Computational Mechanics

Chapter 12 Stress Update Algorithm

Return Mapping for Rate-Independent Plasticity

- Small strain/deformation simplification:
  1. Negligible difference in stress measurement
  2. Objective rate unnecessary
  3. $\boldsymbol{D}$ is the same as $\dot{\boldsymbol{\varepsilon}}$

- For large deformation, one critical change is consideration of rigid rotation (rate)

- Hypoelasto-plasticity for small strain:

$$\dot{\boldsymbol{\sigma}} = \boldsymbol{C}:\dot{\boldsymbol{\varepsilon}}^e = \boldsymbol{C}:(\dot{\boldsymbol{\varepsilon}} - \dot{\boldsymbol{\varepsilon}}^p)$$

$$\dot{\boldsymbol{\varepsilon}}^p = \dot{\lambda}\boldsymbol{r}$$

$$\dot{\boldsymbol{q}} = \dot{\lambda}\boldsymbol{h}$$

$$0 = f_{\boldsymbol{\sigma}}:\dot{\boldsymbol{\sigma}} + f_{\boldsymbol{q}} \cdot \dot{\boldsymbol{q}}$$

$$\dot{\lambda} \geq 0, \qquad f \leq 0, \qquad \dot{\lambda} f = 0$$

- Mapping for Hyperelasto-plasticity is similar with different stress and deformation (rates) measurement

- Numerical integration of rates for total values

- Goal – given $(\boldsymbol{\varepsilon}_n, \boldsymbol{\varepsilon}_n^p, \boldsymbol{q}_n)$ at time n and the strain increment $\Delta\boldsymbol{\varepsilon} = \Delta t\dot{\boldsymbol{\varepsilon}}$, compute $(\boldsymbol{\varepsilon}_{n+1}, \boldsymbol{\varepsilon}_{n+1}^p, \boldsymbol{q}_{n+1})$

Forward Euler Integration Scheme

- Consistency condition for plastic rate parameter calculation under small strain:

$$\dot{\lambda} = \frac{f_{\boldsymbol{\sigma}} : \boldsymbol{C}_{el}^{\sigma J} : \dot{\boldsymbol{\varepsilon}}}{-f_{\boldsymbol{q}} \cdot \boldsymbol{h} + f_{\boldsymbol{\sigma}} : \boldsymbol{C}_{el}^{\sigma J} : \boldsymbol{r}}$$

- Update variables at discrete time $n+1$ purely based on variables at time n:

$$\boldsymbol{\varepsilon}_{n+1} = \boldsymbol{\varepsilon}_n + \Delta\boldsymbol{\varepsilon}$$

$$\boldsymbol{\varepsilon}_{n+1}^p = \boldsymbol{\varepsilon}_n^p + \Delta\lambda_n \boldsymbol{r}_n = \boldsymbol{\varepsilon}_n^p + \dot{\lambda}_n \Delta t \boldsymbol{r}_n$$

$$\boldsymbol{q}_{n+1} = \boldsymbol{q}_n + \Delta\lambda_n \boldsymbol{h}_n$$

$$\boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}_n + \Delta\boldsymbol{\sigma} = \boldsymbol{\sigma}_n + \underline{\boldsymbol{C}_n^{ep}} : \Delta\boldsymbol{\varepsilon}$$

Elasto-plastic tangent modulus at time n

Yield condition $f_{n+1} = f(\boldsymbol{\sigma}_{n+1}, \boldsymbol{q}_{n+1}) = 0$ is often not satisfied as deformation from time n to $n+1$ is not considered!

Fully Implicit Backward Euler Scheme

- Enforce $f_{n+1} = 0$ to avoid drift from yield surface

- Calculation with variables at time $n + 1$:

$$\boldsymbol{\varepsilon}_{n+1} = \boldsymbol{\varepsilon}_n + \Delta\boldsymbol{\varepsilon}$$

$$\boldsymbol{\varepsilon}^p_{n+1} = \boldsymbol{\varepsilon}^p_n + \Delta\lambda_{n+1}\boldsymbol{r}_{n+1}$$

$$\boldsymbol{q}_{n+1} = \boldsymbol{q}_n + \Delta\lambda_{n+1}\boldsymbol{h}_{n+1}$$

$$\boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}_n + \boldsymbol{C}^{ep}_{n+1}:\Delta\boldsymbol{\varepsilon}$$

$$f_{n+1} = f(\boldsymbol{\sigma}_{n+1}, \boldsymbol{q}_{n+1}) = 0$$ – Requirement at $n + 1$

