Determination of the yield behavior of a 2-D polycarbonate honeycomb structure

G.A. van Erp

Bachelor Final Project, MT09.06
## Contents

1 Introduction ................................................................. 1
2 Defining the model ......................................................... 2
    2.1 2D Mesh ................................................................. 2
    2.2 Periodic boundary conditions ...................................... 3
    2.3 Loads ........................................................................ 4
    2.4 Input for simulations (Marc Mentat) ......................... 5
        2.4.1 Element type and technology ............................. 5
        2.4.2 Material model ................................................ 6
3 Data processing ............................................................. 7
    3.1 Homogenisation ....................................................... 7
    3.2 Yield criterium .......................................................... 8
4 Compression in main directions .......................................... 9
    4.1 Vertical compression ................................................ 9
    4.2 Horizontal compression ............................................ 11
        4.2.1 Basic model .................................................... 11
        4.2.2 Using constant dilatation ................................. 13
        4.2.3 Directional sensitivity ...................................... 14
        4.2.4 Checking the parameters of the Neo-Hookean model 15
        4.2.5 Using EGP for elastic and plastic simulations ......... 17
5 Imperfection sensitivity ................................................... 18
    5.1 High horizontal stiffness .......................................... 18
    5.2 Choice of imperfection .............................................. 19
6 Loading the imperfect model ............................................. 22
    6.1 Horizontal loading .................................................. 22
    6.2 Checking vertical loading ......................................... 23
7 Full description of the deformation tensor ......................... 24
    7.1 Full deformation tensor using a constant volume .......... 24
    7.1.1 Determine the components of the deformation tensor 24
    7.2 Rotated deformation tensor ...................................... 27
        7.2.1 Rotating the deformation tensor ....................... 27
        7.2.2 Applying the rotated deformation tensor to a one element model 29
        7.2.3 Applying the rotated deformation tensor to the RVE 31
8 Loading in different angles .............................................. 34
9 Adapting the Matlab script ............................................. 38
10 Conclusions and recommendations ................................. 39
References ............................................................................. 40
Appendix A ................................................................. i
    Matlab script ............................................................. i
1 Introduction

Structural polymers, e.g. foams and honeycombs, are used in many engineering applications because of their high strength to weight ratio and their shock and energy absorbing properties. Although a lot of research has been done on structural polymers, not everything is properly described yet. In the future it is desirable to have a constitutive material model for these foams to predict the properties of the products made of these foams. In order to create such a model intrinsic material properties have to be combined with the underlying structure. A first approach to capture a model is to determine the properties of a 2D honeycomb. In this case, a hexagonally packed circular structure combined with the Eindhoven Glassy Polymer model (EGP) is examined. The EGP model contains the intrinsic material behavior of polycarbonate.

An existing finite element model of the 2D structure [Wismans] is modified to examine the yield behavior of this structure. This structure uses periodic boundary conditions in order to preserve periodicity during deformation. To find the macroscopic yield point of this material an elastic model is used besides the EGP. By comparing the response of the elastic model with the results of the EGP, a yield point can be found.

Chapter two describes the used model, including loads and boundary conditions. In chapter three it is described how the simulation data are processed. The first simulations, done in the main directions (horizontal and vertical), are described in chapter four. This leads to some numerical instabilities in horizontal direction. To solve these problems, in chapter five an imperfection in the ideal geometry is suggested and tested. The imperfection accounted for in the loadcases in chapter six. In chapter seven a full description of the deformation tensor is suggested to have full control of the deformation of the structure. This deformation tensor is then rotated to describe loading in other directions. The simulations in other directions are performed in chapter eight and chapter nine describes the adaptations in the Matlab model. Chapter 10 gives a short conclusion and some recommendations for future investigation.
2 Defining the model

2.1 2D Mesh

A 2D model of a hexagonal structure is required to perform simulations on the polymer structure. This model is already available [Wismans]. It is a representative volume element (RVE) of a hexagonally packed circular honeycomb structure. The geometry of the model is based on a CT-scan of a polymer honeycomb. An ideal model is made, based on this scan, to be able to apply periodic boundary conditions. Therefore some values are extracted from the scanned structure: $D_r$, $t_r$, $w_{tr}$, $w_{wr}$ (Fig. 2.1)., where $D_r$ is the outer diameter of each cell, $t_r$ is the wall thickness of a cell and $w_{tr}$ and $w_{wr}$ represent the welding properties of the cells. With these values and giving the ideal RVE the same volume as the scanned honeycomb, a perfect honeycomb structure is created. This perfect honeycomb will be modified for further simulations.

![Figure 2.1: Ideal RVE with some dimensions.](image)
2.2 Periodic boundary conditions

Periodic boundary conditions are used in this model. These boundary conditions preserve periodicity during deformation. To accomplish this, the following periodic boundary conditions are prescribed:

\[
\begin{align*}
\bar{x}_{L} &= \bar{x}_{L} + \bar{x}_2 - \bar{x}_1, \\
\bar{x}_{R} &= \bar{x}_{R} + \bar{x}_4 - \bar{x}_1,
\end{align*}
\]  

(2.1) (2.2)

where \( \Gamma_i \), with \( i = L, R, T, B \) are the boundaries and \( \bar{x}_i \), with \( i = 1, 2, 3, 4 \) are the corner nodes as defined in Fig. 2.2. The first equation links the right boundary to the left boundary. The distance between the left and right boundary during deformation is defined by the distance between node 2 (bottom right) and node 1 (bottom left). The second equation links the top boundary to the bottom boundary. The distance between the top and bottom boundary is defined by the distance between node 4 (top left) and node 1 (bottom left).

![Figure 2.2: RVE with replaced edge and corner and boundary names.](image)

To apply periodic boundary conditions to the model and considering contact when loading in vertical direction, a small edge (one element size thick) is taken from the bottom of the RVE and sticked to the top of the RVE [Wismans]. Since the structure will be loaded in different directions, from horizontal to vertical and some angles in between, this periodicity is also necessary in horizontal direction. This again is done by moving a small edge of the structure, this time from the left side to the right (Fig. 2.2). Now the boundaries of the model correspond to the material boundaries.

The applied boundary conditions for the vertical compression mode are a fixation in y-direction for node 2, a fixation in the x-direction for node 4 and a fixation in x and y direction for node 1. For horizontal compression the same boundary conditions are applied. Deformation in vertical direction is prescribed to node 4, when loading in horizontal direction this is done to node 2.
2.3 Loads

For loading the geometry, the displacement is prescribed. The displacement is subjected such that a true strain criterium is fulfilled, where

\[ \varepsilon_{ln} = \ln(\lambda) = \dot{\varepsilon} \cdot t, \quad (2.3) \]

which, after rewriting, leads to

\[ \Delta l = l - l_0 = l_0 (e^{\dot{\varepsilon} \cdot t} - 1). \quad (2.4) \]

Here \( \varepsilon_{ln} \) is the logarithmic strain, \( \lambda \) the stretch, \( \dot{\varepsilon} \) the strain rate, \( t \) the time and \( l_0 \) the initial length of the RVE in the loading direction. The time versus displacement graph that follows from this formula can be seen in Fig. 2.3, where, \( \dot{\varepsilon} = -1 \) [s] and \( l_0 = 1 \) [mm].

