

CHALMERS



Finite Element Method - Structures, TME245

Computer assignment 1: large deformation analysis with six noded triangular elements

Instructions

Magnus Ekh, Walther Ahl and Robin Larsson

Department of Mechanical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
TME245 FEM - Structures 2025/2026

Preface

The learning outcomes of this project is to learn to implement a hyperelastic constitutive model as well as a FEM solver for large deformation problems.

All derivations need to be well structured and easy to read (written by hand or using a word-processor). Also, submit your Python (or Matlab) code as part of the hand-in (Abaqus files are not to be submitted).

The project work is to be carried out in groups with preferably, and **at most, two students in each group**. The students should contribute equally to solving the tasks. **Please write names, group number and course code clearly on the front page of the report and submit it with attachments on CANVAS.**

Since the projects prepare you for the examination, it is essential that each group solves the tasks separately and practice writing code as well as doing any necessary derivations. To clarify,

It is OK to:

- Collaborate and discuss around derivations asked for in different subtasks. **HOWEVER**, each group should in the end do the derivations themselves and hand-in their own written solutions.
- Discuss around implementation strategies in Python for different subtasks. **HOWEVER**, limit the discussions to pen and paper otherwise you will know that you are on the border of what is not OK.

It is NOT OK to:

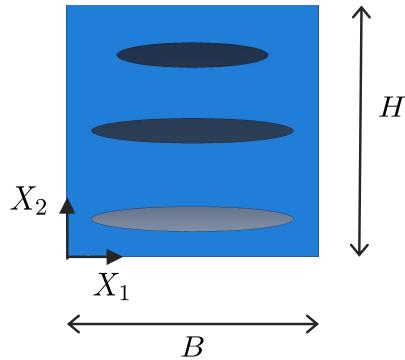
- Directly copy other groups' derivations and submit them as your own.
- Share any amount of Python-code or files from an FE software (e.g. Abaqus) between groups.
- Copy and submit code that is partly written by another group. This is true for code written by other students this year as well as from previous years.

Any suspected cheating (including things like copying Python or MATLAB code or parts of someone else report) will be directly reported to the Disciplinary Committee.

Göteborg in January 2025
Magnus and Robin

Problem definition

The mechanical behaviour of a rubber block with three elliptical defects (voids) shall be analyzed. Both deformations and stresses shall be analyzed.



The numerical values of the geometry parameters are given as: $H = B = 100$ mm and thickness (out of the plane) $h = 100$ mm.

In Task 1 you will do a 2D large deformation analysis of the rubber profile using Abaqus. In Task 2, you will analyse the same block using your own code (Python or Matlab).

Complete the tasks and comment on the results in the report. Write a report following the instructions on `Report_instructions.pdf` on Canvas. The points for each task are indicated in the text.

Task 1: Nonlinear elastic analysis in 2D using Abaqus

Perform FE-analysis the 2D section provided in the Abaqus CAE file `ca2026_formulation.cae`. Please note that the file only contains the definition of the section geometry, with all dimensions in mm.
Please complete the following subtasks:

- (a) Define appropriate boundary conditions such that the following load case can be approximated:

LC1 Uniaxial in-plane tensile loading in the X_2 -direction (the assumption of plane strain will give a stress also out of the plane) by defining a displacement u_Γ on the upper boundary.

Also make your own sketch of the rubber cross section and clearly define and motivate your choice of boundary conditions.

LC2 A second loadcase LC2 is defined by changing to compressive loading in the X_2 -direction (i.e. u_Γ negative).

Please note that to ease the analysis (primarily in Python in Task 2) the loading should be applied as an increasing displacement u_Γ rather than an increasing traction. Prescribed displacements will most likely lead to easier convergence in the Newton iterations. Identify parts of the boundary surface where you constrain the cross-section from moving (in a suitable way) such that a nearly uniaxial in-plane loading case is achieved. You also need to make sure that your boundary conditions take away the risk of rigid body motion.

