

## Example 5: Isotropic Hardening Plasticity

### Governing Equations

- Elasticity:

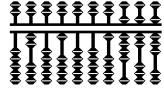
$$\sigma_{ij} = \lambda \delta_{ij} \epsilon_{kk}^{el} + 2\mu \epsilon_{ij}^{el},$$

or in a Jaumann (corotational) rate form:

$$\dot{\sigma}_{ij}^J = \lambda \delta_{ij} \dot{\epsilon}_{kk}^{el} + 2\mu \dot{\epsilon}_{ij}^{el}.$$

- The Jaumann rate equation is integrated in a corotational framework:

$$\Delta \sigma_{ij}^J = \lambda \delta_{ij} \Delta \epsilon_{kk}^{el} + 2\mu \Delta \epsilon_{ij}^{el}.$$



- Plasticity:
  - Yield function:

$$\sqrt{\frac{3}{2}} S_{ij} S_{ij} - \sigma_y(\bar{\epsilon}^{pl}) = 0, \quad S_{ij} = \sigma_{ij} - \frac{1}{3} \delta_{ij} \sigma_{kk}.$$

- Equivalent plastic strain:

$$\bar{\epsilon}^{pl} = \int_0^t \dot{\bar{\epsilon}}^{pl} dt, \quad \dot{\bar{\epsilon}}^{pl} = \sqrt{\frac{2}{3}} \dot{\epsilon}_{ij}^{pl} \dot{\epsilon}_{ij}^{pl}.$$

- Plastic flow law:

$$\dot{\epsilon}_{ij}^{pl} = \frac{3 S_{ij}}{2 \sigma_y} \dot{\bar{\epsilon}}^{pl}.$$

## Integration Procedure

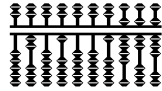
- We first calculate the von Mises stress based on purely elastic behavior (elastic predictor):

$$\bar{\sigma}^{pr} = \sqrt{\frac{3}{2} S_{ij}^{pr} S_{ij}^{pr}}, \quad S_{ij}^{pr} = S_{ij}^o + 2\mu \Delta e_{ij}.$$

- If the elastic predictor is larger than the current yield stress, plastic flow occurs. The backward Euler method is used to integrate the equations.
  - After some manipulation we can reduce the problem to a single equation in terms of the incremental equivalent plastic strain:

$$\bar{\sigma}^{pr} - 3\mu \Delta \bar{\epsilon}^{pl} = \sigma_y(\bar{\epsilon}^{pl}).$$

- This equation is solved with Newton's method.



- After the equation is solved, the following update equations for the stress and the plastic strain can be used:

$$\sigma_{ij} = \eta_{ij}\sigma_y + \frac{1}{3}\delta_{ij}\sigma_{kk}^{pr}, \quad \Delta\epsilon_{ij}^{pl} = \frac{3}{2}\eta_{ij}\Delta\bar{\epsilon}^{pl}$$

$$\eta_{ij} = S_{ij}^{pr} / \bar{\sigma}^{pr}.$$

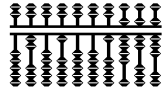
- In addition, you can readily obtain the consistent Jacobian:

$$\Delta\dot{\sigma}_{ij} = \lambda^*\delta_{ij}\Delta\dot{\epsilon}_{kk} + 2\mu^*\Delta\dot{\epsilon}_{ij} + \left(\frac{h}{1 + h/3\mu} - 3\mu^*\right)\eta_{ij}\eta_{kl}\Delta\dot{\epsilon}_{kl}$$

$$\mu^* = \mu\sigma_y/\bar{\sigma}^{pr}, \quad \lambda^* = k - \frac{2}{3}\mu^*, \quad h = d\sigma_y/d\bar{\epsilon}^{pl}.$$

- A detailed discussion about the isotropic plasticity integration algorithm can be found in Section 4.2.2 of the ABAQUS Theory Manual.

The appropriate coding is shown on the following pages.