![Figure 2.3: Graph of time versus displacement.](image)

Since \( l_0 \) is chosen to be one, it can be multiplied by the lengths of the RVE in the corresponding x- and y-direction and in this way be used as input for the simulations.
2.4 Simulation input

2.4.1 Element type and technology

Shell elements are better suited to describe bending behavior than quad elements. Nevertheless quad elements are chosen to solve the model. This is because the model of the structure is based on a CT-scan. From this scan the pixels are directly converted to elements. This gives quad elements, shell elements need more attention to make this conversion possible and are not used in this case. Afterwards an ideal model is made with help of the CT-scan based model. This is also made with quad elements to have a good comparison of the ideal model with the real model.

The assumed strain formulation is used in Marc Mentat to better describe the bending of the elements. In Marc Mentat the standard linear interpolation function is replaced. The drawback however are the increased computational costs.

Constant dilatational elements are often used when the material becomes approximately incompressible, this is the case in large strain plasticity. Conventional elements can produce volumetric locking due to overconstraints for nearly incompressible behavior. Since large plasticity occurs in these simulations this option is used. [Marc Mentat]

As the structure is compressed, at a given moment the cell walls will come in contact with each other. To simulate this contact, it has to be accounted for within Marc Mentat. Assuming that no friction will occur between the surfaces, a friction coefficient of zero is used. The contact table is set to “touching” so if cells collide they will not penetrate each other and can freely slide along each other without friction.
2.4.2 Material model

The simulations of the elastic model are done with a Neo-Hookean model. In comparison with elastic simulations with the Eindhoven Glassy Polymer (EGP) model, this can reduce the simulation time significantly. Nevertheless, it gives nearly the same results. The elastic model is based on the initial stiffness of the EGP model. In the EGP model a spring damper system is used to represent the viscoplastic behavior of polycarbonate(fig 2.4). The following parameters are used in this model: \( G_R = 26 \text{ [MPa]} \), \( G = 308 \text{ [MPa]} \). [Klompen]

![Figure 2.4: Spring-damper system representing the material behavior.](image)

With the multiplicative decomposition option on, Marc Mentat uses a decomposed version of the deformation tensor. It calculates a separate elastic and plastic deformation tensor. These multiplicative decomposed tensors are necessary as input for the EGP model. [Klompen]. Furthermore the updated lagrange option is used because the model is nonlinear in both material behavior and deformation.

Hypela2 is a user subroutine that makes it possible to implement arbitrary material models in conjunction with the hypoelastic model definition option. The used subroutine is the EGP model [EGP]. This model calculates the response of polycarbonate to the applied displacements. The parameters that are used are shown in table 2.1.

| Material parameters used in the Eindhoven Glassy Polymer model. |
|-----------------------|-----------------|-----------------|-----------------|-----------------|
| \( K \) [MPa] | \( G \) [MPa] | \( \eta_{0,r} \) [MPa \cdot s] | \( \tau_0 \) [MPa] | \( \mu \) [-] | \( S_a \) [\text{-}] |
| 3750 | 308 | 2.1\cdot10^{11} | 0.7 | 0.08 | 31.7 |
| \( r_0 \) [-] | \( r_1 \) [-] | \( r_2 \) [-] | \( G_r \) [MPa] | \( \Delta U_a \) [KJ/mol/K] | \( \rho_a \) [\text{m}^3/\text{mol}] |
| 0.965 | 50 | -5 | 26 | 205 | 1.33 \cdot 10^{-3} |
3 Data processing

3.1 Homogenisation

A homogenization scheme is used to determine the macroscopic behavior of the RVE. The scheme is based on averaging the microscopic first Piola Kirchhoff stress tensor $P_m$ [Kouznetsova].

$$P_M = \frac{1}{V_0} \int_{V_0} P_m dV_0, \quad (3.1)$$

where subscripts $M$ and $m$ represent macroscopic and microscopic quantities, respectively, and $V_0$ is the volume of the RVE in the reference state. By incorporating the periodic boundary conditions given in Equations (2.1) and (2.2), the relation is simplified to

$$P_M = \frac{1}{V_0} \sum_{i=1,2,4} \tilde{f}_i \tilde{x}_0, \quad (3.2)$$

where $\tilde{f}_i$ are the reaction forces of the corner nodes in the current state and $\tilde{x}_0$ the position vectors of these nodes in the reference state (Fig 2.2). Finally, the macroscopic first Piola-Kirchhoff stress tensor $P_M$ is related to the macroscopic Cauchy stress tensor $\sigma_M$ by

$$\sigma_M = \frac{1}{\det(F_M)} P_M \cdot F_M^\circ, \quad (3.3)$$

The macroscopic deformation gradient tensor $F_M$ is determined by volume averaging of the microscopic deformation gradient tensor $F_m$, according to

$$F_M = \frac{1}{V_0} \int_{V_0} F_m dV_0 = \frac{1}{V_0} \int_{\Gamma_0} \tilde{x}\tilde{n}_0 d\Gamma_0, \quad (3.4)$$

with $\tilde{n}_0$ the outward normal of the boundary $\Gamma_0$ in the initial state.
3.2 Yield criterium

To determine the yield behavior of the material, acquired from multiple simulations, a yield criterium is needed. First a simulation is done with an elastoplastic material model, the EGP model. The stress-strain curve that results from this simulation can be plotted. By doing the same simulation with an elastic material model, another stress-strain curve can be created, this time purely elastic. Now the two stress-strain curves can be compared. The elastic area should be the same for both plots. At the point where the two graphs split up a macroscopic yield point has been passed. A point can be extracted from the plots and a yield point in one loading direction can be determined.
4 Compression in main directions

4.1 Vertical compression

The first simulation is a vertical compression. The RVE is loaded as discussed in chapter 2.3 with a displacement constraint. Simulations are done with both the Neo-Hookean and the EGP model. Some figures of the structure at different times, done with the Neo-Hookean model, can be seen in fig 4.1.

Figure 4.1a : Undeformed structure (t = 0s).

Figure 4.1b : Deformed structure (t = 0.25s).

Figure 4.1c : Deformed structure (t = 0.5s).

