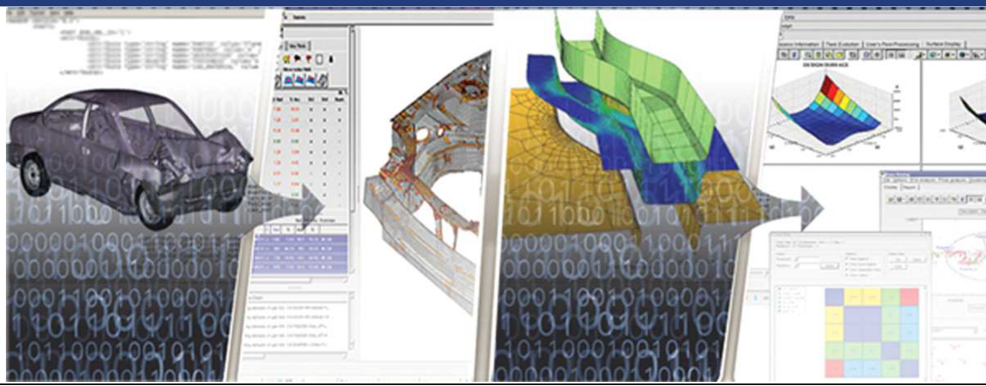


RADIOSS User's Code Interface

2017 version – January 2017

Extended User Material Laws

Chapter 3

Altair

Altair Engineering, Inc., World Headquarters: 1820 E. Big Beaver Rd., Troy MI 48083-2031 USA
Phone: +1.248.614.2400 • Fax: +1.248.614.2411 • www.altair.com • info@altair.com

TABLE OF CONTENTS

3.0 Extended User Material Laws

In RADIOSS, 99 material user laws can be defined for 3D and 2D solid elements and 3D shell elements. User laws for beam or truss elements are not yet available.

To define a user law, two subroutines for each law must be provided. One must be linked with RADIOSS Starter and the other with RADIOSS Engine.

- The Starter subroutine is called LECMUSERnn (where nn = 01, 02, 03, … 99) and reads material data and initializes material parameters.

- The Engine subroutine for solids is called LUSERnn (where nn = 01, 02, 03, … 99) and computes the solid element stress tensor at the integration point. The corresponding shell subroutine is called LUSERnnC (where nn = 01, 02, 03, … 99).

Note:	All communication between RADIOSS and the User's subroutines takes place within the argument list.

3.1 Starter Subroutine LECMUSERnn

This subroutine reads the user law input data. The first seven material cards (see RADIOSS Starter Input Manual 3.1 to 4.1) are read before this subroutine is called. The numbers and formats of specific material cards are free.

The argument list of LECMUSERnn is as follows:

```
C-----------------------------------------------------------------------
   SUBROUTINE LECMUSERnn(IIN, IOUT, UPARAM, MAXUPARAM,NUPARAM, NUVAR,IFUNC,MAXFUNC,NFUNC,PARMAT, USERBUF )
C-----------------------------------------------------------------------
```

Argument	Format	Description
IIN	Integer read only scalar	Input file unit (ROOTD00, ROOT_nnnn.rad) on which the data are read.
IOUT	Integer read only format	Output file unit (ROOT_nnnn.lis).
UPARAM	Float array	Array with a size NUPARAM used to store failure material data.
MAXUPARAM	Integer read only scalar	Maximum possible size of UPARAM.
NUPARAM	Integer scalar	Effective size of UPARAM. (MAXUPARAM, NUPARAM, MAXUPARAM are set to 1000).
NUVAR	Integer scalar	Number of extra variables needed for each integration point of each elements.
IFUNC	Integer array	Array with a size of NFUNC containing the list of RADIOSS functions used in failure model. The function numbers are stored in this array (not in UPARAM) due to a possible renumbering of the function's numbers.
MAXFUNC	Integer read only scalar	Maximum possible size of IFUNC.
NFUNC	Integer scalar	Number of RADIOSS functions.
STIFINT	Float scalar	Modulus needed to compute the interface stiffness. The Young's or bulk modulus is used to define the value, which estimates the time step.
USERBUF	Data structure read write	Defined in LAW_USER.mod. This module is used inside LECMUSERnn.

This **USERBUF** data structure contains:		
NAME	Character array read only	Array of size 100 containing material name.
ID	Integer read only scalar	Material ID defined in Starter input deck.

3.2 Engine Subroutine LUSERnn for Solid Elements

This subroutine calculates the stress tenor versus the strain tensor, strain rate tensor, density, volume, internal energy, or user variables.

The argument list of LUSERnn and its individual arguments and descriptions are as follows:

```
C-----------------------------------------------------------------------
   SUBROUTINE LUSERnn (
      1    NEL    ,NUPARAM,NUVAR   ,NFUNC   ,IFUNC   ,NPF    ,
      2    TF     ,TIME   ,TIMESTEP,UPARAM    ,RHO    , VOLUME ,
      3    EINT  , NGL, SOUNDSP,VISCMAX,UVAR    ,OFF  ,
      4    SIGY  , PLA  , USERBUF    )
C-----------------------------------------------------------------------
```

The user's material law can be used in isotropic or orthotropic modes.

- With isotropic mode, the directions XX, YY, ... are the global reference frame axis. The old elastoplastic stresses (arrays SIGOXX, SIGOYY, ...) are already rotated to take into account the rigid body rotation.
- With orthotropic mode, the directions are the orthotropic frame axis.

You must use the Fortran float external function `FINTER` (shown below) to get the value Y of the function for the abscissa X.

```
Y=FINTER(IFUNC(I),X,NPF,TF,DYDX)
```

where:	**Variable**	**Description**
	Y	Interpolated value
	X	Abscissa value of the function
	I	The i^{th} user's function
	DYDX	Slope
	NPF, TF	Private function parameters

The SOUNDSP array is used in the calculation of the stability time step, the hourglass forces, and the artificial viscous pressure Q.

For isotropic materials, the sound speed value should be equal to the plane wave speed.

For elastic or elastoplastic materials, the sound speed is given by the following equation.

$$c = \sqrt{\frac{K + 4G/3}{\rho_0}} = \sqrt{\frac{\lambda + 2\mu}{\rho_0}}$$

where:	Variable	Description	Equation
	K	Bulk modulus	$K = \frac{E}{3(1-2\upsilon)}$
	G	Shear modulus	$G = \mu = \frac{E}{2(1+\upsilon)}$
	λ and μ	Lame parameters	$\lambda + 2\mu = \frac{E(1-\upsilon)}{(1+\upsilon)(1-2\upsilon)}$

Use VISCMAX to calculate the time step stability when the material law formulation is viscous.

