Computational Nonlinear Mechanics

Assignment 2: Hyperelasticity and FE implementation of large strains

Rostyslav Skrypnyk*

August 23, 2019

Contents

*Department of Mechanics and Maritime Sciences, rostyslav.skrypnyk@chalmers.se

Computer assignment 2: Hyperelasticity and FE implementation of large strains

The goal of this computer assignment is that you should practise how to program a hyperelastic constitutive model and a finite element program that takes into account large deformations.

a Implemement the *Yeoh* hyperelastic material model defined by the following free energy:

$$\Psi = \frac{\mu}{2}\left(I_C - 3\right) + c_2\left(I_C - 3\right)^2 + c_3\left(I_C - 3\right)^3 - \mu \ln J + \frac{\lambda}{2}\left(\ln J\right)^2$$

i.e. write a subroutine (function in Matlab) that computes the 2nd Piola-Kirchhoff stress $\boldsymbol{S}$ and the stiffness $\mathcal{C}$ given the right Cauchy-Green deformation tensor $\boldsymbol{C}$.

Plot the Cauchy stress component σ_{11} against $\epsilon_{11} = F_{11} - 1$ for a uniaxial strain control $\boldsymbol{F} = F_{11}\boldsymbol{e}_1 \otimes \boldsymbol{E}_1$ with the material parameters suitable for natural rubber (see http://polymerfem.com) $\mu = 6.9$ [MPa], $\lambda = 62.1$ [MPa], $c_2 = -\mu/10$ and $c_3 = \mu/30$.

b Program a subroutine (function in Matlab) that computes the shape functions N_a and their derivatives $\nabla_0 N_a$, ∇N_a as well as the deformation gradient $\boldsymbol{F}$ for a CST triangle (plane strain). Input to the subroutine is the initial (material) positions of the three nodes $\boldsymbol{X}_1$ $\boldsymbol{X}_2$, $\boldsymbol{X}_3$ together with their current (spatial) positions $\boldsymbol{x}_1$ $\boldsymbol{x}_2$, $\boldsymbol{x}_3$. Check that your code give the results given in Example 9.1 and 9.2.

$$\frac{\mathrm{d}N}{\mathrm{d}\boldsymbol{X}} = \begin{bmatrix} -0.25000 & 0.25000 & 0 \\ -0.33333 & 0 & 0.33333 \end{bmatrix}$$

$$\frac{\mathrm{d}N}{\mathrm{d}\boldsymbol{x}} = \begin{bmatrix} -0.12500 & 0.12500 & 0 \\ 0 & -0.16667 & 0.16667 \end{bmatrix}$$

and that (for the Neo-Hookean model with $\mu = 3$ and $\lambda = 2$) the 2nd PK-stress becomes:

$$\boldsymbol{S} = \begin{bmatrix} 2.8421 & 2.9431 & 2.7726 & 7.5804 \cdot 10^{-2} & 0 & 0 & 0 & 7.5804 \cdot 10^{-2} & 0 \end{bmatrix}^{\mathrm{T}}$$

c Program a subroutine that computes the equivalent nodal forces $\boldsymbol{T}_a^{(e)}$ for a CST element. Check that your code give the results given in Example 9.3.

$$\boldsymbol{T}_a^{(e)} = [-24.829 \ -12.000 \ 8.8294 \ 0.22741 \ 16.000 \ 11.773]^{\mathrm{T}}$$

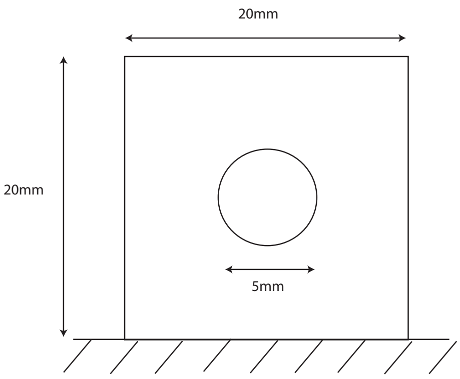
d Program a subroutine that computes the tangent matrices (constitutive) $\boldsymbol{K}_{c,ab}$ and (initial stress) $\boldsymbol{K}_{\sigma,ab}$. Check that your code give the results given in Example 9.4 and 9.5.

