Algorithm Main: Forward Euler method for multi-surface plasticity

Input: σ^t, k_s^t, k_c^t, $d\epsilon^{t+1}$

Output: σ^{t+1}, k_s^{t+1}, k_c^{t+1}

$\sigma^{trial} = \sigma^t + [K]d\epsilon^{t+1}$; $F_s^t = F_s(\sigma^{trial}, k_s^t)$; $F_c^t = F_c(\sigma^{trial}, k_c^t)$

if $F_s^t \leq 0$ *and* $F_c^t \leq 0$ **then**

  $d\epsilon^e = d\epsilon^{t+1}$; $k_s^{t+1} = k_s^t$; $k_c^{t+1} = k_c^t$

else if $F_s^t > 0$ *and* $F_c^t \leq 0$ **then**

  Call **Shear mapping**

else if $F_s^t \leq 0$ *and* $F_c^t > 0$ **then**

  Call **Cap mapping**

else if $F_s^t > 0$ *and* $F_c^t > 0$ **then**

$$[H] = \begin{bmatrix} -\frac{\partial F_s}{\partial k_s}\frac{\partial k_s}{\partial \epsilon^p}\frac{\partial Q_s}{\partial \sigma} & -\frac{\partial F_s}{\partial k_s}\frac{\partial k_s}{\partial \epsilon^p}\frac{\partial Q_c}{\partial \sigma} \\ -\frac{\partial F_c}{\partial k_c}\frac{\partial k_c}{\partial \epsilon^p}\frac{\partial Q_s}{\partial \sigma} & -\frac{\partial F_c}{\partial k_c}\frac{\partial k_c}{\partial \epsilon^p}\frac{\partial Q_c}{\partial \sigma} \end{bmatrix}$$

$$[A] = \begin{bmatrix} \frac{\partial F_s}{\partial \sigma}[K]\frac{\partial Q_s}{\partial \sigma} + H_{ss} & \frac{\partial F_s}{\partial \sigma}[K]\frac{\partial Q_c}{\partial \sigma} + H_{sc} \\ \frac{\partial F_c}{\partial \sigma}[K]\frac{\partial Q_s}{\partial \sigma} + H_{cs} & \frac{\partial F_c}{\partial \sigma}[K]\frac{\partial Q_c}{\partial \sigma} + H_{cc} \end{bmatrix}$$

$$\begin{bmatrix} d\lambda_s \\ d\lambda_c \end{bmatrix} = [A]^{-1} \begin{bmatrix} \frac{\partial F_s}{\partial \sigma}[K]d\epsilon \\ \frac{\partial F_c}{\partial \sigma}[K]d\epsilon \end{bmatrix}$$

  if $d\lambda_s > 0$ *and* $d\lambda_c \leq 0$ **then**

    Call **Shear mapping**

  else if $d\lambda_s > 0$ *and* $d\lambda_c \leq 0$ **then**

    Call **Cap mapping**

  else if $d\lambda_s > 0$ *and* $d\lambda_c \leq 0$ **then**

    Concave yield surface

    Quit Algorithm

  else

    $d\epsilon^p = d\lambda_s \frac{\partial Q_s}{\partial \sigma} + d\lambda_c \frac{\partial Q_c}{\partial \sigma}$ $d\epsilon^e = d\epsilon^{t+1} - d\epsilon^p$ $dk_i^{t+1} = dk_i(d\epsilon^p)$

  end

end

$\sigma^{t+1} = \sigma^t + [K]d\epsilon^e$ $k_i^{t+1} = k_i^t + dk_i^{t+1}$

Algorithm Subroutine1: Shear mapping

Input: F_s^t, k_s^t, $d\epsilon^{t+1}$

Output: $d\epsilon^p$, $d\epsilon^e$, k_s^{t+1}

$H_{ss} = -\frac{\partial F_s}{\partial k_s}\frac{\partial k_s}{\partial \epsilon^p}\frac{\partial Q_s}{\partial \sigma}$

$d\lambda_s = \frac{\frac{\partial F_s}{\partial \sigma}[K]d\epsilon}{\frac{\partial F_s}{\partial \sigma}[K]\frac{\partial Q_s}{\partial \sigma}+H_{ss}}$

$d\epsilon^p = d\lambda_s \frac{\partial Q_s}{\partial \sigma}$

$d\epsilon^e = d\epsilon^{t+1} - d\epsilon^p$

$dk_s^{t+1} = dk_s(\epsilon^p)$

Algorithm Subroutine2: Cap mapping

Input: F_c^t, k_c^t, $d\epsilon^{t+1}$

Output: $d\epsilon^p$, $d\epsilon^e$, k_c^{t+1}

$H_{cc} = -\frac{\partial F_c}{\partial k_c}\frac{\partial k_c}{\partial \epsilon^p}\frac{\partial Q_c}{\partial \sigma}$

$d\lambda_c = \frac{\frac{\partial F_c}{\partial \sigma}[K]d\epsilon}{\frac{\partial F_c}{\partial \sigma}[K]\frac{\partial Q_c}{\partial \sigma}+H_{cc}}$

$d\epsilon^p = d\lambda_c \frac{\partial Q_c}{\partial \sigma}$

$d\epsilon^e = d\epsilon^{t+1} - d\epsilon^p$

$dk_s^{t+1} = dk_s \epsilon^p)$