How to make UMATs for soils in ABAQUS

Geotechnical Research Group
University of Los Andes
http://geotecnia.uniandes.edu.co

December 21, 2008

Introduction

Example:

MODIFIED CAM-CLAY Plasticity constitutive equation:

$$\mathbf{T} = \mathbb{C}^{ep} : \boldsymbol{\varepsilon} \tag{1}$$

with $\mathbb{C}^{ep} =$

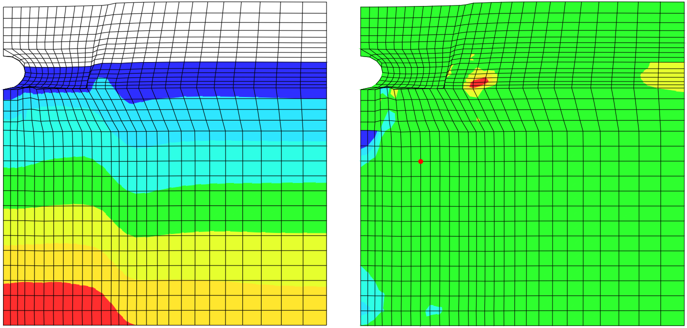
$$\begin{cases} \mathbb{C}^{e}, & \gamma = 0 \\ \mathbb{C}^{ep} = \mathbb{C}^{e} - \dfrac{1}{\chi}\left[a_1 \mathbf{1} \otimes \mathbf{1} + a_2 \left[\overrightarrow{\mathbf{T}}^* \otimes \mathbf{1} + \mathbf{1} \otimes \overrightarrow{\mathbf{T}}^*\right] + a_3 \overrightarrow{\mathbf{T}}^* \otimes \overrightarrow{\mathbf{T}}^*\right], & \gamma > 1 \end{cases}$$

where a_1, a_2, a_3 and χ are function of $\mathbf{T}$ and material constant,and

$$\overrightarrow{\mathbf{T}}^* = \frac{\mathbf{T} - \frac{1}{3}\mathbf{1}}{\parallel \mathbf{T} - \frac{1}{3}\mathbf{1} \parallel} \tag{2}$$

You make an UMAT and build a FEM model in ABAQUS!

FEM models



(c) Pore pressure u

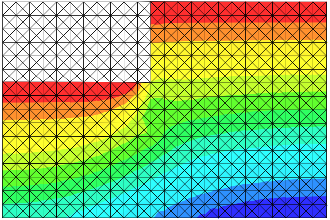
(d) Shear stress T_{xy}

Figure: CAM-CLAY model

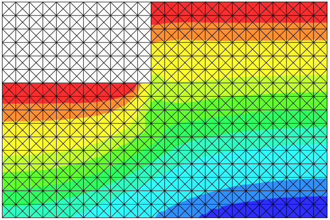
1

[1] UMAT from FUENTES

FEM models



(a) CAM-CLAY model, T_y

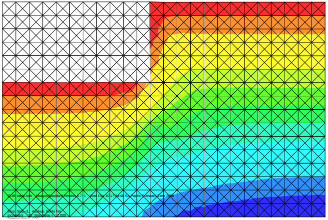


(b) Hypoelastic model, T_y

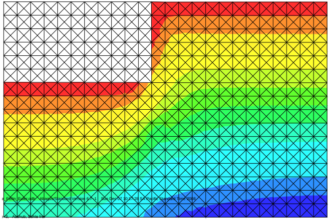
2 3 4

[2]Model from FELLIN in http://www.uibk.ac.at/geotechnik/res/hypopl.html
[3]UMATs from Fuentes.
[4]In Hypoelasticity, the RICHART's proposal is taken.

FEM models



(c) Hypoplasticity model FUENTES's UMAT, T_y

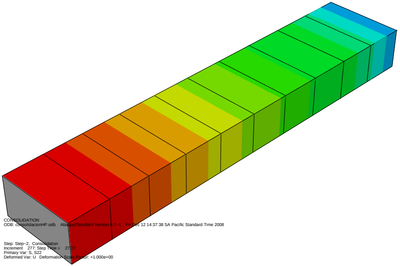


(d) Hypoplasticity model FELLIN's UMAT, T_y

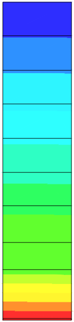
[5]

[5] WOLFFERSDORFF equation

FEM models



(e) T_y (kPa)



(f) u (kPa)

Figure: Consolidation in 3D elements, Hypoplasticity FUENTES's UMAT

INCREMENTAL DRIVER, from (NIEMUNIS, 2007)

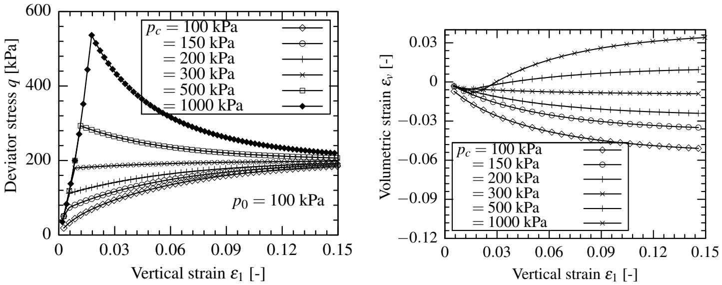


Figure: Drained triaxial path, variation of preconsolidated pressure p_c', CAM-CLAY model

6

[6] For INCREMENTAL DRIVER see [Niemunis, 2008]

INCREMENTAL DRIVER, from (NIEMUNIS, 2007)

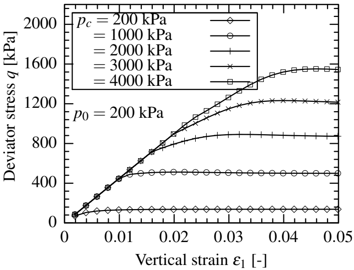
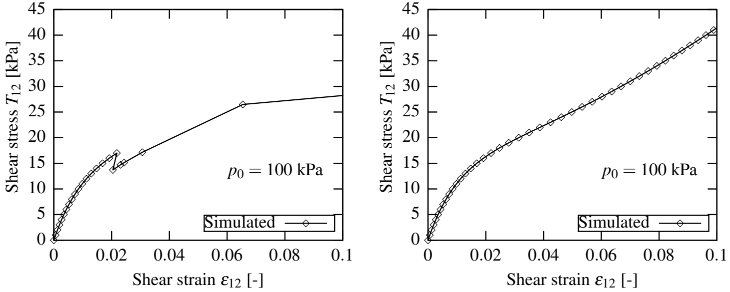


Figure: Undrained triaxial path, variation of preconsolidated pressure p_c, CAM-CLAY model

Some errors in programming



(a) FELLIN's UMAT (b) FUENTES's UMAT

Figure: Proportional path in shear stress control in Hypoplasticity

Outline

Programming in FORTRAN

Fortran advantages

- Developed by IBM at 1957.
- Then, versions FORTRAN IV, 77, 80 90, and 2003 were developed.
- Oriented to rapid calculations in numeric implementations.
- Not well oriented for other purposes, such as data bases.
- In some calculations, results indicates that what VISUAL BASIC takes 80 hours long, is equivalent to 10 seconds in FORTRAN.
- ABAQUS accepts FORTRAN subroutines.