- Challenge – obtain right values of variables at time

Meaning of Fully Implicit Backward Euler Scheme

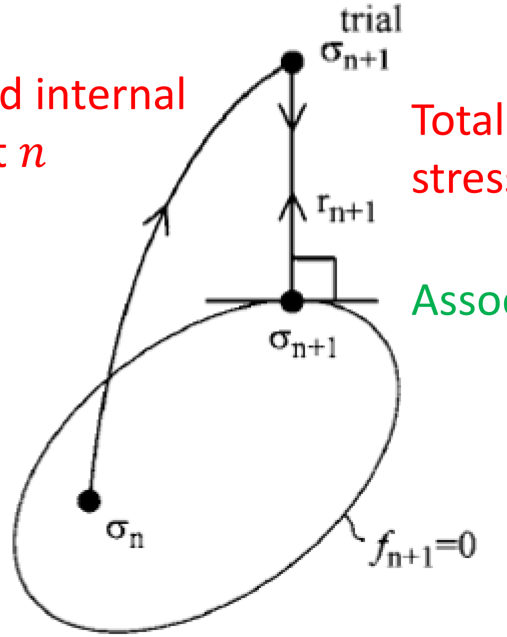
$$\Delta\boldsymbol{\varepsilon}_{n+1}^{p} = \boldsymbol{\varepsilon}_{n+1}^{p} - \boldsymbol{\varepsilon}_{n}^{p} = \Delta\lambda_{n+1}\boldsymbol{r}_{n+1}$$

$$\Rightarrow \boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}_n + \boldsymbol{C}:\left(\Delta\boldsymbol{\varepsilon} - \Delta\boldsymbol{\varepsilon}_{n+1}^{p}\right) = (\boldsymbol{\sigma}_n + \boldsymbol{C}:\Delta\boldsymbol{\varepsilon}) - \boldsymbol{C}:\Delta\boldsymbol{\varepsilon}_{n+1}^{p} = \underline{(\boldsymbol{\sigma}_n + \boldsymbol{C}:\Delta\boldsymbol{\varepsilon})} - \underline{\Delta\lambda_{n+1}\boldsymbol{C}:\boldsymbol{r}_{n+1}}$$

Trial stress of elastic predictor $\boldsymbol{\sigma}_{n+1}^{trial}$

Plastic corrector for plastic flow

- Plastic corrector returns overshoot trial stress onto yield surface along the direction of $\boldsymbol{r}_{n+1}$



Plastic strain and internal variables stay at n

Total strain fixed, correction of stress $\Delta\boldsymbol{\sigma}_{n+1} = -\Delta\lambda_{n+1}\boldsymbol{C}:\boldsymbol{r}_{n+1}$

Associative flow rule

Introduction of Newton's Method

- Numerical solution method for nonlinear algebraic equations, such as $f(x) = 0$:
  1. Selection of initial trial root:
  $$x^{(0)} = 0\ (can\ be\ other\ values)$$
  2. Iteration via linearization (k is the iteration number):
  $$\underline{f\left(x^{(k)}\right)} + f'\left(x^{(k)}\right)\delta x^{(k)} = 0$$
  Denoted as $f^{(k)}$ for simplification
  $$x^{(k+1)} = x^{(k)} + \delta x^{(k)}$$
  3. Solved once convergence condition met:
  $$x^{(k+1)} - x^{(k)} = \delta x^{(k)} < \Delta$$

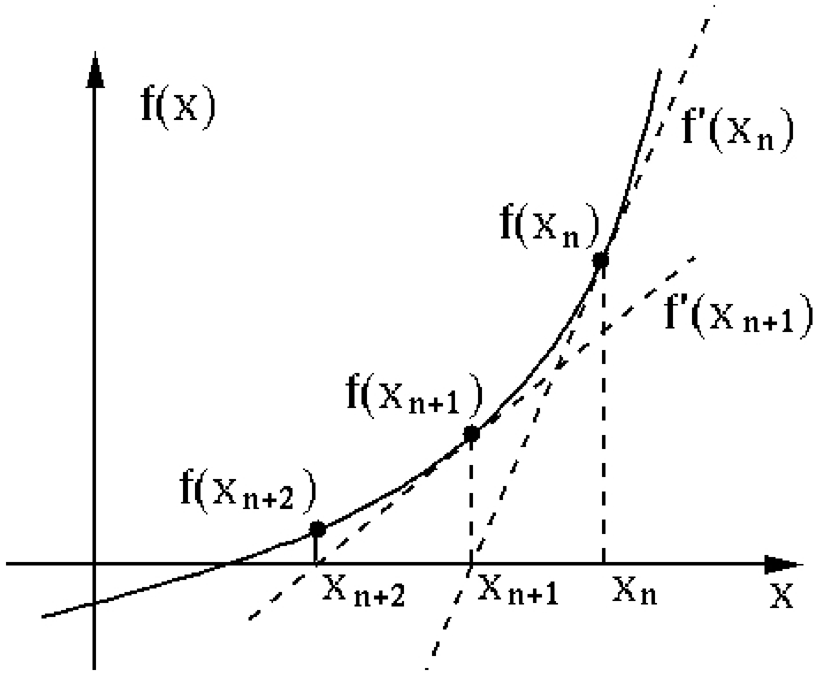


Illustration of Newton's method [1]

[1] https://github.com/Lehmannhen/Bisection-and-Newton-method

Newton's Method to Update Stress at $n + 1$ (1/3)

