Table a36 List of the file, **usrfab.soccerball.plastic.src**. This is the "fleshed out" version of **SUBROUTINE USRFAB** valid for the elastic-plastic 180-degree "soccerball" model displayed in Fig. a2. The difference between usrfab.soccerball.plastic.src and usrfab.plastic.src is analogous to the difference between wall.soccerball.plastic.src and wall.plastic.src. (See Table a33). **SUBROUTINE USRFAB is always used in connection with a "GCP" model, that is, when NGCP = 1 in the STAGS input file, *.inp . NOTE: From the experience gained in generating the results for the generic case, equivellipse, (see especially Fig. 175 and the discussion associated with Fig. 175), the writer urges future STAGS users to use USRFAB rather than WALL.**

```
==============================================================
c=deck    usrfab
c=purpose Template for user-written subroutine USRFAB
c=author  F.A. Brogan (with W.A. Loden revisions)
c=version May, 2002
c
c=This particular version is for an isogrid-stiffened
c=torispherical head optimized by GENOPT.
c=The isogrid stiffeners are internal and smeared.
c=The shell skin is layer 2; the internal isogrid is layer1.
c=The skin thickness and isogrid height vary in the meridional
c=direction only. (Meridional direction=XYs(1) coordinate.)

#include "keydefs.h"

#if   _usage_
*
*     Calling sequence:
*
*         call USRFAB ( t,      Pa,     Pb,     iunit,
*                       ielt,   kelt,   kfab,   eltip,
*                       XYZg,   XYs,    ntvals, tvals,
*                       nlayrs, lays,   laymat, laythk,
*                       layint, layang, zeta,   ecz,
*                       ilin,   iplas )
*
*     Input Arguments
*     ===============
*     t      = Time (seconds)
*     Pa     = Load factor for system A
*     Pb     = Load factor for system B
*     iunit  = Unit number; unit = 0 specifies the entire model
*     ielt   = Local element number within the specified unit; when
*                   unit = 0, elt specifies the global elt number
*     kelt   = 1 -- Unit is a shell unit
*            = 2 -- Unit is an element unit
```

```
*     kfab   = Fabrication number assigned for this element
*     eltip  = Surface (volume) integration point number in element
*     XYZg   = Global coordinates at integration point
*     XYs    = Shell X,Y coordinates at integration point
*     ntvals = Number of temperature sampling points
*     tvals  = Temperature gradient at sampling points
*     nlayrs = Number of layers in fabrication KFAB
*     lays   = Integer array for (optional) use in call to MATSET
*
*     Output Arguments
*     ================
*     laymat(j) = Material identifier for layer j
*     layint(j) = # of through-layer integration pts for layer j
*     laythk(j) = Thickness of layer j
*     layang(j) = Fabrication orientation angle of layer j
*     zeta      = Angle from wall-ref coord to fabrication coord
*     ecz       = Eccentricity in Z' dirn (Z' coord of mid surface)
*     ilin      = 0 -- Non-inear strain-displacement relations
*               = 1 -- Linear strain-displacement relations
*     iplas     = 0 -- Elastic material properties used
*               = 1 -- Plasticity theory enforced at all integ pts
*               = 2 -- Plasticity theory enforced at elt centroid
*
#endif

************************************************************************
         subroutine USRFAB ( t,      Pa,     Pb,     iunit,
     &                       ielt,   kelt,   kfab,   eltip,
     &                       XYZg,   XYs,    ntvals, tvals,
     &                       nlayrs, lays,   laymat, laythk,
     &                       layint, layang, zeta,   ecz,
     &                       ilin,   iplas )