- (b) Based on the geometrical model in `ca2026_formulation.cae`, define the complete finite element model in terms of the following steps:

- **Define the rubber hyperelastic behaviour** (see hint at the end of this task). The material model for the rubber is defined by the nearly incompressible Yeoh strain energy:

$$U_0(\mathbf{C}) = c_{10} \left(J^{-2/3} \text{tr}(\mathbf{C}) - 3 \right) + c_{20} \left(J^{-2/3} \text{tr}(\mathbf{C}) - 3 \right)^2 + c_{30} \left(J^{-2/3} \text{tr}(\mathbf{C}) - 3 \right)^3 + \frac{1}{D_1} (J - 1)^2 + \frac{1}{D_2} (J - 1)^4 + \frac{1}{D_3} (J - 1)^6$$

(see hint at the end of this task where also parameter values are given).

- **Create a section and assign that to the part**
- **Create an instance of the part in the assembly module**
- **Create a static load case for each LC** (automatic time stepping over a quasi-time period from 0 to 1 should suffice).
- **Define boundary conditions** according to what you specified in subtask (a). For LC1 assume final $u_\Gamma = +20$ mm while for LC2 assume final displacement $u_\Gamma = -15$ mm.
- **Discretise the domain in a suitable way** using the element type of your choice. Here it is essential that you **choose plane strain approximation**. Make sure that your solution has converged (in (c)) for the tensile loading with respect to the spatial discretisation (number of elements), both for maximum reaction force and maximum von Mises stress. Please provide results in the report that show that your solution has converged.

Please make sure to briefly explain how you did each of these steps in the report.

- (c) **Run the model up to the prescribed final displacements (in tension and compression).**

Then, **for each load case, produce the following results** (to be included in the report):

- A contour plot of the von Mises stress distribution in the section.
- A contour plot that illustrates the normal strain in X_2 -direction (variable LE22).
- A graph illustrating the force-displacement relation at the boundary where you have the varying prescribed displacement. The force should be computed as the sum of all nodal reaction forces in the relevant direction along that part of the boundary (see pdf on Canvas for instructions).

2p

Some Abaqus hints:

1. Check links to useful video clips and instruction pdfs for Abaqus available on Canvas.
2. To obtain the nearly incompressible Yeoh model with parameter values ($c_{10} - c_{30}$ in MPa and D_1-D_3 in 1/MPa) in Abaqus see the figure below.

Hyperelastic

Material type: Isotropic Anisotropic

Strain energy potential:

Input source: Test data Coefficients

Moduli time scale (for viscoelasticity):

Strain energy potential order:

Use temperature-dependent data

	C10	C20	C30	D1	D2	D3
1	3.45	-0.69	0.23	0.02	0.01	0.01

3. **Voluntary:** Try compressing the profile further by using self-contact, see help on Canvas.

Task 2: Nonlinear elastic analysis in 2D using Matlab/Python

- (a) Combine your elasticity code from Computer Exercise 1 (CE1) and the nonlinear bar structure algorithm in the lecture notes to solve the FE problem defined in Task 1 as a linear isotropic elastic problem with linear kinematics (in this sub-task). To be specific, extend the code from CE1 to include time incrementation and iteration for equilibrium.

Geometry and element topology for two meshes can be found on Canvas (`topology_coarse_3node.mat`, and `topology_fine_3node.mat`). The dof's of the upper, lower, left and right boundaries as well as in the lower left corner are given in the files. Function for reading .mat files in Python see `read_matfiles.py` on Canvas.

By using the given geometry matrices define the matrices/vectors needed for defining what dof's that the boundary conditions are applied on.

Apply the vertical displacements histories according to LC1 and LC2 defined in Task 1 (although they give large deformations).

For this task, assume plane strain and the following isotropic linear elastic properties: $E = 20$ MPa and $\nu = 0.45$. For this task adopt CST triangles.

Present the vertical reaction force (obtained by summation of nodal reaction forces) on the upper boundary vs u_Γ .

