Exercise 5: Implementation of a three-dimensional von Mises viscoplastic model with linear hardening at small strains

Problem 6.1

The stress-update algorithm for a von Mises viscoplastic model with linear isotropic and kinematic hardening and a linear overstress function is given in the algorithmic box 1

Alg. 1 Radial return stress-update algorithm for J_2-viscoplasticity with linear hardening

... Given: history information $\{\boldsymbol{\varepsilon}_n^{\mathrm{vp}}, \boldsymbol{\alpha}_n, \alpha_n\}$ and current total strain $\boldsymbol{\varepsilon}_{n+1}$ ...

... Compute trial elastic state: (elastic prediction) ...

$\operatorname{dev}(\boldsymbol{\varepsilon}_{n+1}) = \boldsymbol{\varepsilon}_{n+1} - \frac{1}{3}\operatorname{tr}(\boldsymbol{\varepsilon}_{n+1})\,\mathbf{I}$

$\operatorname{dev}\left(\boldsymbol{\sigma}_{n+1}^{\mathrm{trial}}\right) = 2\,\mu\operatorname{dev}\left(\boldsymbol{\varepsilon}_{n+1}^{\mathrm{e,trial}}\right) = 2\,\mu\left[\operatorname{dev}(\boldsymbol{\varepsilon}_{n+1}) - \boldsymbol{\varepsilon}_n^{\mathrm{vp}}\right]$

$\boldsymbol{\beta}_{n+1}^{\mathrm{trial}} = \mathrm{H}\boldsymbol{\alpha}_n, \quad \beta_{n+1}^{\mathrm{trial}} = \mathrm{h}\alpha_n$

$\boldsymbol{\xi}_{n+1}^{\mathrm{trial}} = \operatorname{dev}\left(\boldsymbol{\sigma}_{n+1}^{\mathrm{trial}}\right) - \boldsymbol{\beta}_{n+1}^{\mathrm{trial}}$

$\mathbf{n}_{n+1}^{\mathrm{trial}} = \frac{\boldsymbol{\xi}_{n+1}^{\mathrm{trial}}}{\|\boldsymbol{\xi}_{n+1}^{\mathrm{trial}}\|}$

$\phi_{n+1}^{\mathrm{trial}} = \|\boldsymbol{\xi}_{n+1}^{\mathrm{trial}}\| - \sqrt{\frac{2}{3}}\left[\sigma_{\mathrm{y}_0} + \beta_{n+1}^{\mathrm{trial}}\right]$

... Check for elastic-viscoplastic loading ...

if $\phi_{n+1}^{\mathrm{trial}} < 0$ **then**

... elastic step ...

$\operatorname{dev}(\boldsymbol{\sigma}_{n+1}) = \operatorname{dev}\left(\boldsymbol{\sigma}_{n+1}^{\mathrm{trial}}\right)$

$\mathbb{C}_{n+1}^{\mathrm{dev}} = 2\,\mu\,\mathbb{P}_{\mathrm{sym}}$

$\boldsymbol{\varepsilon}_{n+1}^{\mathrm{vp}} = \boldsymbol{\varepsilon}_n^{\mathrm{vp}}, \quad \boldsymbol{\alpha}_{n+1} = \boldsymbol{\alpha}_n, \quad \alpha_{n+1} = \alpha_n$

else

... Radial return (viscoplastic correction) ...

$\gamma_{n+1} = \frac{\phi_{n+1}^{\mathrm{trial}}}{2\,\mu\frac{\tau}{\Delta t} + 2\,\mu + \mathrm{H} + \frac{2}{3}\mathrm{h}} \qquad \text{with} \qquad \tau = \frac{\eta}{2\,\mu}$

$\operatorname{dev}(\boldsymbol{\sigma}_{n+1}) = \operatorname{dev}\left(\boldsymbol{\sigma}_{n+1}^{\mathrm{trial}}\right) - 2\,\mu\,\gamma_{n+1}\mathbf{n}_{n+1}^{\mathrm{trial}}$

$\boldsymbol{\varepsilon}_{n+1}^{\mathrm{vp}} = \boldsymbol{\varepsilon}_n^{\mathrm{vp}} + \gamma_{n+1}\mathbf{n}_{n+1}^{\mathrm{trial}}, \quad \boldsymbol{\alpha}_{n+1} = \boldsymbol{\alpha}_n + \gamma_{n+1}\mathbf{n}_{n+1}^{\mathrm{trial}}, \quad \alpha_{n+1} = \alpha_n + \sqrt{\frac{2}{3}}\gamma_{n+1}$

$\beta_1 = 1 - \frac{\phi_{n+1}^{\mathrm{trial}}}{\|\boldsymbol{\xi}_{n+1}^{\mathrm{trial}}\|}\frac{1}{\frac{\tau}{\Delta t} + 1 + \frac{\mathrm{H}}{2\,\mu} + \frac{\mathrm{h}}{3\,\mu}} \qquad \beta_2 = \left[1 - \frac{\phi_{n+1}^{\mathrm{trial}}}{\|\boldsymbol{\xi}_{n+1}^{\mathrm{trial}}\|}\right]\frac{1}{\frac{\tau}{\Delta t} + 1 + \frac{\mathrm{H}}{2\,\mu} + \frac{\mathrm{h}}{3\,\mu}}$

$\mathbb{C}_{n+1}^{\mathrm{dev}} = 2\,\mu\,\beta_1\mathbb{P}_{\mathrm{sym}} - 2\,\mu\,\beta_2\mathbf{n}_{n+1}^{\mathrm{trial}} \otimes \mathbf{n}_{n+1}^{\mathrm{trial}}$

end if

... Add spherical contribution to stress and moduli: ...

$\boldsymbol{\sigma}_{n+1} = \operatorname{dev}(\boldsymbol{\sigma}_{n+1}) + \kappa\operatorname{tr}(\boldsymbol{\varepsilon}_{n+1})\,\mathbf{I}$

$\mathbb{C}_{n+1} = \mathbb{C}_{n+1}^{\mathrm{dev}} + \kappa\,\mathbf{I} \otimes \mathbf{I}$

(a) Implement the stress-update algorithm in the function `vmises_perzyna.m`, test your implementation for different hardening parameters by running the isochoric tensile test given in `test_suite_isochoric_tension.m`. Compare your results with the analytic solution.

(b) Extent your function `vmises_perzyna.m` to include the algorithmic tangent. Run the uniaxial tensile test, implemented in `drive.m`, check the convergence and compare to the analytic solution (rate-independent) by choosing an appropriate loading rate.