Table 40 How to run **STAGS** to obtain results such as those displayed in the next table and in Figs. 6 - 10 and Figs. 16 and 17 and Figs. 20 – 23, etc.

==========

The following steps are taken in order to run STAGS cases involving the user-written SUBROUTINE WALL or SUBROUTINE USRFAB. A user-written SUBROUTINE WALL or **a user-written SUBROUTINE USRFAB is required whenever the thickness of the shell varies over the surface of the shell.**

I have written three versions of SUBROUTINE WALL, one for elastic material (wall.elastic.src), one for elastic-plastic material (wall.plastic.src), and one for the "soccerball" STAGS model (wall.soccerball.plastic.src. See Tables a20 – a22 and a32 in the Appendix for lists of SUBROUTINE WALL.

I have written two versions of SUBROUTINE USRFAB, one for a 360-degree STAGS model (usrfab.plastic.src) and one for the "soccerball" STAGS model (usrfab.soccerball.plastic.src). See Tables a34 – a36 and a39 in the Appendix for lists of SUBROUTINE USRFAB and SUBROUTINE LAME (Table a39). (SUBROUTINE LAME must be used in connection with "soccerball" STAGS models.)

To set up STAGS for running a case, we do the following:

1. Run GENOPT (OPTIMIZE) **with a fixed design (ITYPE=2 in the *.OPT file)** in order to generate the file called "*.STAGS" (eqellipse.STAGS in the specific case called "eqellipse". In general: <casename>.STAGS).

2. Go to the directory from which you want to run STAGS.

3a.**For the use of STAGS with SUBROUTINE WALL:**
   Copy SUBROUTINE WALL into the directory where you want to run STAGS:
   Example: cp /home/progs/genopt/case/torisph/wall.elastic.src wall.F
   NOTE: You must ALWAYS use the name, "wall.F".

3b.**For the 360-degree elastic-plastic STAGS model:**
   Copy SUBROUTINE USRFAB into the directory where you want to run STAGS:
   Example: cp /home/progs/genopt/case/torisph/usrfab.plastic.src usrfab.F
   NOTE: You must ALWAYS use the name, "usrfab.F".

3c.**For the 180-degree elastic-plastic "soccerball" STAGS model:**
   Copy SUBROUTINE USRFAB into the directory where you want to run STAGS:
   Example:
 cp /home/progs/genopt/case/torisph/usrfab.soccerball.plastic.src usrfab.F
   NOTE: You must ALWAYS use the name, "usrfab.F".
   Copy SUBROUTINE LAME into the directory where you want to run STAGS:
   Example: cp /home/progs/genopt/case/torisph/lame.src lame.F

NOTE: You must ALWAYS use the name, “lame.F”.

4. Copy <casename>.STAGS in an analogous manner:
   Example: cp /home/progs/genoptcase/eqellipse.STAGS WALLTHICK.STAGS
   NOTE: You must ALWAYS use the name, "WALLTHICK.STAGS". WALLTHICK.STAGS
   is “called” by both SUBROUTINE WALL and SUBROUTINE USRFAB .

5. We must "**source**" the STAGS code now. At the writer’s facility:
   source /home/weiler/stags5/prc/initialize (for feynman computer)
   source /home/stag5/prc/initialize (for teller computer)

6. We must "**make**" both s1 (STAGS preprocessor) and s2 (STAGS mainprocessor) in the directory where we want to run STAGS cases. The appropriate commands are:

   makeuser s1 (generates an executable element called us1)
   makeuser s2 (generates an executable element called us2)

7. We must "**make**" the utilities STAPL and XYTRANS (STAGS postprocessors) in the directory where we want to run STAGS cases. The appropriate commands are:

   makeuser stapl (generates an executable element called ustapl)
   makeuser xytrans (generates an executable element called uxytrans)

   ********** **NOTE** **********
   When we use STAPL and XYTRANS with the new us1 and us2, we must type "ustapl <casename>" and "uxytrans" instead of "stapl" and "xytrans".
   ****

8. **Generate the *.inp and *.bin input files for STAGS.** These files are called here <casename>.inp1 and <casename>.bin1 (for linear buckling) and <casename>.inp2 and <casename>.bin2 (for nonlinear collapse) here. See the discussion below for some tips on generating a valid *.inp file. Examples are given of *.inp & *.bin in this table.

9. cp /home/progs/genopt/case/torisph/<casename>.{bin1,inp1} <casename.{bin,inp} and later, after completion of the linear buckling run or runs, cp /home/progs/genopt/case/torisph/<casename>.{bin2,inp2} <casename>.{bin,inp} for the nonlinear collapse analysis of the shell with one or more buckling modal imperfections.

10. Run STAGS via the command:

    stags <casename> -1 <path-to-us1> -2 <path-to-us2> -b

    Example: stags eqellipse -1 /home/bush/us1 -2 /home/bush/us2 -b

NOTE: In order to run STAGS you have to have two valid input files, <casename>.bin and <casename>.inp . Examples are listed below.

NOTE; Whether STAGS runs an ELASTIC model or an ELASTIC-PLASTIC model depends on SUBROUTINE WALL (or SUBROUTINE USRFAB).

There are three SUBROUTINE WALL files included in the directory, /home/progs/genopt/case/torisph :

wall.elastic.src	(listed in Table a21)
wall.plastic.src	(listed in Table a22)
wall.soccerball.plastic.src	(listed in Table a32)

There are two SUBROUTINE USRFAB files included in the directory, /home/progs/genopt/case/torisph :

For the 360-degree STAGS models:

usrfab.plastic.src	(listed in Table a35)

For the 180-degree STAGS “soccerball” models:

usrfab.soccerball.plastic.src	(listed in Table a36)

The STAGS “soccerball” models require also the presence of SUBROUTINE LAME. (listed in Table a39)

HOW TO RUN STAGS AND GET PLOTS

To run STAGS you need a <casename>.bin file and a <casename>.inp file. For example, for this case (eqellipse) the <casename>.bin file for linear bifurcation buckling analysis is called :

/home/progs/genopt/case/torisph/eqellipse.stiffened.stags.bin1

in which /home/progs/genopt/case/torisph is the directory where the "bin1" file is located.