Check that the Newton iteration converges after maximum one iteration (since it is a linear problem). **1p**

- (b) In Python (or Matlab), implement the nearly incompressible Yeoh hyperelastic material model defined by the following strain energy (see lecture notes):

$$U_0(\mathbf{C}) = c_{10} \left(J^{-2/3} \text{tr}(\mathbf{C}) - 3 \right) + c_{20} \left(J^{-2/3} \text{tr}(\mathbf{C}) - 3 \right)^2 + c_{30} \left(J^{-2/3} \text{tr}(\mathbf{C}) - 3 \right)^3 + \frac{1}{D_1} (J - 1)^2 + \frac{1}{D_2} (J - 1)^4 + \frac{1}{D_3} (J - 1)^6$$

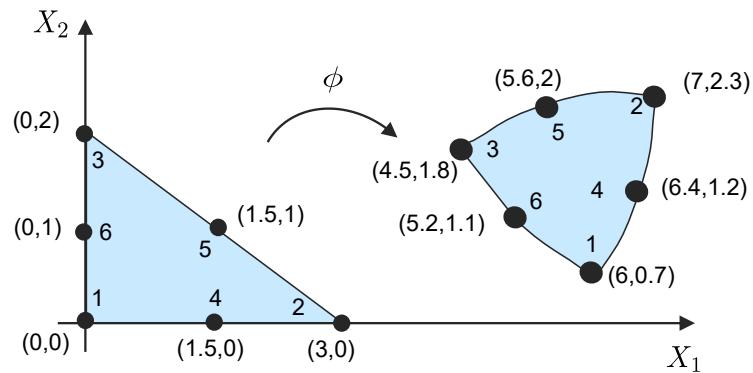
Plot the Cauchy stress component σ_{11} against $0.5 \leq F_{11} \leq 1.5$ for a pure elongation/contraction condition with the material parameters suitable for natural rubber $E = 20$ MPa, $\nu = 0.45$, $G = E/(2(1+\nu))$, $\lambda = E\nu/((1+\nu)(1-2\nu))$, $c_{10} = G/2$, $c_{20} = -G/10$, $c_{30} = G/30$, $D_1 = 0.02$ [1/MPa], $D_2 = 0.01$ [1/MPa] and $D_3 = 0.01$ [1/MPa]. In the same graph show the corresponding result for the neo-Hooke model by using the code in Section 5.2.10 (the values for μ and λ according to above).

At $F_{11} = 1.5$, the neo-Hooke model should give the normal Cauchy stress $\sigma_{11} = 2.2525\text{e}01$ MPa whereas the Yeoh model should give $\sigma_{11} = 1.2142\text{e}02$ MPa.

The implementation shall be such that also \underline{P} and $d\underline{P}/d\underline{F}$ are outputs (for a plane strain situation, see lecture notes) to prepare for the FE implementation. In addition the Cauchy stress needs to be computed for the stress analysis.

1p

- (c) **Generate your own element function** for computation of deformation gradient, 1st Piola-Kirchhoff stress, internal element force and internal element stiffness for a 6-node triangle see lecture notes. For the geometry and deformation:



