Progress on modelling two-dimensional biopolymer networks

B. Roossien

Supervisors:
Dr. Ir. P.R. Onck, Dr. T. van Dillen,
Prof. Dr. Ir. E. van der Giessen

Micromechanics of Materials, Materials Science Center
University of Groningen

Master's Thesis
26th January 2006

Abstract

A network composed of actin filaments exhibits an increasing stiffness under mechanical loading. This thesis expands the original 2D network model [9] to a fully periodic liquid cell network model (RVE). The RVE model has the same network shear response as the initial 2D network model, independent of the boundary prescription. However, the normal stress on the network does differ when volume conservation is prescribed. Tensile and compression of the network cell is totally dominated by the liquid cell, but the network does show stiffening behaviour. A compressive pre-strain applied on a network before a shear strain is applied will therefore return a different response than a network on which only a shear strain is applied. The origin is explained by using the orientation distribution of the filaments and slack in the network.

Contents

1 Introduction **3**

2 Biological overview of the cytoskeleton **5**

2.1 The biological cell 5

2.2 Cytoskeleton & cell motility 6

2.3 Actin 8

3 The unit cell model **9**

3.1 Network generation 9

3.2 Euler-Bernoulli beam elements 12

4 The representative volume element model **15**

4.1 RVE implementation 15

4.2 Unit cell response versus RVE response 18

4.3 RVE response under unconstrained shear 20

5 The RVE network in a liquid cell **23**

5.1 Implementation of a liquid cell 23

5.2 Network response with a liquid cell under simple shear 24

5.3 Tension and compression 27

6 The effect of pre-strain on the network shear response **29**

6.1 New network quantities 29

6.1.1 Discrete orientation distribution function 29

6.1.2 Slack orientation function 30
6.1.3 Principal directions 30
6.2 Pre-strain results 32

7 Conclusions and recommendations 38
7.1 Conclusions 38
7.2 Recommendations 39

References 41

A Fortran code changes 42

CHAPTER 1

Introduction

In cell mechanics, biological tissues show an intriguing active response to mechanical loading, for which a great interest has been shown in the last decade. A quantitative understanding of these complex mechanical responses, such as cell motility and mechanotransduction, is of crucial importance for many biomedical applications. Tissues that have a network-like structure respond to mechanical loading by exhibiting an increasing stiffness, i.e. the ratio between the change of stress and change of strain increases [11]. This has for example been demonstrated by microtwisting experiments on individual cells [13] and by rheological experiments on in-vitro gels of cytoskeletal filaments like actin [3, 8]. These network-like tissues or biological gels are classified as being semiflexible polymers and have also attracted theoretical studies [7], but these mainly apply on the small strain regions. Besides experiments and theoretical studies, one other method of understanding the behaviour of mechanical loaded networks is the use of numerical calculations. A two-dimensional network model, which makes use of the finite element method, has been developed by the *micromechanics* research group at the University of Groningen in trying to understand network behaviour under mechanical loading [9]. In this master's thesis, the model is extended with new features, resulting in a more in-depth understanding of the behaviour of filamenteous networks.

To give the reader an idea about the real life importance of these networks, a brief explanation of the structure of the biological cell and in particular, the construction of the (actin) cytoskeleton, is given in chapter 2. Chapter 3 discusses the details about the original two-dimensional network model [9] before any modifications were made on it. It describes the network generation and explains how the network response is calculated. In chapter 4, the model is extended with the introduction of new degrees of freedom for the network, resulting in a fully periodic model with no longer fixed box sizes. The unexpected problems that arise from these modifications are solved in chapter 5, where a liquid cell is introduced into the model to account for volume conservation. This chapter is not only considering the response for networks under an applied shear strain, but also under pure compression and tension. Chapter 6 discusses what the effect is on the network response of any pre-straining. This network behaviour is explained using the orientation distribution function and slack orientation function quantities. Finally, chapter 7 gives a summary of the results, conclusions and recommendations for further research with the two-dimensional network model. The appendix explains the most important changes applied in the Fortran code, which were needed for the extension of the network model. These can save the group a substantial amount of work in finding out how to apply these changes in their three-dimensional network model, which is currently under development [4].

This master's thesis could not have been written without the help of a few people. First of all, I want to thank Erik van der Giessen for the opportunity given to graduate in his research group and for his guidance, support and continueous encouragement during the graduation period.

Secondly, I want to thank Patrick Onck and Teun van Dillen for their co-operation in this project, ideas and support. Patrick was a good help in writing new pieces of programming code and I had a lot of interesting discussions with Teun.

Special thanks to Teun Koeman with whom, as a room mate and fellow-nerd, I had interesting conversations about whatever nerds talk about. I learned a great deal from him.

Finally, warm thanks to the other (ex-)members of the μMeχ group with whom I had a great and enjoyable time: Andreas, Angelo, Anja, Antonia, Ashok, Cihan, Florence, Kodanda, Liesbeth, Maarten, Maneesh, Monica, Ranjeet, Ratna, Ria, Serge and Yaroslav.

Bart Roossien
January 2006

CHAPTER 2

Biological overview of the cytoskeleton

There are a great many varieties of living systems on this planet, which have been brought to existence by evolution over many millions of years. These living systems are all of a very different kind and therefore, it is very hard to make generalisations in biology. However, in all these varieties of living systems and functions, a few commonalities can be found. One of the most interesting commonalities is the cellular construction. Every living system is composed out of one or more biological cells. Each of these cells is a living system of its own, and as long as it lives in favourable conditions (temperature, food), it can sustain itself. In multi-cell organisms like humans, cells build up the organs, muscles and skin. The cells also take care of logistical functions, like the transport of oxygen and carbon dioxide, and defend the multi-cell creatures from badwilling bacteria and virusses. Because of this variety in functions, the biological cell is therefore one of the most interesting and complex systems in biology to look at. This chapter will give some background information on the cell, with special attention on the construction of the cytoskeleton, which plays an important part during the mechanical loading of the cell.

2.1 The biological cell

The biological cell is a most intriguing and complex system. The basic components that can be found in all cells are the plasma membrane and the cytoplasma. The membrane is the outer 'skin' of the cell and separates the cell interior from its environment, but allows certain types of molecules to get through in order to supply the cell with fuel (oxygen, glucose, amino-acids) and dispose its waste to the blood vessels. The membrane is impermeable for water itself and

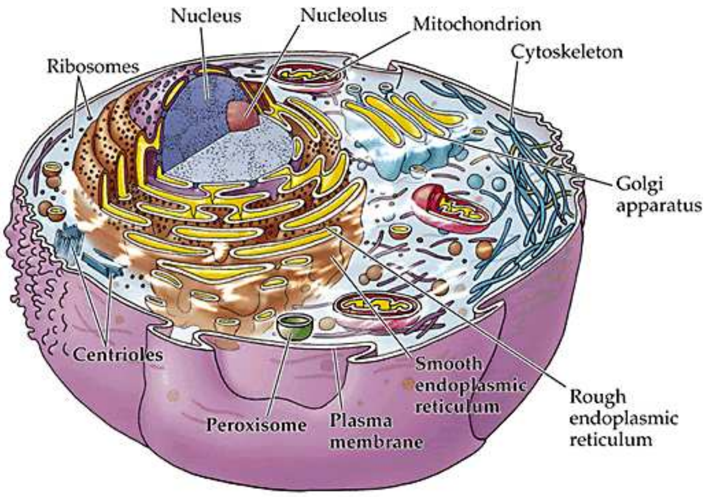


Figure 2.1: Schematic drawing of an eukaryotic cell with all its components.

prevents the cytoplasm from flowing into the environment, which results in the cell conserving its volume. The cytoplasm is anything that can be found within the boundaries of the plasma membrane, except for a possible nucleus. Simplistic cells or prokaryotic cells are commonly found in the bacteria class and distinguish themselves from the more complex eukaryotic cells, which can be found in plants, animals and also humans, by the lack of a membrane-bound nucleus. The genetic material in eukaryotic cells is held inside the nucleus in the shape of chromosomal DNA, but in prokaryotic cells, the genetic material is found in circular loops called plasmids, which are floating in the cytoplasma.

Figure 2.1 is a schematic drawing of an eukaryotic cell from an animal, where a variety of enclosed systems with their own membrane can be found. These systems, like the mitochondria, the endoplasmic reticulum, the vacuole and the ribosomes, are called organelles. These organelles perform basic but important functions in the cell, like the generation of ATP out of ADP (energy) and the construction of necessary proteins [5]. The cytoplasm without these organelles is called the cytosol and in this watery fluid, large arrays of fibrous proteins are contained, which are collectively called the cytoskeleton. This cytoskeleton gives the cell its strength and rigidity, and takes part in mechanotransduction, cell motility and cell division.