More information can be encountered in [Chivers et. al., 2006].

Variable types

- **Integer**: 1, 243, -15, ...
- **Real simple precision**: 7 spaces + sign+ exponent. $1.0d2$, $-4.5678d-25$
- **Real double precision**: 13 spaces + sign+ exponent. $2.0d4$, $-4.567893456321d-25$
- **Complex double precision**: $(-123.65, 1.5E+2)\longrightarrow$ (real,imaginary)
- **Character**: Hello, hola
- **Logical**: .TRUE., .FALSE.

([Chivers et. al., 2006].)

Variable types

```
  INTEGER a,b,c, NTENS, I, J

  REAL a,b,c

  Double Precision a,b,c, STRESS(NTENS), DSTRAN(NTENS), dtime
& DDSDDE(NTENS, NTENS),

  LOGICAL a,b,c, info

  CHARACTER*4 hola

  PARAMETER (pi=3.1415926)
```

[7] [8]

---

[7] in CHARACTER type 4 is the number of characters

[8] & indicates that a the line above continues in that line. Any symbol can be used.

Function and math operations

- Power `**`; example: `a=5.0d0**2`
- Multiplication and division `*` and `/`; example: `b=3.5d0/2`
- Addition and substraction `+` and `-`; example: `c=4.41d0+ 5.98d0`
- Absolute value `ABS(X)`; example: `a=ABS(-10.0d0)`
- Square root `SQRT(X)`; example: `b=SQRT(17.8d0)`
- SIN, COS, TAN `SIN(X), COS(X), TAN(X)`; example: `c=SIN(1.2375d0)`
- ASIN, ACOS, ATAN `ASIN(X), ACOS(X), ATAN(X)`; example: `c=ASIN(0.536d0)`
- Integer conversion `INT(X)`; example: `c=INT(12.817865d0)`
- Real conversion `REAL(X)`; example: `a=REAL(c)`
- MAX, MIN `MAX(X,,...), MIN(X,,...)`; example: `a=MAX(1,10,5)`
- Modulus `MOD(X, Y)`; example: `a=MOD(10.5,2)` $(=10.5 - \mathrm{INT}(10.5/2))$

Data management

- Read data `READ*`; example: `READ*, a`, then you introduce the data for `a` and press enter
- Print data `PRINT*`; example: `PRINT*, a`; `PRINT*',Hello World!!"`"
- pause `PAUSE`

Data management

- Begin Program `PROGRAM [...name...]`; example: `PROGRAM [Trial]`, This is the first sentence.
- End Program `END[PROGRAM [...name...]]`; example: `END[PROGRAM [Trial]]`
- Stop the program `STOP`
- Stop subroutine or function `RETURN`

Exercise No 1

Open MICROSOFT VISUAL STUDIO 2008[9]⟹New⟹Project⟹Intel Fortran Project⟹ Empty project.
Project⟹Add new element⟹Source.

[9] Intel Visual Fortran Compiler needed.

Exercise No 1.

(Compile the executable first, then run the program).

```
PROGRAM Trial
double precision a, b, c
READ*,a
READ*,b
c=sqrt(a+b)
PRINT*, c
pause
END PROGRAM Trial
```

TIPS

- Needs a Tab space before each instruction.
- Can change the source file name by right-clicking in the archive.

Logical operators

Description	Symbol	Alternative
Less than	$<$	.LT.
Greater than	$>$	.GT.
Equal than	$==$	.EQ.
Less or equal than	$<=$	.LE.
Greater or equal than	$>=$	.GE.
Different than	$/=$	.NE.

Description	Symbol
Negation	.NOT.
Conjunction	.AND.
Disjunction	.OR.

Condition IF and Cycle DO

Conditional IF,

```
IF ((a>3).AND.(a<5)) then
b=1.0d0
ELSEIF (a>5)
b=2.0d0
ELSE
b=3.0d0
ENDIF
```

Cycle For,

```
DO i=1,3
    DO j=1,3
        A(i,j)=B(i,k)*C(k,j)
    END DO
END DO
```

SUBROUTINEs and FUNCTIONs in Fortran

SUBROUTINEs for dividing the program into sub-programs.

```
SUBROUTINE dyadic(am,bm,Cm)
integer i, j
double precision am(6,1), bm(6,1), Cm(6,6)
Cm=0
do i=1,6
  do j=1,6
      Cm(i,j) =Cm(i,j)+am(i,1)*bm(j,1)
  enddo
enddo
return
END SUBROUTINE dyadic
```

In Program:

```
CALL dyadic(a,b,C)
```

SUBROUTINEs and FUNCTIONs in Fortran

FUNCTIONs to obtain one single output variable.

```
REAL FUNCTION fun(x,y)
REAL x
INTEGER y
if (x.LE.(1./y.)) then
    fun=1.-y*x
    else
    fun=0.
    end if
END FUNCTION fun
```

In program,

```
a=fun(d,e)
```

Function `RETURN` for exiting the subroutine or function.

Matrix operations

Declaration:

```
double precision A(3,1), B(3,1), C(3,1)
```

Filling data:

```
data A/1.0d0,2.0d0,3.0d0/
B=0.0d0
```

Operations:

- Addition and subtraction `+, -`; example: `A+B`
- Component multiplication `*`; example: `A=B*C`
- Matrix Multiplication `MATMUL( , )`; example: `A=MATMUL(B,C)`
- Transpose `Transpose( )`; example: `Transpose(A)`

Exercise No 2.

```
PROGRAM Exercise2
double precision A(3,1), B(3,1), C(3,1)
integer i
data A/1.0d0,2.0d0,3.0d0/
do i=1,3
B(i,1)=1.0d0
C(i,1)=0.0d0
end do
CALL contract(A, B, C)
PRINT*, C
Pause
END PROGRAM Exercise2
SUBROUTINE contract(A, B, C)
double precision A(3,1), B(3,1), C(3,1)
if ((B(1,1).NE.0.).AND.(B(2,1).NE.0.)) then
C(1:2,1)=A(1:2,1)/B(1:2,1)
C(3,1)=0
endif
return
END SUBROUTINE contract
```

Elastic model

Constitutive equation

$$\mathbf{T} = \left[\frac{\nu E}{(1+\nu)(1-2\nu)} \mathbf{1} \otimes \mathbf{1} + \frac{E}{(1+\nu)} \mathbb{I} \right] : \boldsymbol{\varepsilon} \tag{3}$$

where,

- $\mathbf{T}$ CAUCHY stress tensor.
- $\boldsymbol{\varepsilon}$ strain tensor (infinitesimal).
- $\mathbf{1}$ Unit second order tensor.
- $\mathbb{I}$ Unit fourth order tensor for symmetric tensors.
- E and ν material parameters.