The "bin1" file, eqellipse.stiffened.stags.bin1, contains the following STAGS input data:

```
------------------------------------------------------------
 eqellipse.bin: linear bifurcation buckling
 1, $ INDIC=1 is bifur.buckling; INDIC=3 is nonlinear BEGIN B-1
 1, $ IPOST=1 means save displacements every IPOSTth step
 0, $ ILIST =0 means normal batch-oriented output
 0, $ ICOR  =0 means projection in; 1 means not in.
 1, $ IMPTHE=index for imperfection theory.
 0, $ ICHIST=index for crack archive option
```

```
 0, $ IFLU  =0 means no fluid interaction.
 -1 $ ISOLVR= 0 means original solver; -1 new solver.END B-1 rec
 1.000E+00, $ STLD(1) = starting load factor, System A. BEGIN C-1 rec.
 0.000E+00, $ STEP(1) = load factor increment, System A
 1.000E+00, $ FACM(1) = maximum load factor, System A
 0.000E+00, $ STLD(2) = starting load factor, System B
 0.000E+00, $ STEP(2) = load factor increment, System B
 0.000E+00, $ FACM(2) = maximum load factor, System B
 0  $ ITEMP =0 means no thermal loads. END C-1 rec.
 10000, $ NSEC= number of CPU seconds before run termination
 0., $ DELEV is eigenvalue error tolerance (0=.00001)
 0  $ IPRINT=0 means print modes, iteration data, END D-2 rec.
  8, $ NEIGS= number of eigenvalues sought.  BEGIN D-3 rec.
 2.83, $ SHIFT=initial eigenvalue shift
 0.000E+00, $ EIGA =lower bound of eigenvalue range
 0.000E+00  $ EIGB =upper bound of eigenvalue range.    END D-3 rec.
-----------------------------------------------------------------------
```

