Project 3 of MAEG 5130 (Deadline: 23:59, April 26, 2023)

Application of MATLAB for linear isotopic and kinematic hardening

Problem Statement (100’):

In this project, you need to complete a MATLAB script of 3D linear isotropic hardening law using material properties given in Project 2. For boundary conditions, please apply cyclic loading, the same as the one in Project 2, in one direction and 0 strain along other directions.

You need to

1) obtain stress-strain curve from MATLAB model.

2) compare the curve obtained in MATLAB with Abaqus results. If not consistent, please explain why.

3) For inconsistency between MATLAB and Abaqus, please change the setup of Abaqus to make their results the same, and then show the results.

Bonus (100’):

Similar to linear isotropic hardening law, you need to finish a 3D linear kinetic hardening law in MATLAB using the material properties given in Project 2 with cyclic loading provided in Project 2 along one direction and 0 strain along other directions.

You need to

1) obtain stress-strain curve from MATLAB model.

2) compare the curve obtained in MATLAB with Abaqus results. If not consistent, please explain why.

3) For inconsistency between MATLAB and Abaqus, please change the setup of Abaqus to make their results the same, and then show the results.

Some tips you should know before the start of MALTAB modelling:

(1) Abaqus uses Von Mises yield surfaces and associate flow rule, so the J2 flow model should be utilized.

(2) In one direction loading without rotation, the rate-of-deformation D will be the same as the true strain rate epsilon-dot, and there is no need to use objective rates as no rotation needs to be considered.

(3) Radial return algorithm, shown in the below figure, possesses much higher accuracy than forward Euler method in terms of yield state prediction. Please implement radial return algorithm in stress update in yield region.

Box 5.14 Radial return method

1. Initialization:

$$k = 0: \quad \varepsilon^{p^{(0)}} = \varepsilon_n^p, \quad \bar{\varepsilon}^{(0)} = \bar{\varepsilon}_n, \quad \Delta\lambda^{(0)} = 0, \quad \boldsymbol{\sigma}^{(0)} = \mathbf{C} : (\varepsilon_{n+1} - \varepsilon^{p(0)})$$

2. Check yield condition at kth iteration:

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

   If: $f^{(k)} < TOL_1$ then converged

   Else: go to 3

3. Compute increment in plasticity parameter:

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

4. Update plastic strain and internal variables:

$$\hat{\mathbf{n}} = \boldsymbol{\sigma}_{\text{dev}}^{(0)} / \left\| \boldsymbol{\sigma}_{\text{dev}}^{(0)} \right\|, \quad \Delta\varepsilon^{p^{(k)}} = -\delta\lambda^{(k)} \sqrt{\frac{3}{2}}\hat{\mathbf{n}}, \quad \Delta\bar{\varepsilon}^{(k)} = \delta\lambda^{(k)}$$

$$\varepsilon^{p(k+1)} = \varepsilon^{p(k)} + \Delta\varepsilon^{p(k)}$$

$$\boldsymbol{\sigma}^{(k+1)} = \mathbf{C}:\left(\varepsilon_{n+1} - \varepsilon^{p(k+1)}\right) = \boldsymbol{\sigma}^{(k)} + \Delta\boldsymbol{\sigma}^{(k)} = \boldsymbol{\sigma}^{(k)} - 2\mu\delta\lambda^{(k)}\sqrt{\frac{3}{2}}\hat{\mathbf{n}}$$

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

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

   $k \leftarrow k+1$, go to 2