Also, an equivalent formulation can be written (see [Lai et.al., 1993]),

$$\mathbf{T} = \left[\left(k - \frac{2}{3} G \right) \mathbf{1} \otimes \mathbf{1} + 2G\mathbb{I} \right] : \boldsymbol{\varepsilon} \tag{4}$$

with G and k material parameters.

Representation

Matrix representation chosen, takes advantage of symmetry, (Coincides with ABAQUS representation.)

$$[\mathbf{T}] = \begin{bmatrix} T_{11} \\ T_{22} \\ T_{33} \\ T_{12} \\ T_{13} \\ T_{23} \end{bmatrix}, \qquad [\boldsymbol{\varepsilon}] = \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ 2\varepsilon_{12} \\ 2\varepsilon_{13} \\ 2\varepsilon_{23} \end{bmatrix} = \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \gamma_{12} \\ \gamma_{13} \\ \gamma_{23} \end{bmatrix} \tag{5}$$

Notice the selection of shear strain. Special care to keep invariance when making matrix operations.

Representation

Tangent moduli fourth order tensor representation:

$$[\mathbb{C}^e] = \begin{bmatrix} \overline{\lambda}+2\mu & \overline{\lambda} & \overline{\lambda} & 0 & 0 & 0 \\ \overline{\lambda} & \overline{\lambda}+2\mu & \overline{\lambda} & 0 & 0 & 0 \\ \overline{\lambda} & \overline{\lambda} & \overline{\lambda}+2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix} \quad (6)$$

with $\overline{\lambda} = \dfrac{\nu E}{(1+\nu)(1-2*\nu)}$ and $\mu = \dfrac{E}{2(1+\nu)}$

CAM-CLAY Elasticity and Hypoelasticity

CAM-CLAY Elasticity:

$$E = 3(1-2\nu)(1+e) \parallel (p/\kappa) \parallel \tag{7}$$

being:

- e the void ratio.
- $p = (T_{11} + T_{22} + T_{33})/3$ the mean pressure.
- κ the compression index in semi-log space.

Hypoelasticity:

$$G = G_0 p_{atm} \frac{(2.97-e)^2}{(1+e)\sqrt{(\parallel p/p_{atm} \parallel)}} \tag{8}$$

being:

- G_0 initial shear modulus.
- p_{atm} atmospheric pressure.

Making the UMAT for elastic model

Description

The UMAT is an ABAQUS user's subroutine in order to define new materials constitutive model. For more information see [Hibbit et.al., 1995].

- Accepts FORTRAN and C++ subroutines.
- NEWTON's method used for numeric integration. Other methods disponibles.
- Will be called by ABAQUS for each integration point that belongs to the material defined.
- Can use and view the results of state variables.
- Temperature dependent and time dependent constitutive models can be implemented.
- Needs to provide stresses and Jacobian $\mathbb{J} = \dfrac{\partial \Delta \mathbf{T}}{\partial \Delta \varepsilon}$ at the end of each increment.

Convention

The UMAT is an ABAQUS user's subroutine in order to define new materials constitutive model.

$$
\left[\begin{array}{c} T_{11} \\ T_{22} \\ T_{33} \\ \hline T_{12} \\ T_{13} \\ T_{23} \end{array}\right] = \left[\begin{array}{ccc|ccc} C_{1111} & C_{1122} & C_{1133} & C_{1112} & C_{1113} & C_{1123} \\ & C_{2222} & C_{2233} & C_{2212} & C_{2213} & C_{2223} \\ & & C_{3333} & C_{3312} & C_{3313} & C_{3323} \\ \hline & & & C_{1212} & C_{1213} & C_{1223} \\ & sym & & & C_{1313} & C_{1323} \\ & & & & & C_{2323} \end{array}\right] \left[\begin{array}{c} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \hline \gamma_{12} \\ \gamma_{13} \\ \gamma_{23} \end{array}\right] \tag{9}
$$

Dimensions can change according to the element type.

- `NDI`: number of direct normal components
- `NSHR`: number of direct shear components
- `NTENS`: total components $=$`NDI`$+$`NSHR`

ABAQUS variables for UMAT

Most relevant variables for UMATS for soils:

- `DDSDDE(NTENS,NTENS)`: Jacobian matrix.
- `STRESS(NTENS)`: Stress at the beginning of the increment.
- `STATEV(NSTATV)`: State variables at beginning of the increment.
- `NSTATV`: number of state variables.
- `PNEWDT`: ratio of the suggested `DTIME` with respect to the current `DTIME`.
- `STRAN(NTENS)`: Total strains at beginning of the increment.

ABAQUS variables for UMAT

Most relevant variables for UMATS for soils:

- `DSTRAN(NTENS)`: Increment of strain for current iteration.
- `DTIME`: Increment of time.
- `TIME(2)`: Time at beginning and at the end of increment.
- `PROPS(NPROPS)`: Material parameters
- `NPROPS`: Number of material parameters.
- `COORDS`: Coordinates of current integration point.
- `NOEL`: Number of the current element.

UMAT sintax

Always the subroutine begins with:

```
*USER SUBROUTINES
      SUBROUTINE UMAT(STRESS,STATEV,DDSDDE,SSE,SPD,SCD,
     1 RPL,DDSDDT,DRPLDE,DRPLDT,
     2 STRAN,DSTRAN,TIME,DTIME,TEMP,DTEMP,PREDEF,DPRED,CMNAME,
     3 NDI,NSHR,NTENS,NSTATEV,PROPS,NPROPS,COORDS,DROT,PNEWDT,
     4 CELENT,DFGRD0,DFGRD1,NOEL,NPT,LAYER,KSPT,KSTEP,KINC)
C
      INCLUDE 'ABA_PARAM.INC'
   !   implicit real(8) (a-h,o-z)
C
      CHARACTER*80 CMNAME
      DIMENSION STRESS(NTENS),STATEV(NSTATEV),
     1 DDSDDE(NTENS,NTENS),DDSDDT(NTENS),DRPLDE(NTENS),
     2 STRAN(NTENS),DSTRAN(NTENS),TIME(2),PREDEF(1),DPRED(1),
     3 PROPS(NPROPS),COORDS(3),DROT(3,3),DFGRD0(3,3),DFGRD1(3,3)
```

Parameters:

1. E YOUNG's modulus.
2. ν POISSON ratio.

State variables

1. Type of elasticity, 1=linear, 2=CAM-CLAY, 3=Richart.
2. e void ratio.
3. p_{atm} atmospheric pressure (RICHART'S).
4. G_0 initial shear modulus (RICHART'S).
5. κ swelling index according to CAM-CLAY.