The corresponding STAGS input file, <casename>.inp, is called here,

 /home/progs/genopt/case/torisph/eqellipse.stiffened.stags.inp1

For the optimized isogrid-stiffened equivalent ellipse, this file contains the following input data:

```
-----------------------------------------------------------------------
perfect isogrid-stiffened equivalent ellipsoidal head X_320
  0  0  0  0  0  0  0   $B-1 IGRAV,ICHECK,ILIST,INCBC,NRUNIT,NROTS,KDEV
  12  1  0  23  0,  $B-2 NUNITS,NUNITE,NSTFS,NINTS,NPATS,
  0  0  0  0  0  0  $B-2 NCONST,NIMPFS,INERT,NINSR,NPATX,NSTIFS
  2  0  0  0  0  0  $B-3 NTAM,NTAB,NTAW,NTAP,NTAMT,NGCP
  5 91,              $F-1 NROWS(1),NCOLS(1)
  5 91,              $F-1 NROWS(1),NCOLS(1)
  5 91,              $F-1 NROWS(1),NCOLS(1)
  5 91,              $F-1 NROWS(1),NCOLS(1)
  5 91,              $F-1 NROWS(1),NCOLS(1)
  5 91,              $F-1 NROWS(1),NCOLS(1)
  5 91,              $F-1 NROWS(1),NCOLS(1)
  5 91,              $F-1 NROWS(1),NCOLS(1)
  5 91,              $F-1 NROWS(1),NCOLS(1)
  5 91,              $F-1 NROWS(1),NCOLS(1)
  5 91,              $F-1 NROWS(1),NCOLS(1)
  5 91               $F-1 NROWS(2),NCOLS(1)
  1  3  2  1         $G-1 MUNIT,MBOUND,NUNIT,NBOUND
  2  3  3  1         $G-1 MUNIT,MBOUND,NUNIT,NBOUND
  3  3  4  1         $G-1 MUNIT,MBOUND,NUNIT,NBOUND
  4  3  5  1         $G-1 MUNIT,MBOUND,NUNIT,NBOUND
  5  3  6  1         $G-1 MUNIT,MBOUND,NUNIT,NBOUND
  6  3  7  1         $G-1 MUNIT,MBOUND,NUNIT,NBOUND
```

```
  7  3  8  1         $G-1 MUNIT,MBOUND,NUNIT,NBOUND
  8  3  9  1         $G-1 MUNIT,MBOUND,NUNIT,NBOUND
  9  3 10  1         $G-1 MUNIT,MBOUND,NUNIT,NBOUND
 10  3 11  1         $G-1 MUNIT,MBOUND,NUNIT,NBOUND
 11  3 12  1         $G-1 MUNIT,MBOUND,NUNIT,NBOUND
  1  2  1  4         $G-1 unit 1 is a closed shell
  2  2  2  4         $G-1 unit 2 is a closed shell
  3  2  3  4         $G-1 unit 2 is a closed shell
  4  2  4  4         $G-1 unit 2 is a closed shell
  5  2  5  4         $G-1 unit 2 is a closed shell
  6  2  6  4         $G-1 unit 2 is a closed shell
  7  2  7  4         $G-1 unit 2 is a closed shell
  8  2  8  4         $G-1 unit 2 is a closed shell
  9  2  9  4         $G-1 unit 2 is a closed shell
 10  2 10  4         $G-1 unit 2 is a closed shell
 11  2 11  4         $G-1 unit 2 is a closed shell
 12  2 12  4         $G-1 unit 2 is a closed shell
  -1   -1            $H-1 For pole, rigid links (-1's let computer do the
$                         counting for you!)
  1  7  1  1  0  0  $I-1 ITAM,NESP,IPLST,ITANST,ICREEP,IPLANE
 16.E+06  0.25  0.0  0.16  0.0  16.E+06  0. $I-2 E1,U12,G,RHO,A1,E2,A2
  .0075 120000.,    $I-3 E(i), S(i)
  .0088 138000.,    $I-3 E(i), S(i)
  .0102 148000.,    $I-3 E(i), S(i)
  .0122 156000.,    $I-3 E(i), S(i)
  .0156 164000.,    $I-3 E(i), S(i)
  .0200 165000.,    $I-3 E(i), S(i)
  .0400 166000.     $I-3 E(i), S(i)
  2  7  1  1  0  0  $I-1 ITAM,NESP,IPLST,ITANST,ICREEP,IPLANE
 496894.4 .333 0. .004969 496894.4 0. $I-2 E1,U12,G,RHO,A1,E2,A2
  .0075 3726.710,    $I-3 E(i), S(i)
  .0088 4285.710,    $I-3 E(i), S(i)
  .0102 4596.270,    $I-3 E(i), S(i)
  .0122 4844.720,    $I-3 E(i), S(i)
  .0156 5093.170,    $I-3 E(i), S(i)
  .0200 5124.220,    $I-3 E(i), S(i)
  .0400 5155.280     $I-3 E(i), S(i)
C unit 1 = the spherical cap
  7  0  0  0  0  0  $M-1 ISHELL,IGLOBE,NROWS,NCOLS,NLAYS,NFABS
  0.00 2.958103 0.0  360.0  49.5 $M-2 PHI1, PHI2, THETA1, THETA2, R
  0  0              $M-5 IWALL,IWIMP
  410               $N-1 KELT
  0  6  6  6  0     $P-1 IBLN(i), i=1,4, IBOND
 111  111           $P-2 ITRA, IROT (conditions at pole)
  1  0  0  0  0  0  0 $Q-1 NSYS,NICS,NAMS,NUSS,NHINGE,etc.
  1  1  0           $Q-2 ISYS,NN,IFLG
 -460.  5  3  0  0  0 $Q-3 P,LT,LD,LI,LJ,LAX
  0  0  0  0  0     $R-1 IPRD,IPRR,IPRE,IPRS,IPRP
```

```
C unit 2 = toroidal
  8  0  0  0  0  0  $M-1 ISHELL,IGLOBE,NROWS,NCOLS,NLAYS,NFABS
  2.957441 6.69448 0. 360. .08364234 47.890324 $M-2 PH1,PH2,THET1,THET2,
                                                $    Ra,Rb
  0  0              $M-5 IWALL,IWIMP
  410               $N-1 KELT
  6  6  6  6  0     $P-1 IBLN(i), i=1,4, IBOND
  1  0  0  0  0  0  0 $Q-1 NSYS,NICS,NAMS,NUSS,NHINGE,etc.
  1  1  0           $Q-2 ISYS,NN,IFLG
 -460.  5  3  0  0  0 $Q-3 P,LT,LD,LI,LJ,LAX
  0  0  0  0  0     $R-1 IPRD,IPRR,IPRE,IPRS,IPRP
C unit 3 = toroidal
  8  0  0  0  0  0  $M-1 ISHELL,IGLOBE,NROWS,NCOLS,NLAYS,NFABS
  6.67782 10.67682 0. 360. .4623073 44.752884 $M-2 PH1,PH2,THET1,THET2,
                                               $    Ra,Rb
  0  0              $M-5 IWALL,IWIMP
  410               $N-1 KELT
  6  6  6  6  0     $P-1 IBLN(i), i=1,4, IBOND
  1  0  0  0  0  0  0 $Q-1 NSYS,NICS,NAMS,NUSS,NHINGE,etc.
  1  1  0           $Q-2 ISYS,NN,IFLG
 -460.  5  3  0  0  0 $Q-3 P,LT,LD,LI,LJ,LAX
  0  0  0  0  0     $R-1 IPRD,IPRR,IPRE,IPRS,IPRP
C unit 4 = toroidal
  8  0  0  0  0  0  $M-1 ISHELL,IGLOBE,NROWS,NCOLS,NLAYS,NFABS
  10.65673 15.12016 0. 360. 1.338907 40.095947 $M-2 PH1,PH2,THET1,THET2,
                                                $    Ra,Rb
  0  0              $M-5 IWALL,IWIMP
  410               $N-1 KELT
  6  6  6  6  0     $P-1 IBLN(i), i=1,4, IBOND
  1  0  0  0  0  0  0 $Q-1 NSYS,NICS,NAMS,NUSS,NHINGE,etc.
  1  1  0           $Q-2 ISYS,NN,IFLG
 -460.  5  3  0  0  0 $Q-3 P,LT,LD,LI,LJ,LAX
  0  0  0  0  0     $R-1 IPRD,IPRR,IPRE,IPRS,IPRP
C unit 5 = toroidal
  8  0  0  0  0  0  $M-1 ISHELL,IGLOBE,NROWS,NCOLS,NLAYS,NFABS
  15.08829 20.32144 0. 360. 2.895449 34.199043 $M-2 PH1,PH2,THET1,THET2,
                                                $    Ra,Rb
  0  0              $M-5 IWALL,IWIMP
  410               $N-1 KELT
  6  6  6  6  0     $P-1 IBLN(i), i=1,4, IBOND
  1  0  0  0  0  0  0 $Q-1 NSYS,NICS,NAMS,NUSS,NHINGE,etc.
  1  1  0           $Q-2 ISYS,NN,IFLG
 -460.  5  3  0  0  0 $Q-3 P,LT,LD,LI,LJ,LAX
  0  0  0  0  0     $R-1 IPRD,IPRR,IPRE,IPRS,IPRP
C unit 6 = toroidal
  8  0  0  0  0  0  $M-1 ISHELL,IGLOBE,NROWS,NCOLS,NLAYS,NFABS
  20.26536 26.78145 0. 360. 5.259145 27.465466 $M-2 PH1,PH2,THET1,THET2,
                                                $    Ra,Rb
```

```
  0  0              $M-5 IWALL,IWIMP
  410               $N-1 KELT
  6  6  6  6  0     $P-1 IBLN(i), i=1,4, IBOND
  1  0  0  0  0  0  0 $Q-1 NSYS,NICS,NAMS,NUSS,NHINGE,etc.
  1  1  0           $Q-2 ISYS,NN,IFLG
 -460.  5  3  0  0  0 $Q-3 P,LT,LD,LI,LJ,LAX
  0  0  0  0  0     $R-1 IPRD,IPRR,IPRE,IPRS,IPRP
C unit 7 = toroidal
  8  0  0  0  0  0  $M-1 ISHELL,IGLOBE,NROWS,NCOLS,NLAYS,NFABS
  26.79548 32.96853 0. 360. 7.971097 21.436380 $M-2 PH1,PH2,THET1,THET2,
                                                $    Ra,Rb
  0  0              $M-5 IWALL,IWIMP
  410               $N-1 KELT
  6  6  6  6  0     $P-1 IBLN(i), i=1,4, IBOND
  1  0  0  0  0  0  0 $Q-1 NSYS,NICS,NAMS,NUSS,NHINGE,etc.
  1  1  0           $Q-2 ISYS,NN,IFLG
 -460.  5  3  0  0  0 $Q-3 P,LT,LD,LI,LJ,LAX
  0  0  0  0  0     $R-1 IPRD,IPRR,IPRE,IPRS,IPRP
C unit 8 = toroidal
  8  0  0  0  0  0  $M-1 ISHELL,IGLOBE,NROWS,NCOLS,NLAYS,NFABS
  32.94721 39.85107 0. 360. 10.52211 16.758169 $M-2 PH1,PH2,THET1,THET2,
                                                $    Ra,Rb
  0  0              $M-5 IWALL,IWIMP
  410               $N-1 KELT
  6  6  6  6  0     $P-1 IBLN(i), i=1,4, IBOND
  1  0  0  0  0  0  0 $Q-1 NSYS,NICS,NAMS,NUSS,NHINGE,etc.
  1  1  0           $Q-2 ISYS,NN,IFLG
 -460.  5  3  0  0  0 $Q-3 P,LT,LD,LI,LJ,LAX
  0  0  0  0  0     $R-1 IPRD,IPRR,IPRE,IPRS,IPRP
C unit 9 = toroidal
  8  0  0  0  0  0  $M-1 ISHELL,IGLOBE,NROWS,NCOLS,NLAYS,NFABS
  39.77901 48.82777 0. 360. 13.07984 12.785950 $M-2 PH1,PH2,THET1,THET2,
                                                $    Ra,Rb
  0  0              $M-5 IWALL,IWIMP
  410               $N-1 KELT
  6  6  6  6  0     $P-1 IBLN(i), i=1,4, IBOND
  1  0  0  0  0  0  0 $Q-1 NSYS,NICS,NAMS,NUSS,NHINGE,etc.
  1  1  0           $Q-2 ISYS,NN,IFLG
 -460.  5  3  0  0  0 $Q-3 P,LT,LD,LI,LJ,LAX
  0  0  0  0  0     $R-1 IPRD,IPRR,IPRE,IPRS,IPRP
C unit 10 = toroidal
  8  0  0  0  0  0  $M-1 ISHELL,IGLOBE,NROWS,NCOLS,NLAYS,NFABS
  48.74254 60.90592 0. 360. 15.55374 9.5117826 $M-2 PH1,PH2,THET1,THET2,
                                                $    Ra,Rb
  0  0              $M-5 IWALL,IWIMP
  410               $N-1 KELT
  6  6  6  6  0     $P-1 IBLN(i), i=1,4, IBOND
  1  0  0  0  0  0  0 $Q-1 NSYS,NICS,NAMS,NUSS,NHINGE,etc.
```

```
  1  1  0           $Q-2 ISYS,NN,IFLG
 -460.  5  3  0  0  0 $Q-3 P,LT,LD,LI,LJ,LAX
  0  0  0  0  0     $R-1 IPRD,IPRR,IPRE,IPRS,IPRP
C unit 11 = toroidal
  8  0  0  0  0  0  $M-1 ISHELL,IGLOBE,NROWS,NCOLS,NLAYS,NFABS
  60.95361 75.15099 0. 360. 17.45365 7.3341379 $M-2 PH1,PH2,THET1,THET2,
                                               $    Ra,Rb
  0  0             $M-5 IWALL,IWIMP
  410               $N-1 KELT
  6  6  6  6  0     $P-1 IBLN(i), i=1,4, IBOND
  1  0  0  0  0  0  0 $Q-1 NSYS,NICS,NAMS,NUSS,NHINGE,etc.
  1  1  0           $Q-2 ISYS,NN,IFLG
 -460.  5  3  0  0  0 $Q-3 P,LT,LD,LI,LJ,LAX
  0  0  0  0  0     $R-1 IPRD,IPRR,IPRE,IPRS,IPRP
C unit 12 = toroidal
  8  0  0  0  0  0  $M-1 ISHELL,IGLOBE,NROWS,NCOLS,NLAYS,NFABS
  75.3152 89.91051 0.0 360.0 18.40842 6.3415871 $M-2 PH1,PH2,THET1,THET2,
                                                $    Ra,Rb
  0  0             $M-5 IWALL,IWIMP
  410               $N-1 KELT
  6  6  0  6  0     $P-1 IBLN(i), i=1,4, IBOND
 001  000           $P-2 ITRA, IROT (conditions at pole)
  1  0  0  0  0  0  0 $Q-1 NSYS,NICS,NAMS,NUSS,NHINGE,etc.
  1  1  0           $Q-2 ISYS,NN,IFLG
 -460.  5  3  0  0  0 $Q-3 P,LT,LD,LI,LJ,LAX
  0  0  0  0  0     $R-1 IPRD,IPRR,IPRE,IPRS,IPRP
$
$      ELEMENT UNIT for RIGID LINKS
$
$ S-1 records...
$USRPT unit row col  ignore coords  freedoms  AUX  #defs  layer
  1     1    1   1   3*0.           2*111     0     90      0
  1     0    0   1   $ Increment variable above by value
END                  $ Computer does the counting for you!
$
$ Element records, "command method"
E120_ELEMENTS        $ Ask for rigid link element
$N1 N2 N3 Kelt Ndefs, increment N1,N2,N3. N3 must be unity.
 1  2  1  120   89    1  1  0  $  See T1 record. Want 89 elements
 1. $ SCALE
END  $ Computer did the counting, incrementation
0   $ No loads
0   $ No printed output
-------------------------------------------------------------
```