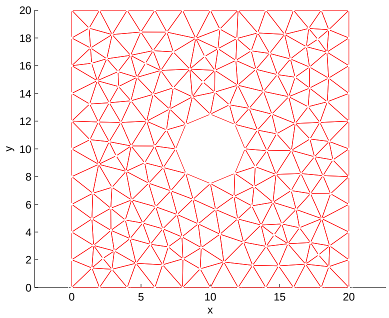
$$\boldsymbol{K}_{c,ab} = \begin{bmatrix} 0.23014 & 0 & -0.23014 & 0.25000 & 0 & -0.25000 \\ 0 & 0.021320 & 0.028426 & -0.021320 & -0.028426 & 0 \\ -0.23014 & 0.028426 & 0.26804 & -0.27843 & -0.037902 & 0.25000 \\ 0.25000 & -0.021320 & -0.27843 & 0.43046 & 0.028426 & -0.40914 \\ 0 & -0.028426 & -0.037902 & 0.028426 & 0.037902 & 0 \\ -0.25000 & 0 & 0.25000 & -0.40914 & 0 & 0.40914 \end{bmatrix}$$

$$
\boldsymbol{K}_{\sigma,ab} = \begin{bmatrix} 3.1037 & 0 & -1.1037 & 0 & -2.0000 & 0 \\ 0 & 3.1037 & 0 & -1.1037 & 0 & -2.0000 \\ -1.1037 & 0 & 1.0658 & 0 & 0.037902 & 0 \\ 0 & -1.1037 & 0 & 1.0658 & 0 & 0.037902 \\ -2.0000 & 0 & 0.037902 & 0 & 1.9621 & 0 \\ 0 & -2.0000 & 0 & 0.037902 & 0 & 1.9621 \end{bmatrix}
$$

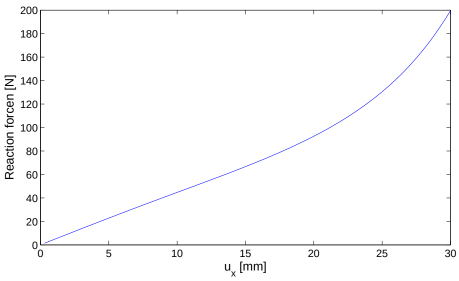
$$
\boldsymbol{K}_{ab} = \begin{bmatrix} 3.3338 & 0 & -1.3338 & 0.25000 & -2.0000 & -0.25000 \\ 0 & 3.1250 & 0.028426 & -1.1250 & -0.028426 & -2.0000 \\ -1.3338 & 0.028426 & 1.3338 & -0.27843 & 0 & 0.25000 \\ 0.25000 & -1.1250 & -0.27843 & 1.4962 & 0.028426 & -0.37124 \\ -2.0000 & -0.028426 & 0 & 0.028426 & 2.0000 & 0 \\ -0.25000 & -2.0000 & 0.25000 & -0.37124 & 0 & 2.3712 \end{bmatrix}
$$

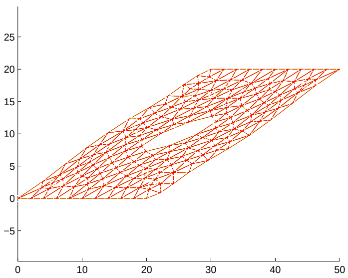
e Finally, combine your subroutines developed in a-d with your code in Computer assignment 1 (together with CALFEM routines) to solve the following boundary values problem The lower boundary is fixed while the nodes on the





upper boundary are displaced in the horizontal (x) direction to $u_{\max} = 30$ [mm]. Assume the thickness $t = 1$ [mm]. Plot the deformed mesh as well as the total horizontal reaction force vs displacement on the upper nodes.





The CALFEM variables for this example

```
Ex,Ey,Coord,Edof,Dof
```

are given in the file `cass2_mesh_data.mat`.

