect plate1.OPM before doing more design iterations.
 ************************************************************


 STRUCTURAL ANALYSIS FOR DESIGN ITERATION NO.   0:
0
  STRUCTURAL ANALYSIS WITH UNPERTURBED DECISION VARIABLES
 VAR. DEC. ESCAPE LINK. LINKED  LINKING   LOWER     CURRENT    UPPER       DEFINITION
  NO. VAR.  VAR.  VAR.    TO   CONSTANT   BOUND      VALUE     BOUND
   1   Y     Y     N      0    0.00E+00  3.00E-02 9.5234E-02  1.00E+00  thickness of the
plate: THICK
   2   Y     N     N      0    0.00E+00  5.00E+00 1.0000E+01  1.00E+02  Length of the
plate: LENGTH
   3   Y     N     N      0    0.00E+00  5.00E+00 5.0000E+00  1.00E+01  Width of the
plate: WIDTH
```

```
  BEHAVIOR FOR  1 ENVIRONMENT (LOAD SET)

 CONSTRAINT  BEHAVIOR              DEFINITION
   NUMBER     VALUE

  BEHAVIOR FOR LOAD SET NUMBER, ILOADX= 1
    1        10500.46      Maximum effective (von Mises) stress: STRESS(1 )
    2        1.249080      Buckling load factor: BUCKLE(1 )
    3       0.1000000E+11  Fundamental frequency of unloaded plate: FREQ(1 )
    4       0.1000000E-09  Normal deflection under uniform pressure: W(1 )

 ***** RESULTS FOR LOAD SET NO.  1  ******
 PARAMETERS WHICH DESCRIBE BEHAVIOR (e.g. stress, buckling load)

 BEH.   CURRENT
 NO.     VALUE            DEFINITION
  1    1.050E+04    Maximum effective (von Mises) stress: STRESS(1 )
  2    1.249E+00    Buckling load factor: BUCKLE(1 )
  3    1.000E+10    Fundamental frequency of unloaded plate: FREQ(1 )
  4    1.000E-10    Normal deflection under uniform pressure: W(1 )

 ******* NOTE ******* NOTE ******* NOTE ****** NOTE ******
 The phrase, "NOT APPLY", for MARGIN VALUE means that that
 particular margin value is exactly zero.
 *** END NOTE *** END NOTE *** END NOTE *** END NOTE *****


 ***** RESULTS FOR LOAD SET NO.  1  ******
 MARGINS CORRESPONDING TO CURRENT DESIGN (F.S.= FACTOR OF SAFETY)

 MARGIN CURRENT
 NO.     VALUE            DEFINITION
  1    1.000E+01  1-1.00*V(2)+100.0*V(3)**-1 -1.
  2   -2.384E-05  1.- (1-1.00*V(2)+50.0*V(3)**-1)
  3    5.000E+00  1.- (1-1.00*V(2)+1.00*V(3))
  4    6.150E-01  1-(STRESS(1 )/MAXSTR(1 )) X FSTRES(1 ); F.S.=  1.10
  5    4.090E-02  (BUCKLE(1 )/MINBUC(1 )) / FBUCKL(1 )-1; F.S.=  1.20
  6    1.000E+00  1-(W(1 )/AW(1 )) X FW(1 ); F.S.=  1.00
0
  STRUCTURAL ANALYSIS WITH UNPERTURBED DECISION VARIABLES
 VAR. DEC. ESCAPE LINK. LINKED  LINKING   LOWER     CURRENT    UPPER       DEFINITION
  NO. VAR.  VAR.  VAR.    TO   CONSTANT   BOUND      VALUE     BOUND
   1   Y     Y     N      0    0.00E+00  3.00E-02 9.5234E-02  1.00E+00  thickness of the
plate: THICK
   2   Y     N     N      0    0.00E+00  5.00E+00 1.0000E+01  1.00E+02  Length of the
plate: LENGTH
   3   Y     N     N      0    0.00E+00  5.00E+00 5.0000E+00  1.00E+01  Width of the
plate: WIDTH
  BEHAVIOR FOR  2 ENVIRONMENT (LOAD SET)

