TECHNISCHE UNIVERSITÄT
BERGAKADEMIE FREIBERG

Die Ressourcenuniversität. Seit 1765.

Non-Linear Finite Element Methods

Assignment Summer Semester 2020

Submitted by:

Harihara Rajan Ramaswamy

Matriculation Number: 65076

harihara-rajan.ramaswamy@student.tu-freiberg.de

Contents

1 Problem Statement

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

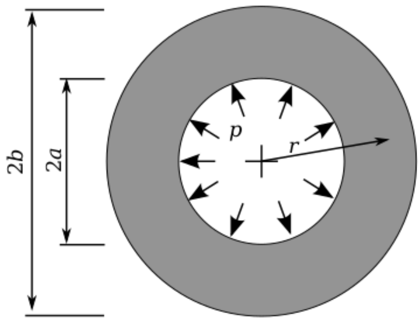


Figure 1: Thick-walled pipe

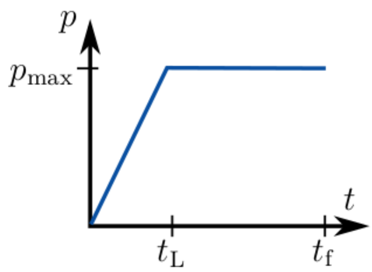


Figure 2: Load sequence

axisymmetric conditions, the only non-vanishing equilibrium condition is

$$0 = \frac{\partial \left(r\sigma_{rr} \right)}{\partial r} - \sigma_{\phi\phi}$$

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

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

with stresses and strains written in Voigt notation as

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

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 $\nVdash$ 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

$$\begin{aligned} \underline{\sigma} &= \underline{\mathbf{C}} \cdot \underline{\varepsilon} + \underline{\sigma}^{\mathrm{ov}} \\ \dot{\sigma}^{\mathrm{ov}} &= Q \, \mathrm{dev}(\underline{\dot{\varepsilon}}) - \frac{1}{T} \underline{\sigma}^{\mathrm{ov}} \end{aligned}$$

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

2 Brief Overview

2.1 Weak Form Discretization in Space

Given the weak form in the problem

$$\delta W = \int_a^b \underline{\delta\varepsilon}^{\mathrm{T}} \underline{\sigma} r \partial r - [r\sigma_{rr}\delta u_r]_a^b \tag{1}$$

$$\delta W = \int_a^b \underbrace{(B\delta u^e)^T \sigma * r * \partial r}_{W_{internal}} - \underbrace{[r\sigma_{rr}\delta u_r]_a^b}_{W_{external}} \tag{2}$$

$$(W_{internal}) = \int_a^b (B\delta u^e)^T \, \sigma \cdot r \cdot \partial r \tag{3}$$

$$(W_{internal}) = \int_a^b \delta u^{e^T} B^T \cdot \sigma \cdot r \cdot \partial r \tag{4}$$

converting from physical domain to unit domain,

$$(W_{internal}) = \int_{-1}^1 \delta u^{e^T} B^T \cdot \sigma \cdot r \cdot \underbrace{\frac{\partial r}{\partial \xi}}_{J} \cdot \partial \xi \tag{5}$$

where, J is the jacobian that maps element from the physical coordinates 'r' to the unit coordinate system. The radial line in the physical domain can be discretized as an iso-parametric line elements. The coordinates r and the radial displacement $u_r(r)$ in the elements are interpolated through the same Ansatz function via the nodal coordinates and nodal displacement respectively. The interpolation function $N(\xi)$ given in the problem is,

$$N(\xi) = \left[\frac{1-\xi}{2}, \frac{1+\xi}{2}\right]^T \tag{6}$$

$$\begin{aligned} \underline{u}_r &= N^T * \widetilde{u} = N_1 u_1 + N_2 u_2 \\ r &= N^T * \widetilde{r} = N_1 r_1 + N_2 r_2 \end{aligned} \tag{7}$$

substituting (7) equation in (5) we get,

$$(W_{internal}) = \delta u^{e^T} \int_{-1}^1 \underbrace{B^T \cdot \sigma \cdot N^T \cdot \underline{r}^T \cdot J}_{(F_{internal})^e} \cdot \partial \xi \tag{8}$$

The above $(F_{internal})^e$ can be integrated by using the Gauss Quadrature method. For this problem, integration is done exactly at the mid point with $w_\alpha = 2$ and $\xi = 0$.