UMAT sintax

Declare your own variables:

```
      double precision T(6), DeltaE(6), ELMOD(6,6),G0, void, patm
     1 K, p, max, kappa, lambda, mu
      integer model
      PARAMETER (ONE=1.0D0, TWO=2.0D0)
```

Initialize variables:

```
C ---------------------------------------------------------------
C     Initializes in fixed R6 arrays.
      CALL Initial(NTENS, NDI, NSHR, T, STRESS, DeltaE, DSTRAN)
      p=(T(1)+T(2)+T(3))/3.0d0
C ---------------------------------------------------------------
```

UMAT sintax

Establish type of elasticity:

```
C       Type of elasticity, ==1 1==linear, 2==Cam-Clay, 3==Richart
        model=STATEV(1)
C       Void ratio
        void=STATEV(2)
C       Compression index cam-clay
        kappa=STATEV(5)
C       Initial shear modulus
        G0=0.0d0
C       Atmospheric pressure
        patm=0.0d0
C       Young modulus
        E=PROPS(1)
C       Poisson ratio
        ANU=PROPS(2)
C       Maximum ratio for E or G allowed.
        max=5000.0d0
```

UMAT sintax

Establish type of elasticity:

```
C ---------------------------------------------------------------
      if (model==2) then
C     Cam-Clay elasticity
      E=3.0d0*(1.0d0-2.0d0*ANU)*(1+void)*abs(p/kappa)
      if (E<PROPS(1)) then
      E=PROPS(1)
      endif
      endif
```

UMAT sintax

Establish type of elasticity (continuation):

```
C ----------------------------------------------------------------
      if (model==3) then
C     Hipoelasticity Richart
C     atmospheric pressure
      patm=abs(STATEV(3))
C     Initial shear modulus
      G0=abs(STATEV(4))
C     Current shear modulus according to Richart
      G=G0*patm*(2.97d0-void)**2.0d0/(1.0d0+void)*sqrt(abs(p/patm))
C     Limits of shear modulus
      if (G<G0) then
      G=G0
      endif
      if (G>max*G0) then
      G=max*G0
      endif
C     Bulk modulus and Young modulus
      K=G*2.0d0*(1.0d0+ANU)/(3.0d0*(1.0d0-2.0d0*ANU))
      E=3.0d0*K*(1.0d0-ANU)
      endif
```

UMAT sintax

Tangent moduli in 6x6 matrix:

```
      lambda=ANU*E/ (ONE+ANU)/(ONE-TWO*ANU)
      mu=E/TWO/(ONE+ANU)
      ELMOD=0.0d0
C ---------------------------------------------------------------
C     Elastic moduli
      ELMOD(1,1)=lambda+TWO*mu
      ELMOD(2,2)=ELMOD(1,1)
      ELMOD(3,3)=ELMOD(1,1)
      ELMOD(4,4)=mu
      ELMOD(5,5)=mu
      ELMOD(6,6)=mu
      ELMOD(1,2)=lambda
      ELMOD(1,3)=lambda
      ELMOD(2,3)=lambda
      ELMOD(2,1)=ELMOD(1,2)
      ELMOD(3,1)=ELMOD(1,3)
      ELMOD(3,2)=ELMOD(2,3)
```

UMAT sintax

Next stress in 6x1 vector

```
C      Euler's forward integration scheme
       DO I=1,NTENS
         DO J=1,NTENS
         T(I)=T(I)+ELMOD(I,J)*DeltaE(J)
         ENDDO
       ENDDO
C      Subroutine for filling the stress and Jacobian matrix
       Call Solution(NTENS, NDI, NSHR, T, STRESS, ELMOD, DDSDDE)
```

The `Solution` subroutine transform the stress vector `T` and the jacobian `ELMOD` according to the real size using `NTENS,NDI,NSHR`.

UMAT sintax

State variables computations and end of subroutine UMAT

```
C      Void ratio
       void=void+(1.+void)*(DeltaE(1)+DeltaE(2)+DeltaE(3))
C      Type of model
       STATEV(1)=model
C      Void ratio
       STATEV(2)=void
C      Atmospheric mean pressure
       STATEV(3)=patm
C      Initial Shear modulus
       STATEV(4)=G0
C      Atmospheric mean pressure
       STATEV(5)=kappa
C ----------------------------------------------------------------
       RETURN
       END
```

UMAT sintax

Subroutine Initial:

```
C ---------------------------------------------------------------
      SUBROUTINE Initial(NTENS, NDI, NSHR, T, STRESS, DeltaE, DSTRAN)
      integer NTENS, NDI, NSHR, i, j, k, l
      double precision T(6), ELMOD(6,6), STRESS(NTENS),
     1 DSTRAN(NTENS), DeltaE(6)
C     Stress and delta strain in a 6-vector
      T=0.0d0
      T(1:ndi)=STRESS(1:ndi)
      T(4:3+NSHR)=STRESS(ndi+1:NTENS)
      DeltaE=0.0d0
      DeltaE(1:ndi)=DSTRAN(1:ndi)
      DeltaE(4:3+NSHR)=DSTRAN(ndi+1:NTENS)
      Return
      END SUBROUTINE Initial
C ---------------------------------------------------------------
```

UMAT sintax

Subroutine Solution:

```
C ---------------------------------------------------------------
C ---------------------------------------------------------------
       SUBROUTINE Solution(NTENS, NDI, NSHR, T, STRESS, ELMOD, DDSDDE)
       integer NTENS, NDI, NSHR
       double precision T(6), ELMOD(6,6), STRESS(NTENS),
     1 DDSDDE(NTENS,NTENS)
C ---------------------------------------------------------------
      STRESS(1:ndi)=T(1:ndi)
      STRESS(ndi+1:NTENS)=T(4:NSHR)
```

UMAT sintax

Subroutine Solution (continuation):

```
    do i=1,ndi
    do j=1,ndi
      ddsdde(i,j)=ELMOD(i,j)
    enddo
  enddo
  do i=ndi+1,ndi+nshr
    do j=1,ndi
      ddsdde(i,j)=ELMOD(i,j)
    enddo
  enddo
  do i=1,ndi
    do j=ndi+1,ndi+nshr
      ddsdde(i,j)=ELMOD(i,j)
    enddo
  enddo
  do i=ndi+1,ndi+nshr
    do j=ndi+1,ndi+nshr
      ddsdde(i,j)=ELMOD(i,j)
    enddo
  enddo
   END SUBROUTINE Solution
```

Modified CAM-CLAY MCC plasticity model

MCC plasticity model

Yield surface:

$$f(q,p) := \frac{q^2}{M^2} + p(p - p_c) \tag{10}$$

where M is the material parameter which defines the slope of the critic state line, p_c the overconsolidated mean stress and q the deviator stress. see [Schofield et.al., 1968].