 CONSTRAINT  BEHAVIOR              DEFINITION
   NUMBER     VALUE

  BEHAVIOR FOR LOAD SET NUMBER, ILOADX= 2
    1        27280.98      Maximum effective (von Mises) stress: STRESS(2 )
    2        1.374368      Buckling load factor: BUCKLE(2 )
    3       0.1000000E+11  Fundamental frequency of unloaded plate: FREQ(2 )
    4       0.1000000E-09  Normal deflection under uniform pressure: W(2 )
```

```
 ***** RESULTS FOR LOAD SET NO.  2  ******
 PARAMETERS WHICH DESCRIBE BEHAVIOR (e.g. stress, buckling load)

 BEH.   CURRENT
 NO.     VALUE             DEFINITION
  1    2.728E+04    Maximum effective (von Mises) stress: STRESS(2 )
  2    1.374E+00    Buckling load factor: BUCKLE(2 )
  3    1.000E+10    Fundamental frequency of unloaded plate: FREQ(2 )
  4    1.000E-10    Normal deflection under uniform pressure: W(2 )

 ******* NOTE ******* NOTE ******* NOTE ****** NOTE ******
 The phrase, "NOT APPLY", for MARGIN VALUE means that that
 particular margin value is exactly zero.
 *** END NOTE *** END NOTE *** END NOTE *** END NOTE *****


 ***** RESULTS FOR LOAD SET NO.  2  ******
 MARGINS CORRESPONDING TO CURRENT DESIGN (F.S.= FACTOR OF SAFETY)

 MARGIN CURRENT
 NO.     VALUE             DEFINITION
  1   -3.028E-04  1-(STRESS(2 )/MAXSTR(2 )) X FSTRES(2 ); F.S.=  1.10
  2    1.453E-01  (BUCKLE(2 )/MINBUC(2 )) / FBUCKL(2 )-1; F.S.=  1.20
  3    1.000E+00  1-(W(2 )/AW(2 )) X FW(2 ); F.S.=  1.00
0
  STRUCTURAL ANALYSIS WITH UNPERTURBED DECISION VARIABLES
 VAR. DEC. ESCAPE LINK. LINKED  LINKING   LOWER     CURRENT    UPPER       DEFINITION
  NO. VAR.  VAR.  VAR.    TO   CONSTANT   BOUND      VALUE     BOUND
   1   Y     Y     N      0    0.00E+00  3.00E-02 9.5234E-02  1.00E+00  thickness of the
plate: THICK
   2   Y     N     N      0    0.00E+00  5.00E+00 1.0000E+01  1.00E+02  Length of the
plate: LENGTH
   3   Y     N     N      0    0.00E+00  5.00E+00 5.0000E+00  1.00E+01  Width of the
plate: WIDTH
  BEHAVIOR FOR  3 ENVIRONMENT (LOAD SET)

 CONSTRAINT  BEHAVIOR              DEFINITION
   NUMBER     VALUE

  BEHAVIOR FOR LOAD SET NUMBER, ILOADX= 3
    1        17915.93      Maximum effective (von Mises) stress: STRESS(3 )
    2       0.1000000E+11  Buckling load factor: BUCKLE(3 )
    3        137.3047      Fundamental frequency of unloaded plate: FREQ(3 )
    4       0.9674953E-01  Normal deflection under uniform pressure: W(3 )

 ***** RESULTS FOR LOAD SET NO.  3  ******
 PARAMETERS WHICH DESCRIBE BEHAVIOR (e.g. stress, buckling load)

 BEH.   CURRENT
 NO.     VALUE             DEFINITION
  1    1.792E+04    Maximum effective (von Mises) stress: STRESS(3 )
  2    1.000E+10    Buckling load factor: BUCKLE(3 )
  3    1.373E+02    Fundamental frequency of unloaded plate: FREQ(3 )
  4    9.675E-02    Normal deflection under uniform pressure: W(3 )

 ******* NOTE ******* NOTE ******* NOTE ****** NOTE ******
 The phrase, "NOT APPLY", for MARGIN VALUE means that that
```

```
particular margin value is exactly zero.
*** END NOTE *** END NOTE *** END NOTE *** END NOTE *****


***** RESULTS FOR LOAD SET NO.  3  ******
MARGINS CORRESPONDING TO CURRENT DESIGN (F.S.= FACTOR OF SAFETY)

MARGIN CURRENT
NO.     VALUE            DEFINITION
 1    4.028E-01  1-(STRESS(3 )/MAXSTR(3 )) X FSTRES(3 ); F.S.=  1.00
 2    5.619E-02  (FREQ(3 )/MINCPS(3 )) / FSFREQ(3 )-1; F.S.=  1.00
 3    3.250E-02  1-(W(3 )/AW(3 )) X FW(3 ); F.S.=  1.00