- Solution target for time $n+1$ (subscript $n+1$ is omitted for simplification):

$$\boldsymbol{\varepsilon}^p = \boldsymbol{\varepsilon}_n^p + \Delta\lambda \boldsymbol{r} \Rightarrow \boldsymbol{a} = -\boldsymbol{\varepsilon}^p + \boldsymbol{\varepsilon}_n^p + \Delta\lambda \boldsymbol{r} = \boldsymbol{0}$$

$$\boldsymbol{q} = \boldsymbol{q}_n + \Delta\lambda \boldsymbol{h} \Rightarrow \boldsymbol{b} = -\boldsymbol{q} + \boldsymbol{q}_n + \Delta\lambda \boldsymbol{h} = \boldsymbol{0}$$

$$f = f(\boldsymbol{\sigma}, \boldsymbol{q}) = 0$$

- Linearization following Newton's method:

$$\boldsymbol{a}^{(k)} - \delta\boldsymbol{\varepsilon}^{p(k)} + \Delta\lambda^{(k)} \delta\boldsymbol{r}^{(k)} + \delta\lambda^{(k)} \boldsymbol{r}^{(k)} = \boldsymbol{0}$$

$$\boldsymbol{b}^{(k)} - \delta\boldsymbol{q}^{(k)} + \Delta\lambda^{(k)} \delta\boldsymbol{h}^{(k)} + \delta\lambda^{(k)} \boldsymbol{h}^{(k)} = \boldsymbol{0}$$

$$f^{(k)} + f_{\boldsymbol{\sigma}}^{(k)} : \delta\boldsymbol{\sigma}^{(k)} + f_{\boldsymbol{q}}^{(k)} \cdot \delta\boldsymbol{q}^{(k)} = 0$$

$$\delta\boldsymbol{\varepsilon}^{p(k)} = -\boldsymbol{C}^{-1} : \delta\boldsymbol{\sigma}^{(k)}$$

$$\delta\boldsymbol{r}(\boldsymbol{\sigma}, \boldsymbol{q})^{(k)} = \boldsymbol{r}_{\boldsymbol{\sigma}}^{(k)} : \delta\boldsymbol{\sigma}^{(k)} + \boldsymbol{r}_{\boldsymbol{q}}^{(k)} : \delta\boldsymbol{q}^{(k)}$$

$$\delta\boldsymbol{h}(\boldsymbol{\sigma}, \boldsymbol{q})^{(k)} = \boldsymbol{h}_{\boldsymbol{\sigma}}^{(k)} : \delta\boldsymbol{\sigma}^{(k)} + \boldsymbol{h}_{\boldsymbol{q}}^{(k)} : \delta\boldsymbol{q}^{(k)}$$

Newton's Method to Update Stress at $n + 1$ (2/3)

$$\boldsymbol{a}^{(k)} + \boldsymbol{C}^{-1} : \delta\boldsymbol{\sigma}^{(k)} + \Delta\lambda^{(k)} \left(\boldsymbol{r}_{\boldsymbol{\sigma}}^{(k)} : \delta\boldsymbol{\sigma}^{(k)} + \boldsymbol{r}_{\boldsymbol{q}}^{(k)} : \delta\boldsymbol{q}^{(k)} \right) + \delta\lambda^{(k)} \boldsymbol{r}^{(k)} = \boldsymbol{0}$$

$$\boldsymbol{b}^{(k)} - \delta\boldsymbol{q}^{(k)} + \Delta\lambda^{(k)} \left(\boldsymbol{h}_{\boldsymbol{\sigma}}^{(k)} : \delta\boldsymbol{\sigma}^{(k)} + \boldsymbol{h}_{\boldsymbol{q}}^{(k)} : \delta\boldsymbol{q}^{(k)} \right) + \delta\lambda^{(k)} \boldsymbol{h}^{(k)} = \boldsymbol{0}$$

$$f^{(k)} + f_{\boldsymbol{\sigma}}^{(k)} : \delta\boldsymbol{\sigma}^{(k)} + f_{\boldsymbol{q}}^{(k)} \cdot \delta\boldsymbol{q}^{(k)} = 0$$

Update of 3 unknowns in each iteration, $x^{(k+1)} = x^{(k)} + \delta x^{(k)}$

- Matrix format (just symbolizing for simiplification, not representing true dimension):

$$\underbrace{\begin{bmatrix} \boldsymbol{C}^{-1} + \Delta\lambda^{(k)} \boldsymbol{r}_{\boldsymbol{\sigma}}^{(k)} & \Delta\lambda^{(k)} \boldsymbol{r}_{\boldsymbol{q}}^{(k)} \\ \Delta\lambda^{(k)} \boldsymbol{h}_{\boldsymbol{\sigma}}^{(k)} & -\boldsymbol{I} + \Delta\lambda^{(k)} \boldsymbol{h}_{\boldsymbol{q}}^{(k)} \end{bmatrix}}_{[\boldsymbol{A}^{(k)}]^{(-1)}} \begin{bmatrix} \delta\boldsymbol{\sigma}^{(k)} \\ \delta\boldsymbol{q}^{(k)} \end{bmatrix} = -\underbrace{\begin{bmatrix} \boldsymbol{a}^{(k)} \\ \boldsymbol{b}^{(k)} \end{bmatrix}}_{[\tilde{\boldsymbol{a}}^{(k)}]} - \delta\lambda^{(k)} \underbrace{\begin{bmatrix} \boldsymbol{r}^{(k)} \\ \boldsymbol{h}^{(k)} \end{bmatrix}}_{[\tilde{\boldsymbol{r}}^{(k)}]}$$