2.2 Cytoskeleton & cell motility

Although most biological cells in living systems are in stationary positions, some cells require the ability to displace themselves or to make changes in their morphology. The mobility of cells is called cell motility and can be seen for example in the contraction of the muscle cells,

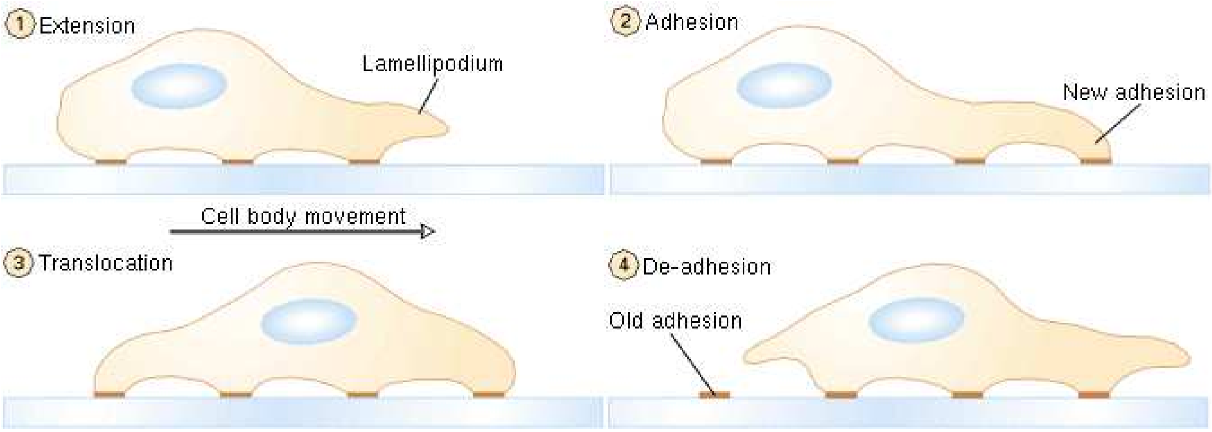


Figure 2.2: Schematic drawing of cell locomotion. (1) The leading edge lamellipodia are forwarded. (2) Some of the forwarded lamellipodia adhere to the underlying substratum via focal adhesion. (3) The bulk of the cytoplasm in the cell body flows forward. (4) Adhesion contacts at the trailing end of the cell are detached and the tail retracts into the body cell. [5]

the elongation of nerve axons and the constriction of a dividing cell during mitosis. Even more subtle are the movements inside the cell itself, like the seperation of chromosomes. The critical element in these movements is the cytoskeleton. The cytoskeleton is a cytoplasmic system of three different types of fibers: the microfilaments ($\varnothing$ 7 − 9nm), the intermediate filaments ($\varnothing$ 10nm) and microtubules ($\varnothing$ 24nm). These fibers are well ordered polymers build from small protein subunits. The network-like structure of these fibers supports the cell membrane in order to give the cell its shape. The fibers also form tracks in the cell itself, along which organelles can move in the cytosol.

Cell motility takes place by the assembly and disassembly of microfilaments and microtubules at the front and the rear end of the cell. Cell locomotion results from a coordination of motions generated by different parts of the cell and starts with the forwarding of the lamellipodium, which is a large broad membrane protrusion, as well as slender “fingers” of the membrane called the filopodia. This is illustrated in figure 2.2. Although it is not clear what propels the membrane to go forward, one hypothesis is that the membrane is pushed forward by the rapid assembling microfilaments. After the forwarding of the cell membrane, pieces of the membrane will attach itself firmly onto the substratum. This process involves focal adhesion points that anchor the cell to the substratum, prevent the lamellipodia from retracting into the cell body. After the forward attachments are made, the bulk contents with the cell body will move forward. As the cell continues the travel, its tail is pulled forward by the contraction of the stress fibers in the tail or by elastic tension, often leaving small parts of the cell behind which are stuck to the substratum by the focal adhesion.

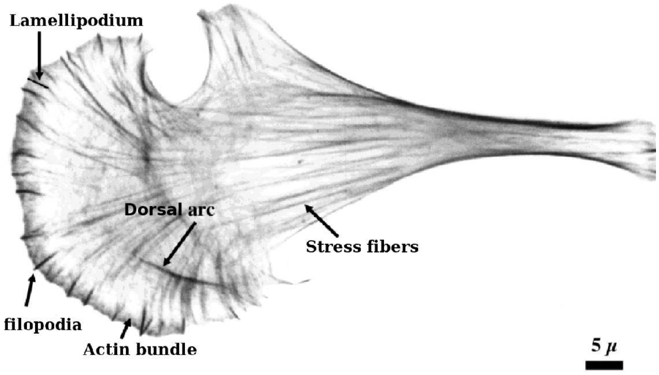


Figure 2.3: Microfilaments in a chick heart fibroblast, labelled with rhodamine-conjugated phalloidin. [10]

2.3 Actin

The migration as well as the morphology of the cell is mainly enabled by the microfilament cytoskeleton, which is build-up of actin subunits. This actin cytoskeleton is larger than any organelle in the cell and because the actin cytoskeleton is so big, it can change the cell morphology by just assembling or dissasembling itself. It can even make up to 5% of the cellular protein. During cell motility, actin must assemble rapidly and does not always have the chance to form ordered structures. It has a great variety of filament lengths, and the filaments are cross-linked into bundles and networks. The ratio of cytoskeletal proteins is not rigidly maintained and from all these properties, the cell can take its many shapes and vary them easily.

In figure 2.3 filament bundles of actin can be seen at the leading edge while axial bundles, called stress fibers, are underlying the cell body. The cell body is filled with a network of actin filaments, but the individual filaments of this network are difficult to resolve under the microscope.

Both the bundles and networks of filaments in the actin cytoskeleton have functionally identical roles. They both provide a framework that supports the plasma membrane and therefore determines the cell's shape. Structurally, bundles and networks are not identical and differ in organization of actin filaments. In bundles, actin filaments are closely packed in parallel arrays, whereas in a network, the actin filaments are loosely packed and more or less randomly oriented and placed. There are two types of actin networks. One is associated with the plasma membrane and is planar, the other is present in the cell and is three-dimensional, giving the cytosol its gel-like properties. In both the bundles and networks, the filaments are held together by actin cross-linking proteins. Short cross-linking proteins hold the filaments close together, thereby forming bundles of filaments. In contrast, long cross-linking proteins are able to adapt to any arrangement of actin filaments and therefore act as glue on the intersection of two filaments.

CHAPTER 3

The unit cell model

The two-dimensional network model considers a filamentous network contained in a cell or box of a certain size. The cell represents a small element of the entire network that can be found in the biological gel sample, but still has the same physical properties as the entire sample. However, the cell does not have fully periodic boundaries and cannot be represented as a representative volume element. Therefore, this model is referred to as a *unit cell* model. The response of this cell under mechanical loading is computed by using a finite element analysis on the filaments, which are considered to be Euler-Bernoulli beam elements. This chapter will explain the network generation and how mechanical loading is applied in the model.

3.1 Network generation

The unit cell network model is based on a two-dimensional periodic unit cell of dimensions $l_x \times l_y$ in which N filaments with length L are randomly placed with a random orientation. The proper periodicity is taken into account, that is, if a filament crosses a boundary, it continues further from the opposite boundary. The box size is chosen large enough that it does not affect the results of the network deformation behaviour much, as determined by Xie [15]. The density of the network is characterized by the line density ρ, which is defined as the total length of the filaments in the unit cell, divided by the unit cell area, i.e.

$$\rho = \frac{NL}{l_x l_y} \tag{3.1}$$

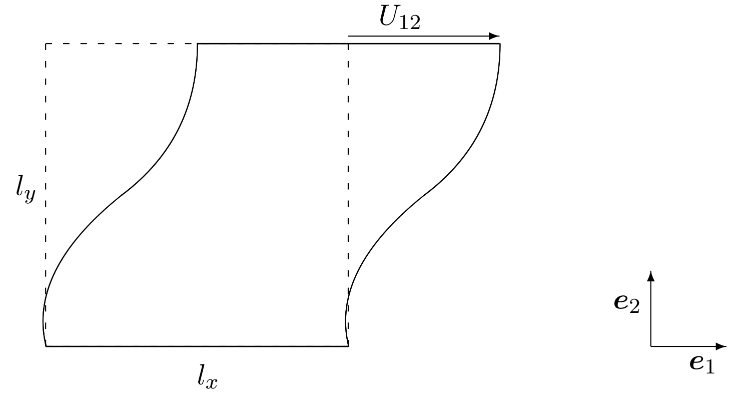


Figure 3.1: Schematic drawing of displacements in a network unit cell under shear

The filaments are treated as elastic rods with stretching stiffness AE, which relates the axial strain to the axial stress needed to induce that axial strain, and a bending stiffness EI, which relates the radius of curvature to the bending moment needed to induce that radius. In an isotropic elastic rod, the bending and stretching stiffness are related through their cross-sectional geometry, but in this model they are taken to be independent. Points where filaments overlap are considered to be crosslinked, in a similar fashion as done by Head*et al.* [2] and Wilhelms and Frey [14]. The crosslinks are assumed to be stiff, i.e. both displacements and rotation of the two filaments at the crosslink remain the same. The network generated this way is taken as the initial stress-free configuration.

Figure 3.1 is a schematic of the unit cell under an applied shear strain, without the actual filaments being shown. Both top and bottom boundaries of the unit cell remain flat. This is in contrast to the left and right boundaries of the network, which are able to distort. Because this kind of deformation is not allowed on the top and bottom boundaries, the unit cell is only fully periodic in the $\boldsymbol{e}_1$ direction. Therefore it cannot represent a representative volume element, which should have full periodicity in the $\boldsymbol{e}_2$ direction as well. The volume of the unit cell during deformation remains constant, which means that the diagonal components of the strain tensor, ε_1 and ε_2, are zero[1]. To maintain periodicity in the $\boldsymbol{e}_1$ direction, the left and right boundary nodes are 'tied' together. Tying can be considered as a very stiff spring between the two opposite boundary nodes. Note that due to periodicity, the nodes only differ by a distance l_x in horizontal direction and that their displacements are equal.

$$\underline{u}(l_x, y) = \underline{u}(0, y) \tag{3.2}$$

[1] Actually, volume conservation means that the trace of the strain tensor ε equals zero. However, the top and bottom boundary nodes are prescribed in such a way that ε_2 equals zero anyway, which immediately implies that ε_1 equals zero as well.

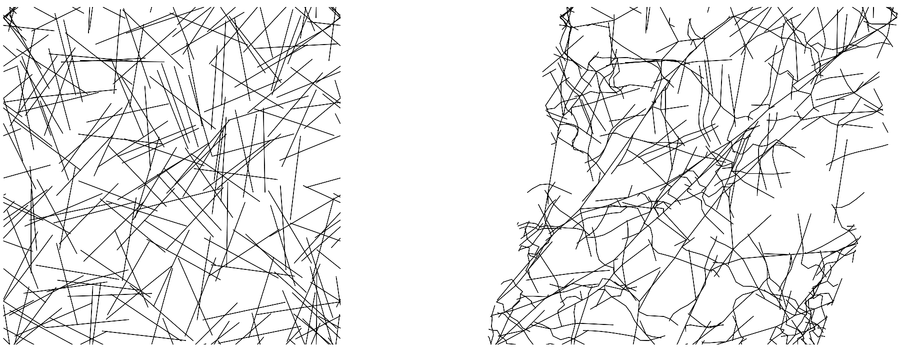


Figure 3.2: The unit cell network model. Left the initial stress free network and right the network under an applied shear strain $\Gamma = 0.225$.