Argument	**Format**	**Description**
NEL	Integer read only scalar	Number of elements per group. In RADIOSS Engine subroutines, the element data are treated by groups for vectorization. This argument is machine-dependent and set by RADIOSS.
NUPARAM	Integer read only scalar	Size of the user parameter array.
NUVAR	Integer read only scalar	Number of user element variables.
NFUNC	Integer read only scalar	Number of functions used for material law.
IFUNC	Integer array read only	Array of size NFUNC with function indexes.
NPF	Integer array private data	Array used by FINTER (float external function).
TF	Integer array private data	Array used by FINTER (float external function).
TIME	Float read only	Current time.
TIMESTEP	Float read only	Current time step.
UPARAM	Float array read only	User material parameter array of size NUPARAM.
RHO	Float array read only	Array of size NEL containing current densities.
VOLUME	Float array read only	Array of size NEL containing current element volumes.
EINT	Float array read only	Array of size NEL containing total internal energy.
SOUNDSP	Float array write only	Array of size NEL containing sound speed.
VISCMAX	Float array write only	Array of size NEL containing the maximum damping modulus.
UVAR	Float array read write	Array of size NEL*NUVAR containing user element variables.

Argument	Format	Description
OFF	Float array read write	Array of size NEL containing deleted element flags. The value is 0 if the element if OFF; the value is 1 if the element is ON.
USERBUF	Data structure read write	Defined in LAW_USERSO.mod. This module is used inside LUSERnn.

This **USERBUF** data structure contains:

Argument	Format	Description
ID	Integer read only scalar	Material ID defined in Starter input deck.
NCYCLE	Integer read only	Cycle number. First cycle is equal to 0.
IPTR	Integer read only	Integration point (direction r).
IPTS	Integer read only	Integration point (direction r).
IPTT	Integer read only	Integration point (direction r).
EPSPXX, EPSPYY, EPSPZZ, EPSPXY, EPSPYZ, EPSPZX	Float array read only	Arrays of size NEL containing ε strain rates in directions XX, YY, and ZZ and γ strain rates in directions XY, YZ, ZX.
DEPSXX, DEPSYY, DEPSZZ, DEPSXY, DEPSYZ, DEPSZX	Float array read only	Arrays of size NEL containing ε strain increments in directions XX, YY, and ZZ and γ strain increments in directions XY, YZ, and ZX.
EPSXX, EPSYY, EPSZZ, EPSXY, EPSYZ, EPSZX	Float array read only	Array of size NEL containing ε strains in directions XX, YY, and ZZ and γ strains in directions XY, YZ, and ZX.
SIGOXX, SIGOYY, SIGOZZ, SIGOXY, SIGOYZ, SIGOZX	Float array read only	Array of size NEL with old (previous time step) elastoplastic stresses in directions XX, YY, ZZ, XY, YZ, and ZX.
R11, R12, R13, R21, R22, R23, R32, R33	Float array write only	Array of size NEL containing rotation matrices from the global skew system to an element skew system.
SIGNXX, SIGNYY, SIGNZZ, SIGNXY, SIGNYZ, SIGNZX	Float array write only	Array of size NEL with new computed elastoplastic stresses in directions XX, YY, ZZ, XY, YZ, and ZX.
SIGVXX, SIGVYY, SIGVZZ, SIGVXY, SIGVYZ, SIGVZX	Float array write only	Array of size NEL containing viscous stresses in directions XX, YY, ZZ, XY, YZ, and ZX.
RHO0	Float array read only	Array of size NEL containing initial densities.
DPLA	Float array write only	Array of size NEL containing incremental plastic strain.
TEMP	Float array read only	Array of size NEL containing temperature.

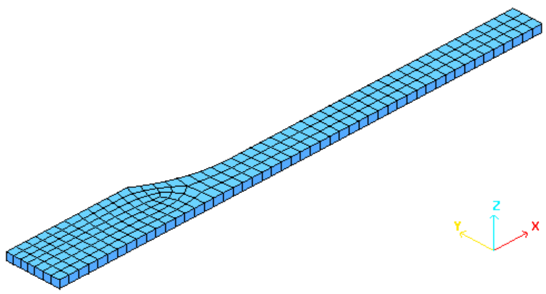
3.2.1 Additional Data Necessary for Compatibility with HEPH

Argument	Format	Description
PLA	Float array write only	Array of size NEL containing plastic strain.
SIGY	Float array read only	Array of size NEL containing yield stress.

3.3 Example of User Material Law for Solid Elements

Example: An elastic material law is defined for solid elements. Input user data includes density, Young's modulus, and Poisson ratio.

This model is made of solid elements:



3.3.1 User's Input Data (/MAT/USERnn/ option)

```
[…]
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
/MAT/USER01/2
user's elastic material law
#             RHO
           0.0027
#               E               Nu
            60400             0.33
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
 […]
```

3.3.2 Starter User's Subroutine LECMUSER01

```
C=================================================================
C    This subroutine reads the user material parameters.
C=================================================================
SUBROUTINE LECMUSER01(IIN,IOUT,UPARAM,MAXUPARAM,NUPARAM,
    .              NUVAR,IFUNC,MAXFUNC,NFUNC,STIFINT,
    .              USERBUF)
     USE LAW_USER
C-----------------------------------------------
C   I m p l i c i t   T y p e s
C-----------------------------------------------
C     IMPLICIT NONE
C-----------------------------------------------
C   D u m m y   A r g u m e n t s
C-----------------------------------------------
     INTEGER IIN,IOUT,MAXUPARAM,NUPARAM,NUVAR,MAXFUNC,NFUNC,
    .       IFUNC(MAXFUNC)
     DOUBLE PRECISION   UPARAM(MAXUPARAM),STIFINT(100)
C-----------------------------------------------
TYPE(ULAWBUF) :: USERBUF
```

```
C-----------------------------------------------
C   L o c a l   V a r i a b l e s
C-----------------------------------------------
      DOUBLE PRECISION E,NU,A11,A12,A44
C
C=======================================
C     ELASTIC LAW WITH SOLIDS
C=======================================
C
C-----------------------------------------------
C     INPUT FILE READING (USER DATA)
C-----------------------------------------------
      READ(IIN,|(2F20.0)|)E,NU
      A11 = E * (1.-NU) / (1.+NU) / (1.-2.*NU)
      A12 = E * NU / (1.+NU) / (1.-2.*NU)
      A44 = E / 2. / (1.+NU)
C
C-----------------------------------------------
C     DATA CHECKING
C-----------------------------------------------
      IF(NU.LT.0.0.OR.NU.GE.0.5)THEN
      WRITE(IOUT,*)' ** ERROR : WRONG NU VALUE'
      ENDIF
      NUPARAM = 3
      IF(NUPARAM.GT.MAXUPARAM)THEN
      WRITE(IOUT,*)' ** ERROR : NUPARAM GT MAXUPARAM'
        WRITE(IOUT,*)'      NUPARAM =',NUPARAM,
     .              ' MAXUPARAM =',MAXUPARAM
      ELSE
C-----------------------------------------------
C     USER MATERIAL PARAMETERS DEFINITION
C-----------------------------------------------
C used in sigeps29 (solid 2d,3d)
      UPARAM(1) = A11
      UPARAM(2) = A12
      UPARAM(3) = A44
      ENDIF
C
C-------------------------------------------------
C     NUMBER OF USER ELEMENT VARIABLES AND CURVES
C-------------------------------------------------
      NUVAR = 0
      NFUNC = 0
C
C-----------------------------------------------
C     USED FOR SOLIDS
C-----------------------------------------------
C used for interface (solid+shell)
      STIFINT = A11
C
C-------------------------------------------------
C     OUTPUT FILE PRINT
C-------------------------------------------------
      WRITE(IOUT,1000)
      WRITE(IOUT,1100)E,NU
C
 1000 FORMAT(
     & 5X,' ELASTIC USER LAW 29',/,
     & 5X,'  ----------------- ',//)
```

```
 1100 FORMAT(
    & 5X,'E . . . . . . . . . . . . . . . . . . . .=',E12.4/
    & 5X,'NU. . . . . . . . . . . . . . . . . . . .=',E12.4//)
C
C-----------------------------------------------
     RETURN
     END
```