NOTE: We must provide input data for each shell segment.
For example, for Shell Segment No. 8 (Unit 8 in STAGS jargon),

we must supply the following geometrical input data:

```
32.94721 39.85107 0. 360. 10.52211 16.758169 $M-2 PH1,PH2,THET1,THET2,
                                             $    Ra,Rb
```

Where do we obtain the numerical values for PH1, PH2, THET1, THET2, Ra, Rb?

THET1 and THET2 are easy: in the 360-degree STAGS model the shell is a closed shell of revolution so that THET1 = 0 and THET2 = 360 degrees.

For a toroidal segment the quantities, Ra and Rb, are defined as follows:

Ra= radius from axis of revolution to the center of meridional curvature
Rb= meridional radius of curvature

The angles, PH1 and PH2, are measured in degrees from the axis of revolution to the beginning (PH1) and to the end (PH2) of the shell segment.

The four quantities, PH1, PH2, Ra, Rb, can be read from BIGBOSOR4 output. For example, for Shell Segment No. 8, BIGBOSOR4 lists the following in the <casename>.OUT file:

```
 SEGMENT NO.  8 IS SPHERICAL OR TOROIDAL.
  END POINT COORDINATES (0.196363E+02,-.753289E+01) AND
                        (0.212607E+02,-.633536E+01)
  AND CENTER (  0.1052211E+02,  0.6530096E+01)
 RADIUS =    1.6758169E+01 ALPHA1 =    3.2947208E+01
 ALPHA2 =   3.9851074E+01  INCREASING ARC LENGTH ANTICLOCKWISE
```

In the above output, the **first of the two** (r,z) "**CENTER**" coordinates, (**0.1052211E+02**, 0.6530096E+01), is the same as Ra. The other BIGBOSOR4 variables correspond to the STAGS required input data as follows:
RADIUS = Rb; **ALPHA1** = PH1; **ALPHA2** = PH2
NOTE: The string, "INCREASING ARC LENGTH ANTICLOCKWISE" refers to the BIGBOSOR4 (BOSOR4) model, not to the STAGS model.

