```
module ModNoNLinearElasticity
contains

Subroutine ESM_NONLINEAR(NPT,NOEL,IDSET,STRESS,EUNLOADING,PLASTICMULTIPLIE
DSTRAN,NSTATEV,STATEV,NADDVAR,ADDITIONALVAR,CMNAME,NPROPS,PROPS,NUMBEF


implicit double precision (a-h, o-z)
integer :: NTENS, NSTATEV, NADDVAR, NPROPS, NPT, NOEL, IDSET, NUMBEROFPHASES
double precision :: EUNLOADING, PLASTICMULTIPLIER
CHARACTER*80 CMNAME
DIMENSION STRESS(NTENS), DSTRAN(NTENS),STATEV(NSTATEV),ADDITIONALVAR(NADD


!---Local variables required in standard UMAT
integer :: IStep, TimeStep
double precision, dimension(:), allocatable :: ddsddt ! only for fully coupled thermal analysis: varia
double precision, dimension(:), allocatable :: drplde ! only for fully coupled thermal analysis: varia
double precision, dimension(:), allocatable :: stran
double precision, dimension(:), allocatable :: time
double precision, dimension(:), allocatable :: predef
double precision, dimension(:), allocatable :: dpred
double precision, dimension(:), allocatable :: coords
double precision, dimension(:,:), allocatable :: ddsdde ! Jacobian matrix of the constitutive model (
double precision, dimension(:,:), allocatable :: drot
double precision, dimension(:,:), allocatable :: dfgrd0
double precision, dimension(:,:), allocatable :: dfgrd1
double precision :: sse, spd, scd ! specific elastic strain energy, plastic dissipation, creep dissipatic
double precision :: rpl ! only for fully coupled thermal analysis: volumetric heat generation
double precision :: drpldt ! only for fully coupled thermal analysis: variation of volumetric heat gene
double precision :: pnewdt, dtime, temp, dtemp, celent
double precision :: Value ! auxiliary variable holding any real valued number
double precision :: Porosity, WaterPressure, WaterPressure0, GasPressure, GasPressure0, Degre


integer :: ndi, nshr, layer, kspt, kstep, kinc

!---Local variables defned by the user
! e.g. integer :: var_local
!---User can define here additional variables

allocate( ddsddt(ntens), drplde(ntens), stran(ntens), time(2), predef(1), dpred(1), &
coords(3), ddsdde(ntens,ntens), drot(3,3), dfgrd0(3,3), dfgrd1(3,3) )

!Initialization
Eunloading = 0.0
PlasticMultiplier = 0.0

!Rename additional variables
Porosity = AdditionalVar(1)
WaterPressure = AdditionalVar(2)
WaterPressure0 = AdditionalVar(3)
GasPressure = AdditionalVar(4)
GasPressure0 = AdditionalVar(5)
DegreeSaturation = AdditionalVar(6)
time(1) = AdditionalVar(7) !TotalRealTime
time(2) = AdditionalVar(8) !OverallTotalTime
dtime = AdditionalVar(9) !TimeIncrement
IStep = AdditionalVar(10)
TimeStep = AdditionalVar(11) !Note: Very first time and load step: Istep=1 and TimeStep=1
!Call the UMAT
call umat(stress, statev, ddsdde, sse, spd, scd, rpl, ddsddt, drplde, drpldt, stran, dstran, time, dtim
dtemp, predef, dpred, cmname, ndi, nshr, ntens, nstatev, props, nprops, coords, drot, pnewdt, cele
dfgrd1, noel, npt, layer, kspt, kstep, kinc)


!---Definition of Eunloading -> required to define the max time step
Eunloading = max(ddsdde(1,1),ddsdde(2,2),ddsdde(3,3))
!---Always define this value to run the simulation

! PlasticMultiplier can be given as an output because plastic points can be plotted as a result




return

end subroutine ESM_NONLINEAR
```

```
SUBROUTINE UMAT(STRESS,STATEV,DDSDDE,SSE,SPD,SCD, &
RPL,DDSDDT,DRPLDE,DRPLDT, &
STRAN,DSTRAN,TIME,DTIME,TEMP,DTEMP,PREDEF,DPRED,CMNAME, &
NDI,NSHR,NTENS,NSTATEV,PROPS,NPROPS,COORDS,DROT,PNEWDT, &
CELENT,DFGRD0,DFGRD1,NOEL,NPT,LAYER,KSPT,KSTEP,KINC)

implicit double precision (a-h, o-z)
! !DEC$ ATTRIBUTES DLLEXPORT, ALIAS:"UMAT" :: UMAT
! INCLUDE 'ABA_PARAM.INC'
integer :: NTENS, NSTATEV, NPROPS
CHARACTER*80 CMNAME
DIMENSION STRESS(NTENS),STATEV(NSTATEV), &
DDSDDE(NTENS,NTENS),DDSDDT(NTENS),DRPLDE(NTENS), &
STRAN(NTENS),DSTRAN(NTENS),TIME(2),PREDEF(1),DPRED(1), &
PROPS(NPROPS),COORDS(3),DROT(3,3),DFGRD0(3,3),DFGRD1(3,3)



! Arguments:
! I/O Type
! PROPS I R() : List with model parameters
! DSTRAN I R() : Strain increment
! DDSDDE O R(,) : Material stiffness matrix
! STRESS I/O R() : stresses
! STATEV I/O R() : state variables

!--- Local variables
dimension dSig(NTENS), Sig(NTENS)

! Contents of PROPS(3)
! 1 : E Young Modulus
! 2 : ENU Poisson's ratio
!   3   : PLIM

 E = PROPS(1)
 ENU = PROPS(2)
 PLIM = PROPS(3)
 one = 1.0d0
 two = 2.0d0
 PRESS = (STRESS(1) + STRESS(2) + STRESS(3))/3
 EDEG = E/(1+PRESS/PLIM)
 G = EDEG/two/(one+ENU)
 ! calculate elastic stress increment (DSigE = elastic stiffness D * strain increment DEps
 FAC = two * G /( one - two * ENU )
 D1 = FAC * ( one - ENU )
 D2 = FAC * ENU
 DSTRANVOL = DSTRAN(1) + DSTRAN(2) + DSTRAN(3)
 dSig(1) = (D1 - D2) * DSTRAN(1) + D2 * DSTRANVOL
 dSig(2) = (D1 - D2) * DSTRAN(2) + D2 * DSTRANVOL
 dSig(3) = (D1 - D2) * DSTRAN(3) + D2 * DSTRANVOL
 dSig(4) = G * DSTRAN(4)
 if (NTENS == 6) then
 dSig(5) = G * DSTRAN(5)
 dSig(6) = G * DSTRAN(6)
 end if
 ! elastic stress
 Sig = STRESS + dSig


 ! stress state parameters update
 do i = 1, NTENS
 STRESS(i) = Sig(i)
 end do

 DDSDDE = 0.0
 DDSDDE(1:3,1:3) = D2
 DDSDDE(1,1) = D1
 DDSDDE(2,2) = D1
 DDSDDE(3,3) = D1
 DDSDDE(4,4) = G
 if (NTENS == 6) then
 DDSDDE(5,5) = G
 DDSDDE(6,6) = G
 end if

 return
 end subroutine umat

 end module ModNoNLinearElasticity
```