Final Project

About the project:

- You must do Exercise 0
- You can choose two out of three from Exercise 1–3.
- If an exercise asks you for text answers, write them in a separate document (PDF, Word or text file).

Submission: Upload the entire folder `soofeam_project` on the TeachCenter.

Exercise 0

Answer the following questions in a separate document:

1. What was the most exciting/surprising/interesting thing you learned in this lecture?
2. Optional: Choose one *specific* thing you did not understand well in this lecture. You will not be asked about this in the oral exam.

Exercise 1

Implement the compressible Neo-Hookean material. This material is well suited for describing rubber at large deformations. Elastic potential, second Piola Kirchhoff tensor and material elasticity tensor of this material read

$$\Psi(\boldsymbol{C}) = \frac{\mu}{2}\left(I_{\boldsymbol{C}} - 3\right) - \mu \ln\left(\sqrt{III_{\boldsymbol{C}}}\right) + \frac{\lambda}{2}\left(\ln\left(\sqrt{III_{\boldsymbol{C}}}\right)\right)^2$$

$$\boldsymbol{S} = \frac{\partial \Psi}{\partial \boldsymbol{E}}(\boldsymbol{E}) = 2\frac{\partial \Psi}{\partial \boldsymbol{C}}(\boldsymbol{C}) = \mu\left(\boldsymbol{I} - \boldsymbol{C}^{-1}\right) + \lambda \ln\left(\sqrt{III_{\boldsymbol{C}}}\right)\boldsymbol{C}^{-1}$$

$$\mathbb{C} = \frac{\partial^2 \Psi}{\partial \boldsymbol{E}\partial \boldsymbol{E}}(\boldsymbol{E}) = 4\frac{\partial^2 \Psi}{\partial \boldsymbol{C}\partial \boldsymbol{C}}(\boldsymbol{C}) = \lambda \boldsymbol{C}^{-1} \otimes \boldsymbol{C}^{-1} + 2\left(\mu - \lambda \ln\left(\sqrt{III_{\boldsymbol{C}}}\right)\right)\boldsymbol{\mathcal{I}}\,.$$

The tensor invariants of the Cauchy Green deformation tensor $\boldsymbol{C} = \boldsymbol{F}^T\boldsymbol{F}$ are defined as

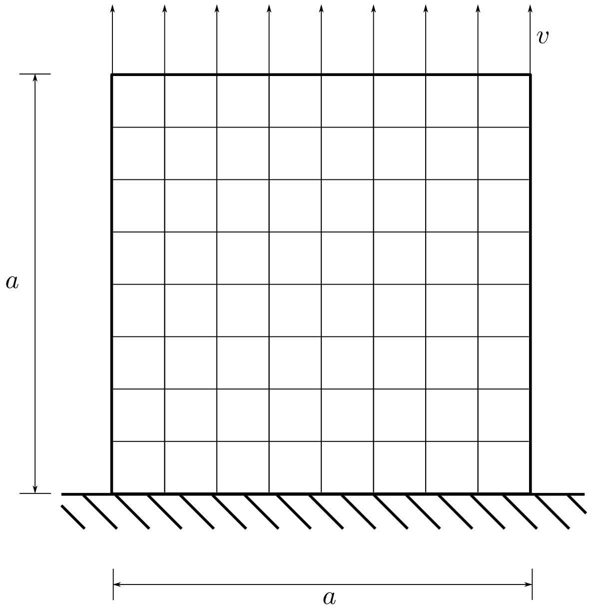
$$\begin{aligned} I_{\boldsymbol{C}} &= \operatorname{tr}(\boldsymbol{C}) = \boldsymbol{C} : \boldsymbol{I} = C_{ii} \\ II_{\boldsymbol{C}} &= \operatorname{tr}\left(\boldsymbol{C}^{\mathrm{T}}\boldsymbol{C}\right) = \boldsymbol{C} : \boldsymbol{C} = C_{ij}C_{ij} \\ III_{\boldsymbol{C}} &= \det(\boldsymbol{C}) = \det(\boldsymbol{F})^2\ , \end{aligned}$$

and the fourth order tensor $\boldsymbol{\mathcal{I}}$ reads

$$\mathcal{I}_{ijkl} = \frac{1}{2}\left(\left(C^{-1}\right)_{ik}\left(C^{-1}\right)_{jl} + \left(C^{-1}\right)_{il}\left(C^{-1}\right)_{jk}\right)\,.$$

Testing

Test the class by creating a material of the Neo-Hookean type and checking, if an input of $E = 0$ yields a stress tensor of $S = 0$. Furthermore, for $E = 0$ the material elasticity tensor must have the same components as the small strain elasticity tensor from the linear theory. Why does this have to be fulfilled?



a	100 mm
E	210 000 N/mm^2
ν	0.3

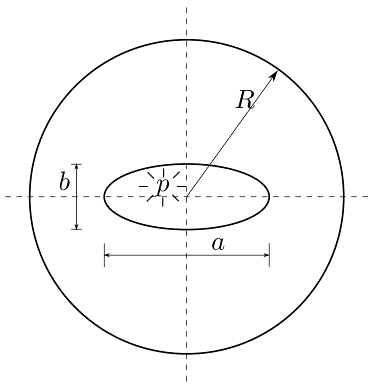
Numerical Example

Simulate the displayed two dimensional body using the given simulation parameters. The bottom boundary of the body is fixed. On the top boundary, a total vertical displacement v is prescribed, while the horizontal movement is not constrained.