Figure 4.1d : Deformed structure (t = 1s).
The stress-strain curves that result from this experiment are plotted on top of each other. From this graphs a clear yielding point can be determined. However, there is a small part in the elastic region of the graphs where the Elastic (Neo-Hookean) and plastic (EGP) models do not match. This is due to numerical errors. To check if this presumption is true the simulation is executed again now with 3000 time steps in stead of 2000. This indeed decreases the error so an even further refinement is done with 6000 time steps in the simulation. The result is shown in fig 4.2, where $\sigma_{22}$ is the stress in vertical direction.

![Stress-strain curve of vertical compression.](image)

Figure 4.2: Stress-strain curve of vertical compression.

Now the two graphs are almost on top of each other in the elastic regime, so this is a satisfying result. Further refinement will probably even out the last differences in the elastic area but this is not necessary for investigating the yield point of the structure.
4.2 Horizontal compression

4.2.1 Basic model

The next step is finding the yield point when loading in horizontal direction. Compressing the structure in horizontal direction gives another deformation mode where square shaped structures form. The deformation mode, performed with the Neo-Hookean model, can be seen in Fig. 4.3 at two different times.

At $t = 0.33s$ the simulation does not converge anymore. Apparently at this point the time step must be very small to be able to calculate further deformation. When increasing the time steps the same error occurs and the simulation comes to an end again. If the structure is already in the plastic region at this time it is not a problem that the simulation does not converge anymore. If the yield point would already be passed then, there is no need to find the cause of the problem and adapt the model for further simulations. Processing the data from the simulation gives a smooth elastic stress-strain curve and a plastic stress-strain curve that is jagged from a certain strain level.
It is not possible to extract a yield point from these curves. A simulation with even more time steps gives a smoother plastic curve (Fig. 4.4), where $\sigma_{11}$ is the stress in horizontal direction, but still does not show a clear yield point. It even shows that the plastic stress-strain curve lies slightly above the elastic one. This can never be true, since the elastic region has a higher stiffness than the plastic region. Since it is not possible to get a yield point from these simulations, a solution has to be found to make a correct model where the plastic curve is under the elastic one and progresses far enough to be able to point out a clear yielding point.

Figure 4.4: Stress-strain curve of horizontal compression (3000 increments).
4.2.2 Using constant dilatation

A possibility of why the plastic model is stiffer than the elastic model is that volumetric locking occurs. The option constant dilatation (chapter 2.4) is used (in both Neo-Hookean and EGP model) to prevent volumetric locking. This time the simulation runs until 0.44s but then ends with the same problem as before. The time steps can not be made small enough to be able to calculate the deformation from that point. The stress-strain curves from these simulations are shown in Fig. 4.5.

![Stress-strain curve of horizontal compression using constant dilatation.](image)

Figure 4.5: Stress-strain curve of horizontal compression using constant dilatation.

The curve resulting from the EGP model is even further above the curve resulting from the Neo-Hookean model. The used models do not even seem to have a similar elastic response. From time $t = 0s$ they slowly drift away from each other. So there is another problem in the model that causes these results. It should be solved and therefore a couple of tests are carried out that can point out the error.
4.2.3 Directional sensitivity

The vertical loading works like expected but when the structure is compressed in horizontal direction the Neo-Hookean and EGP model do not seem to agree in the elastic area. This is an unexpected result so before trying anything else a directional test is performed. Perhaps there is a directional dependency in Marc Mentat that causes these effects in horizontal direction.

To test if there is a directional error in the model the structure is rotated by 90 degrees, like shown in fig 4.6.

![Figure 4.6: RVE rotated 90 degrees counterclockwise.](image)

The boundary conditions are rotated together with the model, so it is exactly the same as before. The only difference is that it is rotated 90 degrees counterclockwise in comparison to the original model. Since there is no difference except for the orientation, the results from loading should be the same as before.

![Figure 4.7: Stress-strain curve of horizontal compression (original RVE and rotated RVE).](image)

This test shows the same effects as before, when loading in horizontal direction fig 4.7, So the problem is still present and there is no directional error that causes these results.
4.2.4 Checking the parameters of the Neo-Hookean model

A second test is carried out to test if the elastic (Neo-Hookean) model agrees with the initial stiffness of the EGP model. If the $S_a$ value of the EGP is raised (Fig. 4.8), the elastic region is extended. If this value is raised far enough the plastic region will not be reached during the simulation.

![Figure 4.8: Effect on the Stress-strain curve by raising $S_a$.](image)
The test is carried out with $S_a = 2700 [-]$ and $S_a = 27000 [-]$. Originally $S_a$ equals 27 [-]. In both cases the elastic region is indeed stretched far enough to make the simulation fully elastic. The result of the simulation with $S_a = 2700 [-]$ is shown in figure 4.9.

![Figure 4.9: Stress-strain curve of horizontal loading comparison of Neo-Hookean and EGP($S_a = 2700$).](image)

Thus the parameters used for the Neo-Hookean model agree with the EGP model and are therefore chosen correctly.
4.2.5 Using EGP for elastic and plastic simulations

None of the tested cases seemed to make a difference in the simulations. However a problem does occur in horizontal loading. The stress-strain curve of the elastic model should always be above the plastic stress-strain curve, since the slope of the stress-strain curve always gets smaller when entering the plastic area. To exclude all model differences, from now on only the EGP material model is used. The model has to be adapted to do an elastic simulation with it. This can be done by stretching the elastic area to higher strain levels as explained in chapter 4.2.4.

By performing simulations with this model the mechanical response of the structure with an elastic material model is calculated. This replaces the simulations previously done with the Neo-Hookean model.

Another adaptation is done to the settings in Marc Mentat to make the simulations go more smoothly. The time step is not set to a ‘constant time step’ anymore but to ‘multi-criteria’. A minimum and maximum time step is chosen. In this way Marc Mentat automatically chooses the time step as large as possible but smaller when necessary.
5 Imperfection sensitivity

5.1 High horizontal stiffness

The problems with the model concerning the material parameters are solved by using the EGP model for both plastic and elastic simulations. This however, still does not fix all problems with the model. First the simulations came to an end prematurely since the time steps were not small enough to continue the calculations. After the time step was set to multi-criteria and the minimum time step was chosen small enough the simulation could continue past this point. However, the structure behaves very unstable, which is not very realistic. The reason for this instability is most probably the high stiffness of the structure in horizontal direction. Where two rings of the structure connect they almost form a horizontal beam (fig 5.1). This is thus compressed in axial direction. The beam is very stiff in axial direction and the simulation does not converge. It is only when the numerical error becomes large enough that the structure has a deformation preference (go up or down) and buckles.