3.3.3 Engine User's Subroutine LUSER01

```
C=================================================================
C    This subroutine computes elastic stresses.
C=================================================================
SUBROUTINE LUSER01 (
    1    NEL    ,NUPARAM,NUVAR   ,NFUNC   ,IFUNC   ,NPF    ,
    2    TF     ,TIME   ,TIMESTEP,UPARAM  ,RHO     ,VOLUME ,
    3    EINT   ,NGL    ,SOUNDSP ,VISCMAX ,UVAR    ,OFF    ,
    4    SIGY   ,PLA    ,USERBUF)
C -----------------------------------------------------------------
     USE LAW_USERSO
C
C   INPUT DATA
C
     INTEGER NEL, NUPARAM, NUVAR,NGL(NEL)
     DOUBLE PRECISION
    .  TIME,TIMESTEP,UPARAM(NUPARAM),
    .  RHO(NEL),VOLUME(NEL),EINT(NEL),
    .  EPSPXX(NEL),EPSPYY(NEL),EPSPZZ(NEL),
    .  EPSPXY(NEL),EPSPYZ(NEL),EPSPZX(NEL),
    .  DEPSXX(NEL),DEPSYY(NEL),DEPSZZ(NEL),
    .  DEPSXY(NEL),DEPSYZ(NEL),DEPSZX(NEL),
    .  EPSXX(NEL) ,EPSYY(NEL) ,EPSZZ(NEL),
    .  EPSXY(NEL) ,EPSYZ(NEL) ,EPSZX(NEL),
    .  SIGOXX(NEL),SIGOYY(NEL),SIGOZZ(NEL),
    .  SIGOXY(NEL),SIGOYZ(NEL),SIGOZX(NEL),
    .  RHO0(NEL)
C----------------------------------------------
C  O U T P U T  DATA
C----------------------------------------------
     DOUBLE PRECISION
    .   SOUNDSP(NEL),VISCMAX(NEL),
    .   SIGNXX(NEL),SIGNYY(NEL),SIGNZZ(NEL),
    .   SIGNXY(NEL),SIGNYZ(NEL),SIGNZX(NEL),
    .   SIGVXX(NEL),SIGVYY(NEL),SIGVZZ(NEL),
    .   SIGVXY(NEL),SIGVYZ(NEL),SIGVZX(NEL),
    .   DPLA(NEL)
C----------------------------------------------
C  I N P U T  O U T P U T  A r g u m e n t s
C----------------------------------------------
     DOUBLE PRECISION
    . UVAR(NEL,NUVAR), OFF(NEL),PLA(NEL), SIGY(NEL)
C----------------------------------------------
     TYPE(ULAWINTBUF) :: USERBUF
C----------------------------------------------
     INTEGER NPF(*), NFUNC, IFUNC(NFUNC)
     DOUBLE PRECISION
    . FINTER ,TF(*)
     EXTERNAL FINTER
C       Y = FINTER(IFUNC(J),X,NPF,TF,DYDX)
```

```
C       Y      : y = f(x)
C       X      : x
C       DYDX    : f'(x) = dy/dx
C       IFUNC(J): FUNCTION INDEX
C            J : FIRST(J=1), SECOND(J=2) .. FUNCTION USED FOR THIS LAW
C       NPF,TF  : FUNCTION PARAMETER
C ---------------------------------------------
C----------------------------------------------
C   L o c a l   V a r i a b l e s
C----------------------------------------------
     INTEGER I,J
     DOUBLE PRECISION
    .   A11,A12,G
C----------------------------------------------
C    USER VARIABLES INITIALIZATION
C----------------------------------------------
       A11      = UPARAM(1)
       A12      = UPARAM(2)
       G       = UPARAM(3)
C    Input Data structure
     SIGOXX = USERBUF%SIGOXX
     SIGOYY = USERBUF%SIGOYY
     SIGOZZ = USERBUF%SIGOZZ
     SIGOXY = USERBUF%SIGOXY
     SIGOYZ = USERBUF%SIGOYZ
     SIGOZX = USERBUF%SIGOZX
C
     EPSPXX = USERBUF%EPSPXX
     EPSPYY = USERBUF%EPSPYY
     EPSPZZ = USERBUF%EPSPZZ
     EPSPXY = USERBUF%EPSPXY
     EPSPYZ = USERBUF%EPSPYZ
     EPSPZX = USERBUF%EPSPZX
C
     EPSXX = USERBUF%EPSXX
     EPSYY = USERBUF%EPSYY
     EPSZZ = USERBUF%EPSZZ
     EPSXY = USERBUF%EPSXY
     EPSYZ = USERBUF%EPSYZ
     EPSZX = USERBUF%EPSZX
C
     DEPSXX = USERBUF%DEPSXX
     DEPSYY = USERBUF%DEPSYY
     DEPSZZ = USERBUF%DEPSZZ
     DEPSXY = USERBUF%DEPSXY
     DEPSYZ = USERBUF%DEPSYZ
     DEPSZX = USERBUF%DEPSZX
C
     SIGNXX = USERBUF%SIGNXX
     SIGNYY = USERBUF%SIGNYY
     SIGNZZ = USERBUF%SIGNZZ
     SIGNXY = USERBUF%SIGNXY
     SIGNYZ = USERBUF%SIGNYZ
     SIGNZX = USERBUF%SIGNZX
C
     SIGVXX = USERBUF%SIGVXX
     SIGVYY = USERBUF%SIGVYY
     SIGVZZ = USERBUF%SIGVZZ
     SIGVXY = USERBUF%SIGVXY
```

```
      SIGVYZ = USERBUF%SIGVYZ
      SIGVZX = USERBUF%SIGVZX
      RHO0   = USERBUF%RHO0
      DPLA   = USERBUF%DPLA
C
C --- Trial stress
C
      DO I = 1,NEL
      SIGNXX(I)=SIGOXX(I) + A11*DEPSXX(I)
     .+ A12*(DEPSYY(I) + DEPSZZ(I))
      SIGNYY(I)=SIGOYY(I) + A11*DEPSYY(I)
     .+ A12*(DEPSXX(I) + DEPSZZ(I))
      SIGNZZ(I)=SIGOZZ(I) + A11*DEPSZZ(I)
     .+ A12*(DEPSYY(I) + DEPSXX(I))
      SIGNXY(I)=SIGOXY(I) + G*DEPSXY(I)
      SIGNYZ(I)=SIGOYZ(I) + G*DEPSYZ(I)
      SIGNZX(I)=SIGOZX(I) + G*DEPSZX(I)
C sound velocity
      SOUNDSP(I) = SQRT(A11/RHO0(I))
      VISCMAX(I) = ZERO
      ENDDO
C     Outp data structure
      USERBUF%SIGNXX = SIGNXX
      USERBUF%SIGNYY = SIGNYY
      USERBUF%SIGNZZ = SIGNZZ
      USERBUF%SIGNXY = SIGNXY
      USERBUF%SIGNYZ = SIGNYZ
      USERBUF%SIGNZX = SIGNZX
C
      USERBUF%SIGVXX = SIGVXX
      USERBUF%SIGVYY = SIGVYY
      USERBUF%SIGVZZ = SIGVZZ
      USERBUF%SIGVXY = SIGVXY
      USERBUF%SIGVYZ = SIGVYZ
      USERBUF%SIGVZX = SIGVZX
      USERBUF%DPLA  =  DPLA
      RETURN
      END
```

