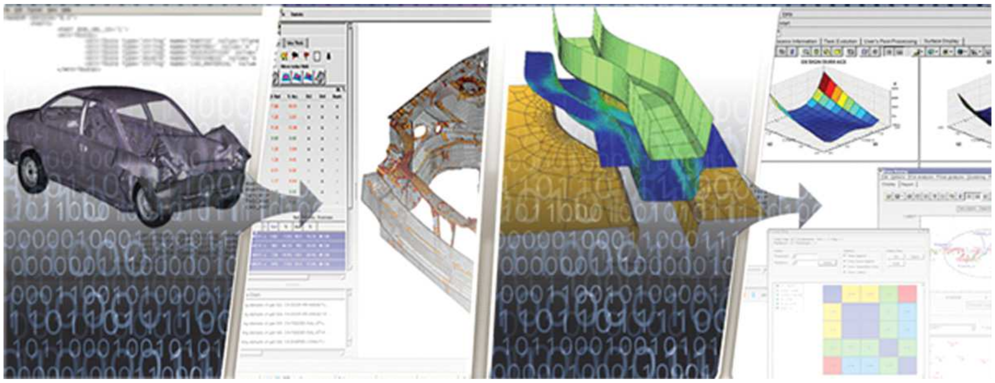


RADIOSS User's Code Interface

2017 version – January 2017

Material Laws

Chapter 2

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

2.0 User's Material Laws 29, 30, 31

Up to 3 material user's laws (law 29, 30 and 31) can be defined for 3D, 2D solid elements and 3D shell elements. User's laws for beam or truss elements are not available.

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

The Starter subroutine is called LECM29, LECM30, or LECM31 and is used to read material data and to initialize material parameters.

The Engine subroutine for solids is called SIGEPS29, SIGEPS30, or SIGEPS31 and is used to compute the solid element stress tensor at the integration point.

The corresponding shell subroutines are SIGEPS29C, SIGEPS30C or SIGEPS31C.

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

2.1. Starter Subroutine LECMnn

This subroutine reads the user's law input data. The seven first material cards (see RADIOSS Starter Input Manual 3.1 to 4.1) are read before this subroutine is called.

The number of specific material cards and their formats is free.

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

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

Argument	Format	Description
IIN	Integer read only scalar	Input file unit (Starter input file) on which the data are read.
IOUT	Integer read only scalar	Output file unit (Starter listing file).
UPARAM	Float array	Array with a size NUPARAM used to store material data.
MAXNUPARAM	Integer read only scalar	Maximum possible size of UPARAM.
NUPARAM	Integer scalar	Effective size of UPARAM. (MAXNUPARAM~NUPARAM, MAXNUPARAM set to 1,000).
NUVAR	Integer scalar	Number of extra variables needed for each element in SIGEPSnn.
IFUNC	Integer array	Array with a size NFUNC with the list of RADIOSS functions used for this law. The function numbers have to be stored in this array (and not in UPARAM) because of a possible renumbering of function numbers.
MAXFUNC	Integer read only scalar	Maximum possible size of IFUNC.
NFUNC	Integer scalar	Number of RADOISS functions. (MAXFUNC, NFUNC, MAXFUNC set to 100).
PARMAT	Float array of size 3	Modulus needed to compute interface stiffness. The Young's or bulk modulus can define this value. The value is also used in RADOISS Starter to make an estimation of the time step.