$$(F_{internal})^e = w_\alpha * \left(\sum_{i=1}^n B^T \cdot \sigma \cdot N^T \cdot \underline{r}^T \cdot J \right) \tag{9}$$

we know that $\delta u^{e^T} = (A.\delta U)^T$, substituting this in 8, and performing integration using Gauss Quadrature scheme we get,

$$(F_{internal})^g = \delta U^T \int_{-1}^1 \underbrace{A^T \cdot B^T \cdot \sigma \cdot N^T \cdot \underline{r}^T \cdot J}_{(F_{external})^g} \cdot \partial \xi \tag{10}$$

$$(F_{internal})^g = w_\alpha * \left(\sum_{i=1}^n A^T \cdot B^T \cdot \sigma \cdot N^T \cdot \underline{r}^T \cdot J \right) \tag{11}$$

$$(F_{external}) = \begin{bmatrix} pa \\ 0 \\ \vdots \\ 0 \end{bmatrix}_{numele+1} \tag{12}$$

we know $\sigma = C\varepsilon$, and B matrix connects strain (ε) matrix and displacement(u) vector and by $\varepsilon = B * u$, substituting this relation in (9) we get, element stiffness matrix $(\mathrm{K})^e$

$$(K)^e = w_\alpha * \left(\sum_{i=1}^{n} B^T * C^T * B * N^T * r^T * J \right) \tag{13}$$

where, $\boldsymbol{J}$ is the Jacobian and $\boldsymbol{B}$ is the strain displacement matrix and

$$J = \frac{r_2 - r_1}{2} \tag{14}$$

$$B = \begin{bmatrix} \frac{-1}{2*J} & \frac{1}{2*J} \\ \frac{N_1}{(N_1*r_1+N_2*r_2)} & \frac{N_2}{(N_1*r_1+N_2*r_2)}) \end{bmatrix} \tag{15}$$

from (13) global stiffness matrix can be obtained by using the following relation,

$$(K)^g = \sum_{i=1}^{n} (A^e)^T * K^e * (A^e) \tag{16}$$

2.2 Time Discretization of Material Law

In our problem Stress (σ) is not only function of strain (ε), but stress (σ) is also a function of internal state variable over stress (σ_{ov}), which can be seen from the material law below,

$$\begin{aligned} \underline{\sigma} &= \underline{\mathbf{C}} \cdot \underline{\varepsilon} + \underline{\sigma}^{\mathrm{ov}} \\ \underline{\dot{\sigma}}^{\mathrm{ov}} &= Q\,\mathrm{dev}(\underline{\dot{\varepsilon}}) - \frac{1}{T}\underline{\sigma}^{\mathrm{ov}} \end{aligned} \tag{17}$$

in order to find stress (σ) we need to perform time integration on the internal state variable equation $(17)_2$.In this problem, **Euler's Backward Method** is used as a time integration scheme.

$$\int_m^{m+1} (\dot{\sigma})_{ov} \cdot \partial\tau = Q \int_m^{m+1} dev(\dot{\epsilon}) \cdot \partial\tau - 1/T * \int_m^{m+1} (\sigma)_{ov} \cdot \partial\tau \tag{18}$$

$$(\sigma)_{ov}^{m+1} = (\sigma)_{ov}^{m} + Q\,\mathrm{dev}\,(\varepsilon_{m+1} - \varepsilon_m) - 1/T \left[\sigma_{ov}(\underbrace{t_{m+1} - t_m}_{\Delta t}) \right] \tag{19}$$

$$(\sigma)_{ov}^{m+1} = (\sigma_{ov}^{m} + Q\,\mathrm{dev}(\Delta\varepsilon)) - \frac{\Delta t}{T}\left((1-\gamma)(\sigma)_{ov}^{m} + \gamma(\sigma)_{ov}^{m+1}\right). \tag{20}$$

Grouping stresses of current time step $(\sigma)_{ov}^{m+1}$ on one side and stresses of previous time step $(\sigma)_{ov}^{m}$ on the other side we get,

$$(\sigma)_{ov}^{m+1} + \frac{\Delta t}{T}\left(\gamma(\sigma)_{ov}^{m+1}\right) = (\sigma)_{ov}^{m} + Q\,\mathrm{dev}(\delta\varepsilon) - \frac{\Delta t}{T}(1-\gamma)(\sigma)_{ov}^{m} \tag{21}$$

$$(\sigma)_{ov}^{m+1} = \left(\frac{1}{1 + \frac{\Delta t * \gamma}{T}} \right) \left(Q\,\mathrm{dev}(\Delta\varepsilon) + (\sigma)_{ov}^{m} - \frac{\Delta t}{T}(1-\gamma)(\sigma)_{ov}^{m} \right) \tag{22}$$

In our problem, since we have used Euler Backward integration scheme for time integrations of internal state variable, the value of $\gamma = 1$ and hence the last term of (22) vanishes.

$$(\sigma)_{ov}^{m+1} = \left(\frac{1}{1 + \frac{\Delta t * \gamma}{T}} \right) (Q\,\mathrm{dev}(\Delta\varepsilon) + (\sigma)_{ov}^{m}) \tag{23}$$

And, finally the elemental stress can be obtained by substituting (23) in $(17)_1$. and this value of elemental stress is used in calculating the $(\mathrm{F}_{internal})^e$.