Assume that now we have valid <casename>.bin and <casename>.inp files in the directory from which we want to run STAGS.

The appropriate command to run STAGS at the writer's facility is:

```
stags <casename> -1 /home/bush/us1 -2 /home/bush/us2 -b
```

If during execution a STAGS case fails, look at the <casenam>.log file first. Then look at the <casename>.out2 file. Often, almost nothing will be in the <casename>.out2 file because the error occurs during execution of "s1", that is, the error will be noted in the file <casename>.out1. If that is so, inspect the <casename>.out1 file. Search for the words, "ERROR" or "WARNING". Inspect the end of the <casename>.out1 file first.

If the STAGS execution is satisfactory, look at the <casename>.out2 file. If it is a buckling analysis, search for the string, "roots". Then search for the string, "CONVERGENCE HAS", to find the eigenvalue(s). If you ask for 8 eigenvalues and STAGS fails to find all 8, then you can set SHIFT (near the end of the <casename>.bin file) to some value that will probably lead to convergence to all 8 requested eigenvalues. Set SHIFT to a value near the center of the cluster of eigenvalues successfully determined by STAGS. Look at the value of "roots" to make sure that you haven't missed any lower eigenvalues.

You may set the initial eigenvalue "SHIFT" too high. You can sometimes tell if you have done this by searching for the string, "roots" and seeing if, in factoring the stability matrix, STAGS counts more "roots" than the number given upon the first occurrence of the string, "roots", in the <casename>.out2 file.

The most significant output from STAGS (<casename>.out2 file) is the following (in the particular case of linear bifurcation buckling, that is, **INDIC = 1** in the <casename>.bin file):

Output from STAGS (abridged) eqellipse.stiffened.stags.linearbuck.out2

CONVERGENCE HAS BEEN OBTAINED FOR EIGENVALUES 1 THROUGH 8

		CRITICAL LOAD FACTOR COMBINATION		
NO.	EIGENVALUE	LOAD SYSTEM A	LOAD SYSTEM B	@DOF (writer's comments)
1	2.835021E+00	2.835021E+00	0.000000E+00	3 <--axisymmetric mode1
2	3.004836E+00	3.004836E+00	0.000000E+00	3795 <-- n = 1 circ. wave
3	3.004836E+00	3.004836E+00	0.000000E+00	3393 <-- n = 1 circ. wave
4	3.483754E+00	3.483754E+00	0.000000E+00	8277 <-- n = 2 circ. waves
5	3.483755E+00	3.483755E+00	0.000000E+00	8211 <-- n = 2 circ. waves
6	3.505017E+00	3.505017E+00	0.000000E+00	3 <--axisymmetric mode2
7	3.551819E+00	3.551819E+00	0.000000E+00	5241 <-- n = 1 circ. wave
8	3.551820E+00	3.551820E+00	0.000000E+00	5109 <-- n = 1 circ. wave

NOTE: The comments on the right-hand side of the above list were added by the writer and are not part of the STAGS output. This abridged list of eqellipse.out2 is also contained in Table 41.

The eigenvectors (buckling mode shapes) corresponding to EIGENVALUEs 1,

2, 4, 6, and 7 are displayed in Figs. 6, 7, 8, 9, and 10, respectively, of the paper, sdm50.report.pdf.

Eigenvalues 2 and 3 correspond to non-axisymmetric buckling with n = 1 circumferential wave. These two modes are the same, except that one mode is rotated about the axis of revolution with respect to the other. Eigenvalue 1 corresponds to axisymmetric buckling and is, in GENOPT jargon, called "(mode 1)". This first mode from STAGS is the one we want to use as an imperfection shape in the nonlinear equilibrium (**INDIC=3**) STAGS run that follows (described below). We use this lowest AXISYMMETRIC mode first because we want to compare load-deflection curves of an axisymmetrically imperfect equivalent ellipsoidal shell from STAGS with those from BIGBOSOR4 and BOSOR5 where only axisymmetric imperfections can be handled. Later, we also use the second axisymmetric mode (EIGENVALUE no. 6).

To get a plot of an eigenvector (<casename>.pdf file), generate a <casename>.pin file using "ustapl". A typical <casename>.pin file for plotting the first eigenvector in a linear buckling analysis (**INDIC=1**) is as follows:

```
-------------------------------------------------------------
linear buckling of perfect shell from STAGS
  1  0  1  0  $PL-2  NPLOT,IPREP,IPRS,KDEV
     1     0     4     0     1  $PL-3  KPLOT,NUNIT,ITEM,STEP,MODE
    0.0   3  $PL-5  DSCALE,NROTS
  1   -35.84   $PL-6  IROT,ROT
  2   -13.14   $PL-6  IROT,ROT
  3    35.63   $PL-6  IROT,ROT
-------------------------------------------------------------
```

For the 2nd eigenvector, set "MODE" in the third line to 2 , and so on for eigenvectors corresponding to higher eigenvalues.

To get a plot of the prebuckling distribution of axial resultant, Nx, use the following <casename>.pin file:

```
-------------------------------------------------------------
eqellipse plot of prebuckling Nx
 1  0  1  0  $PL-2  NPLOT,IPREP,IPRS,KDEV
 2   0  5   0  0  0  0  1 $PL-3 KPLOT,VIEW,ITEM,STEP,
                          $     MODE,IFRNG,COLOR,ICOMP
 0.0   0  0.0  0.0  0.0 $PL-5 DSCALE,NROTS,LWSCALE,RNGMIN,RGMAX
-------------------------------------------------------------
```

To get a plot of Ny change ICOMP to 2 ; to get a plot of Nxy set ICOMP = 3 .

If we already have a valid <casename>.pin (input data for ustapl)

file, we type the command:

ustapl <casename>

This command, if completed successfully, generates a <casename>.pdf file, which you can view on the screen by typing the command:

acroread <casename>.pdf

If the <casename>.pdf file looks too short, meaning that ustapl failed, then inspect the <casename>.pout file for error messages.

In this particular case we generate several "pdf" files of the linear buckling modes obtained from STAGS (Figs. 6, 7, 8, 9, 10 in the paper called sdm50.report.pdf).

Figure 6 shows the first AXISYMMETRIC mode shape, which is called "mode 1" in the paper, sdm50.report.pdf. This "mode 1" corresponds to the first eigenvalue (IMMODE = 1) in the STAGS analysis. This axisymmetric mode from STAGS is plotted in Fig. 6. The 2nd axisymmetric mode from STAGS (called "mode 2" in GENOPT jargon) corresponds to the sixth eigenvalue. It is plotted in Fig. 7. Figure 16 shows the STAGS prediction of nonlinear axisymmetric static response of an imperfect isogrid-stiffened equivalent ellipsoidal shell in which the axisymmetric imperfection shapes are (+ and -) "mode 1" and (+ and -) "mode 2", each of the 4 buckling modal imperfections having an amplitude, Wimp = 0.2 inch. Only one imperfection shape is used at a time.

