Nonlinear Finite Element Methods

Assignment for summer term 2020 (Examiner: Geralf Hütter)

1 Task

The problem of creep of a thick-walled pipe under internal pressure p is considered as sketched in Figure 1. The pressure rises linearly up to its final value $p_{\max}$ and is then hold until t_f as shown in Figure 2. Plain strain $\varepsilon_{zz} = 0$ conditions are assumed. Due to

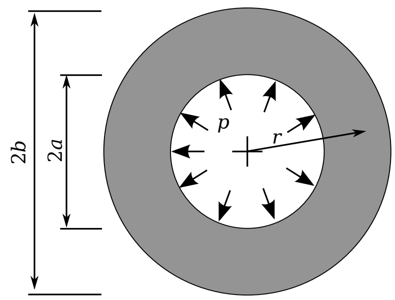

Figure 1: Thick-walled pipe

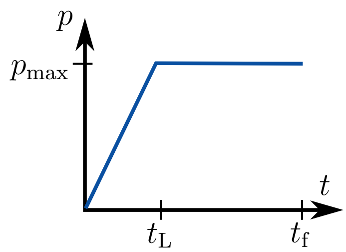

Figure 2: Load sequence

axisymmetric conditions, the only non-vanishing equilibrium condition is

$$0 = \frac{\partial(r\sigma_{rr})}{\partial r} - \sigma_{\phi\phi}\,. \tag{1}$$

Therein, σ_{rr} and $\sigma_{\phi\phi}$ refer to the stress components with respect to a polar coordinate system. The weak form of Eq. (1) reads

$$0 = \delta W = \int_a^b \underline{\delta\varepsilon}^\mathrm{T} \cdot \underline{\sigma}\, r\,\mathrm{d}r - \left[r\sigma_{rr}\delta u_r\right]_{r=a}^{b} \tag{2}$$

with stresses and strains written in Voigt notation as

$$\underline{\sigma} = \begin{bmatrix}\sigma_{rr}\\ \sigma_{\phi\phi}\end{bmatrix}, \qquad \underline{\delta\varepsilon} = \begin{bmatrix}\delta\varepsilon_{rr} = \frac{\partial\delta u_r}{\partial r}\\ \delta\varepsilon_{\phi\phi} = \frac{\delta u_r}{r}\end{bmatrix}, \qquad \text{and analogously } \underline{\varepsilon} = \begin{bmatrix}\varepsilon_{rr} = \frac{\partial u_r}{\partial r}\\ \varepsilon_{\phi\phi} = \frac{u_r}{r}\end{bmatrix}. \tag{3}$$

Therein, the only non-vanishing displacement component is $u_r(r)$ as the displacement in radial direction. The boundary conditions for the problem in Figure 1 are $\sigma_{rr}(r = a) = -p$ and $\sigma_{rr}(r = b) = 0$, respectively.

The linear visco-elastic behavior of the material is described by the equations

$$\underline{\sigma} = \underline{\underline{\mathbf{C}}} \cdot \underline{\varepsilon} + \underline{\sigma}^{\text{ov}} \tag{4a}$$

$$\dot{\underline{\sigma}}^{\text{ov}} = Q\, \text{dev}\left(\dot{\underline{\varepsilon}}\right) - \frac{1}{T}\underline{\sigma}^{\text{ov}} \tag{4b}$$

wherein $\underline{\underline{\mathbf{C}}}$ is the isotropic (long-term) elastic stiffness matrix, expressed by Young's modulus E and Poisson ration ν. The evolution of the overstress $\underline{\sigma}^{\text{ov}}$ (as internal state variable) is governed by the modulus Q and a characteristic time scale T.

Create a program (MatLab/Octave/Python) which solves this static FEM problem. The program has to be verified by comparisons with known analytical solutions and a convergence study shall be performed (see below).

2 Details

The following list gives a brief overview of the features which have to be implemented:

- quasi-static conditions: $\delta W = \delta \hat{\underline{\mathbf{u}}}^{\mathrm{T}} \cdot \left[\hat{\underline{\mathbf{F}}}_{\text{int}} - \hat{\underline{\mathbf{F}}}_{\text{ext}}\right]$
- linear shape functions for $u_r(r)$:

$$[\mathbf{N}] = \left[\frac{1}{2}(1-\xi), \frac{1}{2}(1+\xi)\right]^{\mathrm{T}} \text{ in } \Omega_{\square} = \{\xi \in [-1,1]\} \tag{5}$$

- quadrature with 1 Gauss point per element
- local mesh refinement closer to the interior of the pipe: $h^e(r=a) = \frac{1}{2}h^e(r=b)$ (h^e: element size), see code sniplet in appendix
- time integration with Euler backward method (EB) or modified Euler method (EM)
- variable number of elements and time increment Δt
- Newton-Raphson method with convergence criteria $\left\|\hat{\underline{\mathbf{R}}}\right\|_\infty < 0.005 \left\|\hat{\underline{\mathbf{F}}}_{\text{int}}\right\|_\infty$, $\|\blacktriangle\hat{\underline{\mathbf{u}}}_k\|_\infty < 0.005 \|\hat{\underline{\mathbf{u}}}\|_\infty$ (with $\|\hat{\underline{\circ}}\|_\infty$ denoting the infinity norm, i. e. the maximum component by amount of the column vector $\hat{\underline{\circ}}$)