$$\Rightarrow \begin{bmatrix} \delta\boldsymbol{\sigma}^{(k)} \\ \delta\boldsymbol{q}^{(k)} \end{bmatrix} = -[\boldsymbol{A}^{(k)}][\tilde{\boldsymbol{a}}^{(k)}] - \delta\lambda^{(k)} [\boldsymbol{A}^{(k)}][\tilde{\boldsymbol{r}}^{(k)}] \Rightarrow \delta\lambda^{(k)} = \frac{f^{(k)} - [f_{\boldsymbol{\sigma}}^{(k)} \quad f_{\boldsymbol{q}}^{(k)}][\boldsymbol{A}^{(k)}][\tilde{\boldsymbol{a}}^{(k)}]}{\underbrace{[f_{\boldsymbol{\sigma}}^{(k)} \quad f_{\boldsymbol{q}}^{(k)}]}_{\partial \boldsymbol{f}} [\boldsymbol{A}^{(k)}][\tilde{\boldsymbol{r}}^{(k)}]}$$

Yield equation

Newton's Method to Update Stress at $n + 1$ (3/3)

- Update of calculation targets:

$$\boldsymbol{\varepsilon}^{p(k+1)} = \boldsymbol{\varepsilon}^{p(k)} + \delta\boldsymbol{\varepsilon}^{p(k)} = \boldsymbol{\varepsilon}^{p(k)} - \boldsymbol{C}^{-1}: \delta\boldsymbol{\sigma}^{(k)}$$

$$\boldsymbol{q}^{(k+1)} = \boldsymbol{q}^{(k)} + \delta\boldsymbol{q}^{(k)}$$

$$\Delta\lambda^{(k+1)} = \Delta\lambda^{(k)} + \delta\lambda^{(k)}$$

- Convergence condition – unknown variables converge and yield function becomes to 0:

$$x^{(k+1)} - x^{(k)} = \delta x^{(k)} < tolerance\ 1$$

$$f^{(k+1)} < tolerance\ 2$$

1. Refer to Box 5.13 in textbook for summarized return mapping procedure
2. Reliable but can be difficult to establish for complex constitutive models as updated $\boldsymbol{r}$, $\boldsymbol{h}$ and their derivatives need to be calculated in each iteration

Radial Return for J2 Plasticity

- Fully implicit backward Euler scheme reduces to the radial return method for J2 flow plasticity
- Initial trial stress at 0 iteration of time $n+1$:

$$\boldsymbol{\sigma}^{(0)} = \boldsymbol{\sigma}_n + \boldsymbol{C}:\Delta\boldsymbol{\varepsilon}$$

- Newton iteration for stress:

$$\boldsymbol{\sigma}^{(k)} = \boldsymbol{\sigma}^{(0)} - \Delta\lambda^{(k)}\boldsymbol{C}:\boldsymbol{r}^{(k)}$$

- Plastic flow in J2 models:

$$\boldsymbol{r} = \frac{3}{2\bar{\sigma}}\boldsymbol{\sigma}^{dev} = f_{\boldsymbol{\sigma}}$$

- Von Mises yield surface:

$$\sigma_Y^2 = \frac{3}{2}\boldsymbol{\sigma}^{dev}:\boldsymbol{\sigma}^{dev}$$

Sphere in high-dimensional $\boldsymbol{\sigma}^{dev}$ space, normal direction is along the radius

- Initial unit normal in radial (plastic flow) direction:

$$\hat{\boldsymbol{n}} = \frac{\boldsymbol{r}^{(0)}}{\left|\left|\boldsymbol{r}^{(0)}\right|\right|} = \frac{\boldsymbol{r}^{(0)}}{3/2} \Rightarrow \boldsymbol{r}^{(0)} = \sqrt{\frac{3}{2}}\hat{\boldsymbol{n}}$$

- Yield surfaces share the same centroid (kinematic hardening is the same in $\boldsymbol{\Sigma}$ space) – $\hat{\boldsymbol{n}}$ remains unchanged during iteration at time $n+1$:

$$\boldsymbol{\varepsilon}^p = \boldsymbol{\varepsilon}_n^p + \Delta\lambda\boldsymbol{r}^{(0)}$$

- Isotropic hardening J2 at time $n+1$:

$$\boldsymbol{q} = q_1 = \lambda h_1 = \bar{\varepsilon}, \qquad h_1 = 1$$

$$\Rightarrow q_1 = q_{1n} + \Delta\lambda$$

Essential Components for Radial Return Iteration

- Derivatives for return mapping in J2 models at time $n+1$:

$$\boldsymbol{r}_{\boldsymbol{\sigma}} = \left(\frac{3}{2\bar{\sigma}}\boldsymbol{\sigma}^{dev}\right)_{\boldsymbol{\sigma}} = \frac{3}{2\bar{\sigma}}\hat{\boldsymbol{I}} = \frac{3}{2\bar{\sigma}}\left(\boldsymbol{I}_{4th} - \frac{1}{3}\boldsymbol{I}_{2nd}\boldsymbol{I}_{2nd} - \hat{\boldsymbol{n}}\hat{\boldsymbol{n}}\right)$$