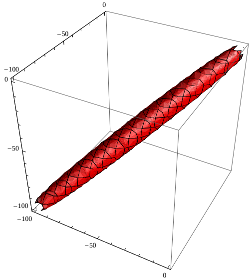


Figure: CAM CLAY surface

Hardening law

The hardening law is taken by the p_c.

$$\dot{p}_c = \vartheta p_c \dot{\varepsilon}_v^p, \qquad \dot{\varepsilon}_v^p = \mathrm{tr}(\dot{\varepsilon}^p), \qquad \vartheta = \frac{1+e}{\lambda^* - \kappa^*} \tag{11}$$

where

- ε_v^p the plastic volumetric strain
- ε^p the plastic strain
- e the void ratio.
- λ^* the compression index (parameter).
- κ^* the swelling index (parameter).

Associative flow rule

$$\frac{\partial f(p,q)}{\partial \mathbf{T}} = \frac{\partial f}{\partial p}\frac{\partial p}{\partial \mathbf{T}} + \frac{\partial f}{\partial q}\frac{\partial q}{\partial \mathbf{T}} \tag{12}$$

Summarizing,

$$\mathbf{D}^p = \gamma \left[(2p - p_c)\frac{1}{3}\mathbf{1} + \sqrt{\frac{3}{2}}\frac{2q}{M^2}\overrightarrow{\mathbf{T}}^* \right] \tag{13}$$

being,

- $\mathbf{D}^p$ the rate of plastic strain tensor.
- $\overrightarrow{\mathbf{T}}^* = \dfrac{\mathbf{T} - \frac{1}{3}\mathbf{1}}{\| \mathbf{T} - \frac{1}{3}\mathbf{1} \|}$
- $\mathbf{1}$ the unit second order tensor.

Parameters:

1. λ^* Compression index
2. κ^* Swelling index.
3. p_{c0} Initial mean overconsolidated stress.
4. M Critical state line slope.
5. ν POISSON ratio.

State variables

1. e void ratio.
2. ε^p_{11} plastic strain component 11.
3. ε^p_{22} plastic strain component 22.
4. ε^p_{33} plastic strain component 33.
5. ε^p_{12} plastic strain component 12.
6. ε^p_{13} plastic strain component 13.
7. ε^p_{23} plastic strain component 23.
8. p_c mean overconsolidated stress.

Implicit Integration

See [Borja, R. I. et.al., 1990] for implicit integration. Some details of the integration scheme:

- Implicit integration using the closest point projection. A return mapping algorithm is used.
- The void ratio e is assumed to be constant in an integration increment.
- 2 nested NEWTON-RAPHSON algorithm needed in order to find the consistency parameter $\Delta\gamma$ and the mean consolidation stress p_c.
- A consistent Jacobian is used in order to preserve the asymptotic rate of quadratic global convergence[Borja, R. I. et.al., 1990].

The following frames presents the implicit integration scheme for the MCC model. Open the archive CamClay.f and follow the frames.

Implicit Integration

Over the plastic range,

$$\mathrm{P_c}\left(p_{c,n+1}\right) = p_{c,n} \exp\left[\vartheta \Delta\gamma \left(\frac{2p^{tr}_{n+1} - p_{c,n+1}}{1 + 2\Delta\gamma k} \right)\right] - p_{c,n+1} = 0 \tag{14}$$

and

$$\frac{\partial f_{n+1}}{\Delta\gamma_{n+1}} = \frac{\partial f}{\partial p}\frac{\partial p}{\partial \Delta\gamma} + \frac{\partial f}{\partial q}\frac{\partial q}{\partial \Delta\gamma} + \frac{\partial f}{\partial p_c}\frac{\partial p_c}{\partial \Delta\gamma}\bigg|_{n+1} \tag{15}$$

where,

$$\frac{\partial p_{n+1}}{\partial \Delta\gamma_{n+1}} = -k \frac{2p - p_c}{1 + (2k + \vartheta p_c)\Delta\gamma}\bigg|_{n+1} \tag{16}$$

$$\frac{\partial q_{n+1}}{\partial \Delta\gamma_{n+1}} = -\frac{q}{\Delta\gamma + M^2/6G}\bigg|_{n+1} \tag{17}$$

$$\frac{\partial p_{c,n+1}}{\partial \Delta\gamma_{n+1}} = \vartheta p_c \frac{(2p - p_c)}{1 + (2k + \vartheta p_c)\Delta\gamma}\bigg|_{n+1} \tag{18}$$

Implicit Integration

So, 2 nested Newton algorithm should be implemented.

1. a trial elastic step is made.
2. if $f < 0$ then the elastic step is ok.
3. if $f >= 0$ a plastic corrector step should be implemented.
   - A global NEWTON algorithm for $f(q, p, p_c, \Delta\gamma) = 0$ is made.
   - a local NEWTON algorithm is made for p_c.
   - q and p are computed.
   - $\Delta\gamma$ is computed.
4. After q, p and $\Delta\gamma$ are computed, the plastic strains are computed.
5. Stress $\mathbf{T}$ is computed.
6. the Jacobian $[\mathbb{J}] = [\mathrm{DDSDDE}]$ is computed.

Implicit Integration

Definition of constants

$$G = \frac{E}{2(1+\nu)} \qquad k = \frac{E}{3(1-2\nu)}$$

$$[1^a] = [1_a] = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}^T$$

$$[C^{e,ab}] = \begin{bmatrix} \bar{\lambda}+2G & \bar{\lambda} & \bar{\lambda} & 0 & 0 & 0 \\ & \bar{\lambda}+2G & \bar{\lambda} & 0 & 0 & 0 \\ & & \bar{\lambda}+2G & 0 & 0 & 0 \\ & & & G & 0 & 0 \\ & sym & & & G & 0 \\ & & & & & G \end{bmatrix}$$

with $\bar{\lambda} = k - \frac{2}{3}G$, $(G = \mu)$

Implicit Integration

Definition of constants (continuation)

$$[I^{ab}] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ & 1 & 0 & 0 & 0 & 0 \\ & & 1 & 0 & 0 & 0 \\ & & & \frac{1}{2} & 0 & 0 \\ & & sym & & \frac{1}{2} & 0 \\ & & & & & \frac{1}{2} \end{bmatrix}$$

$$[I^{dev,ab}] = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} & 0 & 0 & 0 \\ & \frac{2}{3} & -\frac{1}{3} & 0 & 0 & 0 \\ & & \frac{2}{3} & 0 & 0 & 0 \\ & & & \frac{1}{2} & 0 & 0 \\ & & sym & & \frac{1}{2} & 0 \\ & & & & & \frac{1}{2} \end{bmatrix}
\qquad
[I_b^{dev,a}] = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} & 0 & 0 & 0 \\ & \frac{2}{3} & -\frac{1}{3} & 0 & 0 & 0 \\ & & \frac{2}{3} & 0 & 0 & 0 \\ & & & 1 & 0 & 0 \\ & & sym & & 1 & 0 \\ & & & & & 1 \end{bmatrix}$$

Implicit Integration

Definition of initial conditions for state variables p_c, ε^p and stress $\mathbf{T}$.