![Figure 5.1: The horizontal ‘beam’ that gives high stiffness in horizontal compression.](image)
5.2 Choice of imperfection

Since the perfect geometry is not ideal for simulations in horizontal direction an imperfection is introduced into the structure. This imperfection can prevent the instable behavior of the structure by giving the structure a directional preference. In this way the stress will not increase until a numerical error will cause further deformation. The stiff part of the structure pointed out in fig 5.1 is weakened on one side. With the imperfection this part is not a symmetrical stiff beam anymore and the structure can deform without a critical point. The new RVE should therefore represent a realistic situation better than the perfect RVE. The imperfection can be placed on two parts of the structure. Now a choice has to be made to what imperfection will be introduced into the structure, therefore a test is done with different types of imperfections: A dent with a depth of 0.01 mm, a dent with a depth of 0.02 mm (Fig. 5.2), two dents with a depth of 0.01 mm, two dents with a depth of 0.02 mm. A dent of 0.02 mm is approximately one fourth of the total wall thickness of the structure.

![Figure 5.2: Imperfection (dent 0.02mm), the red line shows the original material boundary.](image)

The places where the imperfections are introduced are shown in Fig. 5.3. In case of only one imperfection, the upper red square indicates its location.

![Figure 5.3: Location of the introduced imperfections.](image)
The imperfect structures are now loaded as before in horizontal direction. A Neo-Hookean model is used since this takes less calculation time than the EGP model. The stress-strain curves of these tests are placed in the same figure in order to inspect the differences. The results are shown in Fig. 5.3.

Figure 5.4: Comparison on the effects of different imperfections.
The figure shows that a small imperfection of one eighth of the total wall thickness of the structure does not give the desirable result. It still gives a drop in stress. This happens after building up stress when the “beam” is loaded in axial direction. When the structure prefers a bending direction it buckles and shows this drop in stress which is undesirable. When the imperfection is large enough, one fourth of the wall thickness, this buildup of stress is avoided. The structure immediately has a preference in bending direction due to the imperfection. The structure now deforms in a constant way without pressure build-up. This shows in the stress-strain curve as a smooth curve without sharp transitions. One imperfection gives about the same results as two. Therefore from now on an imperfect structure is used with one imperfection located as shown before in figure 5.4.

Figure 5.5 : The selected imperfection applied to the RVE.
6 Loading the imperfect model

6.1 Horizontal loading

A solution has been found for the pressure drop in the horizontal load case. The new imperfect model will be used for further simulations. A horizontal load is applied again now using the EGP model for both the elastic and the plastic response to the loading. The result from this loadcase is presented in Fig. 6.1.

![Stress-strain curve of horizontal compression using the imperfect model.](image)

This result looks satisfying and a clear yielding point can be pointed out from this graph.
6.2 Checking vertical loading

Using the imperfect model gives a good result in horizontal loading. Next it is checked what the effect of the imperfection is on vertical loading. The vertical loading situation is now applied on the newly acquired model. The results from this simulation are compared with the simulations earlier done without the defect, both using the Neo-Hookean model. The stress-strain curves following from this test are plotted in Fig. 6.2.

![Figure 6.2: Comparison of vertical loading with and without imperfection.](image)

From the figure can be seen that is has no visible effect on the vertical loading situation. The earlier acquired result from this simulation is therefore still valid.
7 Full description of the deformation tensor

7.1 Full deformation tensor using a constant volume

7.1.1 Determine the components of the deformation tensor

In the experiments so far only a part of the deformation tensor is prescribed. When for example loading in vertical direction, only the vertical displacement is used as input, whereas the horizontal movement is let free so the material can freely contract. This works for horizontal and vertical loading. When loading under an angle the boundary conditions are not so trivial to prescribe anymore to get the desired effect and the same conditions as in other loading directions. This problem can be solved by a complete description of the deformation tensor. In this way there are no unknown displacements anymore. For a first attempt to reach this the surface area of the RVE is assumed to be constant during loading. If this is applied in a vertical loading condition, the x displacement of node 2 (dx) can be calculated from the given y displacement of node 4 (dy). Some parameters are shown in Fig. 7.1

\[
x_0 \cdot y_0 = (x_0 + dx) \cdot (y_0 + dy), \tag{7.1}
\]

\[
\frac{x_0 \cdot y_0}{(y_0 + dy)} - x_0 = dx. \tag{7.2}
\]

Filling in the applied y-displacement of node 4, \( dy = y_0(e^{\varepsilon} - 1) \) gives,

\[
\frac{x_0 \cdot y_0}{y_0 + y_0(e^{\varepsilon} - 1)} - x_0 = dx, \tag{7.3}
\]

which leads to:

\[
dx = x_0(e^{\varepsilon} - 1) \tag{7.4}
\]

Figure 7.1: Dimensions of element before and after deformation.
These displacements (dx and dy) can be used in Marc Mentat. It is checked if the surface area of the element is indeed constant throughout the whole deformation. This is true, so the deformation tensor becomes:

\[
F = \begin{pmatrix}
1 & 1 & 1 & 2 & 2 & 1 & 2 & 2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & 4 & 2 & 4 & 2 & 4 & 2 & 4 \\
\end{pmatrix}
\]

\[
(1) + (1) \\
(1) + (1) \\
(1) + (1) \\
\]

\[
(7.5)
\]

\[
= (1+ \frac{n2dx}{x_0}) \hat{e}_1 + n4dx \frac{n2dy}{y_0} \hat{e}_2 + (1+ \frac{n4dy}{y_0}) \hat{e}_3 \\
\]

\[
= (1+ \frac{x_0(e^{-\varepsilon t} - 1)}{x_0}) \hat{e}_1 + (1+ \frac{y_0(e^{\varepsilon t} - 1)}{y_0}) \hat{e}_2 \\
\]

\[
= e^{-\varepsilon t} \hat{e}_1 + e^{\varepsilon t} \hat{e}_2 \\
\]

\[
(7.6)
\]

\[
= e^{-\varepsilon t} \hat{e}_1 + e^{\varepsilon t} \hat{e}_2 \\
\]

\[
(7.7)
\]

Where:

- n2dx is the horizontal displacement of node 2
- n4dx is the horizontal displacement of node 4
- n2dy is the vertical displacement of node 2
- n4dy is the vertical displacement of node 4
- x0 is the initial horizontal position of node 2 (width of the element)
- y0 is the initial vertical position of node 4 (height of the element)

Taking the determinant of the deformation tensor immediately shows that there is no change of the surface (det(F) = e^{-\varepsilon t} \cdot e^{\varepsilon t} = 1).
7.1.2 Applying the deformation tensor to the RVE

The acquired deformation tensor of chapter 7.1.1 can be applied to the RVE.