$$\boldsymbol{r}_{q_1} = \boldsymbol{r}_{\bar{\varepsilon}} = \left(\frac{3}{2\bar{\sigma}}\boldsymbol{\sigma}^{dev}\right)_{\bar{\varepsilon}} = \mathbf{0}$$

$$\boldsymbol{h} = h_1 = 1$$

$$\Rightarrow \boldsymbol{h}_{\boldsymbol{\sigma}} = h_{1\boldsymbol{\sigma}} = \mathbf{0}, \qquad \boldsymbol{h}_{\boldsymbol{q}} = h_{q1} = 0$$

$$f_{\boldsymbol{\sigma}} = \boldsymbol{r}, \qquad f_{\boldsymbol{\sigma}} = -\frac{d\sigma_Y(\bar{\varepsilon})}{d\bar{\varepsilon}} = -H(\bar{\varepsilon})$$

Matrices in Radial Return Iteration

$$\boldsymbol{r_\sigma} = \frac{3}{2\bar{\sigma}}\hat{\boldsymbol{I}}, \qquad \boldsymbol{r_q} = \boldsymbol{0}, \qquad \boldsymbol{h_\sigma} = \boldsymbol{0}, \qquad \boldsymbol{h_q} = 0, \qquad f_{\boldsymbol{\sigma}} = \boldsymbol{r}, \qquad f_{\boldsymbol{\sigma}} = -H$$

- Calculation of iteration parameter matrices:

$$\left[\boldsymbol{A}^{(k)}\right] = \begin{bmatrix} \boldsymbol{C}^{-1} + \Delta\lambda^{(k)}\boldsymbol{r_\sigma}^{(k)} & \Delta\lambda^{(k)}\boldsymbol{r_q}^{(k)} \\ \Delta\lambda^{(k)}\boldsymbol{h_\sigma}^{(k)} & -\boldsymbol{I} + \Delta\lambda^{(k)}\boldsymbol{h_q}^{(k)} \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{C}^{-1} + \Delta\lambda^{(k)}\frac{3}{2\bar{\sigma}}\hat{\boldsymbol{I}} & \boldsymbol{0} \\ \boldsymbol{0} & -1 \end{bmatrix}^{-1}$$

$$\boldsymbol{C}^{-1} + \Delta\lambda^{(k)}\frac{3}{2\bar{\sigma}}\hat{\boldsymbol{I}} = \boldsymbol{C}^{-1} + a\hat{\boldsymbol{I}} \Rightarrow \left(\boldsymbol{C}^{-1} + \Delta\lambda^{(k)}\frac{3}{2\bar{\sigma}}\hat{\boldsymbol{I}}\right)^{-1} = \boldsymbol{C} - 2\mu b\hat{\boldsymbol{I}}$$

Transformation based on isotropic elasticity

$$b = \frac{2\mu a}{1 + 2\mu a}$$

$$\Rightarrow \left[\boldsymbol{A}^{(k)}\right] = \begin{bmatrix} \boldsymbol{C} - 2\mu b\hat{\boldsymbol{I}} & \boldsymbol{0} \\ \boldsymbol{0} & -1 \end{bmatrix}$$

Iteration of $\Delta\lambda^{(k)}$ for Isotropic Materials

$$\delta\lambda^{(k)} = \frac{f^{(k)} - \begin{bmatrix} f_{\boldsymbol{\sigma}}^{(k)} & f_{\boldsymbol{q}}^{(k)} \end{bmatrix} [\boldsymbol{A}^{(k)}] \begin{bmatrix} \boldsymbol{a}^{(k)} \\ \boldsymbol{b}^{(k)} \end{bmatrix}}{\begin{bmatrix} f_{\boldsymbol{\sigma}}^{(k)} & f_{\boldsymbol{q}}^{(k)} \end{bmatrix} [\boldsymbol{A}^{(k)}] \begin{bmatrix} \boldsymbol{r}^{(k)} \\ \boldsymbol{h}^{(k)} \end{bmatrix}} = \frac{f^{(k)} - \begin{bmatrix} \boldsymbol{r}^{(k)} & f_{\bar{\varepsilon}}^{(k)} \end{bmatrix} \begin{bmatrix} \boldsymbol{C} - 2\mu b\hat{\boldsymbol{I}} & \boldsymbol{0} \\ \boldsymbol{0} & -1 \end{bmatrix} \begin{bmatrix} \boldsymbol{a}^{(k)} \\ \boldsymbol{b}^{(k)} \end{bmatrix}}{\begin{bmatrix} \boldsymbol{r}^{(k)} & f_{\bar{\varepsilon}}^{(k)} \end{bmatrix} \begin{bmatrix} \boldsymbol{C} - 2\mu b\hat{\boldsymbol{I}} & \boldsymbol{0} \\ \boldsymbol{0} & -1 \end{bmatrix} \begin{bmatrix} \boldsymbol{r}^{(k)} \\ 1 \end{bmatrix}}$$

- Isotropic elasticity:

$$\boldsymbol{r}:\boldsymbol{C}:\boldsymbol{r} = 3\mu, \qquad \boldsymbol{r}:\left(\hat{\boldsymbol{I}}:\boldsymbol{r}\right) = \boldsymbol{r}:\left(\hat{\boldsymbol{I}}:\sqrt{3/2}\hat{\boldsymbol{n}}\right) = \boldsymbol{r}:\boldsymbol{0} = \boldsymbol{0}, \qquad -f_{\bar{\varepsilon}}^{(k)} = H^{(k)}$$