the results for the nearly incompressible Yeoh model should be (for \mathbf{F} and \mathbf{P} each column represents a Gauss point):

```
>> deformation_gradient_2d=
2.4444e-01  2.4444e-01  3.7778e-01
4.8333e-01  7.8333e-01  8.8333e-01
-8.1667e-01 -7.1667e-01 -8.1667e-01
3.5556e-01  6.2222e-01  7.5556e-01
>> Piola_Kirchoff_2d=
-9.5450e+01 -5.2735e+01 -8.2819e+00
-4.4629e+01 -1.0827e+01  1.1267e+00
6.3764e+01  3.7729e+01  3.8915e+00
-1.6186e+02 -4.5074e+01 -5.0288e+00
>> fe_int=
4.8202e+00
1.1993e+04
1.1864e+03
2.0199e+03
2.0165e+03
-1.7721e+02
-9.7132e+03
-8.7571e+03
-3.0907e+03
-9.4135e+03
9.5962e+03
4.3350e+03
```

```
>>Ke_int=
Columns 1 through 8

1.0125e+03 -1.3198e+03 -1.1052e+02 1.5955e+03 5.3857e+02 -2.6281e+03 2.9076e+02 -1.3472e+04
-1.3198e+03 4.1730e+04 1.9009e+03 6.0745e+03 -2.7925e+03 3.2999e+03 -1.0646e+04 -3.0693e+04
-1.1052e+02 1.9009e+03 3.3792e+03 3.1045e+03 1.8442e+03 -3.0962e+02 -9.0041e+02 -7.7605e+03
1.5955e+03 6.0745e+03 3.1045e+03 3.3992e+03 1.0548e+03 -4.5103e+02 -2.1860e+03 -9.0289e+03
5.3857e+02 -2.7925e+03 1.8442e+03 1.0548e+03 9.6326e+03 -3.4967e+03 -5.4540e+02 3.2515e+02
-2.6281e+03 3.2999e+03 -3.0962e+02 -4.5103e+02 -3.4967e+03 1.8858e+03 1.5252e+03 -1.5779e+03
2.9076e+02 -1.0646e+04 -9.0041e+02 -2.1860e+03 -5.4540e+02 1.5252e+03 2.3028e+04 1.4283e+04
-1.3472e+04 -3.0693e+04 -7.7605e+03 -9.0289e+03 3.2515e+02 -1.5779e+03 1.4283e+04 3.3449e+04
-1.4853e+02 4.8954e+02 -5.6882e+03 -5.2046e+03 -9.3771e+03 -6.0942e+02 -1.7119e+03 1.4378e+04
6.3055e+02 -1.2354e+04 3.6991e+02 -8.7532e+02 -5.1969e+03 3.0951e+03 1.0802e+04 1.3071e+03
-1.5828e+03 1.2368e+04 1.4758e+03 1.6359e+03 -2.0928e+03 5.5186e+03 -2.0161e+04 -7.7534e+03
1.5194e+04 -8.0577e+03 2.6949e+03 8.8151e+02 1.0106e+04 -6.2519e+03 -1.3778e+04 6.5446e+03

Columns 9 through 12

-1.4853e+02 6.3055e+02 -1.5828e+03 1.5194e+04
4.8954e+02 -1.2354e+04 1.2368e+04 -8.0577e+03
-5.6882e+03 3.6991e+02 1.4758e+03 2.6949e+03
-5.2046e+03 -8.7532e+02 1.6359e+03 8.8151e+02
-9.3771e+03 -5.1969e+03 -2.0928e+03 1.0106e+04
-6.0942e+02 3.0951e+03 5.5186e+03 -6.2519e+03
-1.7119e+03 1.0802e+04 -2.0161e+04 -1.3778e+04
1.4378e+04 1.3071e+03 -7.7534e+03 6.5446e+03
1.6749e+04 5.7064e+03 1.7678e+02 -1.4760e+04
5.7064e+03 2.9340e+04 -1.2312e+04 -2.0513e+04
1.7678e+02 -1.2312e+04 2.2184e+04 5.4261e+02
-1.4760e+04 -2.0513e+04 5.4261e+02 2.7396e+04
```

1p

- (d) Now combine the routines developed in (b)-(c) in the same code structure as in (a). Check that your code works with the coarsest FE mesh (using `topology_coarse_6node.mat`) first.

Apply u_Γ by increasing it linearly from 0 to 20 mm. Try different number of time steps (for the coarsest mesh).

Plot the total vertical on the upper boundary of the rubber profile vs u_Γ .

Check mesh convergence by comparing the results from `topology_medium_6node.mat` and, if the computational time is not too long, `topology_fine_6node.mat`

Plot the field of von Mises equivalent stress (of the Cauchy stress) in the rubber profile. Compare the values from the different mesh densities For simplicity, plot the mean value of the stress in each element.

For the coarsest mesh, try also to decrease u_Γ from 0 to -15 mm. Discuss how the large deformations affect the different results in compression and tension as well as by comparing with the results from a). Are more timesteps needed for the compression loading

than for the tension loading?

2p