\[ \ddot{u}_2 = (F - I) \cdot \ddot{x}_{02} = (F - I) \cdot \dddot{e}_1 \]
\[ = e^{\varepsilon \cdot t} \dddot{e}_1 \quad (7.8) \]

\[ \ddot{u}_4 = (F - I) \cdot \ddot{x}_{04} = (F - I) \cdot \dddot{e}_2 \]
\[ = e^{\varepsilon \cdot t} \dddot{e}_2 \quad (7.9) \]

\( \ddot{u}_2 \) and \( \ddot{u}_4 \) represent the displacements of node 2 resp. node 4 that can be used in Marc Mentat. The deformation at different times for the Neo-Hookean model can be seen in Fig. 7.2.
7.2 Rotated deformation tensor

7.2.1 Rotating the deformation tensor

Next the deformation tensor can be rotated by an angle (in this case 60°). The new coordinate system (Fig. 7.3) is expressed in terms of the original coordinate system (equation 7.10 and 7.11).

\[
{e'}_1 = \cos(\alpha)e_1 + \sin(\alpha)e_2 \\
{e'}_2 = -\sin(\alpha)e_1 + \cos(\alpha)e_2
\]  

(7.10)

(7.11)

The elements of the rotated deformation tensor are calculated:

\[
F_{11} = {e'}_1 \cdot F \cdot {e'}_1 \\
= (\cos(\alpha)e_1 + \sin(\alpha)e_2) \cdot (e^{-\varepsilon_t e_1} e_1 + e^{\varepsilon_t e_2} e_2) \cdot (\cos(\alpha)e_1 + \sin(\alpha)e_2)
\]

(7.12)

\[
F_{12} = {e'}_1 \cdot F \cdot {e'}_2 \\
= (\cos(\alpha)e_1 + \sin(\alpha)e_2) \cdot (e^{-\varepsilon_t e_1} e_1 + e^{\varepsilon_t e_2} e_2) \cdot (-\sin(\alpha)e_1 + \cos(\alpha)e_2)
\]

(7.13)
\[ F_{21} = \tilde{e}_2^* \cdot F \cdot \tilde{e}_1^* \quad (7.14) \]
\[ = (-\sin(\alpha) \tilde{e}_1 + \cos(\alpha) \tilde{e}_2) \cdot (e^{-\varepsilon t} \tilde{e}_1 + e^{\varepsilon t} \tilde{e}_2) \cdot (\cos(\alpha) \tilde{e}_1 + \sin(\alpha) \tilde{e}_2) \]
\[ = (-\sin(\alpha) e^{-\varepsilon t} \tilde{e}_1 + \cos(\alpha) e^{\varepsilon t} \tilde{e}_2) \cdot (\cos(\alpha) \tilde{e}_1 + \sin(\alpha) \tilde{e}_2) \]
\[ = -\sin(\alpha) \cos(\alpha) e^{-\varepsilon t} + \sin(\alpha) \cos(\alpha) e^{\varepsilon t} \]

\[ F_{22} = \tilde{e}_2^* \cdot F \cdot \tilde{e}_2^* \quad (7.15) \]
\[ = (-\sin(\alpha) \tilde{e}_1 + \cos(\alpha) \tilde{e}_2) \cdot (e^{-\varepsilon t} \tilde{e}_1 + e^{\varepsilon t} \tilde{e}_2) \cdot (\cos(\alpha) \tilde{e}_1 + \sin(\alpha) \tilde{e}_2) \]
\[ = (-\sin(\alpha) e^{-\varepsilon t} \tilde{e}_1 + \cos(\alpha) e^{\varepsilon t} \tilde{e}_2) \cdot (\cos(\alpha) \tilde{e}_1 + \sin(\alpha) \tilde{e}_2) \]
\[ = \sin^2(\alpha) e^{-\varepsilon t} + \cos^2(\alpha) e^{\varepsilon t} \]

So the rotated deformation tensor becomes:

\[ F^* = (\cos^2(\alpha) e^{-\varepsilon t} + \sin^2(\alpha) e^{\varepsilon t}) \tilde{e}_1 \tilde{e}_1 + (-\sin(\alpha) \cos(\alpha) e^{-\varepsilon t}) \]
\[ + \sin(\alpha) \cos(\alpha) e^{\varepsilon t}) \tilde{e}_2 \tilde{e}_2 + \tilde{e}_1 \tilde{e}_1 \]

(7.16)
7.2.2 Applying the rotated deformation tensor to a one element model

This rotated deformation tensor can be applied to the initial position vectors of the corners of the RVE. To test this method it is first applied to a one element model with a length and height of 1x1. So the initial position vector $\bar{x}_{02}$ of node 2 is $\vec{e}_1$, whereas the initial position vector $\bar{x}_{04}$ of node 4 is $\vec{e}_2$. The displacements of node 2 and 4 are expressed as $\vec{u}_2$ and $\vec{u}_4$.

$\vec{u}_2 = (F - I) \cdot \bar{x}_{02} = (F - I) \cdot \vec{e}_1 \quad (7.17)$

$\vec{u}_2 = (\cos^2(\alpha) e^e + \sin^2(\alpha) e^e - 1) \vec{e}_1 + (-\sin(\alpha)\cos(\alpha) e^{-e} + \sin(\alpha)\cos(\alpha) e^e) \vec{e}_2$

$\vec{u}_4 = (F - I) \cdot \bar{x}_{04} = (F - I) \cdot \vec{e}_2 \quad (7.18)$

$\vec{u}_4 = (-\sin(\alpha)\cos(\alpha) e^{-e} + \sin(\alpha)\cos(\alpha) e^e) \vec{e}_1 + (\sin^2(\alpha) e^{-e} + \cos^2(\alpha) e^e) \vec{e}_2$.

These displacements can be used in the Marc Mentat model. The deformation that follows from these displacements can be seen in Fig. 7.4.

![Figure 7.4: One element model in undeformed state (red) and deformed state (blue).](image-url)
The result from this one element model looks plausible. The structure deforms under an angle and checking for the surface area gives the same result as the initial position. The surface area is thus constant over the deformation time, which was intended. Plotting the stress-strain relation of this loadcase in one figure with the vertical loadcase of this element, compensating for the rotated frame, it gives the result of Fig. 7.5. This confirms that the rotated deformation tensor is applied correctly.