- Linear functions require one Newton's method iteration to obtain roots:

$$\boldsymbol{a} = -\boldsymbol{\varepsilon}^p + \boldsymbol{\varepsilon}_n^p + \Delta\lambda\boldsymbol{r}^{(0)} = \boldsymbol{0} \Rightarrow \boldsymbol{a}^{(k)} = \boldsymbol{0}, \qquad \boldsymbol{b} = b = -q_1 + q_{1n} + \Delta\lambda = 0 \Rightarrow b^{(k)} = 0$$

$$\Rightarrow \delta\lambda^{(k)} = \frac{f^{(k)}}{3\mu + H^{(k)}}$$

Common expression for Iteration of $\Delta\lambda^{(k)}$

$$\boldsymbol{r} = \frac{3}{2\bar{\sigma}}\boldsymbol{\sigma}^{dev} = \sqrt{\frac{3}{2}}\widehat{\boldsymbol{n}} \Rightarrow \boldsymbol{\sigma}^{dev} = \sqrt{\frac{2}{3}}\bar{\sigma}\widehat{\boldsymbol{n}}, \qquad \boldsymbol{\sigma}^{(k)} = \boldsymbol{\sigma}^{(0)} - \Delta\lambda^{(k)}\boldsymbol{C}:\boldsymbol{r}^{(k)} = \boldsymbol{\sigma}^{(0)} - \Delta\lambda^{(k)}2\mu\boldsymbol{r}^{(k)}$$

Plastic deformation only relates to $\boldsymbol{\sigma}^{dev}$

$$\Rightarrow \boldsymbol{\sigma}^{dev(k)} = \boldsymbol{\sigma}^{dev(0)} - \Delta\lambda^{(k)}2\mu\boldsymbol{r}^{(k)} = \left(\sqrt{\frac{2}{3}}\bar{\sigma}^{(0)} - 2\mu\Delta\lambda^{(k)}\sqrt{\frac{3}{2}}\right)\widehat{\boldsymbol{n}}$$

$$\Rightarrow \bar{\sigma}^{(k)} = \bar{\sigma}^{(0)} - 3\mu\Delta\lambda^{(k)}$$

$$\Rightarrow \delta\lambda^{(k)} = \frac{f^{(k)}}{3\mu + H^{(k)}} = \frac{\bar{\sigma}^{(0)} - 3\mu\Delta\lambda^{(k)} - \sigma_Y\left(\bar{\varepsilon}^{(k)}\right)}{3\mu + H^{(k)}}$$

Algorithmic Modulus for Implicit Methods

- Abrupt tangent modulus change at yield can cause spurious loading and unloading in implicit methods

- Introduce algorithmic modulus (consistent tangent modulus) for fully implicit backward Euler update:

$$\boldsymbol{C}^{alg} = \left(\frac{d\boldsymbol{\sigma}}{d\boldsymbol{\varepsilon}}\right)_{n+1}$$

- Incremental form of the integration scheme at time $n+1$ (omit subscript $n+1$):

$$d\boldsymbol{\sigma} = \boldsymbol{C}:(d\boldsymbol{\varepsilon} - d\boldsymbol{\varepsilon}^p)$$

$$d\boldsymbol{\varepsilon}^p = d(\Delta\lambda)\boldsymbol{r} + \Delta\lambda d\boldsymbol{r} \qquad d\boldsymbol{r}(\boldsymbol{\sigma},\boldsymbol{q}) = \boldsymbol{r}_{\boldsymbol{\sigma}}:d\boldsymbol{\sigma} + \boldsymbol{r}_{\boldsymbol{q}}\cdot\boldsymbol{q}$$

$$d\boldsymbol{q} = d(\Delta\lambda)\boldsymbol{h} + \Delta\lambda d\boldsymbol{h} \qquad d\boldsymbol{h}(\boldsymbol{\sigma},\boldsymbol{q}) = \boldsymbol{h}_{\boldsymbol{\sigma}}:d\boldsymbol{\sigma} + \boldsymbol{h}_{\boldsymbol{q}}\cdot\boldsymbol{q}$$

$$df = f_{\boldsymbol{\sigma}}:d\boldsymbol{\sigma} + f_{\boldsymbol{q}}\cdot\boldsymbol{q} = 0$$