2.2. Engine Subroutine SIGEPSnn for Solid Elements

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

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

```
C-----------------------------------------------------------------------------------------
SUBROUTINE SIGEPS29 (
1    NEL    ,NUPARAM,NUVAR   ,NFUNC   ,IFUNC   ,NPF
2    TF     ,TIME   ,TIMESTEP,UPARAM  ,RHO0    ,RHO    ,
3    VOLUME ,EINT   ,
4    EPSPXX ,EPSPYY ,EPSPZZ  ,EPSPXY  ,EPSPYZ  ,EPSPZX ,
5    DEPSXX ,DEPSYY ,DEPSZZ  ,DEPSXY  ,DEPSYZ  ,DEPSZX ,
6    EPSXX  ,EPSYY  ,EPSZZ   ,EPSXY   ,EPSYZ   ,EPSZX  ,
7    SIGOXX ,SIGOYY ,SIGOZZ  ,SIGOXY  ,SIGOYZ  ,SIGOZX ,
8    SIGNXX ,SIGNYY ,SIGNZZ  ,SIGNXY  ,SIGNYZ  ,SIGNZX ,
9    SIGVXX ,SIGVYY ,SIGVZZ  ,SIGVXY  ,SIGVYZ  ,SIGVZX ,
A    SOUNDSP,VISCMAX,UVAR    ,OFF     )
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.

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.
RHO0	Float array read only	Array of size NEL containing initial densities.
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.

Argument	Format	Description
EINT	Float array read only	Array of size NEL containing total internal energy.
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.
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.
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.
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.

2.2.1. Additional Data Necessary for Compatibility with HEPH

You must provide RADIOSS with plasticity information to calculate physical stabilization forces of HEPH. Yield and Plastic strain values must be given back to RADIOSS in the user law (SIGEPS29.F, SIGEPS.F, or SIGEPS.F) to be compatible with the HEPH element. This can be done through a call to routine SET_U_SOLPLAS.

The prototype of this routine and the necessary data to provide are described below.

```
C-----------------------------------------------------------------
C    New routine : SET_U_SOLPLAS allows to return to RADIOSS,
C                          Yield and Plastic strain for all elements.
C
C     Description of the arguments to be given to
C     SET_U_SOLPLAS(NEL,YLD,PLA) :
C---------+-----------+---+------------------------------------------
C VAR    | SIZE     |TYP| DEFINITION
C----------+--------+---+--------------------------------------------
C NEL   |  1        | I  |   NUMBER OF ELEMENTS
C YLD   |  NEL    | F |  YIELD VALUE FOR EACH ELEMENT,
C                         (FOR THE CURRENT INTEGRATION POINT)
C PLA   |  NEL     | F |  PLASTIC STRAIN VALUE FOR EACH ELEMENT,
C                         (FOR THE CURRENT INTEGRATION POINT)
C---------+---------+---+-------------------------------------------
```

The arrays YLD and PLA, defined in the routine SET_U_SOLPLAS, have to be declared as local variables in the user material routines SIGEPS29.F, SIGEPS30.F, and SIGEPS31.F with a sufficiently-long NEL.

It is recommended to use the following statement, if these routines are compiled with a compiler that supports FORTRAN 90.

```
DOUBLE PRECISION YLD(NEL), PLA(NEL)
```

This statement is not possible if the compiler only supports FORTRAN 77, since NEL is set by RADIOSS and given as an argument to the user routine. Therefore, the dynamic allocation is intended in this statement.

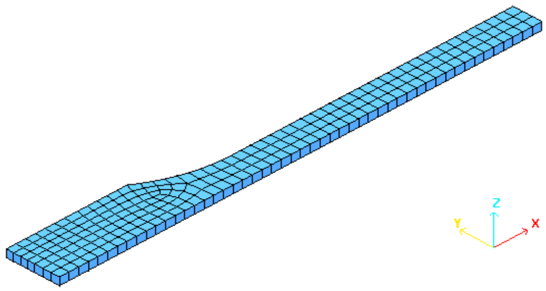
```
DOUBLE PRECISION YLD(4096), PLA(4096)
```

The statement must be sufficient in all cases, meaning the value of NEL given from RADIOSS to the user routine should be less than 4096.

2.3. Example of User's 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:



2.3.1. User's Input Data (/MAT/USERn/ option)

```
[…]
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
/MAT/USER1/2
user's elastic material law
#                RHO
              0.0027
#                  E                  Nu
               60400                0.33
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
 […]
```

2.3.2. Starter User's Subroutine LECMnn

```
C=================================================================
C    This subroutine reads the user material parameters.
C=================================================================
      SUBROUTINE LECM29(IIN  ,IOUT ,UPARAM ,MAXUPARAM,NUPARAM,
     .                 NUVAR,IFUNC,MAXFUNC,NFUNC    ,STIFINT)
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
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
```

2.3.3. Engine User's Subroutine SIGEPSnn