Figure 7.5: Comparing vertical loading and loading under a 60 degree angle of a one element model.
7.2.3 Applying the rotated deformation tensor to the RVE

Since simulations with the one element model looked plausible the deformation tensor from chapter 7.2.1 can now be applied to the RVE.

\[ \ddot{u}_2 = (\mathbf{F} - \mathbf{I}) \cdot \ddot{x}_{02} = (\mathbf{F} - \mathbf{I}) \cdot 7.55174 \ddot{e}_1 \]  
\[ = 7.55174 \left( \cos^2(\alpha) \epsilon^{\epsilon t} - \sin^2(\alpha) \epsilon^{\epsilon t} - 1 \right) \ddot{e}_1 + \left( -\sin(\alpha) \cos(\alpha) \epsilon^{\epsilon t} + \sin(\alpha) \cos(\alpha) \epsilon^{\epsilon t} \right) \ddot{e}_2 \]  
\[ \ddot{u}_4 = (\mathbf{F} - \mathbf{I}) \cdot \ddot{x}_{04} = (\mathbf{F} - \mathbf{I}) \cdot 8.72 \ddot{e}_2 \]  
\[ = 8.72 \left( -\sin(\alpha) \cos(\alpha) \epsilon^{\epsilon t} + \sin(\alpha) \cos(\alpha) \epsilon^{\epsilon t} \right) \ddot{e}_1 + \left( \sin^2(\alpha) \epsilon^{\epsilon t} + \cos^2(\alpha) \epsilon^{\epsilon t} - 1 \right) \ddot{e}_2 \]

These displacements can be used in Marc Mentat. First they are applied to a Neo-Hookean material model. The deformation pattern resulting from this simulation can be seen in Fig. 7.6.

![Figure 7.6a: undeformed structure (t = 0s).](image1)

![Figure 7.6b: deformed structure (t = 0.175s).](image2)
It can be seen from these figures that the deformation pattern is similar to that of vertical loading. This is exactly what is expected from this loading situation, since the stacking of the circles in this direction is the same as in vertical direction.
To check if this situation is indeed the same as the vertical loading case, both stress-strain curves should be plotted in one figure. These two graphs should be on top of each other.
To have the same result from the stress-strain curve under a 60 degree angle as in vertical direction, the Matlab script has to be adapted to plot the stress in this new direction. This is explained in Chapter 9. The resulting stress-strain curve can be seen in Fig. 7.7.

![Figure 7.7: Comparing vertical loading and loading under a 60 degree angle.](image)

From this figure it can be concluded that now all boundary conditions are chosen correctly, since the two loading cases match. The acquired model can be used to do simulations in different kind of angles using the EGP model to do both elastic and plastic simulations.
8 Loading in different angles

Now that the right boundary conditions are available they can be applied to a structure containing the EGP material model. This is done for a normal $S_a$ value and a raised $S_a$ value as explained in chapter 4.2.4. With the standard value the simulation goes as predicted and complete finishes. The simulation with a raised value of $S_a$ however, stops prematurely without a clear reason. The only difference with the other simulation is the difference in $S_a$ so the problem must be there. It is lowered until the simulation finishes in a normal way. The only problem that may occur by doing this is that the simulation is not completely elastic over the whole interval anymore. Therefore this simulation is compared with the simulation done with the Neo-Hookean material model. The stress-strain curves are plotted for the $\sigma_{11}$ direction in the rotated coordinate system (chapter 7.2.1). The result is displayed in Fig. 8.1.

![Figure 8.1](image)

**Figure 8.1** : Comparing the Neo-Hookean material model with the EGP model: $S_a = 50$.~

It can be seen that the stress-strain curves only match for strains up to approximately 0.18[-]. Therefore the result has to be treated with caution.
The resulting stress-strain curves are expressed in the original coordinate system again since these are the directions to eventually create a yielding surface with. The result for the $\sigma_{11}$ direction is given in Fig. 8.2.

![Figure 8.2: Stress-strain curves in $\sigma_{11}$ direction for compression under 60 degrees.](image)

For the $\sigma_{22}$ direction the result is given in Fig. 8.3.

![Figure 8.3: Stress-strain curves in $\sigma_{22}$ direction for compression under 60 degrees.](image)
For both cases a drop in stress can be detected at a strain value of approximately 0.18[-]. This is exactly where the EGP model with an $S_a = 50$ does not match the Neo-Hookean model anymore. It seems that the model passes its yield point here and enters the strain softening area. This is not desirable since it has to be fully elastic in this case. Clearly no yield points can be extracted from these graphs. Plotting the stress-strain curves of the EGP model with the stress-strain curves of the Neo-Hookean model, an idea that was discarded earlier on because of some problems with it, gives the results given in Fig. 8.4 and 8.5.

![Graph](image)

**Figure 8.4**: Stress-strain curves in $\sigma_{11}$ direction for compression under 60 degrees.

![Graph](image)

**Figure 8.5**: Stress-strain curves in $\sigma_{22}$ direction for compression under 60 degrees.
From the plots in the $\sigma_{22}$ direction a yielding point could be pointed out, but in the $\sigma_{11}$ direction there are some problems as before with the Neo-Hookean model. So no reliable yielding limits can be pointed out from these simulations. A solution has to be found to make the simulations with a raised value of $S_a$ more stable.
9 Adapting the Matlab script

The existing Matlab script has been changed to be able to plot figures in other directions than just the vertical one. First the Matlab model is adapted to import two sets of displacement versus reaction force tables. In this way two stress-strain curves can be compared as is done several times during this survey.

For horizontal and vertical simulations, the stress-strain curve could easily be determined. The vertical displacement of node 4 and the horizontal displacement of node 2 are directly imported from Marc Mentat. With these values the stretch can be determined since the initial height and width are known. In this survey loads are applied under an angle. The stretch cannot be determined as straightforward as in the horizontal and vertical case. Therefore a stretch tensor is determined. The Cauchy-Green deformation tensor can be calculated by $C = F^T \cdot F$. Since Matlab cannot work with tensors this is done with symbols by hand. This Cauchy-Green stretch tensor is rewritten in spectral form. By taking the square root of this spectral form the right stretch tensor $U$ is acquired. Taking the natural logarithm of $U$ gives the logarithmic strain tensor $\Lambda$.

Besides the strain also the stress has to be determined in the new main direction. This can be done by rotating the stress tensor in the same way as was done with the deformation tensor in chapter 7.2.1. The Matlab model is now suitable to compare the stress-strain curve of a 60 degree compression expressed in the rotated coordinate system with the stress-strain curve of a vertical compression expressed in the original coordinate system.
10 Conclusions and recommendations

To determine the yield limits in different directions of a two dimensional honeycomb structure first an existing model had to be expanded. First the two main directions were loaded. Vertical compression gave no difficulties from the beginning on, but horizontal loading did give some problems. Some checks were carried out. The Neo-Hookean model did not seem to be suitable anymore for the elastic simulations so the EGP model was taken with a raised value of \( S_a \) to force it to behave elastic. The high stiffness in horizontal direction was fixed by introducing an imperfection to the structure. The next goal was to find a yielding point when loading under an angle of 60 degrees with a vertical line. This direction has the same stacking of circles as the vertical direction. Therefore the vertical direction can be used to validate the stress-strain curves in this direction. To have full control of the deformation of the structure the deformation tensor was fully described using a constant surface area, otherwise the structure did not deform in the right way. Eventually the rotated deformation tensor could be applied to the RVE and it could be proved with the vertical loading that it was correct. Loading under a 60 degree angle using the EGP model still gave some problems. Raising the \( S_a \) far enough to make the model completely elastic made it impossible to complete the simulation.