From $d\varepsilon$ to $d\sigma$

$$d\boldsymbol{\sigma} = \boldsymbol{C} : (d\boldsymbol{\varepsilon} - d\boldsymbol{\varepsilon}^p), \qquad d\boldsymbol{\varepsilon}^p = d(\Delta\lambda)\boldsymbol{r} + \Delta\lambda d\boldsymbol{r}, \qquad d\boldsymbol{q} = d(\Delta\lambda)\boldsymbol{h} + \Delta\lambda d\boldsymbol{h}$$

$$d\boldsymbol{r}(\boldsymbol{\sigma}, \boldsymbol{q}) = \boldsymbol{r}_{\boldsymbol{\sigma}} : d\boldsymbol{\sigma} + \boldsymbol{r}_{\boldsymbol{q}} \cdot \boldsymbol{q}, \qquad d\boldsymbol{h}(\boldsymbol{\sigma}, \boldsymbol{q}) = \boldsymbol{h}_{\boldsymbol{\sigma}} : d\boldsymbol{\sigma} + \boldsymbol{h}_{\boldsymbol{q}} \cdot \boldsymbol{q}$$

$$\Rightarrow \begin{bmatrix} d\boldsymbol{\sigma} \\ d\boldsymbol{q} \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{C}^{-1} + \Delta\lambda \boldsymbol{r}_{\boldsymbol{\sigma}} & \Delta\lambda \boldsymbol{r}_{\boldsymbol{q}} \\ \Delta\lambda \boldsymbol{h}_{\boldsymbol{\sigma}} & -\boldsymbol{I} + \Delta\lambda \boldsymbol{h}_{\boldsymbol{q}} \end{bmatrix}^{-1}}_{\boldsymbol{A}} : \begin{bmatrix} d\boldsymbol{\varepsilon} \\ \boldsymbol{0} \end{bmatrix} - d(\Delta\lambda) \begin{bmatrix} \boldsymbol{C}^{-1} + \Delta\lambda \boldsymbol{r}_{\boldsymbol{\sigma}} & \Delta\lambda \boldsymbol{r}_{\boldsymbol{q}} \\ \Delta\lambda \boldsymbol{h}_{\boldsymbol{\sigma}} & -\boldsymbol{I} + \Delta\lambda \boldsymbol{h}_{\boldsymbol{q}} \end{bmatrix}^{-1} : \underbrace{\begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{h} \end{bmatrix}}_{\tilde{\boldsymbol{r}}}$$

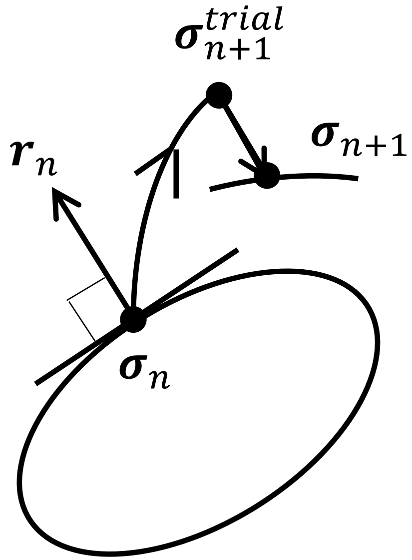
$$df = f_{\boldsymbol{\sigma}} : d\boldsymbol{\sigma} + f_q \cdot \boldsymbol{q} = 0$$

$$\Rightarrow d(\Delta\lambda) = \frac{\begin{bmatrix} f_{\boldsymbol{\sigma}} & f_q \end{bmatrix} : \boldsymbol{A} : \begin{bmatrix} d\boldsymbol{\varepsilon} \\ \boldsymbol{0} \end{bmatrix}}{\underbrace{\begin{bmatrix} f_{\boldsymbol{\sigma}} & f_q \end{bmatrix}}_{\partial \boldsymbol{f}} : \boldsymbol{A} : \tilde{\boldsymbol{r}}} \Rightarrow \begin{bmatrix} d\boldsymbol{\sigma} \\ d\boldsymbol{q} \end{bmatrix} = \overbrace{\left[\boldsymbol{A} - \frac{(\boldsymbol{A} : \tilde{\boldsymbol{r}})(\partial \boldsymbol{f} : \boldsymbol{A})}{\partial \boldsymbol{f} : \boldsymbol{A} : \tilde{\boldsymbol{r}}} \right]}^{\boldsymbol{C}^{alg}} \begin{bmatrix} d\boldsymbol{\varepsilon} \\ \boldsymbol{0} \end{bmatrix}$$

Semi-Implicit Backward Euler Scheme

- Semi-implicit:
  - Implicit in plasticity parameter λ.
  - Explicit in plastic flow direction $\boldsymbol{r}$ and modulus.

    Determined by $\boldsymbol{r}$ and $\boldsymbol{h}$
- Stress update visualization:



Projection scheme for associative plasticity with $\boldsymbol{r}_n \sim f_{\boldsymbol{\sigma}_n}$.

- Integration scheme:

$$\boldsymbol{\varepsilon}_{n+1} = \boldsymbol{\varepsilon}_n + \Delta\boldsymbol{\varepsilon}$$

$$\boldsymbol{\varepsilon}_{n+1}^p = \boldsymbol{\varepsilon}_n^p + \Delta\lambda_{n+1}\boldsymbol{r}_n$$

$$\boldsymbol{q}_{n+1} = \boldsymbol{q}_n + \Delta\lambda_{n+1}\boldsymbol{h}_n$$

$$\boldsymbol{\sigma}_{n+1} = \boldsymbol{C}:\left(\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}_{n+1}^p\right)$$

$$f_{n+1} = f\left(\boldsymbol{\sigma}_{n+1}, \boldsymbol{q}_{n+1}\right) = 0$$

Newton's Method for Semi-Implicit Stress Update

$$\boldsymbol{\varepsilon}_{n+1}^{p} = \boldsymbol{\varepsilon}_{n}^{p} + \Delta\lambda_{n+1}\boldsymbol{r}_{n}$$