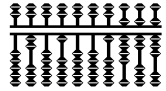


## Coding for Isotropic Mises Plasticity

```

C      LOCAL ARRAYS
C  -----
C      EELAS   - ELASTIC STRAINS
C      EPLAS   - PLASTIC STRAINS
C      FLOW    - DIRECTION OF PLASTIC FLOW
C  -----
C
C      DIMENSION EELAS(6),EPLAS(6),FLOW(6), HARD(3)
C
C      PARAMETER(ZERO=0.D0, ONE=1.D0, TWO=2.D0, THREE=3.D0, SIX=6.D0,
1          ENUMAX=.4999D0, NEWTON=10, TOLER=1.0D-6)
C
C  -----
C      UMAT FOR ISOTROPIC ELASTICITY AND ISOTROPIC MISES PLASTICITY
C      CANNOT BE USED FOR PLANE STRESS
C  -----
C      PROPS(1) - E
C      PROPS(2) - NU
C      PROPS(3..) - SYIELD AN HARDENING DATA
C      CALLS UHARD FOR CURVE OF YIELD STRESS VS. PLASTIC STRAIN
C  -----

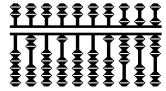
```



```
C
C      ELASTIC PROPERTIES
C
      EMOD=PROPS(1)
      ENU=MIN(PROPS(2), ENUMAX)
      EBULK3=EMOD/(ONE-TWO*ENU)
      EG2=EMOD/(ONE+ENU)
      EG=EG2/TWO
      EG3=THREE*EG
      ELAM=(EBULK3-EG2)/THREE

C
C      ELASTIC STIFFNESS
C

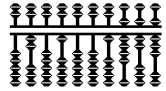
      DO K1=1, NDI
        DO K2=1, NDI
          DDSDE(K2, K1)=ELAM
        END DO
        DDSDE(K1, K1)=EG2+ELAM
      END DO
      DO K1=NDI+1, NTENS
        DDSDE(K1, K1)=EG
      END DO
```



```

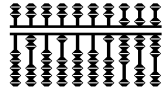
C   RECOVER ELASTIC AND PLASTIC STRAINS AND ROTATE FORWARD
C   ALSO RECOVER EQUIVALENT PLASTIC STRAIN
C
      CALL ROTSIG(STATEV(      1), DROT, EELAS, 2, NDI, NSHR)
      CALL ROTSIG(STATEV(NTENS+1), DROT, EPLAS, 2, NDI, NSHR)
      EQPLAS=STATEV(1+2*NTENS)
C
C   CALCULATE PREDICTOR STRESS AND ELASTIC STRAIN
C
      DO K1=1, NTENS
        DO K2=1, NTENS
          STRESS(K2)=STRESS(K2)+DDSDDE(K2, K1)*DSTRAN(K1)
        END DO
        EELAS(K1)=EELAS(K1)+DSTRAN(K1)
      END DO
C
C   CALCULATE EQUIVALENT VON MISES STRESS
C
      SMISES=(STRESS(1)-STRESS(2))**2+(STRESS(2)-STRESS(3))**2
1      + (STRESS(3)-STRESS(1))**2
      DO K1=NDI+1,NTENS
        SMISES=SMISES+SIX*STRESS(K1)**2
      END DO
      SMISES=SQRT(SMISES/TWO)

```



```
C
C   GET YIELD STRESS FROM THE SPECIFIED HARDENING CURVE
C
      NVALUE=NPROPS/2-1
      CALL UHARD(SYIEL0, HARD, EQPLAS, EQPLASRT, TIME, DTIME, TEMP,
1        DTEMP, NOEL, NPT, LAYER, KSPT, KSTEP, KINC, CMNAME, NSTATV,
2        STATEV, NUMFIELDV, PREDEF, DPRED, NVALUE, PROPS(3))
C
C   DETERMINE IF ACTIVELY YIELDING
C
      IF (SMISES.GT.(ONE+TOLER)*SYIEL0) THEN
C
C   ACTIVELY YIELDING
C   SEPARATE THE HYDROSTATIC FROM THE DEVIATORIC STRESS
C   CALCULATE THE FLOW DIRECTION
C
      SHYDRO=(STRESS(1)+STRESS(2)+STRESS(3))/THREE
      DO K1=1,NDI
        FLOW(K1)=(STRESS(K1)-SHYDRO)/SMISES
      END DO
      DO K1=NDI+1, NTENS
        FLOW(K1)=STRESS(K1)/SMISES
      END DO
```