- Use the symmetry of the example.
- Create examples with names `plateLinear`, `plateStVenant` and `plateNeoHookean` for a linear and two nonlinear analyses with the respective materials.
- Simulate the examples for $v = 0.1$mm, $v = 1$mm and $v = 10$mm. Choose appropriate parameters for the Newton-Raphson solver in the nonlinear analysis.
- Create a table containing the horizontal displacement of the top left corner for the different simulations. Give an interpretation of the results.

Exercise 2

Simulate the displayed two dimensional body using the given simulation parameters. The outer boundary of the body is a free traction surface. On the inside, a constant pressure p is prescribed.



R	150 mm
a	200 mm
b	100 mm
E	1500 N/mm^2
ν	0.4

- Use the symmetries of the example.
- Create an example with name `ringPressure` for a nonlinear analysis using the Neo-Hookean material. You find the .geo-file in the TeachCenter.

- Discretize the body with a structured mesh with 10x10 elements on the quarter system. Choose appropriate parameters for the Newton-Raphson solver in the nonlinear analysis. Run the analysis for $p = 10\text{MPA}$, $p = 50\text{MPA}$, $p = 100\text{MPA}$ and $p = 200\text{MPA}$.
- Create a table containing the vertical displacement of the upper minor vertex of the ellipse for the different simulations. Give an interpretation of the results.

Exercise 3

The goal of this exercise is to calculate and visualize the von Mises stress of the Cauchy stress tensor $\boldsymbol{\sigma} = \frac{1}{J}\boldsymbol{FSF}^{\mathrm{T}}$ with $J = \det \boldsymbol{F}$ in a post-processing step. The von Mises stress is defined as

$$\sigma_{\mathrm{vM}} = \sqrt{\frac{1}{2}\left((\sigma_{11}-\sigma_{22})^2 + (\sigma_{22}-\sigma_{33})^2 + (\sigma_{33}-\sigma_{11})^2\right) + 3\left(\sigma_{12}^2 + \sigma_{23}^2 + \sigma_{31}^2\right)}$$

in 3D. Note that in our 2D examples, we assume a plain strain state, and therefore expect a 3D stress state. The von Mises stress (amongst other stress measures) can be used to reduce the stress tensor to a single number which can be compared to a failure criterion (e.g. the yield stress of the material).

The calculation of the von Mises stress for each element is similar to the caluclation of the stiffness matrix or the force vector. We need the following ingredients:

- Add the property `sigma_vM` to the class `nsModel.Element` and initialize it to zero in the constructor. Add an according set-method `setSigmaVM`.
- Add the following function to the protected methods of your class `nsAnalyzer.nsAnalysis.NonlinearAnalysis.m`:

```
function calcStresses(self)
    for element = self.model.element_dict
        int_sigma_vM = element.type.implementation.calcSigmaVM(element);
        volume = element.type.implementation.calcVolume(element);

        element.setSigmaVM(int_sigma_vM/volume);
    end
end
```

  It calculates the volume-averaged von Mises stress

  $$\bar{\sigma}_{\mathrm{vM}} = \frac{\int_{\Omega^{\mathrm{el}}} \sigma_{\mathrm{vM}} \mathrm{d}V}{\int_{\Omega^{\mathrm{el}}} 1 \mathrm{d}V} = \frac{\int_{\Omega^{\mathrm{el}}} \sigma_{\mathrm{vM}} \mathrm{d}V}{V}$$

  for each element. Call the method `calcStresses` once per time step, before writing the VTK file.
- Implement the methods `calcSigmaVM` and `calcVolume` to the class `nsAnalyzer.nsImplementation.NonlinearElementImpl.m`. You can use the methods `calcStiffness` and `calcLoad` as reference. Make sure that in the 2D case (plane strain), you replace the deformation gradient $\boldsymbol{F}$ with

  $$\boldsymbol{F} = \begin{pmatrix} F_{11} & F_{12} & 0 \\ F_{21} & F_{22} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

  before calculating strain and stress measures.
- Replace your class `nsIO.VTKOutputHandler` with the one from the TeachCenter for visualizing the von Mises stress in ParaView.

Testing

Test your implementation using one of the examples of Exercise 1 or Exercise 2. Interpret your results in a short sentence.