Next, we wish to conduct a nonlinear equilibrium analysis with STAGS (**INDIC=3** in the *.bin file), including an axisymmetric imperfection shape similar to the "-(mode 1)" imperfection shape used in the nonlinear stress and collapse analyses in the GENOPT case enumerated in the file, sdm50.report.pdf.

First, we want to use the NEGATIVE of the buckling modal shape displayed in Fig. 6 as an imperfection shape because that is the most critical case (Fig.16). In the GENOPT jargon this is analogous to the "-(mode 1)" axisymmetric imperfection shape. Therefore, in the <casename>.inp file (called "eqellipse.stiffened.stags.inp2" here), the amplitude of the buckling modal imperfection corresponding to STAGS datum, IMMODE = 1, is set to WIMPFA = -0.2 inches.

NOTE: The only differences between the file,

eqellipse.stiffened.stags.inp1

and the file,

eqellipse.stiffened.stags.inp2

are:

1. NIMPFS is equal to zero in the eqellipse.stiffened.stags.inp1 file.
   NIMPFS is equal to one in the eqellipse.stiffened.stags.inp2 file.

2. There is a new "B5" record. In the eqellipse.stiffened.stags.inp1
   file we have:

```
-----------------------------------------------------------------------
perfect isogrid-stiffened equivalent ellipsoidal head X_320
  0  0  0  0  0  0  0   $B-1 IGRAV,ICHECK,ILIST,INCBC,NRUNIT,NROTS,KDEV
  12  1  0  23  0,  $B-2 NUNITS,NUNITE,NSTFS,NINTS,NPATS,
  0  0  0  0  0  0  $B-2 NCONST,NIMPFS,INERT,NINSR,NPATX,NSTIFS
  2  0  0  0  0  0  $B-3 NTAM,NTAB,NTAW,NTAP,NTAMT,NGCP
  5 91,              $F-1 NROWS(1),NCOLS(1)
-----------------------------------------------------------------------
```

and in the eqellipse.stiffened.stags.inp2 file we have:

```
-----------------------------------------------------------------------
imperfect isogrid-stiffened equivalent ellipsoidal head X_320
  0  0  0  0  0  0  0   $B-1 IGRAV,ICHECK,ILIST,INCBC,NRUNIT,NROTS,KDEV
  12  1  0  23  0,  $B-2 NUNITS,NUNITE,NSTFS,NINTS,NPATS,
  0  1  0  0  0  0  $B-2 NCONST,NIMPFS,INERT,NINSR,NPATX,NSTIFS
  2  0  0  0  0  0  $B-3 NTAM,NTAB,NTAW,NTAP,NTAMT,NGCP
 -0.200   0  1  1   $B-5 WIMPFA, IMSTEP, IMMODE, IMRUN
  5 91,              $F-1 NROWS(1),NCOLS(1)
-----------------------------------------------------------------------
```

We copy the two files,

 eqellipse.stiffened.stags.inp2
 eqellipse.stiffened.stags.bin2

into the directory where we are running STAGS and we rename them

 eqellipse.inp
 eqellipse.bin

First we edit the eqellipse.bin file (making the highest
load factor, FACM(1), equal to unity instead of 2.5 and asking
for zero eigenvalues), so that we have for the eqellipse.bin file
the following:

```
-----------------------------------------------------------------
optimized imperfect shell, nonlinear theory (INDIC=3)
 3, $ INDIC=1 is bifur.buckling; INDIC=3 is nonlinear BEGIN B-1
 1, $ IPOST=1 means save displacements every IPOSTth step
 0, $ ILIST =0 means normal batch-oriented output
```

```
 0, $ ICOR  =0 means projection in; 1 means not in.
 1, $ IMPTHE=index for imperfection theory.
 0, $ IOPTIM=0 means bandwith optimization will be performed
 0, $ IFLU  =0 means no fluid interaction.
 -1 $ ISOLVR= 0 means original solver; -1 new solver.END B-1 rec
 5.000E-02, $ STLD(1) = starting load factor, System A. BEGIN C-1 rec.
 5.000E-02, $ STEP(1) = load factor increment, System A
 1.000E+00, $ FACM(1) = maximum load factor, System A
 0.000E+00, $ STLD(2) = starting load factor, System B
 0.000E+00, $ STEP(2) = load factor increment, System B
 0.000E+00, $ FACM(2) = maximum load factor, System B
 0  $ ITEMP =0 means no thermal loads. END C-1 rec.
   0, $ ISTART=restart from ISTARTth load step.   BEGIN D-1 rec.
 500,$ NSEC= number of CPU seconds before run termination
 15,$ NCUT = number of times step size may be cut
  -20, $ NEWT = number of refactorings allowed
-1,$ NSTRAT=-1 means path length used as independent parameter
 0.00005,$ DELX=convergence tolerance
 0. $ WUND = 0 means initial relaxation factor =1.END D-1 rec.
 0,  0, 0  $ NPATH=0: Riks method, NEIGS=no.of eigs, NSOL=0: contin.ET-1
```

Then we launch the nonlinear equilibrium STAGS run via the command:

```
stags eqellipse -1 /home/bush/us1 -2 /home/bush/us2 -b
```

The results appear in the new eqellipse.out2 file. The most important part of this file is the list of load steps for which a converged solution was determined. Search for the string, "LIST", to find the following output in the file called “eqellipse.out2”:

```
 LIST OF LOAD STEPS AND LOAD FACTORS
 STEP      PA            PB            PX
    0  0.500000E-01  0.000000E+00
    1  0.500000E-01  0.000000E+00
    2  0.100000E+00  0.000000E+00
    3  0.137106E+00  0.000000E+00
    4  0.192128E+00  0.000000E+00
    5  0.273220E+00  0.000000E+00
    6  0.391567E+00  0.000000E+00
    7  0.561420E+00  0.000000E+00
    8  0.797259E+00  0.000000E+00
    9  0.100000E+01  0.000000E+00
```

PA = 1.0 corresponds to the external design pressure, p = 460 psi.

