

## FEM advanced course

### 4. exercise – numerical integration, coding

**Problem 1.** Consider a single quadratic line element. The interpolation functions are in the parent element coordinate  $\xi$  as  $N_1(\xi) = \frac{1}{2}\xi(\xi-1)$ ,  $N_2(\xi) = 1-\xi^2$  and  $N_3(\xi) = \frac{1}{2}\xi(1+\xi)$  and  $\xi \in [-1, 1]$ .

Compute by numerical integration the element e.g.  $K_{11}$  or  $K_{12}$  of the stiffness matrix of a linear bar element

$$K_{ij} = \int_{x_1^{(e)}}^{x_3^{(e)}} \frac{dN_i}{dX} EA \frac{dN_j}{dX} dX.$$

The integration points and weights for the 1,2 and 3 point Gauss-Legendre quadrature are given in the table below for integration of

$$\int_{-1}^1 f(\xi) d\xi \approx \sum_{i=1}^n w_i f(\xi_i)$$

where the residual error  $R$  is given as

$$\frac{2^{2n+1}(n!)^4}{(2n+1)[(2n)!]^3} \frac{d^{2n}f}{d\xi^{2n}}.$$

What can you say about the residual if the nodes are placed uniformly, like  $X_1 = 0$ ,  $X_2 = 50$  mm,  $X_3 = 100$  mm or the middle node is not placed at the center of an element.

$n$	$\xi_i$	$w_i$
1	0	2
2	$-1/\sqrt{3}$	1
	$1/\sqrt{3}$	1
3	$-\sqrt{3/5}$	5/9
	0	8/9
	$\sqrt{3/5}$	5/9

Code the process to compute the elements of the stiffness matrix of an isoparametric element by numerical integration in a way that it can be easily generalized. The process is given in the pseudo-code on the next page.

**Home assignment 4.** Extend your non-linear 1D-truss element code to handle quadratic isoparametric interpolation- and shape functions. Use as a material model the incompressible Mooney-Rivlin model having the strain energy function (3.115) in the study book

$$W(\lambda_1, \lambda_2, \lambda_3) = \frac{1}{2}\mu_1(\lambda_1^2 + \lambda_2^2 + \lambda_3^2 - 3) - \frac{1}{2}\mu_2(\lambda_1^{-2} + \lambda_2^{-2} + \lambda_3^{-2} - 3). \quad (1)$$

The principal PK2-stresses can now be obtained as

$$S_i = \frac{1}{\lambda_i} \frac{\partial W}{\partial \lambda_i}. \quad (2)$$

Solve the problem of a hanging rubber band. Displacement is suppressed at  $X = 0$  and the other end at  $X = L_0$  is free. The loading is now the gravity load  $\rho_0 g$  in the positive  $X$ -axis direction. Use 1, 2 and 100 elements.

Choose  $\mu_1$  and  $\mu_2$  such that the initial response is the same as in before. For the density use 1000 times the density of rubber, i.e. about  $\rho_0 = 1.1 \cdot 10^6 \text{ kg/m}^3$ . You can use same dimensions for initial length and cross-section area as in problem 2 i.e.  $L_0 = 100 \text{ mm}$  and  $A_0 = 10 \text{ mm}^2$ .

Plot also the Cauchy stress curve along the  $X$ -coordinate.

**Solution report should be returned in Moodle prior to exercise 6**

### Pseudo-code for setting up element stiffness matrix and internal force vector

1. Set up the Gauss-points and the weights
2. Zero the element stiffness matrix and internal force vector
3. Loop over the integration points and sum up the contributions
  - (a) Call the routine which gives the values of the local derivatives of the interpolation function (and possibly the interpolation functions if necessary)
  - (b) Compute the Jacobian and the global derivatives of the interpolation functions
  - (c) Zero the B-matrix
  - (d) Compute the  $\mathbf{B}$ -matrix
  - (e) Set up material stiffness matrix  $\mathbb{C}$
  - (f) Multiply the values of  $\mathbb{C}$  by the product of weight and Jacobian determinant
  - (g) Add the contribution of the product  $\mathbf{B}^T \mathbb{C} \mathbf{B}$  to the element stiffness matrix
  - (h) If the problem is nonlinear
    - i. Compute stresses and multiply them by the product of weight and Jacobian determinant
    - ii. Compute  $\mathbf{B}^T$ \*stresses and add the contribution to the element internal force vector
    - iii. Compute the geometric stiffness matrix part and add the contribution to the element stiffness matrix

Since we are using isoparametric element, the element geometry is also interpolated with the same interpolation function as the unknown displacement, i.e.

$$X = N_1 X_1^{(e)} + N_2 X_2^{(e)} + N_3 X_3^{(e)},$$

thus to evaluate it we need just a scalar product of two vector arrays, one having the values of the interpolation functions and the other containing the node coordinates. In this one dimensional case the geometry Jacobian can be computed as

$$J = \frac{1}{\sum_{i=1}^{\text{nodes}} N_{i,\xi} X_i^{(e)}},$$

where  $N_{i,\xi}$  are the interpolation function derivatives w.r.t. the parent element local coordinate  $\xi$ . Then the global derivatives are obtained from

$$N_{i,X} = J^{-1} N_{i,\xi}.$$