Hints:

a For the closely related Neo-Hooke material model the subroutines

`test_const.m, neo_hooke.m, v9_2_m.m, m_2_v9.m, f9_open_u_9.m, f9_open_l_9.m`

solve same thing as you should do in a. Also the push-forward operation of the stiffness $\mathcal{C}$ to $\boldsymbol{c}$ is given in the code using (6.14) which can be written as

$$\boldsymbol{c} = J^{-1} \left[\boldsymbol{F} \overline{\otimes} \boldsymbol{F}\right] : \mathcal{C} : \left[\boldsymbol{F}^{\mathrm{T}} \overline{\otimes} \boldsymbol{F}^{\mathrm{T}}\right]$$

or in index form

$$\boldsymbol{c}_{ijkl} = J^{-1} \left[\boldsymbol{F}_{iM} \overline{\otimes} \boldsymbol{F}_{jN}\right] : \mathcal{C}_{MNOP} : \left[\boldsymbol{F}^{\mathrm{T}}_{Ok} \overline{\otimes} \boldsymbol{F}^{\mathrm{T}}_{Pl}\right]$$

In these files a 2nd order tensor is represented by a 9x1 vector and a 4th order tensor is represented by a 9x9 matrix. Example, 2nd order tensor A_{ij}:

```
A=[A11 A22 A33 A12 A23 A31 A13 A21 A32]';
```

Example, 4th order tensor $\mathcal{B}_{ijkl}$:

```
B=[B1111 B1122 B1133 B1112 B1123 B1131 B1113 B1121 B1132;
   B2211 B2222 B2233 B2212 B2223 B2231 B2213 B2221 B2232;
   B3311 B3322 B3333 B3312 B3323 B3331 B3313 B3321 B3332;
   B1211 B1222 B1233 B1212 B1223 B1231 B1213 B1221 B1232;
   B2311 B2322 B2333 B2312 B2323 B2331 B2313 B2321 B2332;
   B3111 B3122 B3133 B3112 B3123 B3131 B3113 B3121 B3132;
   B1311 B1322 B1333 B1312 B1323 B1331 B1313 B1321 B1332;
   B2111 B2122 B2133 B2112 B2123 B2131 B2113 B2121 B2132;
   B3211 B3222 B3233 B3212 B3223 B3231 B3213 B3221 B3232];
```

By using this representation and the subroutines listed above the following Matlab commands might be useful

$$\boldsymbol{A} : \boldsymbol{B} = A_{ij} B_{ij} = \texttt{sum(A.*B)}$$

$$\boldsymbol{A} \otimes \boldsymbol{B} = A_{ij} B_{kl} = \texttt{A*B'}$$

$$\boldsymbol{A}^{-1} = A_{ij}^{-1} = \texttt{m\_2\_v9(inv(v9\_2\_m(A)))}$$

$$\boldsymbol{A} \overline{\otimes} \boldsymbol{B} = A_{ik} B_{jl} = \texttt{f9\_open\_u\_9(A,B)}$$

$$\boldsymbol{A} \underline{\otimes} \boldsymbol{B} = A_{il} B_{jk} = \texttt{f9\_open\_l\_9(A,B)}$$

$$\det(\boldsymbol{A}) = \texttt{det(v9\_2\_m(A))}$$

$$\boldsymbol{\mathcal{B}} : \boldsymbol{A} = \mathcal{B}_{ijkl} \, A_{kl} = \texttt{B*A}$$

$$\boldsymbol{\mathcal{B}}^{-1} = \texttt{inv(B)}$$

1 Task A

The 2nd Piola-Kirchhoff stress can be obtained from free energy using equation (6.18) in [1]:

$$\underset{\sim}{\boldsymbol{S}} = 2\frac{\partial \Psi}{\partial \underset{\sim}{\boldsymbol{C}}} = 2\frac{\partial \Psi}{\partial I_C}\frac{\partial I_C}{\partial \underset{\sim}{\boldsymbol{C}}} + 2\frac{\partial \Psi}{\partial II_C}\frac{\partial II_C}{\partial \underset{\sim}{\boldsymbol{C}}} + 2\frac{\partial \Psi}{\partial III_C}\frac{\partial III_C}{\partial \underset{\sim}{\boldsymbol{C}}} \tag{6.18}$$

with derivatives of invariants of Green-Lagrange deformation tensor defined as ($J = \det(\underset{\sim}{\boldsymbol{F}})$)

$$\frac{\partial I_C}{\partial \underset{\sim}{\boldsymbol{C}}} = \underset{\sim}{\boldsymbol{I}} \tag{6.19a}$$

$$\frac{\partial II_C}{\partial \underset{\sim}{\boldsymbol{C}}} = 2\underset{\sim}{\boldsymbol{C}} \tag{6.19b}$$

$$\frac{\partial III_C}{\partial \underset{\sim}{\boldsymbol{C}}} = J^2\underset{\sim}{\boldsymbol{C}}^{-1} \tag{6.22}$$