Here $\underline{u}$ denotes the degree-of-freedom vector of a node, which is composed out of the displacements in the $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ directions, and the rotation about the $\boldsymbol{e}_3$-axis (perpendicular to $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$) in this particular order:

$$\underline{u} = \begin{bmatrix} u_x & u_y & l\phi \end{bmatrix}^{\mathsf{T}} \tag{3.3}$$

where the rotation has been multiplied by the beam length l to give it the same dimensions as the other displacements in the vector. The underlining of the quantity $\underline{u}$ means that it is a generalised finite element quantity and not a real vector as opposed to $\boldsymbol{u}$ for example. In this way, the corresponding nodes at $x = 0$ and $x = l_x$ are in fact one and the same node, which results in the proper periodicity. Unless some degrees of freedom are eliminated, it is not possible to add equation (3.2) directly into the system and therefore, the computer model uses the imaginary spring to connect the two nodes. If the stiffness of the spring is going to infinity, the two nodes will behave like one and the same node. For the computer program, it is sufficient to assign a high value to the stiffness of the spring. To be on the safe side, a subroutine checks if the stiffness is indeed chosen high enough to prevent the nodes drifting away from each other. The maximum drift distance between the two nodes that has been allowed was $l/10^3$.

Mechanical loading in terms of an applied shear strain on the unit cell is simulated by prescribing the displacements of the top and bottom boundary nodes. The bottom boundary nodes are fixed in their initial positions, which means that they cannot displace in the $\boldsymbol{e}_1$ or $\boldsymbol{e}_2$ direction nor can they rotate, i.e.

$$\underline{u}(x, 0) = \underline{0} \tag{3.4}$$

The top boundary nodes can only collectively move in the $\boldsymbol{e}_1$ direction by prescribing

$$\underline{u}(x, l_y) = \begin{bmatrix} U_{12} & 0 & 0 \end{bmatrix}^{\mathsf{T}} \tag{3.5}$$

Figure 3.2 shows a network in its initial stress-free configuration (left) and the same network

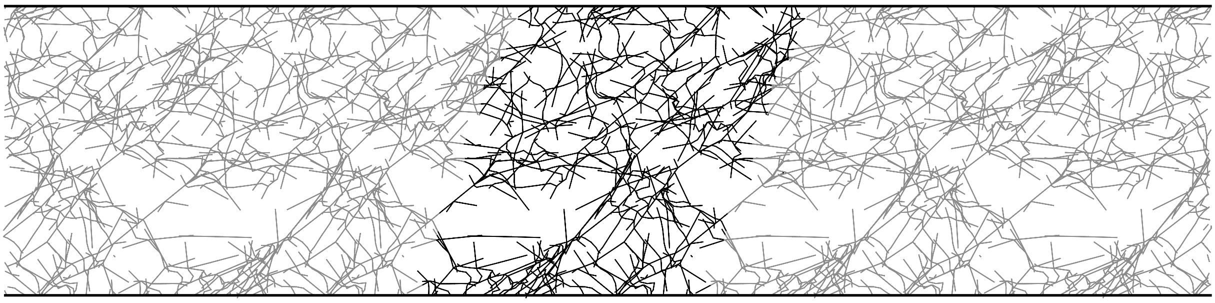


Figure 3.3: Shear strained horizontally periodic actin network (From [4]).

when it is under an applied shear strain with the previously described method and boundary conditions (right). It is clearly seen that the top and bottom boundaries remain flat, while the left and right boundaries deform identically. The periodicity of the network in horizontal direction can be seen clearly in figure 3.3, where the highlighted part is a single unit cell.

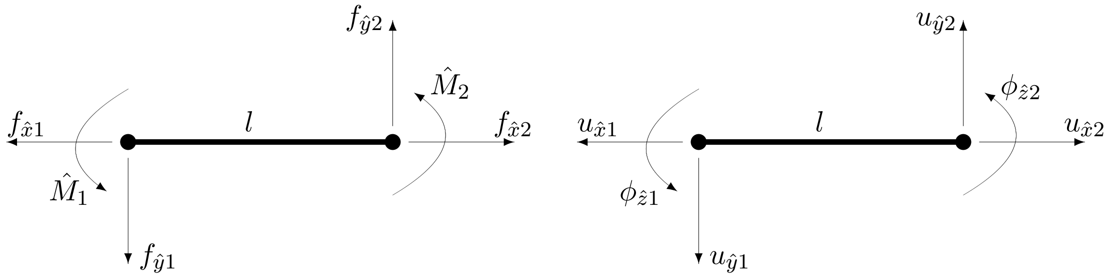
3.2 Euler-Bernoulli beam elements

Every filament in the network is discretized into 10 equal-sized Euler-Bernoulli beam elements, each accounting for stretching and bending of the entire network. Convergence tests have shown that this number is small enough in order not to influence the results [15]. The stiffness matrix of each of these beam elements can be deduced from the well-known Euler-Bernoulli beam theory [1, 6]. Figure 3.4 shows a beam element with the forces and moments that can act upon it and the displacements and rotations that are a response to those forces and moments respectively. To reach equilibrium, the following equations must be satisfied:

$$\begin{aligned} f_{\hat{x}1} + f_{\hat{x}2} &= 0 \\ f_{\hat{y}1} + f_{\hat{y}2} &= 0 \\ \hat{M}_1 + \hat{M}_2 + f_{\hat{y}2} l &= 0 \end{aligned} \tag{3.6}$$

Here the hat denotes the use of a local coordinate system, i.e. the coordinate system of the beam itself when it is placed in a horizontal position as shown in figure 3.4. Vectors and matrices that are expressed in the global coordinate system are written without hat. The external virtual work δw_e is given by $\delta w_e = (\delta \underline{\hat{u}})^\mathsf{T} \underline{\hat{f}}$, with $\underline{\hat{u}}$ and $\underline{\hat{f}}$ being the six point displacement and the six point force respectively. By incorporating the equilibrium equations into the external virtual work an expression for the internal virtual work δw_i is found:

$$\delta w_i = f_{\hat{x}2} \left(\delta u_{\hat{x}2} - \delta u_{\hat{x}1} \right) + f_{\hat{y}1} \left(\delta u_{\hat{y}2} - \delta u_{\hat{y}1} + l \delta \hat{\phi}_1 \right) + \hat{M}_2 \left(\delta \hat{\phi}_2 - \delta \hat{\phi}_1 \right) \tag{3.7}$$



(a) Forces and moments

(b) Displacements and rotations

Figure 3.4: Forces and moments with their corresponding displacements and rotations for a single Euler-Bernouilli beam element.

Because the internal virtual work is also equal to $\delta w_i = (\delta \underline{\varepsilon})^{\mathsf{T}} \underline{\sigma}$, the general stresses are chosen to be[2]

$$\underline{\sigma} = \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \end{bmatrix} = \begin{bmatrix} f_{\hat{x}2} \\ f_{\hat{y}1} \\ \hat{M}_2/l \end{bmatrix} \tag{3.8}$$

making the conjugate internal strains to be

$$\underline{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{bmatrix} = \begin{bmatrix} u_{\hat{x}2} - u_{\hat{x}1} \\ u_{\hat{y}1} - u_{\hat{y}2} + l\phi_{\hat{z}1} \\ \left(\phi_{\hat{z}2} - \phi_{\hat{z}1}\right) l \end{bmatrix} \tag{3.9}$$

The constitutive behaviour of the beam can be found by starting with the well known governing equations for a beam in equilibrium, which follow from the Euler-Bernouilli beam theory [6]:

$$\frac{d\hat{u}_x^2}{d\hat{x}^2}(x) = 0 \qquad EI\frac{d^2\hat{u}_y}{d\hat{x}^2}(x) = M(\hat{x}) \qquad \frac{d\hat{u}_y^4}{d\hat{x}^4}(x) = 0 \tag{3.10}$$

The quantities E and I denote the Young's modulus and the moment of inertia of the cross section, $I = \int_A x_2^2 dx_2 dx_3$, of the beam, respectively. By solving these differential equations with the proper boundary conditions, the stiffness matrix $\underline{\hat{S}}$ and stress-strain relations are obtained:

$$\underline{\sigma} = \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \end{bmatrix} = \frac{E}{l^3} \begin{bmatrix} Al^2 & 0 & 0 \\ 0 & 12I & 6I \\ 0 & 6I & 4I \end{bmatrix} \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{bmatrix} \equiv \underline{\hat{S}}\, \underline{\varepsilon} \tag{3.11}$$

where A is the area of a cross-section of the beam. To transform the stiffness matrix of a beam element from the local coordinate system to the global coordinate system, a rotation matrix is

[2] Other choices are possible, but the same results will be obtained.

required:

$$\begin{bmatrix} u_{xi} \\ u_{yi} \\ l\phi_{zi} \end{bmatrix} = \begin{bmatrix} \cos\beta & -\sin\beta & 0 \\ \sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_{\hat{x}i} \\ u_{\hat{y}i} \\ l\phi_{\hat{z}i} \end{bmatrix} \quad i = 1,2 \tag{3.12}$$

with $i = 1, 2$ the two nodes that correspond to the beam element and β the angle between the beam element and the horizontal axis in the global coordinate system, i.e.

$$\sin\beta = (y_2 - y_1)/l\,, \quad \cos\beta = (x_2 - x_1)/l \tag{3.13}$$

where (x_i, y_i) are the node coordinates in the unit cell and l is the node to node length,

$$l = \sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2} \tag{3.14}$$

which should not be confused with the initial length l_0 of the beam. Designating this transformation matrix as $\underline{T}$, the stiffness matrix $\underline{S}$ for a beam element in the global coordinate system can be calculated as

$$\underline{S} = \underline{T}^{\mathsf{T}}\,\hat{\underline{S}}\,\underline{T} \tag{3.15}$$

The stiffness matrices of all beams are combined in one big stiffness matrix $\underline{A}$, corresponding together with a displacement vector $\underline{U}$ and a force vector $\underline{F}$, which contains the displacement respectively force vectors of all nodes:

$$\underline{AU} = \underline{F} \tag{3.16}$$

Additionally, the tying equations and boundary conditions are added to this equation. In every timestep, this procedure is repeated. Using Gaussian elimination to solve this equation, the change in displacements in the time-step can be found. Updating all parameters like energies, forces and geometry of the network results in the simulation of a deforming network.