3.4 Engine Subroutine LUSERnnC for Shell Elements

This subroutine calculates the stress tensor versus the strain tensor, strain rate tensor, or user variables.

The argument list of LUSERnnC and its individual arguments and descriptions are as follows:

```
C-----------------------------------------------------------------------------
         SUBROUTINE LUSERnnC(
        1 NEL,NUPARAM,NUVAR,NFUNC,IFUNC,
        2 NPF,NGL,TF,TIME,TIMESTEP,
        3 UPARAM,RHO0,AREA,EINT,SHF,
        4 SOUNDSP,VISCMAX,PLA,UVAR,OFF,
        5 SIGY,USERBUF)
C----------------------------------------------------------------------------
```

The user's material law can be used in isotropic mode with PID 1 or in orthotropic mode with PID 9, 10, or 11. The directions XX, YY, ... are the shell local reference frame axis. Stresses are computed at each integration point.

Use the Fortran float external function `FINTER` to get the value Y of the function for the abscissa X.

```
Y=FINTER(IFUNC(I),X,NPF,TF,DYDX)
```

where:	Variable	Description
	Y	Interpolated value
	X	Abscissa value of the function
	I	The i^{th} user's function
	DYDX	Slope
	NPF, TF	Private function parameters

The SOUNDSP array should always be set by you. It is used to calculate the stability time step and the hourglass forces. The sound speed value should be equal to the plane wave speed.

For elastic or elastoplastic materials, the sound speed is given by the following equation.

$$c = \sqrt{\frac{E}{\left(1-\upsilon^2\right)\rho_0}}$$

Use VISCMAX to calculate the time step stability when the material law formulation is viscous.

Argument	**Format**	**Description**
NEL	Integer read only scalar	Number of elements per group. In RADIOSS Engine subroutines, the element data are treated by groups for vectorization. This argument is machine-dependent and set by RADIOSS.
NUPARAM	Integer read only scalar	Size of the user parameter array.
NUVAR	Integer read only scalar	Number of user element variables.
NFUNC	Integer read only scalar	Number of functions used for material law.
IFUNC	Integer array read only	Array of size NFUNC with function indexes.
NPF	Integer array private data	Array used by `FINTER` (float external function).
TF	Integer array private data	Array used by `FINTER` (float external function).
IPT	Integer read only scalar	Current layer or interrogation point.
TIME	Float read only	Current time.
TIMESTEP	Float read only	Current time step.
UPARAM	Float array read only	User material parameter array of size NUPARAM.
RHO0	Float array read only	Array of size NEL with initial densities.
AREA	Float array read only	Array of size NEL with current element surfaces.
EINT	Float array read only	Array of size 2*NEL with internal membrane and bending energy.
SOUNDSP	Float array write only	Array of size NEL containing sound speed.
VISCMAX	Float array write only	Array of size NEL containing the maximum damping modulus.
PLA	Float array read write	Array of size NEL containing plastic strain.
UVAR	Float array read write	Array of size NEL*NUVAR containing user element variables.
OFF	Float array read write	Array of size NEL containing deleted element flags. The value is 0 if the element if OFF; the value is 1 if the element is ON.
NGL	Integer array read only	Array of size NEL containing the external element number.
SIGY	Float array read write	Array of size NEL containing yield stress.
USERBUF	Data structure read writer	Defined in LAW_USERSH.mod. This module should be used to relink the executable inside LUSERnnC.

This USERBUF module contains:

Argument	Format	Description
NCYCLE	Integer read only scalar	Current cycle.
ID	Integer read only scalar	User material ID.
ILAYER	Integer read only scalar	Current cycle.
IG	Integer read only scalar	Integration point (surface).
NPTA	Integer read only scalar	Number of layers or integration points.
IFLAG	Integer array read only scalar	Array of size NEL containing geometric flags.
R11, R12, R13, R21, R22, R23, R31, R32, R33	Float arrays read only scalar	
Rotational matrix		Global ----- > local.
THKLYL	Float array read only	Array of size NEL containing the layer thickness at each integration point.
THKN	Float array read write	Array of size NEL containing total thickness.
EPSPXX, EPSPYY, EPSPXY, EPSPYZ, EPSPZX	Float array read only	Arrays of size NEL containing ε strain rates in directions XX and YY and γ strains in directions XY, YZ, and ZX.
DEPSXX, DEPSYY, DEPSXY, DEPSYZ, DEPSZX	Float array read only	Arrays of size NEL containing ε strain increments in directions XX and YY and γ strain increments in directions XY, YZ, and ZX.
EPSXX, EPSYY, EPSXY, EPSYZ, EPSZX	Float array read only	Array of size NEL containing ε strains in directions XX and YY and γ strains in directions XY, YZ, and ZX.
SIGOXX, SIGOYY, SIGOZZ, SIGOXY, SIGOYZ, SIGOZX	Float array read only	Array of size NEL containing old (previous time step) elastoplastic stresses in directions XX, YY, ZZ, XY, YZ, and ZX.
SIGNXX, SIGNYY, SIGNXY, SIGNYZ, SIGNZX	Float array write only	Array of size NEL containing new computed elastoplastic stresses in directions XX, YY, XY, YZ, and ZX.
SIGVXX, SIGVYY, SIGVZZ, SIGVXY, SIGVYZ, SIGVZX	Float array write only	Array of size NEL containing viscous stresses in directions XX, YY, XY, YZ, and ZX.