***************************************************
******************** DESIGN OBJECTIVE *******************
*************                             ***********
   CURRENT VALUE OF THE OBJECTIVE FUNCTION:

VAR.   CURRENT
NO.     VALUE            DEFINITION
 1    4.762E-01  Weight of the plate: WEIGHT

*************                          ************
******************** DESIGN OBJECTIVE *******************
*****************************************************
***********  ALL 3 LOAD CASES PROCESSED ***********
*****************************************************


PARAMETERS WHICH ARE ALWAYS FIXED.  NONE CAN BE DECISION VARIAB.

VAR.   CURRENT
NO.     VALUE            DEFINITION
 1    1.000E+07  Young's modulus of the plate material: E
 2    3.000E-01  Poisson's ratio of the plate material: NU
 3    1.000E-01  Weight density (e.g. lb/in**3) of the plate material: RHO
 4    2.000E-01  (plate length, LENGTH)/(plate width, WIDTH): AOBAXL(1 )
 5    3.000E-01  (plate length, LENGTH)/(plate width, WIDTH): AOBAXL(2 )
 6    4.000E-01  (plate length, LENGTH)/(plate width, WIDTH): AOBAXL(3 )
 7    6.000E-01  (plate length, LENGTH)/(plate width, WIDTH): AOBAXL(4 )
 8    8.000E-01  (plate length, LENGTH)/(plate width, WIDTH): AOBAXL(5 )
 9    1.000E+00  (plate length, LENGTH)/(plate width, WIDTH): AOBAXL(6 )
10    1.200E+00  (plate length, LENGTH)/(plate width, WIDTH): AOBAXL(7 )
11    1.400E+00  (plate length, LENGTH)/(plate width, WIDTH): AOBAXL(8 )
12    1.600E+00  (plate length, LENGTH)/(plate width, WIDTH): AOBAXL(9 )
13    1.800E+00  (plate length, LENGTH)/(plate width, WIDTH): AOBAXL(10)
14    2.000E+00  (plate length, LENGTH)/(plate width, WIDTH): AOBAXL(11)
15    2.200E+00  (plate length, LENGTH)/(plate width, WIDTH): AOBAXL(12)
16    2.400E+00  (plate length, LENGTH)/(plate width, WIDTH): AOBAXL(13)
17    2.700E+00  (plate length, LENGTH)/(plate width, WIDTH): AOBAXL(14)
18    3.000E+00  (plate length, LENGTH)/(plate width, WIDTH): AOBAXL(15)
19    3.000E+01  (plate length, LENGTH)/(plate width, WIDTH): AOBAXL(16)
20    2.220E+01  buckling coefficient: uniform axial compression: KAXL(1 )
21    1.090E+01  buckling coefficient: uniform axial compression: KAXL(2 )
22    6.920E+00  buckling coefficient: uniform axial compression: KAXL(3 )
23    4.230E+00  buckling coefficient: uniform axial compression: KAXL(4 )
24    3.450E+00  buckling coefficient: uniform axial compression: KAXL(5 )
25    3.290E+00  buckling coefficient: uniform axial compression: KAXL(6 )
```