If we wish to find the meridional stress distribution at the inner fiber of the isogrid layer in Shell Units 8, 9, 10, 11, and 12

of the STAGS model (isogrid layer = layer no. 1 in the
STAGS model, shell skin = layer number 2 in the STAGS model),
we execute "ustapl" with use of the following eqellipse.pin input file:

```
-------------------------------------------------------------------
STAGS: nonlinear meridional stress isogrid inner fiber
  1  0  1  0  $PL-2  NPLOT,IPREP,IPRS,KDEV
  2   5  7   9  0  0  0  1  1  1 $PL-3  KPLOT,VIEW,ITEM,STEP,MODE,
                                 $   IFRNG,COLOR,ICOMP,LAYER,FIBR
  8  9  10  11  12               $ include only Units 8 - 12
   0.0   3  0.0  0.0  0.0 $PL-5  DSCALE,NROTS,LWSCALE,RNGMIN,RGMAX
  1  -0.35840000E+02   $PL-6  IROT,ROT
  2  -0.13140000E+02   $PL-6  IROT,ROT
  3   0.35630001E+02   $PL-6  IROT,ROT
-------------------------------------------------------------------
```

The execution of ustapl with the input data just listed generates a *.pdf file which shows contour plots of the meridional stress at the inner fiber of the isogrid “layer”. The *.pdf file is plotted in Fig. 29 in the report, sdm50.report.pdf.

In order to try to find the axisymmetric collapse load, we must continue the nonlinear equilibrium (**INDIC=3**) STAGS analysis. We edit the eqellipse.bin file so that we can start where the previous nonlinear run left off, at Load Step No. 9. The new eqellipse.bin file is as follows:

```
-------------------------------------------------------------------------
optimized imperfect shell, nonlinear theory (INDIC=3)
 3, $ INDIC=1 is bifur.buckling; INDIC=3 is nonlinear BEGIN B-1
 1, $ IPOST=1 means save displacements every IPOSTth step
 0, $ ILIST =0 means normal batch-oriented output
 0, $ ICOR  =0 means projection in; 1 means not in.
 1, $ IMPTHE=index for imperfection theory.
 0, $ IOPTIM=0 means bandwith optimization will be performed
 0, $ IFLU  =0 means no fluid interaction.
 -1 $ ISOLVR= 0 means original solver; -1 new solver.END B-1 rec
 1.0, $ STLD(1) = starting load factor, System A. BEGIN C-1 rec.
 1.000E-01, $ STEP(1) = load factor increment, System A
 1.200E+00, $ FACM(1) = maximum load factor, System A
 0.000E+00, $ STLD(2) = starting load factor, System B
 0.000E+00, $ STEP(2) = load factor increment, System B
 0.000E+00, $ FACM(2) = maximum load factor, System B
 0  $ ITEMP =0 means no thermal loads. END C-1 rec.
   9, $ ISTART=restart from ISTARTth load step.   BEGIN D-1 rec.
 300,$ NSEC= number of CPU seconds before run termination
 15,$ NCUT = number of times step size may be cut
  -20, $ NEWT = number of refactorings allowed
-1,$ NSTRAT=-1 means path length used as independent parameter
```

```
 0.00005,$ DELX=convergence tolerance
 0. $ WUND = 0 means initial relaxation factor =1.END D-1 rec.
 0,  0, 0  $ NPATH=0: Riks method, NEIGS=no.of eigs, NSOL=0: contin.ET-1
-----------------------------------------------------------------------
```

The eqellipse.inp file remains unchanged. We launch another STAGS run via the command:

```
stags eqellipse -1 /home/bush/us1 -2 /home/bush/us2 -b
```

After this run finishes we have a new eqellipse.out2 file, the most significant part of which is as follows:

```
-------------------------------------------
 LIST OF LOAD STEPS AND LOAD FACTORS
 STEP      PA            PB            PX
    9  0.100000E+01  0.000000E+00
   10  0.101816E+01  0.000000E+00
   11  0.104443E+01  0.000000E+00
   12  0.108156E+01  0.000000E+00
   13  0.113111E+01  0.000000E+00
   14  0.118720E+01  0.000000E+00
   15  0.120000E+01  0.000000E+00
-------------------------------------------
```

Again, we must launch a new nonlinear equilibrium STAGS run (INDIC=3) with use of a new eqellipse.bin file, as follows:

```
-----------------------------------------------------------------
optimized imperfect shell, nonlinear theory (INDIC=3)
 3, $ INDIC=1 is bifur.buckling; INDIC=3 is nonlinear BEGIN B-1
 1, $ IPOST=1 means save displacements every IPOSTth step
 0, $ ILIST =0 means normal batch-oriented output
 0, $ ICOR  =0 means projection in; 1 means not in.
 1, $ IMPTHE=index for imperfection theory.
 0, $ IOPTIM=0 means bandwith optimization will be performed
 0, $ IFLU  =0 means no fluid interaction.
 -1 $ ISOLVR= 0 means original solver; -1 new solver.END B-1 rec
 1.2, $ STLD(1) = starting load factor, System A. BEGIN C-1 rec.
 1.000E-01, $ STEP(1) = load factor increment, System A
 1.400E+00, $ FACM(1) = maximum load factor, System A
 0.000E+00, $ STLD(2) = starting load factor, System B
 0.000E+00, $ STEP(2) = load factor increment, System B
 0.000E+00, $ FACM(2) = maximum load factor, System B
 0  $ ITEMP =0 means no thermal loads. END C-1 rec.
  15, $ ISTART=restart from ISTARTth load step.   BEGIN D-1 rec.
 300,$ NSEC= number of CPU seconds before run termination
 15,$ NCUT = number of times step size may be cut
  -20, $ NEWT = number of refactorings allowed
-1,$ NSTRAT=-1 means path length used as independent parameter
```

```
 0.00005,$ DELX=convergence tolerance
 0. $ WUND = 0 means initial relaxation factor =1.END D-1 rec.
 0,  0, 0  $ NPATH=0: Riks method, NEIGS=no.of eigs, NSOL=0: contin.ET-1
```

Again, we launch the new STAGS run via the command:

```
stags eqellipse -1 /home/bush/us1 -2 /home/bush/us2 -b
```

After this third nonlinear run finishes we have a new eqellipse.out2 file, the most significant part of which is as follows:

```
 LIST OF LOAD STEPS AND LOAD FACTORS
 STEP      PA            PB            PX
   15  0.120000E+01  0.000000E+00
   16  0.120107E+01  0.000000E+00
   17  0.120262E+01  0.000000E+00
   18  0.120546E+01  0.000000E+00
   19  0.121019E+01  0.000000E+00
   20  0.121590E+01  0.000000E+00
   21  0.121596E+01  0.000000E+00 <--This is the collapse load.
   22  0.120181E+01  0.000000E+00    The collapse pressure in psi is
   23  0.116407E+01  0.000000E+00    given by PA x 460
   24  0.110760E+01  0.000000E+00
   25  0.103917E+01  0.000000E+00
   26  0.965515E+00  0.000000E+00
   27  0.892909E+00  0.000000E+00
   28  0.828470E+00  0.000000E+00
   29  0.781610E+00  0.000000E+00
   30  0.763487E+00  0.000000E+00
   31  0.778943E+00  0.000000E+00
   32  0.816120E+00  0.000000E+00
```

To obtain the external pressure in psi, we use the product,

PA*(applied pressure in each shell segment in the <casename>.inp file).