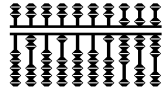




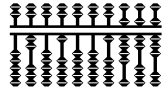
```

C
C      SOLVE FOR EQUIVALENT VON MISES STRESS
C      AND EQUIVALENT PLASTIC STRAIN INCREMENT USING NEWTON ITERATION
C
      SYIELD=SYIEL0
      DEQPL=ZERO
      DO KEWTON=1, NEWTON
        RHS=SMISES-EG3*DEQPL-SYIELD
        DEQPL=DEQPL+RHS/(EG3+HARD(1))
        CALL UHARD(SYIELD,HARD,EQPLAS+DEQPL,EQPLASRT,TIME,DTIME,TEMP,
1        DTEMP,NOEL,NPT,LAYER,KSPT,KSTEP,KINC,CMNAME,NSTATV,
2        STATEV,NUMFIELDV,PREDEF,DPRED,NVALUE,PROPS(3))
        IF(ABS(RHS).LT.TOLER*SYIEL0) GOTO 10
      END DO
C
C      WRITE WARNING MESSAGE TO THE .MSG FILE
C
      WRITE(7,2) NEWTON
2      FORMAT(/,30X,'***WARNING - PLASTICITY ALGORITHM DID NOT ',
1      'CONVERGE AFTER ',I3,' ITERATIONS')
10  CONTINUE

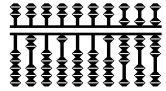
```



```
C
C      UPDATE STRESS, ELASTIC AND PLASTIC STRAINS AND
C      EQUIVALENT PLASTIC STRAIN
C
      DO K1=1,NDI
        STRESS(K1)=FLOW(K1)*SYIELD+SHYDRO
        EPLAS(K1)=EPLAS(K1)+THREE/TWO*FLOW(K1)*DEQPL
        EELAS(K1)=EELAS(K1)-THREE/TWO*FLOW(K1)*DEQPL
      END DO
      DO K1=NDI+1,NTENS
        STRESS(K1)=FLOW(K1)*SYIELD
        EPLAS(K1)=EPLAS(K1)+THREE*FLOW(K1)*DEQPL
        EELAS(K1)=EELAS(K1)-THREE*FLOW(K1)*DEQPL
      END DO
      EQPLAS=EQPLAS+DEQPL
C
C      CALCULATE PLASTIC DISSIPATION
C
      SPD=DEQPL*(SYIEL0+SYIELD)/TWO
```



```
C
C   FORMULATE THE JACOBIAN (MATERIAL TANGENT)
C   FIRST CALCULATE EFFECTIVE MODULI
C
      EFFG=EG*SYIELD/SMISES
      EFFG2=TWO*EFFG
      EFFG3=THREE/TWO*EFFG2
      EFFLAM=(EBULK3-EFFG2)/THREE
      EFFHRD=EG3*HARD(1)/(EG3+HARD(1))-EFFG3
      DO K1=1, NDI
        DO K2=1, NDI
          DDSDE(K2, K1)=EFFLAM
        END DO
        DDSDE(K1, K1)=EFFG2+EFFLAM
      END DO
      DO K1=NDI+1, NTENS
        DDSDE(K1, K1)=EFFG
      END DO
      DO K1=1, NTENS
        DO K2=1, NTENS
          DDSDE(K2, K1)=DDSDE(K2, K1)+EFFHRD*FLOW(K2)*FLOW(K1)
        END DO
      END DO
ENDIF
```



```

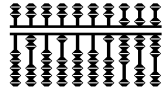
C
C   STORE ELASTIC AND (EQUIVALENT) PLASTIC STRAINS
C   IN STATE VARIABLE ARRAY
C
      DO K1=1, NTENS
        STATEV(K1)=EELAS(K1)
        STATEV(K1+NTENS)=EPLAS(K1)
      END DO
      STATEV(1+2*NTENS)=EQPLAS
C
      RETURN
      END

      SUBROUTINE UHARD(SYIELD,HARD,EQPLAS,EQPLASRT,TIME,DTIME,TEMP,
1      DTEMP,NOEL,NPT,LAYER,KSPT,KSTEP,KINC,
2      CMNAME,NSTATV,STATEV,NUMFIELDV,
3      PREDEF,DPRED,NVALUE,TABLE)

      INCLUDE 'ABA_PARAM.INC'

      CHARACTER*80 CMNAME
      DIMENSION HARD(3),STATEV(NSTATV),TIME(*),
1      PREDEF(NUMFIELDV),DPRED(*)

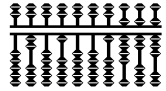
```



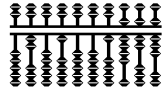
```

C
    DIMENSION TABLE(2, NVALUE)
C
    PARAMETER(ZERO=0.D0)
C
    SET YIELD STRESS TO LAST VALUE OF TABLE, HARDENING TO ZERO
C
    SYIELD=TABLE(1, NVALUE)
    HARD(1)=ZERO
C
    IF MORE THAN ONE ENTRY, SEARCH TABLE
C
    IF(NVALUE.GT.1) THEN
        DO K1=1, NVALUE-1
            EQPL1=TABLE(2,K1+1)
            IF(EQPLAS.LT.EQPL1) THEN
                EQPL0=TABLE(2, K1)
                IF(EQPL1.LE.EQPL0) THEN
                    WRITE(7, 1)
1          FORMAT(//, 30X, '***ERROR - PLASTIC STRAIN MUST BE ` ,
1          'ENTERED IN ASCENDING ORDER')
                    CALL XIT
                ENDIF
            ENDIF
        END DO
    END IF

```

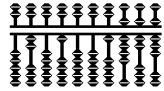


```
C
C      CURRENT YIELD STRESS AND HARDENING
C
      DEQPL=EQPL1-EQPL0
      SYIEL0=TABLE(1, K1)
      SYIEL1=TABLE(1, K1+1)
      DSYIEL=SYIEL1-SYIEL0
      HARD(1)=DSYIEL/DEQPL
      SYIELD=SYIEL0+(EQPLAS-EQPL0)*HARD(1)
      GOTO 10
    ENDIF
  END DO
10  CONTINUE
    ENDIF
    RETURN
  END
```



## Remarks

- This **UMAT** yields exactly the same results as the **\*PLASTIC** option with **ISOTROPIC** hardening.
  - This result is also true for large-strain calculations. The necessary rotations of stress and strain are taken care of by **ABAQUS**.
  - The rotation of elastic and plastic strain, prior to integration, is accomplished by the calls to **ROTSIG**.



- The routine calls user subroutine **UHARD** to recover a piecewise linear hardening curve.
  - It is straightforward to replace the piecewise linear curve by an analytic description.
  - A local Newton iteration is used to determine the current yield stress and hardening modulus.
  - If the data are not given in ascending order of strain, the routine **XIT** is called, which closes all files and terminates execution.