Make a trial elastic step

$$
\begin{aligned}
[T_{n+1}^{tr,a}] &= [T_{n+1}^{a}] + [C^{e,ab}][\Delta\varepsilon_{n+1,b}] \\
\| \, [T_{n+1}^{tr,a}] \, \| &= \mathrm{NORM}[T_{n+1}^{tr,a}] \\
p_{n+1}^{tr} &= \frac{1}{3}\mathrm{TRACE}\left[[T_{n+1}^{tr,a}]\right] \\
[T_{n+1}^{*,tr,a}] &= [I^{dev,a}{}_{b}][T_{n+1}^{tr,b}] \\
\| \, [T_{n+1}^{*,tr,a}] \, \| &= \mathrm{NORM}\left[[T_{n+1}^{*,tr,a}]\right] \\
q_{n+1}^{tr} &= \sqrt{\frac{3}{2}} \, \| \, [T_{n+1}^{*,tr,a}] \, \|
\end{aligned}
$$

Implicit Integration

Check the yielding function.

$$f^{tr}_{n+1} = \frac{(q^{tr}_{n+1})^2}{M^2} + p^{tr}_{n+1}(p^{tr}_{n+1} - p_{c,n+1})$$

if $f^{tr}_{n+1} < 0$, trial elastic step is ok!. Then,

$$\begin{aligned}
[T^a_{n+1}] &= [T^{tr,a}_{n+1}] \\
p_{n+1} &= p^{tr}_{n+1} \\
q_{n+1} &= q^{tr}_{n+1} \\
p_{c,n+1} &= p_{c,n} \\
[\varepsilon^p_{n+1,a}] &= [\varepsilon^p_{n,a}] \\
[C^{ep,ab}] &= [C^{e,ab}] \\
e_{n+1} &= e_n + (1 + e_n)\mathrm{TRACE}[\Delta\varepsilon_{n+1,b}]
\end{aligned}$$

Implicit Integration

if $f^{tr}_{n+1} \geq 0$, then plastic corrector step, define initial values and tolerances for iterations, and maximum number of iterations,

$$\Delta\gamma_{n+1} = 0$$
$$e_{n+1} = e_n + (1 + e_n)\mathrm{TRACE}[\Delta\varepsilon_{n+1,b}]$$
$$\vartheta_{n+1} = (1 + e_{n+1})/(\lambda^* + \kappa^*)$$
$$\mathrm{Tolf} := 10^{-7} p_{c,n+1} \quad \text{(suggestion)}$$

$$\mathrm{TolP_c} := 10^{-7} \parallel [T^{*,tr,a}_{n+1}] \parallel \quad \text{(suggestion)}$$
$$\max = 50 \quad \text{(suggestion)}$$

Implicit Integration

Making NEWTON AND RAPHSON algorithm for p_c

↱ for $i = 1$, until $i = \max$

↱ for $j = 1$, until $j = \max$

$$P_c = p_{c,n} \exp\left[\vartheta_{n+1}\Delta\gamma_{n+1}\left(\frac{2p^{tr}_{n+1} - p_{c,n+1}}{1 + 2\Delta\gamma_{n+1}k}\right)\right] - p_{c,n+1} = 0$$

↱ if $\| P_c \| < \mathrm{TolP_c}$ exit for. else:

$$\frac{\partial P_c}{\partial p_{c,n+1}} = -p_{c,n} \exp\left[\vartheta_{n+1}\Delta\gamma_{n+1}\left(\frac{2p^{tr}_{n+1} - p_{c,n+1}}{1 + 2\Delta\gamma_{n+1}k}\right)\right]\frac{(\Delta\gamma_{n+1})(\vartheta_{n+1})}{2k\Delta\gamma_{n+1} + 1} - 1$$

$$\underbrace{p_{c,n+1}}_{atj+1} = \underbrace{p_{c,n+1}}_{atj} - \frac{P_c}{\dfrac{\partial P_c}{\partial p_{c,n+1}}}$$

↱ end for (j)

Implicit Integration

Making NEWTON AND RAPHSON for yielding funciton f

- calculate current p_{n+1} and q_{n+1}

$$p_{n+1} = \frac{p^{tr}_{n+1} + \Delta\gamma k p_{c,n+1}}{1 + 2\Delta\gamma k}$$

$$q_{n+1} = \frac{q^{tr}_{n+1}}{1 + 6G\Delta\gamma_{n+1}/M^2}$$

- calculate current yielding function f_{n+1}

$$f_{n+1} = \frac{{q_{n+1}}^2}{M^2} + p_{n+1}(p_{n+1} - p_{c,n+1}) \quad (19)$$

Implicit Integration

Making NEWTON AND RAPHSON for yielding funciton f (continuation)

↱ if $\| f_{n+1} \| < \mathrm{Tolf}$ exit for. else:

$$\frac{\partial f_{n+1}}{\partial p_{n+1}} = 2p_{n+1} - p_{c,n+1}$$

$$\frac{\partial f_{n+1}}{\partial q_{n+1}} = 2q_{n+1}/M^2$$

$$\frac{\partial f_{n+1}}{\partial p_{c,n+1}} = -p_{n+1}$$

$$\frac{\partial p_{n+1}}{\partial \Delta\gamma_{n+1}} = -k \left.\frac{2p - p_c}{1 + (2k + \vartheta p_c)\Delta\gamma}\right|_{n+1}$$

$$\frac{\partial q_{n+1}}{\partial \Delta\gamma_{n+1}} = -\left.\frac{q}{\Delta\gamma + M^2/6G}\right|_{n+1}$$

$$\frac{\partial p_{c,n+1}}{\partial \Delta\gamma_{n+1}} = \vartheta p_c \left.\frac{(2p - p_c)}{1 + (2k + \vartheta p_c)\Delta\gamma}\right|_{n+1}$$

and...