```
26    3.400E+00  buckling coefficient: uniform axial compression: KAXL(7 )
27    3.680E+00  buckling coefficient: uniform axial compression: KAXL(8 )
28    3.450E+00  buckling coefficient: uniform axial compression: KAXL(9 )
29    3.320E+00  buckling coefficient: uniform axial compression: KAXL(10)
30    3.290E+00  buckling coefficient: uniform axial compression: KAXL(11)
31    3.320E+00  buckling coefficient: uniform axial compression: KAXL(12)
32    3.400E+00  buckling coefficient: uniform axial compression: KAXL(13)
33    3.320E+00  buckling coefficient: uniform axial compression: KAXL(14)
34    3.290E+00  buckling coefficient: uniform axial compression: KAXL(15)
35    3.290E+00  buckling coefficient: uniform axial compression: KAXL(16)
36    6.667E-01  (plate length, LENGTH)/(plate width, WIDTH): AOBSHR(1 )
37    7.143E-01  (plate length, LENGTH)/(plate width, WIDTH): AOBSHR(2 )
38    8.333E-01  (plate length, LENGTH)/(plate width, WIDTH): AOBSHR(3 )
39    1.000E+00  (plate length, LENGTH)/(plate width, WIDTH): AOBSHR(4 )
40    1.200E+00  (plate length, LENGTH)/(plate width, WIDTH): AOBSHR(5 )
41    1.400E+00  (plate length, LENGTH)/(plate width, WIDTH): AOBSHR(6 )
42    1.500E+00  (plate length, LENGTH)/(plate width, WIDTH): AOBSHR(7 )
43    1.600E+00  (plate length, LENGTH)/(plate width, WIDTH): AOBSHR(8 )
44    1.800E+00  (plate length, LENGTH)/(plate width, WIDTH): AOBSHR(9 )
45    2.000E+00  (plate length, LENGTH)/(plate width, WIDTH): AOBSHR(10)
46    2.500E+00  (plate length, LENGTH)/(plate width, WIDTH): AOBSHR(11)
47    3.000E+00  (plate length, LENGTH)/(plate width, WIDTH): AOBSHR(12)
48    3.000E+01  (plate length, LENGTH)/(plate width, WIDTH): AOBSHR(13)
49    5.840E+00  buckling coefficient: uniform in-plane shear: KSHR(1 )
50    6.000E+00  buckling coefficient: uniform in-plane shear: KSHR(2 )
51    6.580E+00  buckling coefficient: uniform in-plane shear: KSHR(3 )
52    7.750E+00  buckling coefficient: uniform in-plane shear: KSHR(4 )
53    6.580E+00  buckling coefficient: uniform in-plane shear: KSHR(5 )
54    6.000E+00  buckling coefficient: uniform in-plane shear: KSHR(6 )
55    5.840E+00  buckling coefficient: uniform in-plane shear: KSHR(7 )
56    5.760E+00  buckling coefficient: uniform in-plane shear: KSHR(8 )
57    5.590E+00  buckling coefficient: uniform in-plane shear: KSHR(9 )
58    5.430E+00  buckling coefficient: uniform in-plane shear: KSHR(10)
59    5.180E+00  buckling coefficient: uniform in-plane shear: KSHR(11)
60    5.020E+00  buckling coefficient: uniform in-plane shear: KSHR(12)
61    4.400E+00  buckling coefficient: uniform in-plane shear: KSHR(13)
PARAMETERS WHICH ARE ENVIRONMENTAL FACTORS (e.g. loads, temps.)

VAR.   CURRENT
NO.     VALUE           DEFINITION
 1   -1.000E+03  Axial tension/width of the plate (lb/in): Nx(1 )
 2    0.000E+00  Axial tension/width of the plate (lb/in): Nx(2 )
 3    0.000E+00  Axial tension/width of the plate (lb/in): Nx(3 )
 4    0.000E+00  Transverse tension/length of the plate (lb/in): Ny(1 )
 5    0.000E+00  Transverse tension/length of the plate (lb/in): Ny(2 )
 6    0.000E+00  Transverse tension/length of the plate (lb/in): Ny(3 )
 7    0.000E+00  In-plane shear/length applied to the plate: Nxy(1 )
 8    1.500E+03  In-plane shear/length applied to the plate: Nxy(2 )
 9    0.000E+00  In-plane shear/length applied to the plate: Nxy(3 )
10    0.000E+00  Uniform normal pressure on the plate: PRESS(1 )
11    0.000E+00  Uniform normal pressure on the plate: PRESS(2 )
12    1.200E+01  Uniform normal pressure on the plate: PRESS(3 )
PARAMETERS WHICH ARE CLASSIFIED AS ALLOWABLES (e.g. max. stress)

VAR.   CURRENT
NO.     VALUE           DEFINITION
 1    3.000E+04  Maximum effective (von Mises) stress allowed: MAXSTR(1 )
 2    3.000E+04  Maximum effective (von Mises) stress allowed: MAXSTR(2 )
```