DPLA	Float array writ only	Array of size NEL containing the plastic strain.
ETSE	Float array write only	Array of size NEL containing Et/E (tangent modulus divided by Young's modulus).
TEMP	Float array read only	Array of size NEL containing the temperature.

3.4.1 Shell Element Law Output

Unlike solid elements, shell elements do not have any variables specific to user's law that are saved in time-history and animation.

3.4.2 Additional Data Necessary for Compatibility with QEPH

The Yield value and value Et/E (tangent modulus divided by Young modulus) must be given in order for this law to be compatible with QEPH element. The prototype of this routine and the necessary data to provide are described below.

3.5 Example of User's Material Law for Shell Elements

Example: A Johnson-Cook elasto-plastic material law is defined for shell elements. Input user data includes density, Young's modulus, Poisson ratio, yield stress, hardening parameters, hardening modulus, maximum stress, and maximum strain.

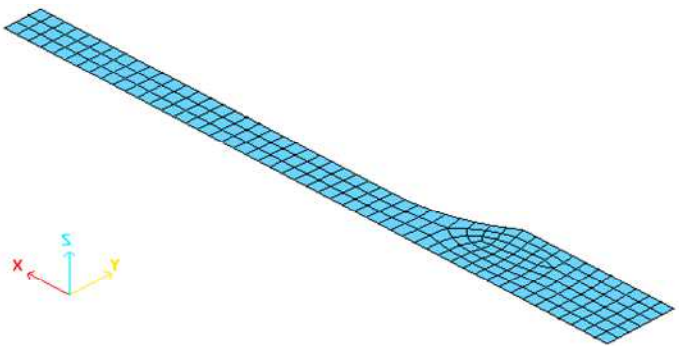
The Johnson Cook model: $\sigma = A + B\varepsilon_p^N$

Maximum stress and plastic strain are taken into account. Shell thickness is variable.

Two methods are available to compute plastically admissible stresses.

- Projection by return radial
- Iterative projection with three Newton iterations

This mesh example is made of shell elements:



3.5.1 User's Input Data (/MAT/USERnn/ option)

```
[…]
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
/MAT/USER01/1
user's elasto-plastic material law
#            RHO
          0.0027
#              E               Nu
           60400             0.33
#              A                B                N            EPSM              SIGM
          90.266           223.14            0.375               0               175
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
[…]
```

3.5.2 Starter User's Subroutine LECMUSER01

```
C=================================================================
C    This subroutine reads the user material parameters.
C=================================================================
      SUBROUTINE LECMUSER01(IIN  ,IOUT ,UPARAM ,MAXUPARAM,NUPARAM,
     .                   NUVAR,IFUNC,MAXFUNC,NFUNC ,PARMAT )
C-----------------------------------------------
C   I m p l i c i t   T y p e s
```

```
C-----------------------------------------------
      IMPLICIT NONE
C-----------------------------------------------
C   D u m m y   A r g u m e n t s
C-----------------------------------------------
      INTEGER IIN,IOUT,MAXUPARAM,NUPARAM,NUVAR,MAXFUNC,NFUNC,
     .        IFUNC(MAXFUNC)
      DOUBLE PRECISION   UPARAM(MAXUPARAM),PARMAT(*)
C
C-----------------------------------------------
C   L o c a l   V a r i a b l e s
C-----------------------------------------------
      DOUBLE PRECISION E,NU,CA,CB,CN,EPSM,SIGM,G
C
C===============================================
C     ELASTO-PLASTIC LAW (Y=A+B*PLA^N)
C===============================================
C
C-----------------------------------------------
C     INPUT FILE READING (USER DATA)
C-----------------------------------------------
      READ(IIN,'(2F20.0)')E,NU
      READ(IIN,'(5F20.0)')CA,CB,CN,EPSM,SIGM
C
C-----------------------------------------------
C     DATA CHECKING
C-----------------------------------------------
      IF(NU.LT.0.0.OR.NU.GE.0.5)THEN
        WRITE(IOUT,*)' ** ERROR : WRONG NU VALUE'
      ENDIF
      IF(CN.EQ.0.0.OR.CN.EQ.1.)   CN = 1.0001
      IF(EPSM.EQ.0.) EPSM  = 1.E+30
      IF(SIGM.EQ.0.) SIGM  = 1.E+30
      NUPARAM = 10
      IF(NUPARAM.GT.MAXUPARAM)THEN
        WRITE(IOUT,*)' ** ERROR : NUPARAM GT MAXUPARAM'
        WRITE(IOUT,*)'      NUPARAM =',NUPARAM,
     .               ' MAXUPARAM =',MAXUPARAM
      ELSE
C-----------------------------------------------
C     USER MATERIAL PARAMETERS DEFINITION
C-----------------------------------------------
        UPARAM(1) = E
        UPARAM(2) = NU
        G = 0.5*E/(1.+NU)
        UPARAM(3) = G
        UPARAM(4) = CA
        UPARAM(5) = CB
        UPARAM(6) = CN
        UPARAM(7) = EPSM
        UPARAM(8) = SIGM
        UPARAM(9) = E/(1.-NU*NU)
        UPARAM(10) = NU*E/(1.-NU*NU)
      ENDIF
C
C-----------------------------------------------
C     USED FOR SHELLS
C-----------------------------------------------
C     PARMAT(1) = C1 (interface for solid)
```

```
      PARMAT(2) = E
      PARMAT(3) = NU
C
C-----------------------------------------------
C     NUMBER OF USER ELEMENT VARIABLES AND CURVES
C-----------------------------------------------
      NUVAR = 4
      NFUNC = 0
C
C-----------------------------------------------
C     OUTPUT FILE PRINT
C-----------------------------------------------
      WRITE(IOUT,1000)
      WRITE(IOUT,1100)E,NU,G,
     .              CA,CB,CN,EPSM,SIGM
C
 1000 FORMAT(
     & 5X,'   ELASTO-PLASTIC LAW (SIG=A+B*EPSp^N) ',/,
     & 5X,'  ------------------------------------'//)
 1100 FORMAT(
     & 5X,'YOUNG MODULUS.  . . . . . . . . . . . . .=',E12.4/
     & 5X,'POISSON RATIO.  . . . . . . . . . . . . .=',E12.4/
     & 5X,'SHEAR MODULUS . . . . . . . . . . . . . .=',E12.4/
     & 5X,'YIELD COEFFICIENT A . . . . . . . . . . .=',E12.4/
     & 5X,'YIELD COEFFICIENT B . . . . . . . . . . .=',E12.4/
     & 5X,'YIELD COEFFICIENT N . . . . . . . . . . .=',E12.4/
     & 5X,'EPS-MAX . . . . . . . . . . . . . . . . .=',E12.4/
     & 5X,'SIG-MAX  . . . . . . . . . . . . . . . . =',E12.4//)
C
C-----------------------------------------------
C     END
C-----------------------------------------------
      RETURN
      END
```