Find the missing derivatives of the free energy (given in the task) with respect to the invariants:

$$\frac{\partial \Psi}{\partial I_C} = \frac{\mu}{2} + 2c_2\left(I_C - 3\right) + 3c_3\left(I_C - 3\right)^2 \tag{1}$$

$$\frac{\partial \Psi}{\partial II_C} = 0 \tag{2}$$

$$\frac{\partial \Psi}{\partial III_C} = \frac{\lambda \ln J - \mu}{2J^2} \tag{3}$$

Lastly, the expression for the 2nd Piola-Kirchhoff stress yields

$$\underset{\sim}{\boldsymbol{S}} = \left[\mu + 4c_2\left(I_C - 3\right) + 6c_3\left(I_C - 3\right)^2\right]\underset{\sim}{\boldsymbol{I}} + \left(\lambda \ln J - \mu\right)\underset{\sim}{\boldsymbol{C}}^{-1} \tag{4}$$

The Lagrangian elasticity tensor can be obtained using equation (6.11):

$$\underset{\approx}{\boldsymbol{C}} = \frac{\partial \underset{\sim}{\boldsymbol{S}}}{\partial \underset{\sim}{\boldsymbol{E}}} = 2\frac{\partial \underset{\sim}{\boldsymbol{S}}}{\partial \underset{\sim}{\boldsymbol{C}}} \tag{6.11}$$

Expanding invariants to have explicit dependence on $\underset{\sim}{\boldsymbol{C}}$ yields (*note the overbar open product*)

$$\frac{\partial \underset{\sim}{\boldsymbol{S}}}{\partial \underset{\sim}{\boldsymbol{C}}} = \left[4c_2 + 12c_3\left(I_C - 3\right)\right]\underset{\sim}{\boldsymbol{I}} \otimes \underset{\sim}{\boldsymbol{I}} + \frac{\lambda}{2}\underset{\sim}{\boldsymbol{C}}^{-1} \otimes \underset{\sim}{\boldsymbol{C}}^{-1} + \left(\mu - \lambda \ln J\right)\underset{\sim}{\boldsymbol{C}}^{-1}\overline{\otimes}\underset{\sim}{\boldsymbol{C}}^{-1} \tag{5}$$

Cauchy stress is a push-forward of the 2nd Piola-Kirchhoff:

$$\underset{\sim}{\boldsymbol{\sigma}} = J^{-1}\underset{\sim}{\boldsymbol{F}} \cdot \underset{\sim}{\boldsymbol{S}} \cdot \underset{\sim}{\boldsymbol{F}}^T \tag{5.45b}$$

Figure 1 shows component of Cauchy stress plotted against engineering strain for the situation of uniaxial strain control $\underset{\sim}{\boldsymbol{F}} = F_{11}\underline{\boldsymbol{e}}_1 \otimes \underline{\boldsymbol{E}}_1$. It is evident that the curve is nonlinear.

The Matlab implementation of this task can be found in `yeoh.m` (see section 6).