For future investigation the problem with this EGP model needs to be fixed. The current model seems to have a problem with the sharp transition caused by strain softening. If it can cope with this transition and the simulation can be completed with an \( S_a \) large enough to make the model elastic, the yielding limit under an angle of 60 degrees can be pointed out and eventually for all directions. Another approach to the problem is to avoid this pressure drop from strain softening. The damper of the spring damper model of chapter 2.4.2 can be removed in the EGP model by changing some parameters. This results in a larger elastic region without altering the strain softening and hardening behavior. Thus the pressure peak that seems to cause a problem is avoided. Another point of attention is the assumption that the surface area is constant during deformation. This is not true in reality and a more suitable value for lateral contraction has to be implemented. Doing these things can eventually lead to a realistic yielding surface.
### References


[Marc Mentat]  MSC.Marc Mentat 2005r3 product documentation


Appendix A

Matlab script

%% Homogenization scheme %%
clear all
close all
clc

%% MSC data imported %%

%read in data: simulation
1%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
model = ['60graden_np'];
alpha = -1/3*pi;
beta = 0;

file_1 = [model,'_n2dx_n2fx.dat'];
file_2 = [model,'_n2dy_n2fy.dat'];
file_3 = [model,'_n4dx_n4fx.dat'];
file_4 = [model,'_n4dy_n4fy.dat'];

[n2dx_1 n2fx_1] = textread(file_1,'%f%f','headerlines',9);
n2dy_1 n2fy_1 = textread(file_2,'%f%f','headerlines',9);
n4dx_1 n4fx_1 = textread(file_3,'%f%f','headerlines',9);
n4dy_1 n4fy_1 = textread(file_4,'%f%f','headerlines',9);

%read in data: simulation
1%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
model2 = ['verticaal'];

file_5 = [model2,'_n2dx_n2fx.dat'];
file_6 = [model2,'_n2dy_n2fy.dat'];
file_7 = [model2,'_n4dx_n4fx.dat'];
file_8 = [model2,'_n4dy_n4fy.dat'];

[n2dx_2 n2fx_2] = textread(file_5,'%f%f','headerlines',9);
n2dy_2 n2fy_2 = textread(file_6,'%f%f','headerlines',9);
n4dx_2 n4fx_2 = textread(file_7,'%f%f','headerlines',9);
n4dy_2 n4fy_2 = textread(file_8,'%f%f','headerlines',9);
% Initial position vectors
x02 = [7.55174 0 ]; % bottom right node (node 2)
x04 = [0 8.72 ]; % top left node (node 4)

% Determine first Piola Kirchhoff stress tensor P % ///// 1 \\\
Pi_1 = zeros(2,2); % Incremental Stress Tensor
V0 = x02(1)*x04(2); % Initial RVE volume
ninc_1 = length(n4fy_1); % number of increments

for i = 2:ninc_1
    Pij_1 = 1/V0*[ n2fx_1(i)*x02(1) n4fx_1(i)*x04(2)
                  n2fy_1(i)*x02(1) n4fy_1(i)*x04(2)]; %
    Determine incremental P stress tensor
    Pi_1 = [Pi_1 Pij_1]; %
    Store all incremental stress tensors
end

% Determine first Piola Kirchhoff stress tensor P % ///// 2 \\\
Pi_2 = zeros(2,2); % Incremental Stress Tensor
V0 = x02(1)*x04(2); % Initial RVE volume
ninc_2 = length(n4fy_2); % number of increments

for i = 2:ninc_2
    Pij_2 = 1/V0*[ n2fx_2(i)*x02(1) n4fx_2(i)*x04(2)
                  n2fy_2(i)*x02(1) n4fy_2(i)*x04(2)]; %
    Determine incremental P stress tensor
    Pi_2 = [Pi_2 Pij_2]; %
    Store all incremental stress tensors
end

% Determine the deformation gradient tensor F % ///// 1 \\\
F1_1 = zeros(2,2); % incremental deformation tensor
E_1 = zeros(2,2);
C_1 = zeros(2,2);

for i = 2:ninc_1
    % general form
    Pij_1 = [ 1+(n2dx_1(i))/x02(1) n4dx_1(i))/x04(2)
             n2dy_1(i))/x02(1) 1+(n4dy_1(i))/x04(2)];
\[ f_{11,1} = F_{ij,1}(1,1); \]
\[ f_{12,1} = F_{ij,1}(1,2); \]
\[ f_{21,1} = F_{ij,1}(2,1); \]
\[ f_{22,1} = F_{ij,1}(2,2); \]

% calculate the right Cauchy-Green deformation tensor \( C \)
\[
C_{ij,1} = \begin{bmatrix}
(f_{11,1}^2+f_{21,1}^2) & (f_{22,1}f_{21,1}+f_{11,1}f_{12,1}) \\
(f_{22,1}f_{21,1}+f_{11,1}f_{12,1}) & (f_{12,1}^2+f_{22,1}^2)
\end{bmatrix};
\]

\([V \ W] = \text{eig}(C_{ij,1}); \quad \% V = \text{eigenvector} ; W = \text{eigenvalue}\]
labda1 = W(1,1);
labda2 = W(2,2);

vector1 = V(:,1);
vector2 = V(:,2);

% calculate spectral form of \( C \) (general case)
\[
C_{ij,1}(1,1) = \lambda_1 \cdot \text{vector1}(1)^2 + \frac{1}{\lambda_2} \cdot \text{vector2}(1)^2; \\
C_{ij,1}(1,2) = \lambda_1 \cdot \text{vector1}(1) \cdot \text{vector1}(2) + \frac{1}{\lambda_2} \cdot \text{vector2}(1) \cdot \text{vector2}(1); \\
C_{ij,1}(2,1) = \lambda_1 \cdot \text{vector1}(1) \cdot \text{vector1}(2) + \frac{1}{\lambda_2} \cdot \text{vector2}(1) \cdot \text{vector2}(1); \\
C_{ij,1}(2,2) = \lambda_1 \cdot \text{vector1}(2)^2 + \frac{1}{\lambda_2} \cdot \text{vector2}(2)^2; \\
\%
\]
% calculate the right stretch tensor \( U \)
\[ U_{1} = \sqrt{C_{ij,1}}; \%
\]
% calculate the logarithmic stain tensor \( E \)
\[ E_{ij,1} = \log(U_{1}); \% \text{general case} \%
\[
\text{Fi}_1 = [\text{Fi}_1 \ F_{ij,1}]; \%
\text{Store all incremental deformation tensors} \%
\text{E}_1 = [\text{E}_1 \ E_{ij,1}]; \\
\text{C}_1 = [\text{C}_1 \ C_{ij,1}]; \\
end
\%
% Determine the deformation gradient tensor \( F \) \%
\text{Fi}_2 = \text{zeros}(2,2); \%
\text{E}_2 = \text{zeros}(2,2); \\
\text{C}_2 = \text{zeros}(2,2); \\
\text{for} \ i = 2:1:ninc_1
\%
genral form
\[
\text{Fij}_2 = [\ 1+(n2dx_2(i))/x02(1) \ (n4dx_2(i))/x04(2) \\
(n2dy_2(i))/x02(1) \ 1+(n4dy_2(i))/x04(2) ];
\]
\[
f_{11,2} = \text{Fi}_2(1,1); \\
f_{12,2} = \text{Fi}_2(1,2); \]
\[ f_{21,2} = Fi_j(2,1); \]
\[ f_{22,2} = Fi_j(2,2); \]