CHAPTER 4

The representative volume element model

To account for full periodicity of the network, the unit cell is reconstructed into a representative volume element (RVE). An RVE is a small element from a macroscopic body with fully periodic boundaries in all translational directions, but with the same physical properties and responses as that macroscopic body. Therefore it can represent the macrosocopic body, while significantly reducing the time needed for stress-strain calculations. Except for the fully periodic boundaries, it is closely related to the unit cell. However, it allows the implementation of stretching and compression of the network, meaning that the dimensions of the box l_x and l_y are not constant under network deformation. This was not possible with the unit cell model.

4.1 RVE implementation

To construct a fully periodic cell, the top and bottom boundary nodes need to be tied together in a same way that already has been done with the left and right boundary nodes. However, this implies that mechanical loading is not possible anymore by prescribing the boundary nodes, as it would restrict the displacements of those nodes and therefore, contradict with the condition of full periodicity. This problem is solved by introducing three new degrees of freedom in the network model, which will represent the macroscopic normal displacements (U_1 and U_2) and the shear displacement (U_{12}) of the entire network as drawn in figure 4.1. These new variables can be inserted into the network model by rewriting the boundary conditions. Nodes that are on the left and right boundaries were having equal displacement vectors in the unit cell model (equation 3.2). This is extended by allowing the right-hand boundary nodes to have an

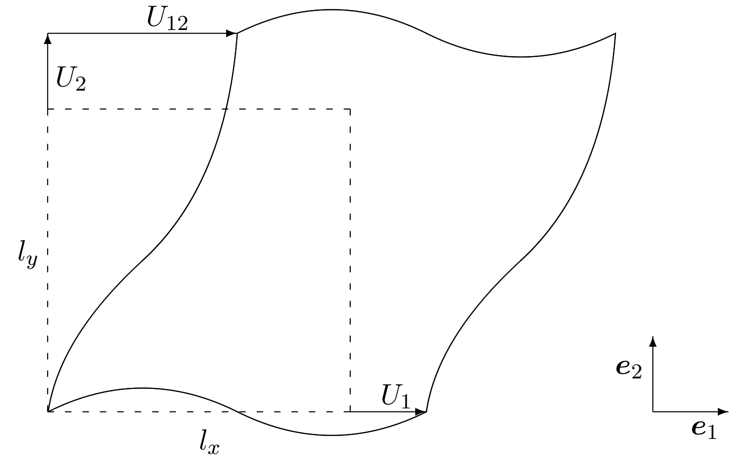


Figure 4.1: Schematic drawing of a deformed representative volume element with its new degrees of freedom.

additional displacement U_1 in the $\boldsymbol{e}_1$ direction with respect to the left boundary nodes:

$$\underline{u}(l_x, y) = \underline{u}(0, y) + U_1 \boldsymbol{e}_1 \tag{4.1}$$

Nodes that are on the top and bottom boundaries were not tied together at all in the unit cell model, but in this model they are. Similar to the left and right boundary nodes, these top boundary nodes have the same displacement vector as the bottom boundary nodes, plus an additional displacement U_2 in the $\boldsymbol{e}_2$ direction. However, in this case the top boundary nodes can also have a displacement in the $\boldsymbol{e}_1$ direction with respect tot the bottom boundary nodes in the shape of a shear displacement U_{12}:

$$\underline{u}(x, l_y) = \underline{u}(x, 0) + U_{12} \boldsymbol{e}_1 + U_2 \boldsymbol{e}_2 \tag{4.2}$$

Because the network is now able to stretch or compress, the dimensions of the cell are subject to change according to

$$l_x = l_x^{(0)} + U_1 \qquad l_y = l_y^{(0)} + U_2 \tag{4.3}$$

where $l_x^{(0)}$ and $l_y^{(0)}$ are the initial box size dimensions.

Since new degrees of freedom have been introduced, the network stiffness matrix has changed both in size and content and therefore needs to be recalculated. This new stiffness matrix can be deduced from the concept of virtual work. For a virtual displacement vector $\delta\underline{u}$, which contains the virtual displacements of all nodes in the network, the internal virtual work δw needed for these displacements is

$$\delta w_i = \underline{\sigma}^{\mathsf{T}} \delta \underline{\varepsilon} \tag{4.4}$$

where $\delta\underline{\varepsilon}$ and $\underline{\sigma}$ denote the virtual strain and stress related to these displacements, which are defined as in equation (3.9) and (3.8) respectively. The principle of virtual work states that the internal work is equal to the external work of the network, the latter being given by

$$\delta w_e = \underline{f}^{\mathsf{T}} \delta \underline{u} \tag{4.5}$$

The new degrees of freedom are introduced in a vector $\underline{Q}$ and are related to the displacements $\underline{u}$ of the nodes according to:

$$e = \underline{D}\underline{u} - \underline{P}\underline{Q} = 0 \tag{4.6}$$

where both $\underline{P}$ and $\underline{D}$ are coupling tensors. The condition (4.6) comes from the nodes being tied stiffly together, because due to periodicity the two nodes are in fact one and the same node. The virtual work component resulting from the strain e can be incorporated by introducing a corresponding multiplier λ. Furthermore, the network strain can be written in terms of the displacement vector by a (differential) matrix $\underline{B}$ as $\underline{\varepsilon} = \underline{B}\underline{u}$. The combination of equations (4.4), (4.5) and (4.6) results into

$$\begin{aligned} \delta w = \underline{F}^{\mathsf{T}} \delta \underline{Q} + \underline{f}^{\mathsf{T}} \delta \underline{u} &= \underline{\sigma}^{\mathsf{T}} \delta \underline{\varepsilon} + \lambda^{\mathsf{T}} \delta e \\ &= \underline{\sigma}^{\mathsf{T}} \underline{B} \delta \underline{u} + \lambda^{\mathsf{T}} \underline{D} \delta \underline{u} - \lambda^{\mathsf{T}} \underline{P} \delta \underline{Q} \end{aligned} \tag{4.7}$$

Since the network only deforms elastically, the multipliers can be written in terms of a stiffness constant times the related strain. By introducing $\underline{\sigma} = \underline{S}\underline{\varepsilon}$ and $\lambda = ke$, equation (4.7) then becomes

$$\begin{aligned} \underline{F}^{\mathsf{T}} \delta \underline{Q} + \underline{f}^{\mathsf{T}} \delta \underline{u} = \underline{u}^{\mathsf{T}} & \left[\underline{B}^{\mathsf{T}} \underline{S} \underline{B} + \underline{D}^{\mathsf{T}} k \underline{D} \right] \delta \underline{u} \\ & - \underline{P}^{\mathsf{T}} \underline{Q}^{\mathsf{T}} k \underline{D} \delta \underline{u} - \underline{u}^{\mathsf{T}} \underline{D}^{\mathsf{T}} k \underline{P} \delta \underline{Q} + \underline{P}^{\mathsf{T}} \underline{Q}^{\mathsf{T}} k \underline{P} \delta \underline{Q} \end{aligned} \tag{4.8}$$

As k is kind of a stiffness constant of a spring between two nodes that are tied together, the condition $e \to 0$ can be accomplished by choosing $k \to \infty$. In the computer model, the value of k is therefore chosen to be very large in order to avoid drifting of nodes which are tied together. Finally, equation (4.8) can be written in a simplified matrix notation:

$$\begin{bmatrix} \underline{B}^{\mathsf{T}} \underline{S} \underline{B} + \underline{D}^{\mathsf{T}} k \underline{D} & -\underline{D}^{\mathsf{T}} k \underline{P} \\ -\underline{P} k \underline{D} & \underline{P}^{\mathsf{T}} k \underline{P} \end{bmatrix} \begin{bmatrix} \underline{u} \\ \boldsymbol{Q} \end{bmatrix} = \begin{bmatrix} \underline{f} \\ \boldsymbol{F} \end{bmatrix} \tag{4.9}$$

where $\boldsymbol{Q}$ and $\boldsymbol{F}$ are composed of

$$\boldsymbol{Q} = \begin{bmatrix} U_1 & U_2 & U_{12} \end{bmatrix}^{\mathsf{T}} = \begin{bmatrix} l_x \varepsilon_1 & l_y \varepsilon_2 & l_y \Gamma \end{bmatrix}^{\mathsf{T}} \tag{4.10a}$$

$$\boldsymbol{F} = \begin{bmatrix} F_1 & F_2 & F_{12} \end{bmatrix}^{\mathsf{T}} = \begin{bmatrix} l_y \sigma_1 & l_x \sigma_2 & l_x \tau \end{bmatrix}^{\mathsf{T}} \tag{4.10b}$$

The submatrix $\underline{B}^{\mathsf{T}} \underline{S} \underline{B}$ is the previous stiffness matrix $\underline{A}$ of all the Euler-Bernoulli beam elements, as derived in section 3.2.