$$\Rightarrow \boldsymbol{a} = -\boldsymbol{\varepsilon}_{n+1}^{p} + \boldsymbol{\varepsilon}_{n}^{p} + \Delta\lambda_{n+1}\boldsymbol{r}_{n} = \boldsymbol{0}$$

$$\boldsymbol{q}_{n+1} = \boldsymbol{q}_{n} + \Delta\lambda_{n+1}\boldsymbol{h}_{n}$$

$$\Rightarrow \boldsymbol{b} = -\boldsymbol{q}_{n+1} + \boldsymbol{q}_{n} + \Delta\lambda_{n+1}\boldsymbol{h}_{n} = \boldsymbol{0}$$

$$f_{n+1} = f(\boldsymbol{\sigma}_{n+1}, \boldsymbol{q}_{n+1}) = 0$$

- Linearization at time $n+1$ (omit $n+1$):

$$\boldsymbol{a}^{(k)} + \boldsymbol{C}^{-1}:\delta\boldsymbol{\sigma}^{(k)} + \delta\lambda^{(k)}\boldsymbol{r}_{n} = \boldsymbol{0}$$

$$\boldsymbol{b}^{(k)} - \delta\boldsymbol{q}^{(k)} + \delta\lambda^{(k)}\boldsymbol{h}_{n} = \boldsymbol{0}$$

$$f^{(k)} + f_{\boldsymbol{\sigma}}^{(k)}:\delta\boldsymbol{\sigma}^{(k)} + f_{\boldsymbol{q}}^{(k)}\cdot\delta\boldsymbol{q}^{(k)} = 0$$

$$\Rightarrow \begin{bmatrix}\delta\boldsymbol{\sigma}^{(k)}\\ \delta\boldsymbol{q}^{(k)}\end{bmatrix} = -\boldsymbol{A}^{(k)}:\begin{bmatrix}\boldsymbol{a}^{(k)}\\ \boldsymbol{b}^{(k)}\end{bmatrix} - \delta\lambda^{(k)}\boldsymbol{A}^{(k)}:\tilde{\boldsymbol{r}}_{n}$$

$$\boldsymbol{A}^{(k)} = \begin{bmatrix}\boldsymbol{C} & \boldsymbol{0}\\ \boldsymbol{0} & -\boldsymbol{I}\end{bmatrix}^{(k)}, \qquad \tilde{\boldsymbol{r}}_{n} = \begin{bmatrix}\boldsymbol{r}_{n}\\ \boldsymbol{h}_{n}\end{bmatrix}$$

$$\Rightarrow \delta\lambda^{(k)} = \frac{f^{(k)}}{\underbrace{\begin{bmatrix}f_{\boldsymbol{\sigma}}^{(k)} & f_{\boldsymbol{q}}^{(k)}\end{bmatrix}}_{\partial\boldsymbol{f}}\boldsymbol{A}^{(k)}:\tilde{\boldsymbol{r}}_{n}}$$

- Similar to fully implicit with unchanged $\boldsymbol{r}$ and $\boldsymbol{h}$ from time n
- Plastic strain update: $\boldsymbol{\varepsilon}^{p(k+1)} = \boldsymbol{\varepsilon}^{p(k)} - \boldsymbol{C}^{-1}:\delta\boldsymbol{\sigma}^{(k)}$

Algorithmic Modulus for Semi-Implicit Methods

- Similar to derivation for fully implicit methods at time $n+1$:

$$\begin{bmatrix} d\boldsymbol{\sigma} \\ d\boldsymbol{q} \end{bmatrix} = \left[\boldsymbol{A} - \frac{(\boldsymbol{A}:\tilde{\boldsymbol{r}})(\partial \boldsymbol{f}:\boldsymbol{A})}{\partial \boldsymbol{f}:\boldsymbol{A}:\tilde{\boldsymbol{r}}} \right] \begin{bmatrix} d\boldsymbol{\varepsilon} \\ \boldsymbol{0} \end{bmatrix}$$

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{C} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{I} \end{bmatrix}, \qquad \tilde{\boldsymbol{r}} = \begin{bmatrix} \boldsymbol{r}_n \\ \boldsymbol{h}_n \end{bmatrix}, \qquad \partial \boldsymbol{f} = [f_{\boldsymbol{\sigma}} \quad f_{\boldsymbol{q}}]$$

$$\Rightarrow \boldsymbol{C}^{alg} = \left(\frac{d\boldsymbol{\sigma}}{d\boldsymbol{\varepsilon}} \right)_{n+1} = \left[\boldsymbol{C} - \frac{(\boldsymbol{C}:\boldsymbol{r}_n)(f_{\boldsymbol{\sigma}}:\boldsymbol{C})}{-f_{\boldsymbol{q}} \cdot \boldsymbol{h}_n + f_{\boldsymbol{\sigma}}:\boldsymbol{C}:\boldsymbol{r}_n} \right]_{n+1}$$

Caution: $\boldsymbol{C}^{alg}$ is asymmetric even for associative flow as $\boldsymbol{r}_n \neq (f_{\boldsymbol{\sigma}})_{n+1}$!

The End