************************************************************************

      _implicit_none_

      Real     t
      Real     Pa
      Real     Pb
      Integer  iunit
      Integer  ielt
      Integer  kelt
      Integer  kfab
      Integer  eltip
      Real     XYZg(3)
      Real     XYs(2)
      Integer  nlayrs
      Integer  ntvals
```

```
      Real     tvals(ntvals)
      Integer  lays(nlayrs)
      Integer  laymat(nlayrs)
      Real     laythk(nlayrs)
      Integer  layint(nlayrs)
      Real     layang(nlayrs)
      Real     zeta
      Real     ecz
      Integer  ilin
      Integer  iplas
      Integer    icap
      Integer    junit
      Integer    icirc

#include "mater1.h"
#include "mater2.h"
#include "mater3.h"
#include "mater4.h"
Cinclude "mater5.m"
Cinclude "mater6.m"
Cinclude "mater7.m"
#include "mater8.h"
#include "mater9.h"
#include "mater10.h"
#include "stndcm.h"

      Logical  debug
      Logical  NTITLE

*     ====================
*     MATERIAL TYPE CODES:
*     ====================
*
*     Code   Items   Description
*     ----   -----   -----------
*       1       7    Linear elastic isotropic material
*       2      18    Linear elastic orthotropic material
*       3      54    Mechanical sub-layer plasticity material
*       4      44    Linear elastic orthotropic brittle material
*       5      12    Shape-memory-alloy material
*       6      54    Plane-strain material
*       7      36    PDCOMP/PDLAM property material
*       8      40    Abaqus umat material
*       9      10    Membrane wrinkling material
*      10      19    Nonlinear elastic orthotropic material
C
      Real SPACNG,EMATL,DNMATL,PHDIFF,XDIFF,RATIO,TDIFF,HDIFF,TATX,HATX
      Real THSKIN,THKSTF,HEIGHT,PHORIG,SARCLT
```

```
      Integer I5,NSEG,ISEG,JSEG,I5I,I,IMORE,IMORE1
      COMMON/ISEGX1/PHORIG(100,30),SARCLT(100,30)
      COMMON/ISEGX2/THSKIN(100,30),THKSTF(100,30),HEIGHT(100,30)
      COMMON/ISEGX3/I5(30)
      COMMON/ISEGX4/SPACNG,EMATL,NUMATL,DNMATL
      REAL NUMATL
      CHARACTER*38 WORD1,WORD2,WORD3,WORD4,WORD5,WORD6,WORD7,WORD8
      CHARACTER*2 WORD3B
C2345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012
C
      character       filnam*33
      integer         iw,       ios,      itime
      data            iw    / 61 /
      data            itime / -1 /

c--------------------------------------------------------------------
c     1st time enter, open the wall thickness file (iw)
c     read the data therein
c     and fill common blocks ISEGX1, ISEGX2, ISEGX3
c--------------------------------------------------------------------
      if ( itime .lt. 0 ) then

        filnam  =  'WALLTHICK.STAGS'

        open ( unit=iw, name=filnam, access='SEQUENTIAL',
     $         form='FORMATTED', iostat=ios)

        if (ios .ne. 0) then
          write(not,3000) iw, filnam, ios
 3000     format (/,'*****ERROR in routine WALL(---) *****',
     $            /,'tried to open file:  iw = ',I4,'   name = ',A,
     $            /,'error return (iostat) = ',I12,/)
          call exit
        endif
c--------------------------------------------------------------------
C  Retrieve angle, PHORIG and arc length SARCLT (X-coordinates),
C  shell skin thickness THSKIN, stringer thickness, THKSTF, and
C  stringer height, HEIGHT
C
        WORD1 = '     Number of shell segments (units)='
        WORD2 = '   Isogrid spacing,modulus,nu,density='
        WORD3 = '               Nodal points in Segment'
        WORD3B= '= '
        WORD4 = '                  Angle (X-coordinate)='
        WORD5 = ' Meridional arc length (X-coordinate)='
        WORD6 = '                  Shell skin thickness='
        WORD7 = '        Stringer (or isogrid)  height='
        WORD8 = '     Stringer (or isogrid)  thickness='
```

```