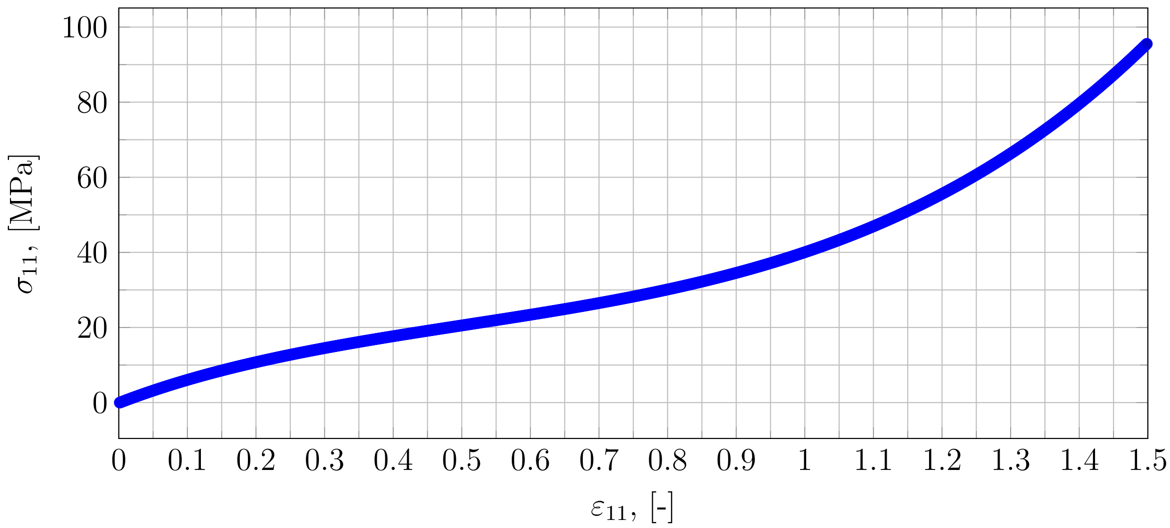


Figure 1: Cauchy stress component σ_{11} versus strain $\varepsilon_{11} = F_{11} - 1$.

2 Task B

For a constant strain triangle (CST) element, the shape functions can be defined as

$$N_1 = 1 - \xi - \eta, \quad N_2 = \xi, \quad N_3 = \eta \tag{6}$$

According to equations (9.6ab), the *material* gradient of shape function reads

$$\underline{\nabla}_0 N_a = \frac{\partial N_a}{\partial \underline{\boldsymbol{X}}} = \left(\frac{\partial \underline{\boldsymbol{X}}}{\partial \underline{\boldsymbol{\xi}}}\right)^{-T} \frac{\partial N_a}{\partial \underline{\boldsymbol{\xi}}} = \left(\frac{\partial \underline{\boldsymbol{X}}}{\partial \underline{\boldsymbol{\xi}}}\right)^{-T} \cdot \underline{\nabla}_\xi N_a, \quad \frac{\partial \underline{\boldsymbol{X}}}{\partial \underline{\boldsymbol{\xi}}} = \sum_{a=1}^{n} \underline{\boldsymbol{X}}_a \otimes \underline{\nabla}_\xi N_a \tag{9.6ab}$$

$$\underline{\nabla}_\xi N_a = \begin{bmatrix} \dfrac{\partial N_a}{\partial \xi} \\ \dfrac{\partial N_a}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial N_1}{\partial \xi} & \dfrac{\partial N_2}{\partial \xi} & \dfrac{\partial N_3}{\partial \xi} \\ \dfrac{\partial N_1}{\partial \eta} & \dfrac{\partial N_2}{\partial \eta} & \dfrac{\partial N_3}{\partial \eta} \end{bmatrix} = \begin{bmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \tag{7}$$

Similarly, the *spatial* shape function gradients are given as

$$\underline{\nabla} N_a = \frac{\partial N_a}{\partial \underline{\boldsymbol{x}}} = \left(\frac{\partial \underline{\boldsymbol{x}}}{\partial \underline{\boldsymbol{\xi}}}\right)^{-T} \cdot \underline{\nabla}_\xi N_a, \quad \frac{\partial \underline{\boldsymbol{x}}}{\partial \underline{\boldsymbol{\xi}}} = \sum_{a=1}^{n} \underline{\boldsymbol{x}}_a \otimes \underline{\nabla}_\xi N_a \tag{9.11ab}$$

Lastly, the deformation gradient can be expressed as

$$\underset{\sim}{\boldsymbol{F}} = \sum_{a=1}^{n} \underline{\boldsymbol{x}}_a \otimes \underline{\nabla}_0 N_a \tag{9.5}$$

The Matlab implementation of this task can be found in `shape_gradients.m` (see section 6).

3 Task C

The equivalent internal forces are given as

$$\underline{\boldsymbol{T}}_a^{(e)} = \int_{v^{(e)}} \underset{\sim}{\boldsymbol{\sigma}} \cdot \underline{\nabla} N_a \, dv \tag{9.15b}$$

For a CST element of thickness t the same expression in local coordinates reads

$$\underline{\boldsymbol{T}}_a^{(e)} = \int\int \underset{\sim}{\boldsymbol{\sigma}} \cdot \underline{\nabla} N_a \, t \, dx \, dy = \int_{-1}^{1}\int_{-1}^{1} \underset{\sim}{\boldsymbol{\sigma}} \cdot \underline{\nabla} N_a \, t \det\left(\frac{\partial \underline{\boldsymbol{x}}}{\partial \underline{\boldsymbol{\xi}}}\right) d\xi \, d\eta \tag{8}$$

Apply Gaussian quadrature rule to compute the integral numerically:

$$\underline{\boldsymbol{T}}_a^{(e)} \approx \sum_{i=1}^{\text{nip}} W_i \underset{\sim}{\boldsymbol{\sigma}} \cdot \underline{\nabla} N_a \, t \det\left(\frac{\partial \underline{\boldsymbol{x}}}{\partial \underline{\boldsymbol{\xi}}}\right), \tag{9}$$

where W_i is a weight for each of the integration points. For a CST element, the number of integration points is 1 with coordinates $\xi = 1/3$, $\eta = 1/3$ and the weight $W = 0.5$.