3.5.3 Engine User's Subroutine LUSER01C

```
     SUBROUTINE LUSER01C(
    1    NEL    ,NUPARAM,NUVAR   ,NFUNC  ,IFUNC    ,
    2    NPF    , NGL   , TF     ,TIME   ,TIMESTEP ,
    3    UPARAM ,RHO0   , AREA   ,EINT   ,SHF      ,
    4    SOUNDSP,VISCMAX, PLA    ,UVAR   , OFF     ,
    5    SIGY     ,USERBUF )
C
     USE LAW_USERSH
C
C-----------------------------------------------
C  I m p l i c i t   T y p e s
C-----------------------------------------------
     IMPLICIT NONE
C-----------------------------------------------
C  I N P U T-OUTP DATA structure
C-----------------------------------------------
C
```

```
     TYPE(ULAWCINTBUF) :: USERBUF
C
C-----------------------------------------------
C   I N P U T  DATA
C-----------------------------------------------
     INTEGER NEL, NUPARAM, NUVAR, NPT, IPT,IFLAG,
    .    NGL(NEL)
     DOUBLE PRECISION
    .    TIME,TIMESTEP,UPARAM(NUPARAM),
    .    AREA(NEL),RHO0(NEL),EINT(2,NEL),
    .    THKLY(NEL),PLA(NEL),SHF(NEL),
    .    EPSPXX(NEL),EPSPYY(NEL),
    .    EPSPXY(NEL),EPSPYZ(NEL),EPSPZX(NEL),
    .    DEPSXX(NEL),DEPSYY(NEL),
    .    DEPSXY(NEL),DEPSYZ(NEL),DEPSZX(NEL),
    .    EPSXX(NEL) ,EPSYY(NEL) ,
    .    EPSXY(NEL) ,EPSYZ(NEL) ,EPSZX(NEL) ,
    .    SIGOXX(NEL),SIGOYY(NEL),
    .    SIGOXY(NEL),SIGOYZ(NEL),SIGOZX(NEL)
C-----------------------------------------------
C   O U T P U T  DATA
C-----------------------------------------------
     DOUBLE PRECISION
    .    SIGNXX(NEL),SIGNYY(NEL),
    .    SIGNXY(NEL),SIGNYZ(NEL),SIGNZX(NEL),
    .    SIGVXX(NEL),SIGVYY(NEL),
    .    SIGVXY(NEL),SIGVYZ(NEL),SIGVZX(NEL),
    .    SOUNDSP(NEL),VISCMAX(NEL)
C-----------------------------------------------
C   I N P U T   O U T P U T   A r g u m e n t s
C-----------------------------------------------
     DOUBLE PRECISION UVAR(NEL,NUVAR),OFF(NEL),THK(NEL)
C-----------------------------------------------
C   VARIABLES FOR FUNCTION INTERPOLATION
C-----------------------------------------------
     INTEGER NPF(*), NFUNC, IFUNC(NFUNC)
     DOUBLE PRECISION FINTER ,TF(*)
     EXTERNAL FINTER
C       Y = FINTER(IFUNC(J),X,NPF,TF,DYDX)
C       Y      : y = f(x)
C       X      : x
C       DYDX    : f’(x) = dy/dx
C       IFUNC(J): FUNCTION INDEX
C            J : FIRST(J=1), SECOND(J=2)
C       NPF,TF  : FUNCTION PARAMETER
```

```
C-----------------------------------------------
C  L o c a l   V a r i a b l e s
C-----------------------------------------------
      INTEGER I,J,INDEX(NEL),NMAX,N,NINDX,IPLAS
      DOUBLE PRECISION
     .       E,NU,G,CA,CB,CN,EPSM,SIGM,
     .       A1,A2,G3,
     .       CH1,QH1,
     .       NNU1,NU1,S1,S2,S3,
     .       R,RR,UMR,DEZZ,UN,EM20,ZERO,
     .       L,M,
     .       S11,S22,P2,S1S2,S122,NNU2,NU4,NU6,
     .       C,S12,F,DF,Q2,YLD_I,NU3,NU2
      DOUBLE PRECISION
     .       SVM(NEL),AA(NEL),BB(NEL),PP(NEL),QQ(NEL),DPLA(NEL),
     .       X1(NEL),Y1(NEL),Z1(NEL),SVM1(NEL),
     .       A(NEL),VM2(NEL),DPLA_J(NEL),DR(NEL),ETSE(NEL),SIGY(NEL)
C
      DATA ZERO/0.0/,UN/1.0/,NMAX/3/,EM20/1.E-20/
C
C===============================================
C
C     ELASTO-PLASTIC LAW (Y=A+B*PLA^N)
C
C===============================================
C
C-----------------------------------------------
C     PARAMETERS READING
C-----------------------------------------------
      E = UPARAM(1)
      NU = UPARAM(2)
      G = UPARAM(3)
      CA = UPARAM(4)
      CB = UPARAM(5)
      CN = UPARAM(6)
      EPSM = UPARAM(7)
      SIGM = UPARAM(8)
      A1 = UPARAM(9)
      A2 = UPARAM(10)
C
C-----------------------------------------------
C     USER VARIABLES INITIALIZATION
C-----------------------------------------------
      IF(TIME.EQ.0.0)THEN
        DO I=1,NEL
```

```
          UVAR(I,1)=0.
          UVAR(I,2)=0.
          UVAR(I,3)=0.
          UVAR(I,4)=0.
        ENDDO
      ENDIF
C
      G3 = 3. * G
      NNU1 = NU / (1. - NU)
      NU1 = 1.-NNU1
      NU2 = 1./(1.+NU)
      NU3 = 1./(1.-NU)
C
C   input data structure
C
      IPT= USERBUF%ILAYER
      NPT = USERBUF%NPTA
      IPLAS =USERBUF%IFLAG
C
      SIGOXX(1:NEL) = USERBUF%SIGOXX(1:NEL)
      SIGOYY(1:NEL) = USERBUF%SIGOYY(1:NEL)
      SIGOXY(1:NEL) = USERBUF%SIGOXY(1:NEL)
      SIGOYZ(1:NEL) = USERBUF%SIGOYZ(1:NEL)
      SIGOZX(1:NEL) = USERBUF%SIGOZX(1:NEL)
C
      EPSPXX(1:NEL) = USERBUF%EPSPXX(1:NEL)
      EPSPYY(1:NEL) = USERBUF%EPSPYY(1:NEL)
      EPSPXY(1:NEL) = USERBUF%EPSPXY(1:NEL)
      EPSPYZ(1:NEL) = USERBUF%EPSPYZ(1:NEL)
      EPSPZX(1:NEL) = USERBUF%EPSPZX(1:NEL)
C
      EPSXX(1:NEL) = USERBUF%EPSXX(1:NEL)