```
 3    3.000E+04  Maximum effective (von Mises) stress allowed: MAXSTR(3 )
 4    1.000E+00  Minimum allowable buckling load factor (use 1.0): MINBUC(1 )
 5    1.000E+00  Minimum allowable buckling load factor (use 1.0): MINBUC(2 )
 6    1.000E+00  Minimum allowable buckling load factor (use 1.0): MINBUC(3 )
 7    0.000E+00  Minimum allowable value for the fundamental frequency: MINCPS(1 )
 8    0.000E+00  Minimum allowable value for the fundamental frequency: MINCPS(2 )
 9    1.300E+02  Minimum allowable value for the fundamental frequency: MINCPS(3 )
10    0.000E+00  Maximum allowable normal deflection under pressure: AW(1 )
11    0.000E+00  Maximum allowable normal deflection under pressure: AW(2 )
12    1.000E-01  Maximum allowable normal deflection under pressure: AW(3 )
PARAMETERS WHICH ARE FACTORS OF SAFETY

VAR.   CURRENT
NO.     VALUE            DEFINITION
 1    1.100E+00  Factor of safety for effective stress: FSTRES(1 )
 2    1.100E+00  Factor of safety for effective stress: FSTRES(2 )
 3    1.000E+00  Factor of safety for effective stress: FSTRES(3 )
 4    1.200E+00  Factor of safety for buckling load factor: FBUCKL(1 )
 5    1.200E+00  Factor of safety for buckling load factor: FBUCKL(2 )
 6    1.000E+00  Factor of safety for buckling load factor: FBUCKL(3 )
 7    1.000E+00  Factor of safety for FREQ: FSFREQ(1 )
 8    1.000E+00  Factor of safety for FREQ: FSFREQ(2 )
 9    1.000E+00  Factor of safety for FREQ: FSFREQ(3 )
10    0.000E+00  Factor of safety for max deflection under pressure: FW(1 )
11    0.000E+00  Factor of safety for max deflection under pressure: FW(2 )
12    1.000E+00  Factor of safety for max deflection under pressure: FW(3 )

   3 INEQUALITY CONSTRAINTS WHICH MUST BE SATISFIED

 1 <1-1.00*V(2)+100.0*V(3)**-1
 1 >1-1.00*V(2)+50.0*V(3)**-1
 1 >1-1.00*V(2)+1.00*V(3)


DESCRIPTION OF FILES USED AND GENERATED IN THIS RUN:

plate1.NAM = This file contains only the name of the case.
plate1.OPM = Output data. Please list this file and inspect
           carefully before proceeding.
plate1.OPP = Output file containing evolution of design and
           margins since the beginning of optimization cycles.
plate1.CBL = Labelled common blocks for analysis.
           (This is an unformatted sequential file.)
plate1.OPT = This file contains the input data for MAINSETUP
            as well as OPTIMIZE. The batch command OPTIMIZE
            can be given over and over again without having
            to return to MAINSETUP because plate1.OPT exists.
URPROMPT.DAT= Prompt file for interactive input.

For further information about files used and generated
during operation of GENOPT, give the command HELPG FILES.

Menu of commands: CHOOSEPLOT, OPTIMIZE, MAINSETUP, CHANGE,
                  DECIDE, SUPEROPT

IN ORDER TO AVOID FALSE CONVERGENCE OF THE DESIGN, BE SURE TO
RUN  "OPTIMIZE"  MANY TIMES DURING AN OPTIMIZATION AND/OR USE
THE "GLOBAL" OPTIMIZING SCRIPT, "SUPEROPT".
```

```
 **** NOTE: It is almost always best to set the number of ****
 **** iterations per execution of "OPTIMIZE" equal to 5   ****
 **** in response to the following prompt in "MAINSETUP": ****
 **** "How many design iterations in this run (3 to 25)?" ****
 **** Hence, the *.OPT file should almost always have the ****
 **** following line in it:                               ****
 **** "5   $ How many design iterations in this run (3 to 25)?"
 ******************** END OF plate1.OPM FILE ****************
------------- end of the plate1.OPM file --------------------
```

PART 18 THE END USER EXECUTES "CHANGE" IN ORDER TO ARCHIVE THE OPTIMUM DESIGN FOR THE SPECIFIC CASE, "plate1"

(Next, archive the optimum design by using the GENOPT processor called CHANGE.)

bush-> change

```
  ENTER THE SPECIFIC CASE NAME: plate1


 ******************   CHANGE   *******************

 You use CHANGE to change parameters without having to go back
 to BEGIN. The parameters you can change are segregated into
 five groups:

    1. parameters elegible to be decision variables
    2. parameters not elegible to be decision variables
    3. parameters that characterize the environment (loads)
    4. allowables (for example, max. strain)
    5. factors of safety.