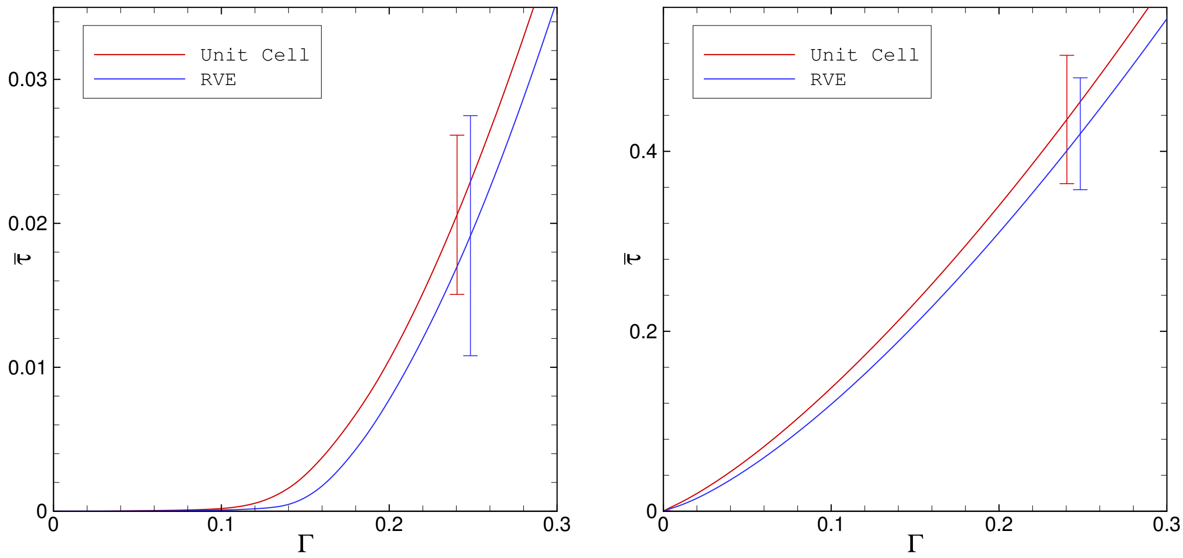


Figure 4.2: Average shear stress response of networks with density $\bar{\rho} = 13$ (left) and $\bar{\rho} = 38$ (right) for both the RVE and unit cell model. Ten different random realizations were used.

4.2 Unit cell response versus RVE response

It is of significant importance to check on differences between the unit cell model and the RVE model. They are best compared with each other when the models have similar properties. The unit cell model has fixed box size dimensions and so, the RVE model should have those as well. This is of course easily achieved by prescribing

$$U_1 = U_2 = 0 \text{ and } U_{12} = l_y\Gamma. \tag{4.11}$$

The network response for both the RVE and unit cell model is plotted in figure 4.2. For simplicity, the results are expressed in the following dimensionless parameters:

$$\bar{\boldsymbol{\sigma}} = \begin{bmatrix} \bar{\sigma}_1 & \bar{\tau} \\ \bar{\tau} & \bar{\sigma}_2 \end{bmatrix} = \frac{L}{EA}\boldsymbol{\sigma} \qquad \bar{\rho} = \rho L \tag{4.12}$$

In the RVE model, the values for σ_2 and τ follow directly from the model, as these values are coupled to the new stretching parameters of the network. However, this is not the case with the unit cell model. In that model, the stresses are found from the sum of the corresponding forces on the top boundary nodes, i.e.

$$\tau = \frac{1}{l_x}\sum_{i\in N_{top}} F_1^{(i)} \qquad \sigma_2 = \frac{1}{l_x}\sum_{i\in N_{top}} F_2^{(i)} \tag{4.13}$$

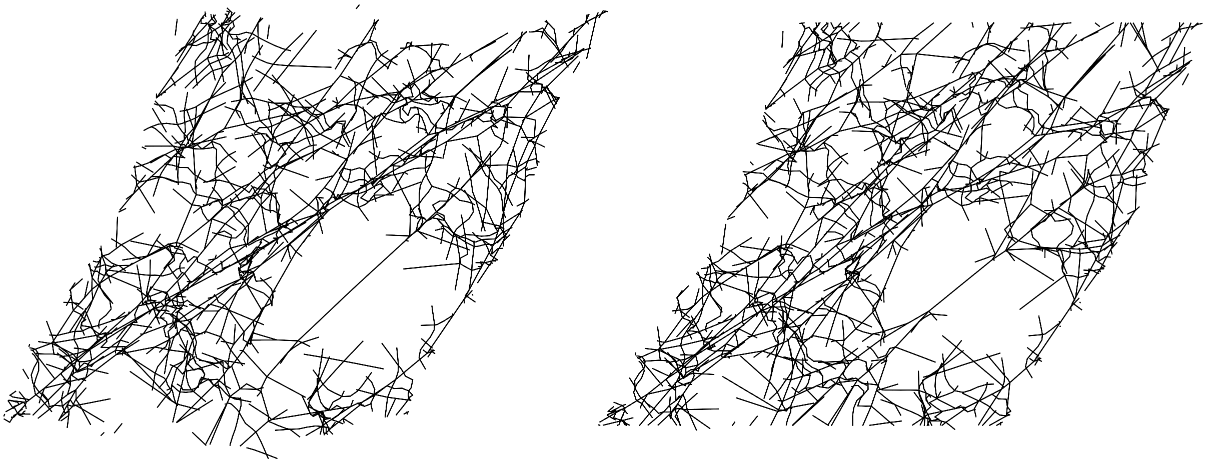


Figure 4.3: Snapshot of the geometry of deformed networks with left, the RVE model and right, the unit cell model. The networks are under an applied shear strain $\Gamma = 0.5$ *and have density* $\bar{\rho} = 13$.

The responses as plotted in figure 4.2 are averages of 10 different random realizations. The stress-strain curves of the two networks differ from each other, because stiffening in the RVE model occurs later than in the unit cell model. However, the difference is only marginally small and in the long run, both networks end up having the same shear stiffness. Stiffening of the networks occurs due to the reorientation of some of the filaments. Because of these orientations, those beams are now axially loaded and, compared to the bending stiffness, the stretching stiffness of a filament is much higher, resulting in a higher overall stiffness. The delayed stiffening in the RVE model is likely to be caused by the model having more freedom at the boundaries. The top and bottom boundaries in the unit cell model are prescribed, but in the RVE model these boundaries can deform and therefore partially adept to forces that act upon it. However, these boundaries play only a minor part in the entire network response, resulting in the difference between the two network responses being small. The relative large error bars indicate that there is a lot of spread among the different network responses. Therefore, any small difference between the two models can be neglected.

Figure 4.3 shows a snapshot of the network geometry for each model with the networks being under an applied shear strain $\Gamma = 0.5$. Before mechanical loading, the network geometries were identical with a line density of $\bar{\rho} = 13$. It is easy to see that the RVE model has a network geometry where the top and bottom boundaries are deformed (left image), while the unit cell model has straight top and bottom boundaries (right image). From the bottom-left to the top-right of the cell, a bundle of filaments under an angle of approximately 45° can be seen in both snapshots. These type of bundels comprise reoriented filaments, which are axial loaded and therefore are the cause for stiffening of these networks. Furthermore, a lot of initially straight

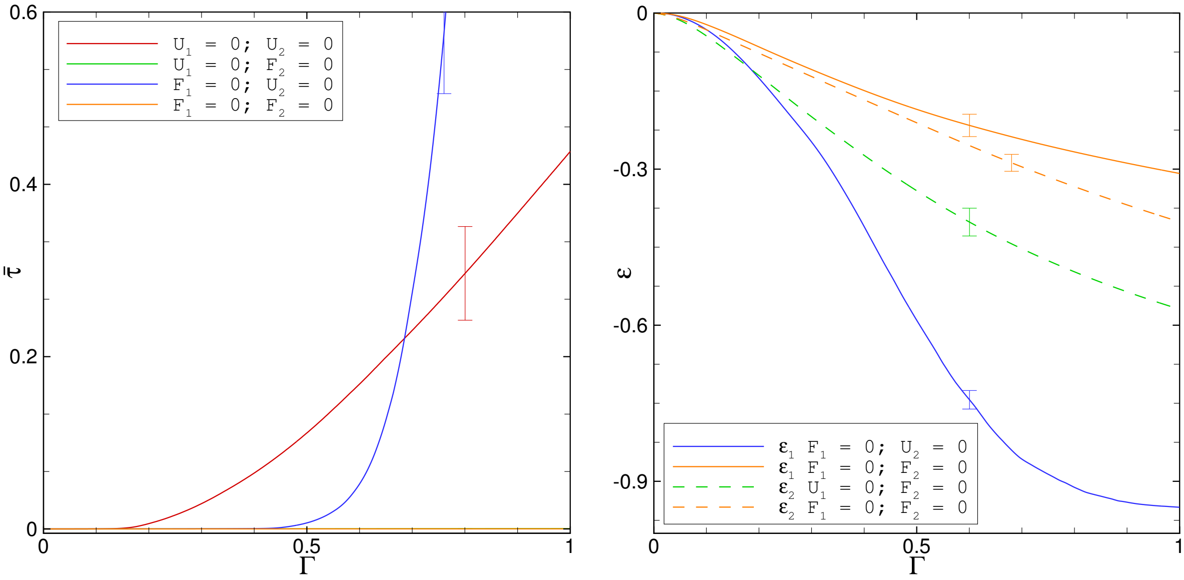


Figure 4.4: Response of networks with density $\bar{\rho} = 13$ and different boundary conditions with left the shear stress and right the normal strains.

filaments have deformed by bending, which corresponds to the low stiffness of the network at small strain levels at these low densities. But at high densities ($\bar{\rho} = 38$), the network stiffens immediately in the small strain region. Because of the higher crosslink density in these high density networks, percolations have already been formed in the initial stress free network. When the network is mechanical loaded, these percolations are immediately axial loaded. All in all, it can be concluded that there are no differences in network response between the unit cell model and the RVE model.