```
C=================================================================
C    This subroutine computes elastic stresses.
C=================================================================
         SUBROUTINE SIGEPS29 (
    1    NEL    ,NUPARAM,NUVAR   ,NFUNC   ,IFUNC   ,NPF    ,
    2    TF     ,TIME   ,TIMESTEP,UPARAM  ,RHO0    ,RHO    ,
    3    VOLUME ,EINT   ,
    4    EPSPXX ,EPSPYY ,EPSPZZ  ,EPSPXY  ,EPSPYZ  ,EPSPZX ,
    5    DEPSXX ,DEPSYY ,DEPSZZ  ,DEPSXY  ,DEPSYZ  ,DEPSZX ,
    6    EPSXX  ,EPSYY  ,EPSZZ   ,EPSXY   ,EPSYZ   ,EPSZX  ,
    7    SIGOXX ,SIGOYY ,SIGOZZ  ,SIGOXY  ,SIGOYZ  ,SIGOZX ,
    8    SIGNXX ,SIGNYY ,SIGNZZ  ,SIGNXY  ,SIGNYZ  ,SIGNZX ,
    9    SIGVXX ,SIGVYY ,SIGVZZ  ,SIGVXY  ,SIGVYZ  ,SIGVZX ,
    A    SOUNDSP,VISCMAX,UVAR    ,OFF     )
C-----------------------------------------------
C   I m p l i c i t   T y p e s
C-----------------------------------------------
C    IMPLICIT NONE
C-----------------------------------------------
C   I N P U T   A r g u m e n t s
C-----------------------------------------------
C
      INTEGER NEL, NUPARAM, NUVAR
      DOUBLE PRECISION TIME,TIMESTEP,UPARAM(NUPARAM),
     .   RHO(NEL),RHO0(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)
C-----------------------------------------------
C   O U T P U T   A r g u m e n t s
C-----------------------------------------------
      DOUBLE PRECISION
     .    SIGNXX(NEL),SIGNYY(NEL),SIGNZZ(NEL),
     .    SIGNXY(NEL),SIGNYZ(NEL),SIGNZX(NEL),
     .    SIGVXX(NEL),SIGVYY(NEL),SIGVZZ(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)
C
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) .. FUNCTION USED FOR THIS LAW
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
      DOUBLE PRECISION A11,A12,A44
C     .    YLD(NEL),PLA(NEL)
C======================================
C
C     ELASTIC LAW FOR SOLID - PROGRAM
C
C======================================
C
C-----------------------------------------------
C     ELASTIC SOLUTION
C-----------------------------------------------
      DO I=1,NEL
C
        A11 = UPARAM(1)
        A12 = UPARAM(2)
        A44 = UPARAM(3)
C
        SIGNXX(I) = A11 * EPSXX(I) + A12 * EPSYY(I) + A12 * EPSZZ(I)
        SIGNYY(I) = A12 * EPSXX(I) + A11 * EPSYY(I) + A12 * EPSZZ(I)
        SIGNZZ(I) = A12 * EPSXX(I) + A12 * EPSYY(I) + A11 * EPSZZ(I)
        SIGNXY(I) = A44 * EPSXY(I)
        SIGNYZ(I) = A44 * EPSYZ(I)
        SIGNZX(I) = A44 * EPSZX(I)
C
        SOUNDSP(I) = SQRT(A11/RHO0(I))
        VISCMAX(I) = 0.
      ENDDO
C
C-----------------------------------------------
C     PLASTIC STRAIN
C    (for compatibility with HEPH elements)
C-----------------------------------------------
C      DO I=1,NEL
C        YLD(I) = 1. E+20
C        PLA(I)= 0.
C      ENDDO
C      CALL SET_U_SOLPLAS(NEL,YLD,PLA)
C-----------------------------------------------
      RETURN
      END
```

2.4. Engine Subroutine SIGEPSnnC for Shell Elements

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

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