% calculate the right Cauchy-Green deformation tensor \( C \)
\[
C_{ij,2} = \begin{bmatrix}
(f_{11,2}^2 + f_{21,2}^2) & (f_{22,2} * f_{21,2} + f_{11,2} * f_{12,2}) \\
(f_{22,2} * f_{21,2} + f_{11,2} * f_{12,2}) & (f_{12,2}^2 + f_{22,2}^2)
\end{bmatrix};
\]

\[ [V \ W] = \text{eig}(C_{ij,2}); \] % \( V = \text{eigenvector} ; \ W = \text{eigenvalue} \)
labdal = \( W(1,1); \)
labda2 = \( W(2,2); \)

vector1 = \( V(:,1); \)
vector2 = \( V(:,2); \)

% calculate spectral form of \( C \) (general case)

\[
C_{ij,2}(1,1) = \text{labdal} * \text{vector1}(1)^2 + 1/\text{labda2} * \text{vector2}(1)^2;
\]
\[
C_{ij,2}(1,2) = \text{labdal} * \text{vector1}(1) * \text{vector1}(2) + 1/\text{labda2} * \text{vector2}(1) * \text{vector2}(1);
\]
\[
C_{ij,2}(2,1) = \text{labdal} * \text{vector1}(1) * \text{vector1}(2) + 1/\text{labda2} * \text{vector2}(1) * \text{vector2}(1);
\]
\[
C_{ij,2}(2,2) = \text{labdal} * \text{vector1}(2)^2 + 1/\text{labda2} * \text{vector2}(2)^2;
\]

% calculate the right stretch tensor \( U \)
\[
U_{2} = \sqrt{C_{ij,2}};
\]

% calculate the logarithmic stain tensor \( E \)
\[
E_{ij,2} = \log(U_{2}); \] % general case

\[ Fi_2 = [Fi_2 \ Fi_j_2]; \] %
Store all incremental deformation tensors
\[ E_2 = [E_2 \ Eij_2]; \]
\[ C_2 = [C_2 \ Cij_2]; \]
end

%% Determine the cauchy stress tensor ///// 1 \\\
\[
Gsi_1 = \text{zeros}(2,2);\] %
Incremental cauchy stress tensor

for \( i = 3:2:\text{length}(Fi_1) \)
    \[ Gsj_1 = 1/\det(Fi_1(:,i:i+1)) * Fi_1(:,i:i+1) * (Fi_1(:,i:i+1)'); \] % Incremental cauchy stress tensor
    \[ \det(Fi_1(:,i:i+1)) \]
    \[ Gsi_1 = [Gsi_1 \ Gsj_1]; \] %
Store all incremental stress tensors
end

%% Determine the cauchy stress tensor ///// 2 \\\

Gsi_2 = zeros(2,2);  

Incremental cauchy stress tensor

for i = 3:2:length(Pi_2)

    Gsij_2 = 1/det(Fi_2(:,i:i+1))*Pi_2(:,i:i+1)*(Fi_2(:,i:i+1)');
  
% Incremental cauchy stress tensor

Gsi_2 = [Gsi_2 Gsij_2];

Store all incremental stress tensors

end

% strain and rotated stress tensor  /// 1 ///

E11_1 = E_1(1,1:2:end);
E22_1 = E_1(2,2:2:end);
E12_1 = E_1(1,2:2:end);

Gs11_1 = Gsi_1(1,1:2:end);
Gs22_1 = Gsi_1(2,2:2:end);
Gs12_1 = Gsi_1(1,2:2:end);
Gs21_1 = Gsi_1(2,1:2:end);

% rotate the stress tensor
Gs11new_1 = Gs11_1*(cos(alpha))^2 + Gs22_1*(sin(alpha))^2 + Gs12_1*sin(alpha)*cos(alpha) + Gs21_1*sin(alpha)*cos(alpha);
% reken sigma11 in de nieuwe richting uit
Gs22new_1 = Gs11_1*(sin(alpha))^2 + Gs22_1*(cos(alpha))^2 - Gs12_1*sin(alpha)*cos(alpha) - Gs21_1*sin(alpha)*cos(alpha);
% reken sigma22 in de nieuwe richting uit

% strain and rotated stress tensor  /// 2 ///

E11_2 = E_1(1,1:2:end);
E22_2 = E_1(2,2:2:end);
E12_2 = E_1(1,2:2:end);

Gs11_2 = Gsi_1(1,1:2:end);
Gs22_2 = Gsi_1(2,2:2:end);
Gs12_2 = Gsi_1(1,2:2:end);
Gs21_2 = Gsi_1(2,1:2:end);

% rotate the stress tensor
Gs11new_2 = Gs11_2*(cos(beta))^2 + Gs22_1*(sin(beta))^2 + Gs12_1*sin(beta)*cos(beta) + Gs21_1*sin(beta)*cos(beta);
% reken sigma11 in de nieuwe richting uit
Gs22new_2 = Gs11_2*(sin(beta))^2 + Gs22_1*(cos(beta))^2 - Gs12_1*sin(beta)*cos(beta) - Gs21_1*sin(beta)*cos(beta);
% reken sigma22 in de nieuwe richting uit

% plot

figure
p1 = plot(abs(E22_1),abs(Gs22new_1), 'r');
hold on
p2 = plot(abs(E22_2), abs(Gs22new_2), 'bx');
legend('60 degree', 'vertical')
set(gca, 'box', 'on', 'linewidth', 2, 'fontsize', 16)
set([p1 p2], 'linewidth', 2)
lx = xlabel('$\varepsilon_{ln}$ $[\text{~}]$');
ly = ylabel('$\sigma_{22}$ $[\text{MPa}]$');
set([lx ly], 'interpreter', 'latex', 'fontsize', 32)