4.3 RVE response under unconstrained shear

In the previous section, the RVE was prescribed with the top and right boundaries fixed relatively to the bottom and left boundaries respectively. This means that the strains are prescribed to be zero as done in equation (4.11). Another kind of boundary condition is to leave the cell free to distort and specify F_i to be zero. By not fixing the boundaries, the network is physically free and can accomodate itself under mechanical loading by either streching or compression. Figure 4.4 shows the network responses under an applied shear strain for four networks with different boundary conditions. The fixed boundaries response (red curves), which equals the unit cell response, serves as a reference for the other networks. The left and right image show the shear stress and the normal strains ε_1 and ε_2 of the networks respectively.

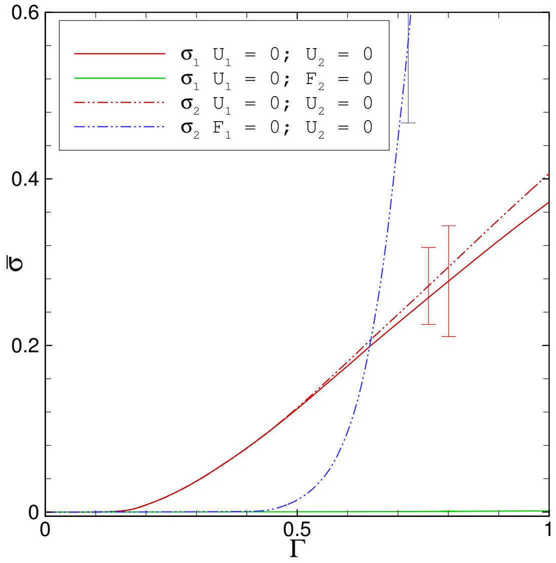


Figure 4.5: Normal stresses response of networks with density $\bar{\rho} = 13$ and different boundary conditions.

For both networks with the top and bottom boundaries free, i.e. $F_2 = 0$, the shear stress is almost zero. This is no surprise as the normal stress σ_2 is closely related to the shear stress of the network. As said previously, the filaments that account for stiffening of the network are under an angle of approximately 45°. The axial force on these filaments can be decomposed into components parallel to and perpendicular on the top and bottom surface, with the components being of approximately the same size. Because the network is prescribed in such a way that F_2 (and of course also σ_2) must equal zero, the network will accomodate itself for any positive normal stress σ_2 (figure 4.5) that acts on the network by compressing itself. As a result, the shear stress is close to zero as well. The network accomodations can be observed in the right image of figure 4.4, where the networks compress between 30% and 50% of their initial box size. Note that due to this compression, the volume of the box reduces and therefore, the line density is actually increasing under mechanical loading.

The network with the left and right boundaries free, but with its top and bottom boundaries fixed, i.e. $F_1 = 0$ and $U_2 = 0$, shows a remarkeble response under the applied shear strain. In the region where the applied shear strain is less than 50%, the shear stress is approximately zero and with that, the stiffness is also very low. In the second region with strains larger than 50%, the network stiffens fast. This results in a high shear stress, even higher than the shear stress of the reference network. The first region of this response can be explained with the amount of freedom the network initially has. So again, the network accomodates itself under an applied shear strain, resulting in the network shear stress almost being zero. The second region can be understood by looking at the network geometries, which are shown in figure 4.6. This figure

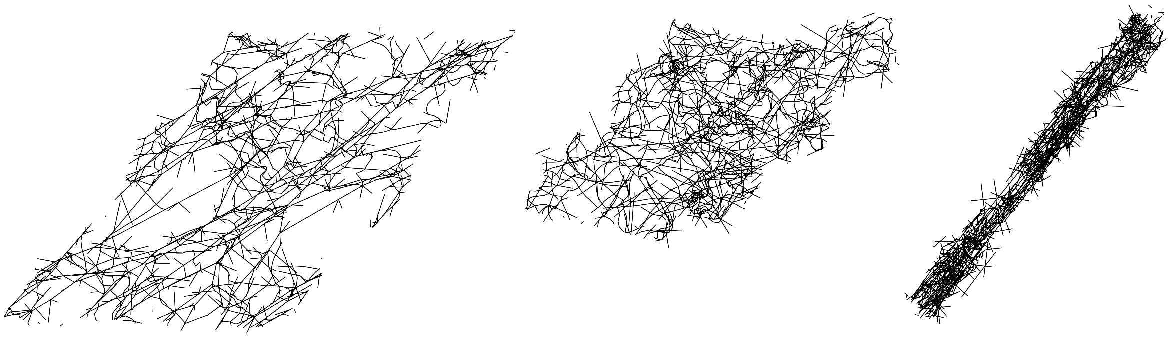


Figure 4.6: Left: $U_1 = U_2 = 0$, middle: $F_1 = F_2 = 0$, right: $F_1 = 0$ $U_2 = 0$. The snapshots are properly scaled and taken under an applied shear strain $\Gamma = 0.8$. The left network has dimensions $40 \times 40 \mu m$.

shows three initially identical networks of density $\bar{\rho} = 13$, which were under an applied shear strain $\Gamma = 0.8$ when the snapshots were taken. The left network is the reference network where the boundaries are completely fixed. The center network has none of its boundaries fixed and compresses in both $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ direction. The right network is the network with the remarkeble response. The latter network compresses with about 90% of its initial length as can also be seen in the right image of figure 4.4. The filaments of the network reorient in such a way, that they form one large bundle of filaments that is now axially loaded. This gives rise to a very large stiffeness.

With higher density networks, the same response has been observed for all four boundary conditions, although stiffening occurs earlier. Concluding this chapter, the stretching of networks in the RVE model seems to be working, but in all cases, the reduction in network volume is far from being realistic and is likely the cause of the ridiculous stress-strain responses of the networks. However, the space between the filaments is empty, which could be the cause for these effects. The next chapter will continue on this.

CHAPTER 5

The RVE network in a liquid cell

The previous chapter showed that the RVE model did work, but that the results were far from being satisfactory. The networks compressed a great amount under mechanical loading, resulting in a much higher or lower stiffness than the reference network. But in the network geometry itself, between the filaments, only empty space can be found. There is no constraint in the model that accounts for volume conservation. However, in *in vitro* experiments, the network is immersed in a liquid. Because a liquid is almost incompressible, it implies volume conservation of the network as a whole. This chapter will continue on the extension of the filamenteous network model, by introducing a liquid cell in the model.

5.1 Implementation of a liquid cell

Initially, the model contained a filamenteous network, which was placed into an empty box with the proper periodicity. Instead of taking an empty box, the model can now be seen as a liquid cell containing the filamenteous network. Any drag forces the liquid acts upon the filaments are neglected as well as the shear stiffness of the liquid, but not the bulk modulus. Therefore, the filaments are not affected by the viscosity of the liquid, so that the liquid and the filamentous network in the cell can be considered as being non-interacting constituents. In a periodic cell, there is no room the liquid can go, so it should accomodate for what the network wants to do. Moreover, the liquid is essentially in rest and therefore it can only have a uniform hydrostatic pressure everywhere, which is determined by the overall volume change ε_v:

$$\sigma_m = \tfrac{1}{2}(\sigma_1 + \sigma_2) = k\varepsilon_v = k(\varepsilon_1 + \varepsilon_2) \tag{5.1}$$

with σ_m the hydrostatic or mean stress of the cell. The internal virtual work in equation (4.4) increases due to the uniform hydrostatic pressure. The additional virtual work $\delta w_{\text{int}}^{\text{liq}}$ is

$$\delta w_{int}^{\text{liq}} = l_x l_y \sigma_m \delta\varepsilon_v = l_x l_y (1-f) k (\varepsilon_1 + \varepsilon_2)(\delta\varepsilon_1 + \delta\varepsilon_2) \tag{5.2}$$

where f is the network volume fraction, defined as the fraction of cell area that is covered by filaments i.e.

$$f = \frac{NLD}{l_x l_y} \tag{5.3}$$

with D the diameter of the filaments. Substituting the definitions for ε_1 and ε_2 in terms of the displacements U_1 and U_2 and rewriting the equation in a matrix notation results into

$$\delta w_{int}^{\text{liq}} = (1-f)k \begin{bmatrix} \delta U_1 & \delta U_2 \end{bmatrix} \begin{bmatrix} \dfrac{l_y}{l_x} & 1 \\ 1 & \dfrac{l_x}{l_y} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} \tag{5.4}$$

This is simply added to the total internal virtual work on the right-hand side of equation (4.7), and thus leads to the stiffness matrix

$$(1-f)k \begin{bmatrix} \dfrac{l_y}{l_x} & 1 \\ 1 & \dfrac{l_x}{l_y} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} \tag{5.5}$$

being added to that of the network.

5.2 Network response with a liquid cell under simple shear

The same situation as in section 4.3 is analyzed, but this time with the liquid cell implemented in the model. Again, the average responses for networks with four different boundary conditions are calculated, each consisting out of ten different random realizations. These responses are plotted in figure 5.1. The left image of figure 5.1 shows the shear stress responses. It is fairly obvious that the shear stresses are identical for all four different boundary conditions and therefore, the shear stress response is independant of how the network boundaries are prescribed. The right-hand image of figure 5.1 shows the normal strains of the networks, which are on average zero (note the large error bars), and it is immediately clear that these are totally different compared to that without liquid. It is also interesting to see that $\varepsilon_2 \approx -\varepsilon_1$, indicating that the liquid is doing its task properly. Because there is no net stretch or compression, the volume of the cell is conserved and with that, the problem of high amounts of network compression as described in the previous chapter is solved, as well the pathological stiffening behaviour of the networks.