        READ(iw,'(/,A38,I4)') WORD1,NSEG
        READ(iw,'(/,A38,1P,4E14.6)')
     1   WORD2,SPACNG,EMATL,NUMATL,DNMATL
        DO 3 ISEG = 1,NSEG
           READ(iw,'(/,A38,I3,A2,I4)') WORD3,JSEG,WORD3B,I5I
           I5(ISEG) = I5I
           READ(iw,'(/,A38,/(1P5E14.6))') WORD4,(PHORIG(I,ISEG),I=1,I5I)
           READ(iw,'(/,A38,/(1P5E14.6))') WORD5,(SARCLT(I,ISEG),I=1,I5I)
           READ(iw,'(/,A38,/(1P5E14.6))') WORD6,(THSKIN(I,ISEG),I=1,I5I)
           READ(iw,'(/,A38,/(1P5E14.6))') WORD7,(HEIGHT(I,ISEG),I=1,I5I)
           READ(iw,'(/,A38,/(1P5E14.6))') WORD8,(THKSTF(I,ISEG),I=1,I5I)
    3   CONTINUE
C23456789012345678901234567890123456789012345678901234567890123456789012
C
c  Test SUBROUTINE WALL (remove the following statements later)
        rewind iw
        WRITE(not,'(/,A38,I4)')
     1  '     Number of shell segments (units)=',NSEG
        WRITE(not,'(/,A38,1P,4E14.6)')
     1  '   Isogrid spacing,modulus,nu,density=',
     1      SPACNG,EMATL,NUMATL,DNMATL
        DO 20 ISEG = 1,NSEG
           I5I = I5(ISEG)
           WRITE(not,'(/,A38,I3,A2,I4)')
     1'               Nodal points in Segment',ISEG,' =',I5I
           WRITE(not,'(/,A38,/(1P5E14.6))')
     1'                 Angle (X-coordinate)=', (PHORIG(I,ISEG),I=1,I5I)
           WRITE(not,'(/,A38,/(1P5E14.6))')
     1' Meridional arc length (X-coordinate)=', (SARCLT(I,ISEG),I=1,I5I)
           WRITE(not,'(/,A38,/(1P5E14.6))')
     1'                 Shell skin thickness=', (THSKIN(I,ISEG),I=1,I5I)
           WRITE(not,'(/,A38,/(1P5E14.6))')
     1'        Stringer (or isogrid)  height=', (HEIGHT(I,ISEG),I=1,I5I)
           WRITE(not,'(/,A38,/(1P5E14.6))')
     1'     Stringer (or isogrid)  thickness=', (THKSTF(I,ISEG),I=1,I5I)
   20   CONTINUE
C23456789012345678901234567890123456789012345678901234567890123456789012
c
        CLOSE(UNIT=iw)
C
        itime  =  0

      endif
C
c  Find thickness, stiffener height at shell coordinate, X:
c  thickness at X = TATX;  stiffener height at X = HATX
c
c BEG NOV 2008
```

```
c  soccerball shell unit number is not the same as the
c  equivalent ellipsoidal shell unit number...
      icirc = 180
      icap = 1
      if (icirc.eq.180) icap = 2
      if (icirc.eq.360) icap = 4
      if (iunit.le.(icap*3)) then
c      We are in the soccerball cap region (Shell Unit 1
c      in the 360-degree "polar coordinate" STAGS model).
c      TATX and HATX must be uniform within the soccerball
c      cap region for this "soccerball" version of usrfab
c      to be valid:
        TATX = THSKIN(1,1)
        HATX = HEIGHT(1,1)
        go to 30
      endif
c
      junit = (iunit - icap*3 + 2*icap-1)/(2*icap) + 1
      I5I = I5(junit)
      DO 10 I = 2,I5I
         IF (XYs(1).LT.PHORIG(I,junit)) THEN
            IMORE = I
            GO TO 11
         ENDIF
   10 CONTINUE
   11 CONTINUE
      IMORE1 = IMORE - 1
      PHDIFF = PHORIG(IMORE,junit) - PHORIG(IMORE1,junit)
      XDIFF  = XYs(1) - PHORIG(IMORE1,junit)
      RATIO  = XDIFF/PHDIFF
      TDIFF  = THSKIN(IMORE,junit) - THSKIN(IMORE1,junit)
      HDIFF  = HEIGHT(IMORE,junit) - HEIGHT(IMORE1,junit)
      TATX   = THSKIN(IMORE1,junit) + RATIO*TDIFF
      HATX   = HEIGHT(IMORE1,junit) + RATIO*HDIFF
c
   30 CONTINUE
C END NOV 2008
c
      ecz = (TATX + HATX)/2. - HATX
      laymat(1) = 2
      laymat(2) = 1
      layint(1) = 3
      if (HATX.GT.0.1*TATX) layint(1) = 5
      layint(2) = 5
      laythk(2)   = TATX
      laythk(1)   = HATX
      layang(1) = 0.
      layang(2) = 0.
```

```
      zeta = 0.
      ilin = 0
      iplas = 1
c
      return
c
c     debug = .false.
c     if (NTITLE('X_UsrFab')) debug = .true.
c     write (not,1000)
c1000  format (//'ERROR: Subroutine USRFAB has not been provided.' )
c     STOP
c
      end
```

==========