The maximum pressure for which converged results were obtained is at Load Step 21, and is:

```
             load      $ Q-3
 pressure   factor    pressure, P
  in psi            in each segment
                     of *.inp file
  p(max)   =  PA   *   (-460)      = 1.21596 * (-460) = -559.34 psi.
```

To obtain the load-deflection plot we execute **uxytrans** with the

following input (stored in the <casename>.pxy file, a list of which follows):

```
-------------------------------------------------------------
 P              $ (P)lotps or (S)pread_Sheet output
 eqellipse      $ STAGS solution 'Case Name'
 F              $ (F)ull or (C)ondensed Model
 Y              $ (Y)es-(N)o: setup data for another plot
  5             $ x-axis variable = choice (1 to 15)
       1        $ node no. (0 = ask for Unit,Row,Col)
 3              $ comp no., dis,vel,acc (1-6) = u,v,w,ru,rv,rw
 S              $ (G)lobal or (S)hell ref surface
 Y              $ (Y)es-(N)o: specify x-variable scale factor
 -1.0           $ x-variable scale factor
  2             $ y-axis variable = choice (1 to 15)
 Y              $ (Y)es-(N)o: specify x-variable scale factor
  0.460000E+03  $ y-variable scale factor
 N              $ (Y)es-(N)o: specify subrange of loadsteps
 Y              $ (Y)es-(N)o: plotted points start at origin
 N              $ (Y)es-(N)o: setup data for another plot
-------------------------------------------------------------
```

The output, <casename>.plt. from **uxytrans** is:

```
---------------------------------------
"Disp(1,w,L) vs. load_PA
   0.000000E+00    0.000000E+00
   2.109927E-02    2.300000E+01
   4.254400E-02    4.600000E+01
   5.869397E-02    6.306869E+01
   8.303329E-02    8.837901E+01
   1.198239E-01    1.256814E+02
   1.757387E-01    1.801207E+02
   2.616668E-01    2.582531E+02
   3.973414E-01    3.667389E+02
   5.436194E-01    4.600000E+02
   5.593396E-01    4.683524E+02
   5.833589E-01    4.804367E+02
   6.206204E-01    4.975161E+02
   6.797398E-01    5.203097E+02
   7.770769E-01    5.461098E+02
   8.126104E-01    5.520000E+02
   8.161874E-01    5.524943E+02
   8.215648E-01    5.532043E+02
   8.323741E-01    5.545132E+02
   8.541937E-01    5.566884E+02
   8.984855E-01    5.593120E+02
   9.657895E-01    5.593420E+02
   1.066471E+00    5.528304E+02
   1.213734E+00    5.354706E+02
```

```
   1.397812E+00    5.094970E+02
   1.626016E+00    4.780168E+02
   1.906073E+00    4.441367E+02
   2.247330E+00    4.107383E+02
   2.662107E+00    3.810960E+02
   3.166491E+00    3.595404E+02
   3.781092E+00    3.512041E+02
   4.475360E+00    3.583138E+02
   5.327549E+00    3.754154E+02
----------------------------------------

The data in the <casename>.plt file is in a form that makes it
very easy to incorporate into the files:

eqellipse.stiffened.bosor4andstags.input
eqellipse.stiffened.mode1mode2.collapse.input

We then obtain postscript files for plotting via the commands:

/home/progs/bin/plotps.linux
     < eqellipse.stiffened.bosor4andstags.input
     > eqellipse.stiffened.bosor4andstags.ps

/home/progs/bin/plotps.linux
     < eqellipse.stiffened.mode1mode2.collapse.input
     > eqellipse.stiffened.mode1mode2.collapse.ps


We obtain plots on the screen via the commands:

gv eqellipse.stiffened.bosor4andstags.ps
gv eqellipse.stiffened.mode1mode2.collapse.ps


Next, we wish to run essentially the same STAGS model, but this time
we want to include plasticity. The runstream is analogous to that
described above. Briefly, we type the following commands:

cp wall.plastic.src wall.F
makeuser s1
makeuser s2
makeuser stapl
makeuser xytrans
'rm' *.out2*

(First, execute the linear bifurcation [INDIC=1] analysis)
cp <casename>.stiffened.stags.bin1 <casename>.bin
cp <casename>.stiffened.stags.inp1 <casename>.inp
```

```
stags <casename> -1 /home/bush/us1 -2 /home/bush/us2 -b
(inspect the <casename>.out2 file)
(set up the proper <casename>.pin file)
ustapl <casename>
acroread <casename>.pdf

(Next, execute the nonlinear equilibrium [INDIC=3] analysis)
cp <casename>.stiffened.stags.bin2 <casename>.bin
cp <casename>.stiffened.stags.inp2 <casename>.inp
stags <casename> -1 /home/bush/us1 -2 /home/bush/us2 -b
(inspect the <casename>.out2 file; search for "LIST")
```
(Unfortunately in this case STAGS bombed for an unknown reason, probably because of the greatly elongated finite elements next to the pole. To run elastic-plastic cases successfully you must use the "soccerball" model.)
```
(If STAGS had run successfully to completion, we would have done
 the following)
uxytrans <casename> (use the input data listed above in <casename>.pxy)
(inspect the output from uxytrans in the <casename>.plt file)
(include the "x,y" values as an additional trace in the
 file called "eqellipse.bosor4andbosor5andstags.plotps.input"
 and obtain a new eqellipse.bosor4andbosor5andstags.plotps.ps file
 from the command,
/home/progs/bin/plotps.linux
   < eqellipse.bosor4andbosor5andstags.plotps.input
   > eqellipse.bosor4andbosor5andstags.plotps.ps
```

Please see Fig. 169 and Figs. a2 – a13 in the appendix for the STAGS model called "soccerball". Unlike the STAGS model based on polar coordinates, the "soccerball" model has no singularity at the apex of the shell. Therefore, one can successfully use the 480 finite element and one can obtain good results for domes in which elastic-plastic material behavior is included. Figures 169 – 276 are results generated with use of the "soccerball" STAGS model.

==========