```
C-----------------------------------------------------------------------------------------
      SUBROUTINE SIGEPS29C(
     1     NEL    ,NUPARAM,NUVAR   ,NFUNC   ,IFUNC  ,
     2     NPF    ,NPT    ,IPT     ,IFLAG   ,
     2     TF     ,TIME   ,TIMESTEP,UPARAM  ,RHO0   ,
     3     AREA   ,EINT   ,THKLY   ,
     4     EPSPXX ,EPSPYY ,EPSPXY  ,EPSPYZ  ,EPSPZX ,
     5     DEPSXX ,DEPSYY ,DEPSXY  ,DEPSYZ  ,DEPSZX ,
     6     EPSXX  ,EPSYY  ,EPSXY   ,EPSYZ   ,EPSZX  ,
     7     SIGOXX ,SIGOYY ,SIGOXY  ,SIGOYZ  ,SIGOZX ,
     8     SIGNXX ,SIGNYY ,SIGNXY  ,SIGNYZ  ,SIGNZX ,
     9     SIGVXX ,SIGVYY ,SIGVXY  ,SIGVYZ  ,SIGVZX ,
     A     SOUNDSP,VISCMAX,THK     ,PLA     ,UVAR   ,
     B     OFF    ,NGL    ,SHF)
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. Stress are computed at each integration point.

You must 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.

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

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).
NPT	Integer read only scalar	Number of layers or integration points.
IPT	Integer read only scalar	Current layer or interrogation point.
IFLAG	Integer array read only scalar	Array of size NEL with geometrical flags.
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.
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.
THKLY	Float array write only	Array of size NEL with the layer thickness at each integration point.
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.

Argument	Format	Description
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.
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.
THK	Float array read write	Array of size NEL containing total thickness.
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.

2.4.1. 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.

2.4.2. Additional Data Necessary for Compatibility with QEPH

With QEPH, you must provide RADIOSS information relative to plasticity so that physical stabilization can be accurately calculated. Yield value and value Et/E (tangent modulus divided by Young's modulus) must be given back to RADIOSS in the user law (SIGEPS29C.F, SIGEPS30C.F, or SIGEPS31C.F) to be compatible with the QEPH element.

This can be done through a call to routine SET_U_SHLPLAS.

The prototype of this routine and the necessary data to provide are described below.

```
C-----------------------------------------------------------------------
C    New routine : SET_U_SHLPLAS allows to return to RADIOSS,
C                              Yield and Et/E for all elements.
C
C     Description of the arguments to be given to
C     SET_U_SHLPLAS(NEL,YLD,ETSE) :
C---------+-----------+---+---------------------------------------------
C VAR    | SIZE     |TYP| DEFINITION
C---------+-----------+---+---------------------------------------------
C NEL    |  1        | I  |   NUMBER OF ELEMENTS
C YLD    |  NEL    | F |  YIELD VALUE FOR EACH ELEMENT,
C                         (FOR THE CURRENT INTEGRATION POINT)
C ETSE   |  NEL    | F |  VALUE FOR EACH ELEMENT,
C                         AND THE CURRENT INTEGRATION POINT OF :
C                       ETSE = 1          in case of an elastic increment
C                              = H/(H+E)   in case of a plastic increment
C        where H : plastic tangent modulus,
C              E : Young modulus
C---------+---------+---+---+-------------------------------------------
```

The arrays YLD and ETSE, defined in the routine SET_U_SHLPLAS, have to be declared as local variables in the user material routines SIGEPS29C.F, SIGEPS30C.F, and SIGEPS30C.F with a sufficiently-long NEL.

Use the following statement in case these routines are compiled with a compiler that supports FORTRAN 90.

```
DOUBLE PRECISION YLD(NEL), ETSE(NEL)
```

This statement is not possible if the compiler only supports FORTRAN 77, since NEL is set by RADIOSS and given as an argument to the user routine. Therefore, the dynamic allocation is intended in this statement. In this case, use the following statement.

```
DOUBLE PRECISION YLD(4096), ETSE(4096)
```

The statement must be sufficient in all cases, meaning the value of NEL given from RADIOSS to the user routine should be less than 4096.

2.5. 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.