 Your interactive input is saved on a file called NAME.CHG, in
 which NAME is the same name you used for BEGIN, DECIDE, etc.
 A summary of the output from CHANGE is stored in NAME.OPC.

 ************************************************************

 Are you correcting, adding to, or using an existing file?=n
 n
 Do you want a tutorial session and tutorial output?=n
 n

 This program permits you to change certain quantities without
 starting over from the beginning (without having to use BEGIN).

 Parameters that you can change are segregated into three sets:

    1. parameters that are "elegible" to be decision variables;
    2. parameters that are always considered to be fixed
       during design iterations: they are not elegible
       to be decision variables;
    3. parameters that describe the environment (loads, temps)
```

```
   4. parameters that are allowables, such as max. stress.
   5. parameters that are factors of safety.

You will next be asked if you want to change any parameters in
set no. 1, and if so, which; then you will be asked the same
questions relative to parameter sets  2, 3, 4, and 5.

Do you want to change any values in Parameter Set No. 1?=y
y
PARAMETERS WHICH CAN BE CHANGED.   CHOOSE ONE OF THE FOLLOWING
VAR.   CURRENT
NO.     VALUE            DEFINITION
 1    9.523E-02  thickness of the plate: THICK
 2    1.000E+01  Length of the plate: LENGTH
 3    5.000E+00  Width of the plate: WIDTH
Number of parameter to change (1, 2, 3, . .)=1
        1
New value of the parameter=0.09523
 0.9523000E-01
Want to change any other parameters in this set?=y
y
PARAMETERS WHICH CAN BE CHANGED.   CHOOSE ONE OF THE FOLLOWING
VAR.   CURRENT
NO.     VALUE            DEFINITION
 1    9.523E-02  thickness of the plate: THICK
 2    1.000E+01  Length of the plate: LENGTH
 3    5.000E+00  Width of the plate: WIDTH
Number of parameter to change (1, 2, 3, . .)=2
        2
New value of the parameter=10.
  10.00000
Want to change any other parameters in this set?=y
y
PARAMETERS WHICH CAN BE CHANGED.   CHOOSE ONE OF THE FOLLOWING
VAR.   CURRENT
NO.     VALUE            DEFINITION
 1    9.523E-02  thickness of the plate: THICK
 2    1.000E+01  Length of the plate: LENGTH
 3    5.000E+00  Width of the plate: WIDTH
Number of parameter to change (1, 2, 3, . .)=3
        3
New value of the parameter=5.0
  5.000000
Want to change any other parameters in this set?=n
n
Do you want to change values of any "fixed" parameters?=n
n
Do you want to change any loads?=n
n
Do you want to change values of allowables?=n
n
Do you want to change any factors of safety?=n
n


DESCRIPTION OF FILES GENERATED BY THIS CASE:

plate1.CHG = Summary of interactive session you have just
```

```
          completed. This file can be edited and used for
          future runs of CHANGE.

 plate1.CBL = Contains part of plate1 data base.

 plate1.OPC = Output from CHANGE. Please list this file and
          inspect it and the plate1.CHG file carefully before
          proceeding.

 For further information about files generated during operation
 of GENOPT give the command HELPG FILES.

 Next, give any of the commands:
 DECIDE, MAINSETUP, OPTIMIZE, or SUPEROPT.
------------ end of the interactive CHANGE session ------------


(The input data for CHANGE has been saved in the file,
 plate1.CHG, a list of which follows. This file can be
 used anytime in the future to re-establish the optimum
 design.)

---------- the plate1.CHG file (input for CHANGE) -----------
      n         $ Do you want a tutorial session and tutorial output?
      y         $ Do you want to change any values in Parameter Set No. 1?
         1      $ Number of parameter to change (1, 2, 3, . .)
  0.9523000E-01 $ New value of the parameter
      y         $ Want to change any other parameters in this set?
         2      $ Number of parameter to change (1, 2, 3, . .)
   10.00000     $ New value of the parameter
      y         $ Want to change any other parameters in this set?
         3      $ Number of parameter to change (1, 2, 3, . .)
   5.000000     $ New value of the parameter
      n         $ Want to change any other parameters in this set?
      n         $ Do you want to change values of any "fixed" parameters?
      n         $ Do you want to change any loads?
      n         $ Do you want to change values of allowables?
      n         $ Do you want to change any factors of safety?
------------- end of the plate1.CHG file -----------------