The particular material parameters (E, ν, Q, T), loading parameters (p_{max}, t_{L}, t_{f}), time integration scheme (EM/EB) and geometric properties a and b to be implemented depend on your variant as given in Table 1. *Each student has to work on the variant that corresponds to the last digit of her or his matriculation number.*

3 Workflow

1. Theory
   - Discretize the weak form (2) in space (i. e. in r).

- Identify the $\underline{\underline{\mathbf{B}}}$ matrix to be defined as $\underline{\varepsilon} = \underline{\underline{\mathbf{B}}} \cdot \underline{\hat{\mathbf{u}}}^e$ for the shape functions in Eq. (5).
- Identify the vectors of internal and external nodal forces $\underline{\hat{\mathbf{F}}}_{\text{int}}^e$ and $\underline{\hat{\mathbf{F}}}_{\text{ext}}$, respectively.
- Discretize the constitutive equations (4) in time and compute the algorithmically consistent material tangent stiffness.

2. Implementation in MatLab/Octave/Python:
   - Implement $\underline{\underline{\mathbf{B}}}$ and $[\mathbf{N}]$ into an element routine to compute $\underline{\hat{\mathbf{F}}}_{\text{int}}^e$ (*Hint:* The Jacobian of the element is identical to the FEM of rods considered in the exercises.)
   - Develop the main program which assembles total nodal forces for each time increment and performs the Newton-Raphson scheme.
   - Note that the material routine requires internal state variables (the overstresses $\underline{\sigma}^{\text{ov}}$) for which memory has to be allocated and which have to be passed through main program and element routine.
3. Verification:

   a) According to classical theory of elasticity, the exact solution of the considered boundary value problem Eqs. (1)–(3) for *linear-elastic material* is

$$u_r^{\text{elast}} = (1+\nu)\frac{p}{E}\frac{a^2}{b^2-a^2}\left[(1-2\nu)r + \frac{b^2}{r}\right], \qquad (6)$$

   compare basic course Engineering Mechanics B. In a first step, use a material routine for the purely elastic case (corresponding formally to $Q = 0$). Perform a convergence study with respect to the number of elements and verify that your FEM solution converges towards the exact solution (6). Verify that the Newton-Raphson method converges within a single iteration for the linear problem.

Table 1: Assignment of parameters

var		E [MPa]	ν	Q [MPa]	T [s]	a [mm]	b [mm]	$p_{\max}$ [MPa]	t_{L} [s]	t_{f} [s]
1	EM	200 000	0.20	100 000	1	50	100	140	2	10
2	EM	70 000	0.25	35 000	2	40	80	50	4	20
3	EM	70 000	0.30	35 000	3	60	120	50	6	30
4	EM	200 000	0.30	100 000	1	30	60	140	2	10
5	EM	100 000	0.30	50 000	2	40	80	70	4	20
6	EB	200 000	0.20	100 000	3	50	100	140	6	30
7	EB	70 000	0.25	35 000	1	40	80	50	2	10
8	EB	70 000	0.30	35 000	2	60	120	50	4	20
9	EB	200 000	0.30	100 000	3	30	60	140	6	30
0	EB	100 000	0.30	50 000	4	40	80	70	8	40

b) In the next step, perform a convergence study with respect to number of elements and time increments Δt for the visco-elastic model. Identify the necessary number of elements and the required Δt.

4. Results:
   - Extract the distributions of $u_r(r)$, σ_{rr} and $\sigma_{\phi\phi}$ at final loading $t = t_\mathrm{f}$ from your FEM simulation.
   - Extract the time history of the widening of the pipe $u_r(r = b, t)$ for $t \in [0, t_\mathrm{f}]$. Verify, that the visco-elastic solution relaxes towards the elastic solution (6).

4 Documentation

In addition to the program code, a short *technical documentation* is to be created (in hard-copy form) containing:

1. a brief overview over the implemented theory
2. an overview over the structure of the program (routines, files, ...), in text form or graphically
3. a short user's manual answering the following questions:
   - How to start the program?
   - Where does the program get its input from?
   - What output does the program generate and where does it store it to?
4. verification: results requested in section 3.3 and 3.4

5 Remarks

- The successful completion of the task is a prerequisite to be admitted to the final examination.
- The deadline for the assignment is Friday, July 10, 2020 when the program has to be sent to Geralf.Huetter@imfd.tu-freiberg.de and the documentation has to be submitted in hard-copy form at secretary of the institute (room WEI-130). The program has to be presented *individually* before the examination. Details on the mode of presentation will be decided and published, depending on the circumstances of teaching at that time.

Appendix

Click **here** in AdobeReader to download code sniplet for generating a local mesh refinement.