The Matlab implementation of this task can be found in `get_intern_eq_forces.m` (see section 6).

4 Task D

The *constitutive* component of the tangent matrix relating node a to node b is

$$\left[\underline{\underline{\boldsymbol{K}}}_{c,ab}\right]_{ij} = \int_{v^{(e)}} \sum_{k,l=1}^{3} \frac{\partial N_a}{\partial x_k} c_{ijkl} \frac{\partial N_b}{\partial x_l} \, dv, \quad i,j = 1,2,3 \tag{9.35}$$

The 4th order elasticity tensor in *spatial* configuration can be computed by pushing forward its counterpart in the *material* configuration $\partial \underset{\sim}{\boldsymbol{S}} / \partial \underset{\sim}{\boldsymbol{E}}$:

$$\underset{\approx}{\boldsymbol{c}} = J^{-1} \underset{\sim}{\boldsymbol{F}} \overline{\otimes} \underset{\sim}{\boldsymbol{F}} \cdot \frac{\partial \underset{\sim}{\boldsymbol{S}}}{\partial \underset{\sim}{\boldsymbol{E}}} \cdot \underset{\sim}{\boldsymbol{F}}^T \overline{\otimes} \underset{\sim}{\boldsymbol{F}}^T \tag{10}$$

For a CST element of thickness t equation (9.35) can be rewritten as

$$\underline{\underline{\boldsymbol{K}}}_{c,ab} \approx \sum_{i=1}^{\text{nip}} W_i \underline{\nabla} N_a \cdot \underline{\underline{\boldsymbol{D}}} \cdot \underline{\nabla} N_b t \det\left(\frac{\partial \underline{\boldsymbol{x}}}{\partial \underline{\boldsymbol{\xi}}}\right), \tag{11}$$

where $\underset{\approx}{\boldsymbol{c}}$ was rearranged into matrix $\underline{\underline{\boldsymbol{D}}}$ that facilitates computations in the matrix form for 2D case:

$$\underline{\underline{\boldsymbol{D}}} = \begin{bmatrix} c_{1111} & c_{1122} & c_{1112} \\ & c_{2222} & c_{2212} \\ \text{sym} & & c_{1212} \end{bmatrix} \tag{12}$$

The components of the *initial stress* matrix are as follows:

$$\left[\underline{\underline{\boldsymbol{K}}}_{\sigma,ab}\right]_{ij} = \int_{v^{(e)}} \sum_{k,l=1}^{3} \frac{\partial N_a}{\partial x_k} \sigma_{kl} \frac{\partial N_b}{\partial x_l} \delta_{ij} \, dv, \quad i,j = 1,2,3 \tag{9.44c}$$

Then for a CST element this becomes

$$\underline{\underline{\boldsymbol{K}}}_{\sigma,ab} \approx \sum_{i=1}^{\text{nip}} W_i \underline{\nabla} N_a \cdot \underset{\sim}{\boldsymbol{\sigma}} \cdot \underline{\nabla} N_b \underset{\sim}{\boldsymbol{I}} t \det\left(\frac{\partial \underline{\boldsymbol{x}}}{\partial \underline{\boldsymbol{\xi}}}\right) \tag{13}$$