Figure 4.5 showed that the normal stress σ_2 of a network with prescribed boundaries is positive and that the axial stiffness $d\sigma_2/d\Gamma$ increases with the applied shear strain. However, a

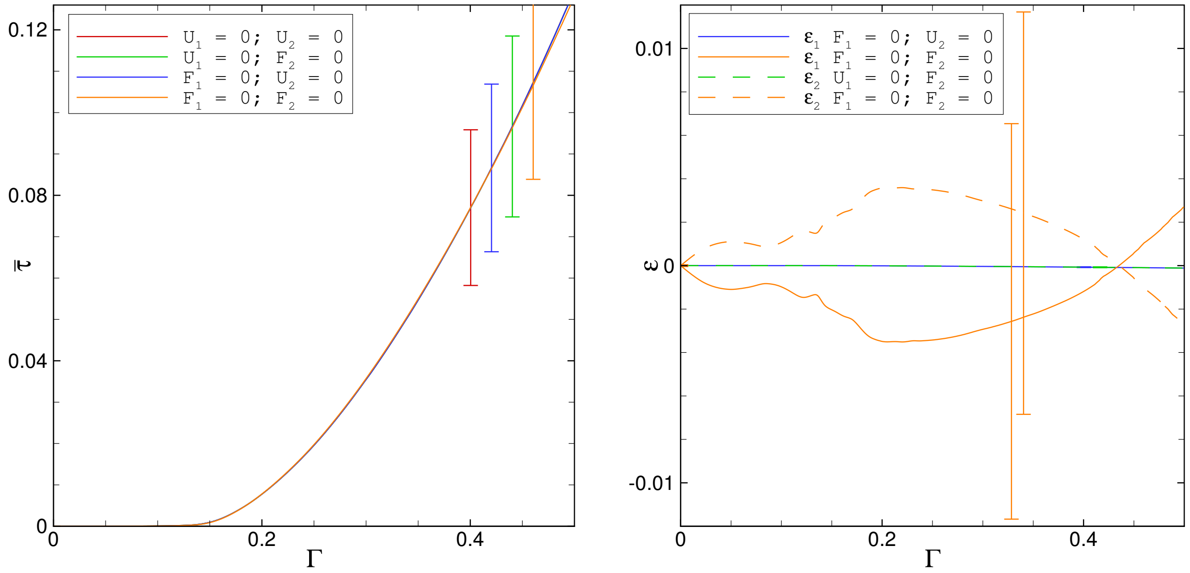


Figure 5.1: Average shear stress (left) and normal strains (right) versus the applied shear strain on networks immersed in a liquid with line density $\bar{\rho} = 13$.

network in a liquid cell with only the top and bottom boundary constrained ($F_1 = 0$ and $U_2 = 0$) has a normal stress which is zero during the whole process of shearing. This response has been plotted together with the response of the reference network ($U_1 = U_2 = 0$) in figure 5.2. But why is this normal stress zero in contrast to the positive stress the reference network has? To satisfy the boundary condition $F_1 = 0$, the network wants to contract in the $\boldsymbol{e}_1$ direction. The liquid in the cell cannot be compressed and wants to stretch the network in the $\boldsymbol{e}_2$ direction in order to conserve its volume. However, the prescription $U_2 = 0$ prohibits the stretching of the cell in the $\boldsymbol{e}_2$ direction and as a result, an external compressive stress σ_1 on the top and bottom boundaries is needed to satisfy this boundary condition. The network itself wants to shrink under simple shear and therefore tension is necessary to keep the boundaries at their place. This is equal to the σ_2 response of the reference network, where the liquid does not play a role. The compressive and tensile stress are approximately of equal magnitude, since $\sigma_2 \approx \sigma_1$ for the network, resulting in a zero net stress on the top and bottom boundaries.

Figure 5.2 also shows the normal stress response for affine behaviour. This behaviour is not calculated with the network model, but from the analytical expression for σ_2 from [12]:

$$\bar{\sigma}_2(\Gamma) = \frac{\bar{\rho}}{EA} \int_0^{\Phi} \lambda(\phi, \Gamma) f_c(\phi, \Gamma) C(\phi, \Gamma) \{\sin^2 \phi - \cos^2 \phi\} d\phi \tag{5.6}$$

with the boundary condition that $\sigma_1 = 0$ and where $\bar{\sigma}_2$ and $\bar{\rho}$ are the same dimensionless paramaters as in equation (4.12). The integration range $0 \leq \Phi \leq \arctan(2/\Gamma)$ is the range

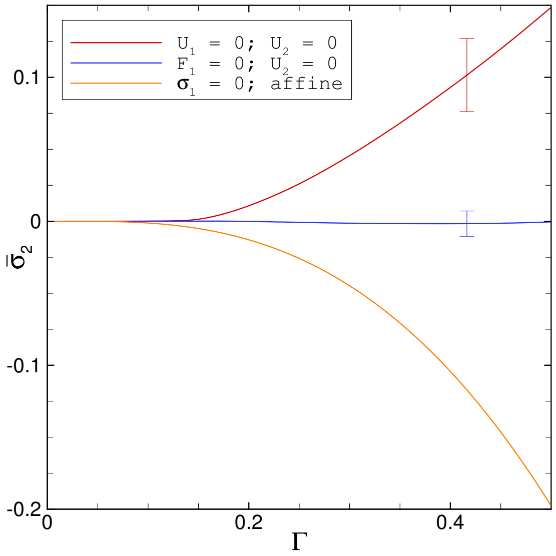


Figure 5.2: A tensile force is found in the simulations if the boundaries are fixed by prescription, but when the liquid must take care of volume conservation, the normal stress is zero. Affine calculations return a compressive stress for which no explanation has been found.

where the filaments are stretched in the affine model. Undulated filaments have almost no contribution and are therefore not taken into account. The function $f_c(\phi, \Gamma)$ is the amount of applied force on a filament oriented under an angle ϕ when the network is under an applied shear strain Γ. This applied force gives rise to a stretch $\lambda(\phi, \Gamma)$ of the filament. The chain orientation distribution function or CODF is given by the function $C(\phi, \Gamma)$ and is the distribution of how the filaments are oriented in the network, normalized by the total number of filaments. As can be seen in figure 5.4, the affine calculation for the normal stress is negative, i.e. a compressive force is acting on the top and bottom boundaries. And although it does show stiffening, it behaves physically completely different from the simulations. In the affine model, the network wants to stretch in the $\boldsymbol{e}_2$ direction, because a compressive force is needed to counter that, but in the discrete network simulations, the network wants to do the opposite.

So, although the shear stresses for networks with different boundary conditions are identical, the normal stresses are not. This is because of the liquid being incompressible and therefore countering the network wanting to compress, resulting in a net normal stress of zero. Networks with their boundaries prescribed have nothing to do with the liquid, because volume conservation is maintained by the boundary conditions itself and the liquid cannot affect the network directly. Analytic calculations on the affine model returns a compressive stress on the network, but no explanation has been found yet for why this is the case.

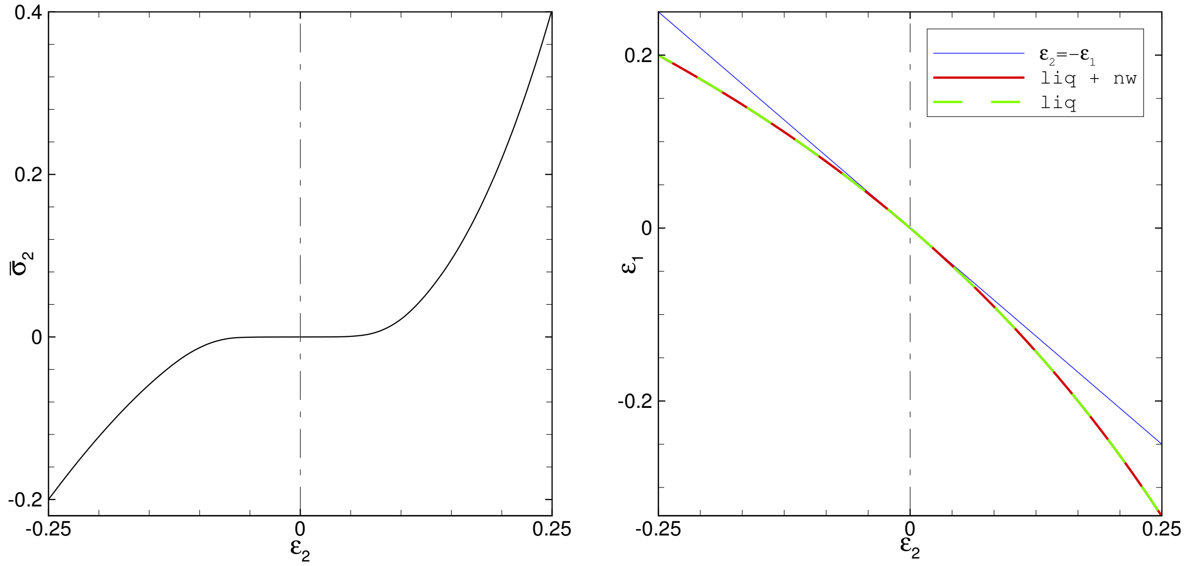


Figure 5.3: Response of a liquid cell with and without filamenteous network under compression and tension. Left: normal stress response. Right: normal strain.

5.3 Tension and compression

Tension and compression simulations are performed by increasing U_2 in time at a constant rate, with $U_{12} = 0$. To allow the network to conserve its volume, the left and right-hand boundaries are kept free, i.e. $F_1 = 0$. The results of these simulations can be found in figure 5.3. The left image shows the normal stress σ_2 which, just as under simple shear, exhibits stiffening. The right image shows the relation between the strains ε_1 and ε_2 for a liquid cell with and without a network. There is no difference between these two, indicating that the liquid cell dominates the network behaviour. But as can be seen in figure 5.4, the network filaments both buckle and reorient. The left snapshot in this figure is the network geometry when the network is in its initially stress free configuration. When a small strain is applied, the filaments buckle, characterizing the low stiffness of the network. When the applied strain is higher, filaments start reorientating in either the $\boldsymbol{e}_2$ direction when under tension or the $\boldsymbol{e}_1$ direction when under compression. These filaments are axially loaded and are the reason of the σ_2 behaviour. Note that the liquid cell itself does not contribute to the normal stress, because the cell only changes shape and does not contain any elastic energy. As the affine model would predict, no shear stress has been observed with both compression and tension simulations.