      EPSYY(1:NEL) = USERBUF%EPSYY(1:NEL)
      EPSXY(1:NEL) = USERBUF%EPSXY(1:NEL)
      EPSYZ(1:NEL) = USERBUF%EPSYZ(1:NEL)
      EPSZX(1:NEL) = USERBUF%EPSZX(1:NEL)
C
      DEPSXX(1:NEL) = USERBUF%DEPSXX(1:NEL)
      DEPSYY(1:NEL) = USERBUF%DEPSYY(1:NEL)
      DEPSXY(1:NEL) = USERBUF%DEPSXY(1:NEL)
      DEPSYZ(1:NEL) = USERBUF%DEPSYZ(1:NEL)
      DEPSZX(1:NEL) = USERBUF%DEPSZX(1:NEL)
      THKLY(1:NEL)  = USERBUF%THKLYL(1:NEL)
      THK(1:NEL)    = USERBUF%THKN(1:NEL)
C       initialisation
```

```
      SIGNXX(1:NEL) = USERBUF%SIGNXX(1:NEL)
      SIGNYY(1:NEL) = USERBUF%SIGNYY(1:NEL)
      SIGNXY(1:NEL) = USERBUF%SIGNXY(1:NEL)
      SIGNYZ(1:NEL) = USERBUF%SIGNYZ(1:NEL)
      SIGNZX(1:NEL) = USERBUF%SIGNZX(1:NEL)
C
      SIGVXX(1:NEL) = USERBUF%SIGVXX(1:NEL)
      SIGVYY(1:NEL) = USERBUF%SIGVYY(1:NEL)
      SIGVXY(1:NEL) = USERBUF%SIGVXY(1:NEL)
      SIGVYZ(1:NEL) = USERBUF%SIGVYZ(1:NEL)
      SIGVZX(1:NEL) = USERBUF%SIGVZX(1:NEL)
      ETSE(1:NEL)   = USERBUF%ETSE(1:NEL)
      DPLA(1:NEL)   = USERBUF%DPLA(1:NEL)
C=========================================================
C     I - ELASTIC STRESSES COMPUTATION
C=========================================================
      DO I=1,NEL
C
        SIGNXX(I)=SIGOXX(I)+A1*DEPSXX(I)+A2*DEPSYY(I)
        SIGNYY(I)=SIGOYY(I)+A2*DEPSXX(I)+A1*DEPSYY(I)
        SIGNXY(I)=SIGOXY(I)+G *DEPSXY(I)
        SIGNYZ(I)=SIGOYZ(I)+G *DEPSYZ(I)
        SIGNZX(I)=SIGOZX(I)+G *DEPSZX(I)
C
        SOUNDSP(I) = SQRT(A1/RHO0(I))
        VISCMAX(I) = 0.
      ENDDO
C
C=========================================================
C     II - ELASTO-PLASTIC COMPUTATION
C=========================================================
C
C================================================
C     A - COMPUTE CURRENT YIELD STRESS
C================================================
      DO I=1,NEL
        IF(UVAR(I,1).LE.0.) THEN
          CH1=CA
        ELSEIF(UVAR(I,1).GT.EPSM) THEN
          CH1=CA+CB*EPSM**CN
        ELSE
          CH1=CA+CB*UVAR(I,1)**CN
        ENDIF
        UVAR(I,2)=MIN(SIGM,CH1)
      ENDDO
```

```
C
C===============================================
C     B- COMPUTE HARDENING MODULUS H
C===============================================
      DO I=1,NEL
        IF(UVAR(I,1).GT.0. AND .CN.GE.1) THEN
          QH1= CB*CN*UVAR(I,1)**(CN-1.)
        ELSEIF(UVAR(I,1).GT.0. AND .CN.LT.1)THEN
          QH1= CB*CN*UVAR(I,1)**(1.-CN)
        ELSE
          QH1=0.
        ENDIF
        UVAR(I,3)=QH1
      ENDDO
C
C
C==============================================================
C     C - STRESSES, PLASTIC STRAIN AND THICKNESS CALCULATION
C
C     COMPUTE PLASTICALLY ADMISSIBLE STRESSES
C     Two available computations according to IPLAS flag
C==============================================================
C
      IF(IPLAS.EQ.0)THEN
C==============================================================
C     1 - PROJECTION by RADIAL RETURN  (Iplas=0)
C==============================================================
C
C      print *, 'PROJECTION by RADIAL RETURN - Iplas=0'
C
C-----------------------------------------------
C     -> Plastic strain evaluation
C-----------------------------------------------
      DO I=1,NEL
        UVAR(I,1) = 0.5*( EPSXX(I)+EPSYY(I)
     .             + SQRT( (EPSXX(I)-EPSYY(I))*(EPSXX(I)-EPSYY(I))
     .             + EPSXY(I)*EPSXY(I) ) )
      ENDDO
C
C-----------------------------------------------
C     -> Von Mises criterion (non principal stresses)
C-----------------------------------------------
      DO I=1,NEL
        SVM(I)=SQRT(SIGNXX(I)*SIGNXX(I)
     .           +SIGNYY(I)*SIGNYY(I)
```

```
     .          -SIGNXX(I)*SIGNYY(I)
     .        +3.*SIGNXY(I)*SIGNXY(I))
      ENDDO
C
C-----------------------------------------------
C    -> Projection on criterion
C-----------------------------------------------
      DO I=1,NEL
        R = MIN(UN,UVAR(I,2)/MAX(EM20,SVM(I)))
        SIGNXX(I)=SIGNXX(I)*R
        SIGNYY(I)=SIGNYY(I)*R
        SIGNXY(I)=SIGNXY(I)*R
      ENDDO
C
C-----------------------------------------------
C    -> Compute plastic strain
C-----------------------------------------------
      DO  I=1,NEL
        UMR = 1.-R
        DPLA(I) = OFF(I)*SVM(I)*UMR/E
        UVAR(I,1) = UVAR(I,1) + DPLA(I)
        PLA(I) = PLA(I) + DPLA(I)
      ENDDO
C
C-----------------------------------------------
C    -> Compute thickness
C-----------------------------------------------
      DO I=1,NEL
        DEZZ = DPLA(I) * 0.5*(SIGNXX(I)+SIGNYY(I)) /UVAR(I,2)
        DEZZ=-(DEPSXX(I)+DEPSYY(I))*NNU1-NU1*DEZZ