Implicit Integration

Making NEWTON AND RAPHSON for yielding funciton f (continuation)

and...

$$\frac{\partial f_{n+1}}{\Delta\gamma_{n+1}} = \frac{\partial f}{\partial p}\frac{\partial p}{\partial \Delta\gamma} + \frac{\partial f}{\partial q}\frac{\partial q}{\partial \Delta\gamma} + \frac{\partial f}{\partial p_c}\frac{\partial p_c}{\partial \Delta\gamma}\bigg|_{n+1}$$

$$\underbrace{\Delta\gamma_{n+1}}_{ati+1} = \underbrace{\Delta\gamma_{n+1}}_{ati} - \frac{f_{n+1}}{\frac{\partial f_{n+1}}{\Delta\gamma_{n+1}}}$$

↱ end for (i)

Implicit Integration

Plastic strains and stresses

Plastic strains are calculated as follows,

$$[\overrightarrow{T}^{*,tr,a}_{n+1}] = \frac{[T^{*,tr,a}_{n+1}]}{\| [T^{*,tr,a}_{n+1}] \|}$$

$$[\overrightarrow{B}_{n+1,a}] = (2p_{n+1} - p_{c,n+1})\frac{1}{3}[1_a] + \sqrt{\frac{3}{2}}\frac{2q_{n+1}}{M^2}[\overrightarrow{T}^{*,tr,a}_{n+1}]$$

$$[\Delta\varepsilon^{p}_{n+1,a}] = \Delta\gamma_{n+1} * [\overrightarrow{B}_{n+1,a}]$$

$$[\varepsilon^{p}_{a,n+1}] = [\varepsilon^{p}_{a,n}] + [\Delta\varepsilon^{p}_{n+1,a}]$$

↱ The stress is calculated according to the return algorithm scheme,

$$[T^{*,a}_{n+1}] = [T^{*,tr,a}_{n+1}] - [C^{e,ab}][\Delta\varepsilon^{p}_{n+1,b}]$$

Implicit Integration

Finally the consistent elastoplastic moduli is given by,

$$f_1 = 2G\sqrt{\frac{2}{3}}\frac{q_{n+1}}{\parallel [T^{*,tr,a}_{n+1}] \parallel}$$

$$f_2 = \left[k(a_1 + a_2 d_1) - \frac{1}{3}2G\sqrt{\frac{2}{3}}\frac{q_{n+1}}{\parallel [T^{*,tr,a}_{n+1}] \parallel}\right]$$

$$f_3 = k(a_2 d_2)$$

$$f_4 = 2G\sqrt{\frac{2}{3}}(c_2 d_1)$$

$$f_5 = 2G\left[\sqrt{\frac{2}{3}}(c_1 + c_2 d_2) - \sqrt{\frac{2}{3}}\frac{q_{n+1}}{\parallel [T^{*,tr,a}_{n+1}] \parallel}\right]$$

$$[C^{ep,ab}_{n+1}] = f_1[I^{ab}] + f_2[1_a][1_a]^T + f_3[1_a][\overrightarrow{T}^{*,tr,a}_{n+1}]^T + f_4[\overrightarrow{T}^{*,tr,a}_{n+1}][1_a]^T + f_5[\overrightarrow{T}^{*,tr,a}_{n+1}][\overrightarrow{T}^{*,tr,a}_{n+1}]^T$$

UMAT ready!! Response envelopes using INCREMENTAL DRIVER (see [Niemunis, 2008]).

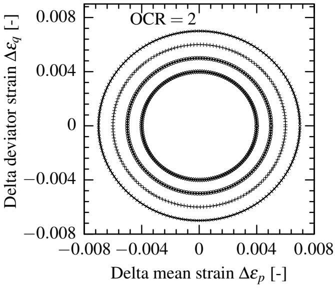


Figure: Reponse envelopes with MCC model

Hypoplasticity model

Constitutive model

- Done for granular materials.
- Assumes non-linear behaviour without an elastic range.
- Time-independent model formulated in a rate-form constitutive equation.
- $\overset{\circ}{\mathbf{T}} = \mathbf{h}(\mathbf{T}, e, \mathbf{D})$, being $\mathbf{D}$ the strain rate tensor and $\overset{\circ}{\mathbf{T}}$ the ZAREMBA-JAUMMAN co-rotational stress rate tensor.
- Model from [Wolffersdorff, V., 1996].
- BAUER's law for compression.
- MATSUOKA-NAKAI surface as the limit surface.

Constitutive model

$$\overset{\circ}{\mathbf{T}} = f_s \left[\mathbb{L} : \mathbf{D} + f_d \mathbf{N} \parallel \mathbf{D} \parallel \right] \tag{20}$$

$\mathbb{L}$ is the fourth order hypoelastic tensor and $\mathbf{N}$ is a second order tensor non-linear with the stretching tensor $\mathbf{D}$.

$$\mathbb{L} = \frac{f_s}{\hat{\mathbf{T}} : \hat{\mathbf{T}}} a^2 \left[\left(\frac{F}{a} \right)^2 \mathbb{I} + \hat{\mathbf{T}} \otimes \hat{\mathbf{T}} \right] \tag{21}$$

$$\mathbf{N} = \frac{f_s}{\hat{\mathbf{T}} : \hat{\mathbf{T}}} a^2 \left[\frac{F}{a} \right] [\hat{\mathbf{T}} + \hat{\mathbf{T}}^*] \tag{22}$$

with,

$$\hat{\mathbf{T}} = \frac{\mathbf{T}}{\mathrm{Tr}\,[\mathbf{T}]} \tag{23}$$

$$\hat{\mathbf{T}}^* = \frac{\mathbf{T} - \mathrm{Tr}\,[\mathbf{T}]\,\mathbf{1}}{\mathrm{Tr}\,[\mathbf{T}]} \tag{24}$$

$$I_{ijkl} = \frac{1}{2}(\delta_{ik}\delta_{jl} + \delta il \delta_{jk}) \tag{25}$$

Constitutive model

$$\overset{\circ}{\mathbf{T}} = f_s \left[\mathbb{L} : \mathbf{D} + f_d \mathbf{N} \parallel \mathbf{D} \parallel \right] \tag{26}$$

f_d is the density factor, f_e is the piknotropy factor, and f_b is the barotropy factor: With,

$$f_e = \left(\frac{e_c}{e} \right) \tag{27}$$

$$f_d = \left(\frac{e - e_d}{e_c - e_d} \right)^{\alpha} \tag{28}$$

$$f_b = \left(\frac{e_{i0}}{e_{c0}} \right)^{\beta} \frac{h_s}{n} \frac{1 + e_i}{e_i} \left(\frac{-\mathrm{tr}\mathbf{T}}{h_s} \right)^{1-n} \left[3 + a^2 - a\sqrt{3} \left(\frac{e_{io} - e_{d0}}{e_{c0} - e_{d0}} \right) \alpha \right]^{-1} \tag{29}$$

$$f_s = f_e f_b \tag{30}$$

e_c, e_d and e_i are the critical, minimum and maximum void ratio according to the BAUER's law. h_s, n α and β are materials parameters. a is function of the critical friction angle φ_c.