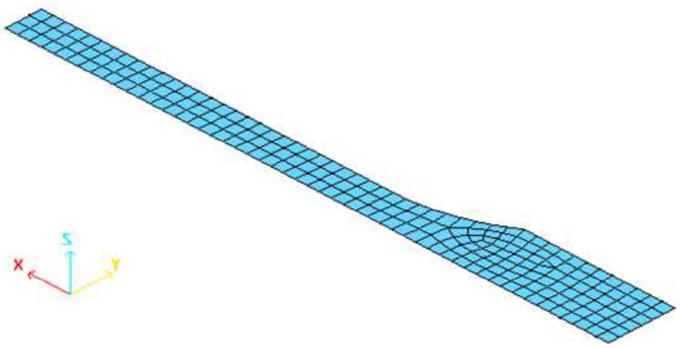
The Johnson Cook model: $\sigma = A + B\varepsilon_p^N$

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:



2.5.1. User's Input Data (/MAT/USERn/ option)

```
[…]
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
/MAT/USER1/1
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----|
[…]
```

2.5.2. Starter User's Subroutine LECMnn

```
C=================================================================
C    This subroutine reads the user material parameters.
C=================================================================
     SUBROUTINE LECM29(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
```

2.5.3. Engine User's Subroutine SIGEPSnnC

```
SUBROUTINE SIGEPS29C(
    1    NEL    ,NUPARAM,NUVAR   ,NFUNC   ,IFUNC  ,
    2    NPF    ,NPT    ,IPT     ,IFLAG   ,
    2    TF     ,TIME   ,TIMESTEP,UPARAM  ,RHO0   ,
    3    AREA   ,EINT   ,THKLY   ,
    4    EPSPXX ,EPSPYY ,EPSPXY  ,EPSPYZ  ,EPSPZX ,
    5    DEPSXX ,DEPSYY ,DEPSXY  ,DEPSYZ  ,DEPSZX ,
    6    EPSXX  ,EPSYY  ,EPSXY   ,EPSYZ   ,EPSZX  ,
    7    SIGOXX ,SIGOYY ,SIGOXY  ,SIGOYZ  ,SIGOZX ,
    8    SIGNXX ,SIGNYY ,SIGNXY  ,SIGNYZ  ,SIGNZX ,
    9    SIGVXX ,SIGVYY ,SIGVXY  ,SIGVYZ  ,SIGVZX ,
    A    SOUNDSP,VISCMAX,THK     ,PLA     ,UVAR   ,
    B    OFF    ,NGL  , SHF)
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   A r g u m e n t s
C-----------------------------------------------
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   A r g u m e n t s
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_I(NEL),X1(NEL),Y1(NEL),Z1(NEL),SVM1(NEL),
     .       A(NEL),VM2(NEL),DPLA_J(NEL),DR(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      Plastically admissible stresses computation:
       IF(IFLAG(1).EQ.0) THEN
        IPLAS = 0
     ELSEIF(IFLAG(1).EQ.1) THEN
        IPLAS = 1
       ELSEIF(IFLAG(1).EQ.2) THEN
        IPLAS = 2
       ENDIF
C
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.
C
      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(I) = OFF(I)*SVM(I)*UMR/E
        UVAR(I,1) = UVAR(I,1) + DPLA_I(I)
        PLA(I) = PLA(I) + DPLA_I(I)
       ENDDO
C
C------------------------------------------------
C     -> Compute thickness
C------------------------------------------------
       DO I=1,NEL
        DEZZ = DPLA_I(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) RETURN
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))
C
        ENDDO
C       NMAX: number of iterations
        DO N=1,NMAX
          DO J=1,NINDX
            I=INDEX(J)
            DPLA_I(I) = DPLA_J(I)
            YLD_I = UVAR(I,2)+UVAR(I,3)*DPLA_I(I)
            DR(I) = 0.5*E*DPLA_I(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(I).GT.0.) THEN
              DPLA_J(I)=MAX(ZERO,DPLA_I(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(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
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(I) = OFF(I)*SQRT(VM2(I))*UMR/(G3)
         S1=0.5*(SIGNXX(I)+SIGNYY(I))
         UVAR(I,1) = UVAR(I,1) + DPLA_I(I)
         PLA(I) = UVAR(I,1)
         DEZZ = DPLA_I(I) * S1 /UVAR(I,2)
         DEZZ=-(DEPSXX(I)+DEPSYY(I))*NNU1-NU1*DEZZ
         THK(I) = THK(I) + DEZZ*THKLY(I)
C
C-----------------------------------------------
        ENDDO
C
      ENDIF
C
C-----------------------------------------------
C
     RETURN
     END
```