        THK(I) = THK(I) + DEZZ*THKLY(I)
      ENDDO
C
C
      ELSEIF(IPLAS.EQ.1)THEN
C==============================================================
C    2 - ITERATIVE PROJECTION  (Iplas =1 )
C       with 3 Newton iterations
C==============================================================
C
C     print *, 'ITERATIVE PROJECTION - Iplas=1'
C
C-----------------------------------------------
C    -> Von Mises criterion (non principal stresses)
C-----------------------------------------------
```

```
      DO  I=1,NEL
        UVAR(I,3) = MAX(ZERO,UVAR(I,3))
        S1=SIGNXX(I)+SIGNYY(I)
        S2=SIGNXX(I)-SIGNYY(I)
        S3=SIGNXY(I)
        AA(I)=0.25*S1*S1
        BB(I)=0.75*S2*S2+3.*S3*S3
        SVM(I)=SQRT(AA(I)+BB(I))
        DEZZ = -(DEPSXX(I)+DEPSYY(I))*NNU1
        THK(I) = THK(I) + DEZZ*THKLY(I)
      ENDDO
C
C-----------------------------------------------
C     -> Gather plastic flow - Plasticity check
C-----------------------------------------------
      NINDX=0
      DO I=1,NEL
        IF(SVM(I).GT.UVAR(I,2).AND.OFF(I).EQ.1.) THEN
          NINDX=NINDX+1
          INDEX(NINDX)=I
        ENDIF
      ENDDO
      IF(NINDX.EQ.0) GOTO 100
C
C-----------------------------------------------
C     -> Plastic plane stress
C-----------------------------------------------
      DO J=1,NINDX
        I=INDEX(J)
        DPLA_J(I)=(SVM(I)-UVAR(I,2))/(G3+UVAR(I,3))
      ENDDO
C    NMAX: number of iterations = 3
      DO N=1,NMAX
       DO J=1,NINDX
       I=INDEX(J)
       DPLA(I) = DPLA_J(I)
       YLD_I = UVAR(I,2)+UVAR(I,3)*DPLA(I)
       DR(I) = 0.5*E*DPLA(I)/YLD_I
       PP(I) = 1./(1.+DR(I)*NU3)
       QQ(I) = 1./(1.+3.*DR(I)*NU2)
       P2 = PP(I)*PP(I)
       Q2 = QQ(I)*QQ(I)
       F = AA(I)*P2+BB(I)*Q2-YLD_I*YLD_I
       DF = -(AA(I)*NU3*P2*PP(I)+3.*BB(I)*NU2*Q2*QQ(I))
     .      *(E-2.*DR(I)*UVAR(I,3))/YLD_I
```

```
    .     -2.*UVAR(I,3)*YLD_I

     IF(DPLA(I).GT.0.) THEN
      DPLA_J(I)=MAX(ZERO,DPLA(I)-F/DF)
     ELSE
      DPLA_J(I)=0.
     ENDIF
C
     ENDDO
C
     ENDDO
C
C-------------------------------------------
C    -> Plastic strain
C    -> Plastically admissible stresses
C    -> Thickness
C-------------------------------------------
C
     DO J=1,NINDX
     I=INDEX(J)
     UVAR(I,1) = UVAR(I,1) + DPLA(I)
     PLA(I) = UVAR(I,1)
     S1=(SIGNXX(I)+SIGNYY(I))*PP(I)
     S2=(SIGNXX(I)-SIGNYY(I))*QQ(I)
     SIGNXX(I)=0.5*(S1+S2)
     SIGNYY(I)=0.5*(S1-S2)
     SIGNXY(I)=SIGNXY(I)*QQ(I)
     DEZZ = - NU1*DR(I)*S1/E
     THK(I) = THK(I) + DEZZ*THKLY(I)
     ENDDO
C
     ELSEIF(IPLAS.EQ.2)THEN
C==============================================================
C    3 - PROJECTION by RADIAL RETURN with correction (Iplas=2)
C==============================================================
C
C     print *, 'PROJECTION by RADIAL RETURN - Iplas=2'
     DO  I=1,NEL
C
     PP(I) = -(SIGNXX(I)+SIGNYY(I))*0.33333333
     S11 = SIGNXX(I)+PP(I)
     S22 = SIGNYY(I)+PP(I)
     S12 = SIGNXY(I)
     P2 = PP(I)*PP(I)
     S1S2 = S11*S22
```

```
      S122 = S12*S12
C
      NNU2 = NNU1*NNU1
      NU4 = 1 + NNU2 + NNU1
      NU6 = 0.5  - NNU2 + 0.5*NNU1
C
      QQ(I) = (1.-NNU1)*PP(I)
      AA(I) = P2*NU4 + 3.*(S122 - S1S2)
      BB(I) = P2*NU6
      C  = QQ(I)*QQ(I)
      VM2(I)= AA(I)+BB(I)+BB(I)+C
C  = C - UVAR(I,2)*UVAR(I,2)
C
      R = MAX(ZERO,BB(I)*BB(I)-AA(I)*C)
      R  = MIN(UN,(-BB(I)+ SQRT(R))/MAX(AA(I) ,EM20))
C
      UMR = 1 - R
      QQ(I) = QQ(I)*UMR
      SIGNXX(I) = SIGNXX(I)*R - QQ(I)
      SIGNYY(I) = SIGNYY(I)*R - QQ(I)
      SIGNXY(I) = S12*R
      DPLA(I) = OFF(I)*SQRT(VM2(I))*UMR/(G3)
      S1=0.5*(SIGNXX(I)+SIGNYY(I))
      UVAR(I,1) = UVAR(I,1) + DPLA(I)
      PLA(I) = UVAR(I,1)
      DEZZ = DPLA(I) * S1 /UVAR(I,2)
      DEZZ=-(DEPSXX(I)+DEPSYY(I))*NNU1-NU1*DEZZ
      THK(I) = THK(I) + DEZZ*THKLY(I)
C
C-----------------------------------------------
      ENDDO
      ENDIF

 100   CONTINUE


C
C outp data structure

      USERBUF%SIGNXX(1:NEL) = SIGNXX(1:NEL)
      USERBUF%SIGNYY(1:NEL) = SIGNYY(1:NEL)
      USERBUF%SIGNXY(1:NEL) = SIGNXY(1:NEL)
      USERBUF%SIGNYZ(1:NEL) = SIGNYZ(1:NEL)
      USERBUF%SIGNZX(1:NEL) = SIGNZX(1:NEL)
c
```

```
      USERBUF%SIGVXX(1:NEL) = SIGVXX(1:NEL)
      USERBUF%SIGVYY(1:NEL) = SIGVYY(1:NEL)
      USERBUF%SIGVXY(1:NEL) = SIGVXY(1:NEL)
      USERBUF%SIGVYZ(1:NEL) = SIGVYZ(1:NEL)
      USERBUF%SIGVZX(1:NEL) = SIGVZX(1:NEL)
      USERBUF%DPLA(1:NEL)  = DPLA(1:NEL)
      USERBUF%ETSE(1:NEL)  = ETSE(1:NEL)
      USERBUF%THKN(1:NEL)  = THK(1:NEL)
C
C-----------------------------------------------
      RETURN
      END
```