BAUER's law for isotropic compression

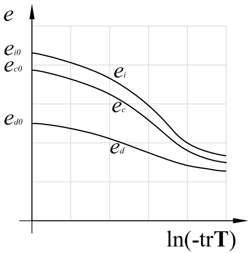


Figure: BAUER's law for isotropic compression

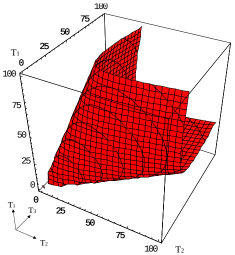
$$\frac{e_i}{e_{i0}} = \frac{e_c}{e_{c0}} = \frac{e_d}{e_{d0}} = \exp\left[-\left(-\frac{\mathrm{tr}\mathbf{T}}{h_s}\right)^n\right] \qquad (31)$$

MATSUOKA-NAKAI's limit surface

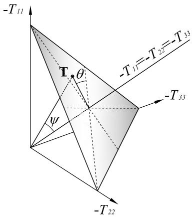
$$y_{M-N}(\mathbf{T}) = -\frac{I_1 I_2}{I_3} + \frac{9-\sin^2(\varphi_c)}{-1+\sin^2(\varphi_c)} = 0 \tag{32}$$

with,

$$\begin{aligned} I_1 &= \mathrm{tr}\mathbf{T} \\ I_2 &= \frac{1}{2}[\mathbf{T}:\mathbf{T}-(I_1)^2] \\ I_3 &= \det(\mathbf{T}) \end{aligned} \tag{33}$$



MATSUOKA-NAKAI's limit surface



$$a = \frac{\sqrt{3}(3 - \sin(\varphi_c))}{2\sqrt{2}\sin(\varphi_c)} \tag{34}$$

$$F = \sqrt{\frac{1}{8}\tan^2(\psi) + \frac{2 - \tan^2(\psi)}{2 + \sqrt{2}\tan(\psi)\cos(3\theta)}} - \frac{1}{2\sqrt{2}}\tan(\psi) \tag{35}$$

$$\tan(\psi) = \sqrt{3} \parallel \hat{\mathbf{T}}^* \parallel \tag{36}$$

$$\cos(3\theta) = -\sqrt{6}\frac{\mathrm{tr}(\hat{\mathbf{T}}^{*\mathbf{3}})}{[\mathrm{tr}(\hat{\mathbf{T}}^*)]^{3/2}} \tag{37}$$

Explicit implementation

- ABAQUS$^{\mathrm{TM}}$establishes a co-rotational frame for UMATs. It is not necessary to make correction by sc Zaremba-Jaumann $\overset{\circ}{\mathbf{T}}$ proposal.
- Eulers forward method with not error control implemented.
- The Jacobian $\mathbb{J}$ is computed according to [Fellin, W. et.al. 2002].
- Tensor $\mathbf{D}$ is calculated using the approximation $\mathbf{D} = \dfrac{\Delta\varepsilon}{\Delta t}$

Jacobian by numeric differentiation

Jacobian:

$$\mathbb{J} = \frac{\partial \Delta \mathbf{T}}{\partial \Delta \boldsymbol{\varepsilon}} \tag{38}$$

The hypoplastic law establishes the following relation for the rate of $\mathbf{T}$:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{T} = \mathbf{h}(\mathbf{T}, \mathbf{D}, \mathbf{Q}) \tag{39}$$

where $\mathbf{Q}$ denotes the additional state variables.
Introducing,

$$B_{ij,kl} = \frac{\partial T_{ij}}{\partial D_{kl}} = \Delta t \cdot \frac{\partial \Delta T_{ij}}{\partial \Delta \varepsilon_{ij}} \tag{40}$$

This means that an aproximation of the Jacobian is:

$$\mathbb{J} \simeq \frac{B_{ij,kl}}{\Delta t} \tag{41}$$

Jacobian by numeric differentiation

Developing numerically $B_{ij,kl}$:

The following is a numeric differentiation of $\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{B}_{ij}$:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{B}_{ij} = \frac{1}{\vartheta}\left[\mathbf{h}(\mathbf{T}+\vartheta\mathbf{B}_{ij}, \mathbf{D}+\vartheta\mathbf{V}_{ij}, \mathbf{Q}+\vartheta\mathbf{G}_{ij}) - \mathbf{h}(\mathbf{T}, \mathbf{D}, \mathbf{Q})\right] \tag{42}$$

where

$$\mathbf{V}_{ij} = \delta_{ik}\delta_{jl} \tag{43}$$

for example,

$$[\mathbf{V}_{11}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{44}$$

and

$$\mathbf{G}_{ij} = \frac{\partial \mathbf{Q}}{\partial \mathbf{D}} \tag{45}$$

Jacobian by numeric differentiation

Arrange of the super-vector solution

For programming the super-vector $\mathbf{y}$ will be employed:

$$\begin{aligned}
[y_1, y_2, ..., y_6]^T &= [T_{11}, T_{22}, T_{33}, T_{12}, T_{13}, T_{23}]^T \\
[y_7, ..., y_{42}]^T &= [B_{11,11}, B_{11,22}, ..., B_{23,23}]^T \qquad \text{(36 components)} \\
[y_{43}, y_{44}]^T &= [\varphi_m, e]^T \\
[y_{45}, ..., y_n]^T &= [G_{e,11}, G_{\varphi_m,11}, G_{e,22}, G_{\varphi_m,22}, ..., G_{e,23}, G_{\varphi_m,23}]^T
\end{aligned} \tag{46}$$

and a rate super-vector $\dot{\mathbf{y}}$ such that, an EULER forward integration can be computed as:

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t \cdot \dot{\mathbf{y}} \tag{47}$$

References

- Chivers. I.D. AND Sleightholme. J
  Introduction to Programming with Fortran.
  Springer. London 2006.
- Niemunis. A.
  Incremental Driver, user's manual.
  University of Karlsruhe KIT,2008.
- Hibbit, Karlsson and Sorensen Inc
  Abaqus 5.6 Theory Manual.
  Simulia, 1995.
- Lai Michael, Rubin David, Krempl Erhard
  Continuum Mechanics.
  Third Edition. Butterworth Heinemann. 1993.
- Schofield. A AND Wroth. P.
  Critical state soil mechanics.
  Mc Graw Hill. London. 1968.

References

Borja, R. I., AND Seung, R. L.
Cam Clay plasticity, Part 1: Implicit integration of elasto-plastic constitutive relations.
Computer methods in applied mechanics and engineering.(78), pags 49-72. 1990.

Wolffersdorff, V.
A hypoplastic relation for granular materials with a predefined limit state surface.
Mechanics of cohesive-frictional materials.(1), pags 251-271. 1996.

Fellin. V., Ostermann. A.
Consistent tangent operators for constitutive rate equations.
Int. J. Numer. Anal. Meth. Geomech..(26), pags 1213-1233. 2002.