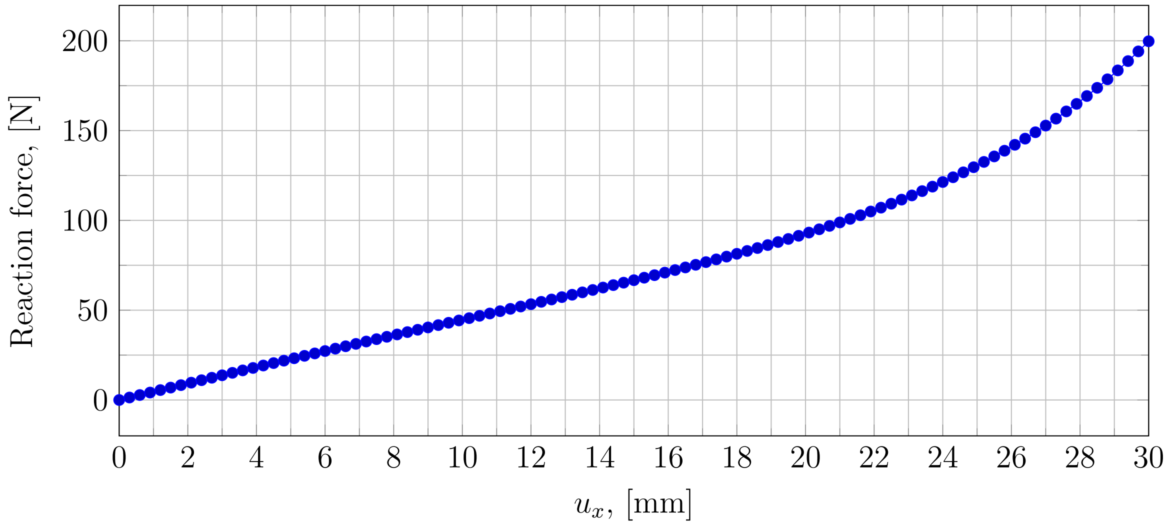
The total tangent matrix is then

$$\underline{\underline{\boldsymbol{K}}}_{ab} = \underline{\underline{\boldsymbol{K}}}_{c,ab} + \underline{\underline{\boldsymbol{K}}}_{\sigma,ab} \tag{14}$$

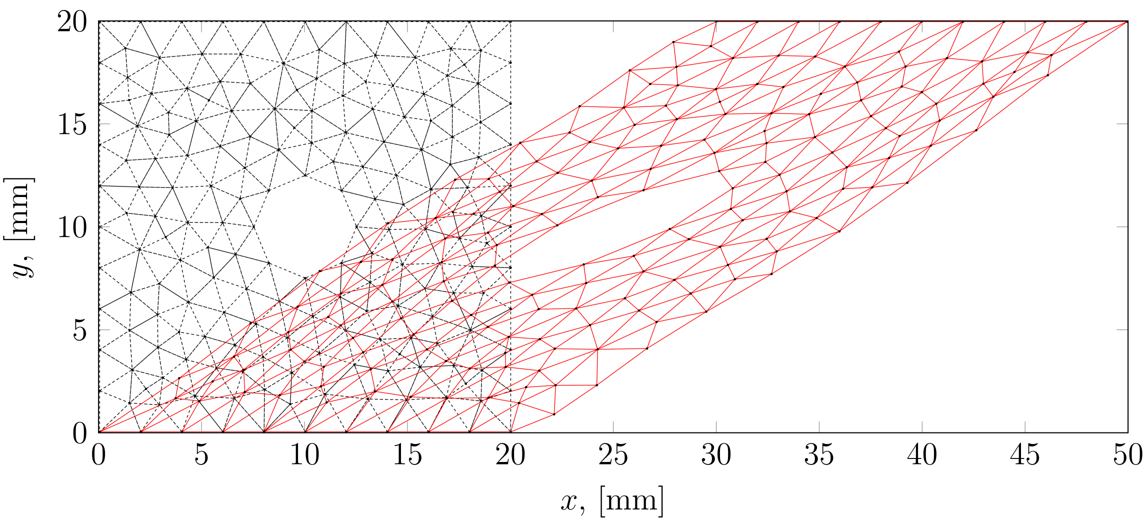
The Matlab implementation of this task can be found in `get_tangent_matrices.m` (see section 6).

5 Task E

The given boundary value problem has been solved using Matlab script `plate_with_hole.m` (see section 6). Figure 2 shows total reaction force plotted against displacement of the top nodes and the shape of the plate before and after the deformation.



(a)



(b)

Figure 2: (a) Reaction force versus displacement and (b) deformed mesh.

6 Matlab code

The Matlab code can be found in the Git repository on Github.

References

[1] J. Bonet and R. D. Wood. *Nonlinear continuum mechanics for finite element analysis.* Cambridge University Press, second edition, 2008.