3 An overview - structure of the implemented program

The entire program is splitted into six *.py scripts,

1. inparams.py
2. elemental-routine.py
3. material-routine.py
4. main.py
5. analytical.py
6. mesh-refinement.py

3.1 Input File

The ***inparams.py*** fileconsist of parameters like Inner radius, Outer radius, Youngs modulus, Number of elements, Maximum pressure, and Poisons ratio.

3.2 Element Routine

The ***elemental − routine.py*** file consist of four functions that returns the following,

```
1. N matrix
       --> N matrix
2. Jacobian Matrix
      --> jacobian for each element
3. Strain Displacement matrix
      --> B matrix for each element
3. Assignment matrix
      --> assignment matrix list, that contains assignment matrix of all element
4. fint_elements_K_elements
      --> elemental internal force
      --> elemental stiffness matrix
      --> elemental overstress
      --> elemental stress
```

3.3 Material Routine

The ***material − routine.py*** file consist of one functions that returns the following,

```
1. material routine
      --> Stress
      --> overstress
      --> tangent Stiffness matrix
```

3.4 Main Program

The ***main.py***, takes the input and results from *.py scripts and for each time step, element routine is called, and with in the elemental routine, material routine is called for computing the stress value. once the stress value is obtained from the material routine, elemental routine now computes and returns $(F_{internal})^e, (K^e), (\sigma)^e$ and $(\sigma_{ov})^e$. Global matrices of $(F_{internal})^e$, (K^e) and Residual are computed in the main program. Once all the values are obtained convergence criteria is checked, the Newton-Raphson scheme is performed until the convergence criteria is met before starting with the next time step.

3.5 Analytical and Mesh Refinement Files

The ***analytical.py*** script contains the function that returns the exact solution. The ***mesh − refinement.py*** scripts returns the nodal position for the given number of elements. Mesh refinement is done in such a way that, close to inner radius the number of elements will be twice as much as the number of elements close to outer radius.

3.6 User Manual

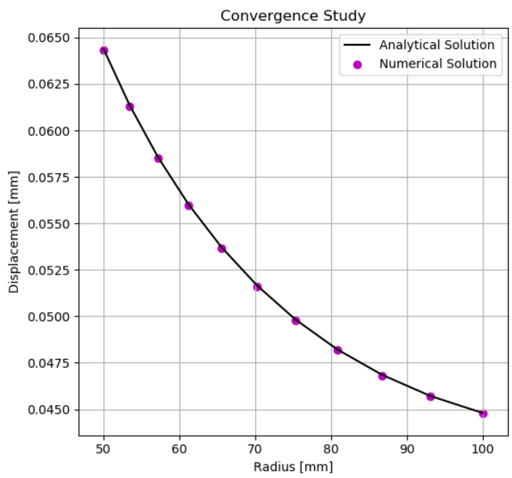
All the files that are required for successful compiling of program is already imported in the main file. Once the program is executed, at the very beginning the program will ask the user to input to whether to solve for linear elastic case or visco-elastic case, when the user gives input as linear, the program automatically takes the viscous modulus Q=0 and performs convergence study, widening study, stress distribution study and plots are generated. If the user give input as non-linear, the program automatically takes the viscous modulus Q = 100,000 and performs all the study that was mentioned in linear case and finally plots are generated.

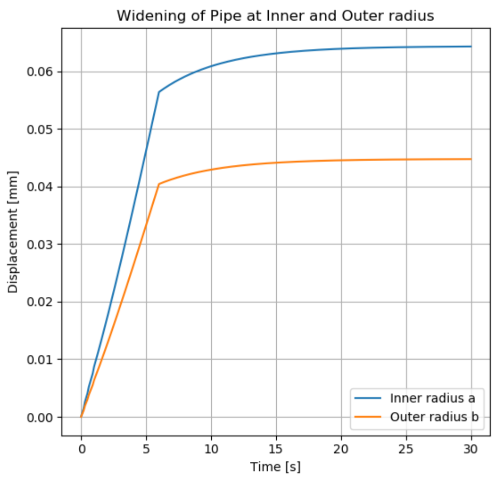
The values that are required for plotting purpose, like radial stress, tangential stress, displacement at outer and inner radius and analytical displacement are stored in the separate array.

4 Results

4.1 Convergence and Widening Study

Convergence study is performed, the convergence occurs at 0.1 time step (dt) for 10 elements. For every time step from 0 second to 30 seconds, the displacement at inner radius 'a' and outer radius 'b' are extracted and widening study is performed. The convergence and widening plots are shown below





4.2 Stress Distribution

The values of both Radial $(\sigma)_{rr}$ and Tangential $(\sigma)_{\phi\phi}$ have been extracted and plotted against the radius. It can be interpreted from the radial stress distribution plot, that the radial stress $(\sigma)_{rr}$ asymptotically reducing towards zero, which is in line with our boundary condition stating radial stress $(\sigma)_{rr}$ vanishes at the outer radius. similarly, distribution of tangential stress $(\sigma_{\phi\phi})$ also been plotted against the radius.