---------------------------------------------------------
```

PART 19 THE FILES, plate1.*, THAT NOW EXIST IN THE WORKING DIRECTORY, .../genoptcase:

```
(The files with the specific name, "plate1" now existing in
 the working directory, .../genoptcase, are as follows:)

    432 Mar  2 16:09 plate1      (input for the last interactive
                                  processor that was executed)
     76 Mar  2 16:09 plate1.010  (Are you correcting.....?)
  24311 Mar  2 16:07 plate1.3.ps (Postscript file: margins)
  14874 Mar  2 16:07 plate1.4.ps (Postscript file: dec. var)
  12705 Mar  2 16:07 plate1.5.ps (Postscript file: objective)
   7873 Mar  1 13:37 plate1.BEG  (input for BEGIN)
```

```
 162156 Mar  2 16:09 plate1.CBL  (plate1 binary database)
    920 Mar  1 16:31 plate1.CHG  (input for CHANGE)
   2184 Mar  1 13:37 plate1.CPL  (input for CHOOSEPLOT)
   3349 Mar  1 13:51 plate1.DEC  (input for DECIDE)
     30 Mar  2 16:02 plate1.NAM  (specific case name)
   9781 Mar  2 16:02 plate1.OPB  (output from BEGIN)
   9570 Mar  2 16:08 plate1.OPC  (output from CHANGE)
   3719 Mar  2 16:03 plate1.OPD  (output from DECIDE)
   4631 Mar  2 16:06 plate1.OPL  (output from CHOOSEPLOT)
  18139 Mar  2 16:09 plate1.OPM  (output from OPTIMIZE)
   9968 Mar  2 16:03 plate1.OPP  (output from OPTIMIZE)
   1056 Mar  2 16:09 plate1.OPT  (input for MAINSETUP)
   1056 Mar  2 16:03 plate1.PL1  (binary file)
   1056 Mar  2 16:03 plate1.PL2  (binary file)
   3550 Mar  2 16:06 plate1.PL3  (plotps input: margins)
    838 Mar  2 16:06 plate1.PL4  (plotps input: dec.var)
    549 Mar  2 16:06 plate1.PL5  (plotps input: objective)
```

PART 20 THE END USER EXECUTES "CLEANSPEC" IN ORDER TO CLEAN UP THE FILES WITH THE SPECIFIC CASE NAME, "plate1"

(Next, clean up the files with the SPECIFIC case name, "plate1":)

bush-> cleanspec

Enter specific case name: plate1

(The remaining "plate1" files after execution of "cleanspec" are as follows:)

```
     76 Mar  2 16:09 plate1.010  (Are you correcting.....?)
  24311 Mar  2 16:07 plate1.3.ps (Postscript  margins)
  14874 Mar  2 16:07 plate1.4.ps (Postscript  dec. var)
  12705 Mar  2 16:07 plate1.5.ps (Postscript  objective)
   7873 Mar  1 13:37 plate1.BEG  (input for BEGIN)
    920 Mar  1 16:31 plate1.CHG  (input for CHANGE)
   2184 Mar  1 13:37 plate1.CPL  (input for CHOOSEPLOT)
   3349 Mar  1 13:51 plate1.DEC  (input for DECIDE)
   1056 Mar  2 16:09 plate1.OPT  (input for MAINSETUP)
```

PART 21 THE END USER EXECUTES "CLEANGEN" IN ORDER TO CLEAN UP THE FILES WITH THE GENERIC CASE NAME, "plate"

(Next, clean up the files with the GENERIC case name, "plate":)

bush-> cleangen

```
Enter generic case name: plate

Next, you have the chance to save the FORTRAN source codes:

         behavior.new      and/or      struct.new

You may have done a lot of work modifying these libraries.
Save them by copying them into other files.

Do you want to copy "behavior.new"? (Y or N) n

Do you want to copy "struct.new"? (Y or N) n


(The remaining "plate" files after execution of "cleangen"
 are as follows:)

  26592 Feb 29  2008 behavior.plate ("fleshed out" behavior)
  13908 Mar  2 10:08 plate.INP      (input for GENTEXT)
=============================================================
```