The straight thin solid line in 5.3, indicates $\varepsilon_2 = -\varepsilon_1$ or in other words where the overall Poisson ratio $\nu = -1$. The cell response deviates from this line, with sometimes even an absolute

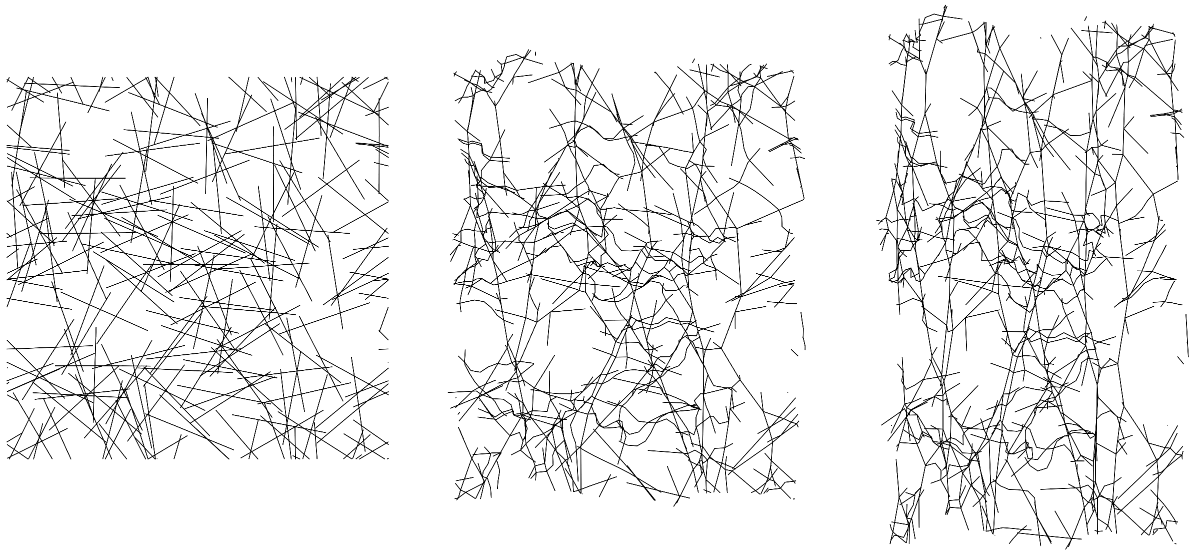


Figure 5.4: Geometries of a network under tension. Left: the initial network geometry. Middle: the network geometry under an applied tensile strain of $\varepsilon_2 = 0.11$*. Right: the network in its final state with* $\varepsilon_2 = 0.25$*.*

Poisson ratio $|\nu|$ that is larger than one. Therefore, it looks like if the cell is not conserving its volume, but this is not true. It has specifically been checked that the volume of the cell is really conserved in the simulations. The non-linear relation between ε_1 and ε_2 can be entirely related to how these strains are defined. For each timestep the strains $d\varepsilon_1$ and $d\varepsilon_2$ are calculated:

$$d\varepsilon_1 = \frac{dU_1}{l_x} \quad \text{and} \quad d\varepsilon_2 = \frac{dU_2}{l_y} \tag{5.7}$$

where dU_1 and dU_2 are the changes in the values of U_1 and U_2 in a single timestep. To find the accumulated strains ε_1 and ε_2 of the network, these relations are integrated, while using the definition for the box size dimensions as in equation (4.3). The result is

$$\varepsilon_1 = \ln\left(\frac{U_1}{l_x^{(0)}} + 1\right) \quad \text{and} \quad \varepsilon_2 = \ln\left(\frac{U_2}{l_y^{(0)}} + 1\right) \tag{5.8}$$

which is a logarithmic defined strain.

Ending this chapter, the most important thing in these tension and compression simulations is that the shape of the box is determined by the liquid cell. Therefore, the network does deform and forms bundles of filaments which stiffens the network. This will have an effect when a pre-strain has been applied on a network that is being deformed under simple shear. The next chapter will look into that.

CHAPTER 6

The effect of pre-strain on the network shear response

With the liquid cell implemented, the RVE model is now working properly. Tension, compression and simple shear simulations have been performed with this model and satisfying results have been found. This chapter will look into a more complex type of mechanical loading. In experiments, the gel is sometimes put under compression before simple shear is applied. To understand what the effect of such a pre-strain is on the network response under simple shear and to explain why this is, some new network quantities are introduced first.

6.1 New network quantities

6.1.1 Discrete orientation distribution function

In equation (5.6), the chain orientation distribution function $C(\phi, \Gamma)$ or CODF was used in the analytic calculations of the network stress in the affine model. To characterize the filament orientation distribution in the simultations, a discrete orientation distribution function $D(\phi, \Gamma)$ or DODF is computed. This is a discrete distribution of the filament orientation in the network geometry, i.e. the number of elements in the network that have an orientation between ϕ and $\phi + \Delta\phi$ at a certain value of Γ:

$$D(\phi, \Gamma) = \frac{1}{2\pi A} \sum_{\alpha} n_\alpha(\Gamma) \qquad \{n_\alpha \in \mathbb{A} : \phi \leq \alpha < \phi + \Delta\phi\} \tag{6.1}$$

with $\mathbb{A}$ the set of A network elements and $0 \leq \phi < \pi$, since the DODF is a π-periodic function,

$$D(\phi + \pi, \Gamma) = D(\phi, \Gamma) \tag{6.2}$$

The DODF is a distribution, i.e. a normalized function, meaning that the total area under the distribution function for the discrete intervals sums up to 1

$$\sum_{\phi=0}^{2\pi} D(\phi, \Gamma)\Delta\phi = 1 \tag{6.3}$$

6.1.2 Slack orientation function

Slack is a quantity that represents the amount of undulations in a network. Here it has been decided to look at the slack in filament sections between two crosslinks instead of the slack in whole filaments. The amount of slack in a filament section between two crosslinks is defined as

$$s_\alpha = \frac{l_c - r}{l_c} \tag{6.4}$$

where l_c is the contour length of the filament section and r the end-to-end distance between the two crosslinks. The total amount of slack is found to increase when the network is being deformed. At first sight, this is counter-intuitive since the network stiffens at the same time, but only a few filaments take part in stiffening by being axially loaded. The filaments that are not axially loaded easily buckle, so that the amount of slack in a filament section depends on its orientation. To avoid such confusion, the amount of slack is discretized over a number of angle intervals. Not only is the amount of slack visible this way, but also under which angle this amount of slack occurs. The numerical function that calculates this quantity is called the slack orientation function $S(\phi, \Gamma)$ or SOF. The function is of course not normalized as that would mean that information about the total amount of slack in a network is lost. The SOF is somewhat similar to the DODF and can be defined as

$$S(\phi, \Gamma) = \sum s_\alpha(\Gamma) \qquad \left\{ s_\alpha \in \widetilde{A} : \phi \leq \alpha < \phi + \Delta\phi \right\} \tag{6.5}$$

where $\widetilde{A}$ is the set of filament sections in the entire network and s_α is the amount of slack in a single filament section as defined in equation (6.4). Just as the DODF, the slack orientation function is a π-periodic function. The total amount of slack in the network, $\bar{S}$, is the area under the slack orientation function between 0 and π

$$\bar{S}(\Gamma) = \sum_{\phi=0}^{\pi} S(\phi, \Gamma)\Delta\phi \tag{6.6}$$

6.1.3 Principal directions

To see in which direction the filaments are maximally stretched, the principal directions are calculated for affine behaviour and then compared with the simulations. In a coordinate sytem with the principal directions as base vectors, the shear strain vanishes by definition. To find the principal directions,we start from the components of the deformation gradient tensor $\boldsymbol{F}$ for a continuum under simple shear:

$$F_{ij} = \begin{bmatrix} 1 & \Gamma \\ 0 & 1 \end{bmatrix} \tag{6.7}$$

Since the $\det(\boldsymbol{F}) > 0$ on physical grounds, the polar decomposition theorem says that $\boldsymbol{F}$ can be uniquely decomposed into

$$\boldsymbol{F} = \boldsymbol{R} \cdot \boldsymbol{U} = \boldsymbol{V} \cdot \boldsymbol{R} \tag{6.8}$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ are the right and left stretch tensor respectively and $\boldsymbol{R}$ is a proper orthogonal rotation matrix. Both stretch tensors have the same principal values λ_i, but the corresponding principal directions $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ differ by a rotation

$$\boldsymbol{v}_i = \boldsymbol{R} \cdot \boldsymbol{u}_i \tag{6.9}$$

Using the Cauchy-Green tensor $\boldsymbol{U}^2 = \boldsymbol{F}^\intercal \cdot \boldsymbol{F}$, the principal values can be found by finding the eigenvalues of $\boldsymbol{U}^2$, which are equal to λ_i^2. Taking the square root, we obtain

$$\lambda_1 = \frac{\Gamma}{2} + \sqrt{\left(\frac{\Gamma}{2}\right)^2 + 1} \quad \wedge \quad \lambda_2 = -\frac{\Gamma}{2} + \sqrt{\left(\frac{\Gamma}{2}\right)^2 + 1} \tag{6.10}$$

with the corresponding eigenvectors

$$\boldsymbol{u}_1 = \left(1 + \lambda_1^2\right)^{-\frac{1}{2}} \begin{bmatrix} 1 \\ \lambda_1 \end{bmatrix} \quad \wedge \quad \boldsymbol{u}_2 = \left(1 + \lambda_1^2\right)^{-\frac{1}{2}} \begin{bmatrix} -\lambda_1 \\ 1 \end{bmatrix} \tag{6.11}$$

which are the principal stretch directions in the undeformed configuration. The principal directions $\boldsymbol{v}_i$ in the deformed configuration are computed from (6.9). The rotation tensor can be calculated by using a similarity transformation for $\boldsymbol{U}$ in terms of its eigenvectors and eigenvalues, leading to

$$\boldsymbol{R} = \boldsymbol{F} \cdot \boldsymbol{U}^{-1} = \boldsymbol{F} \cdot \left(\boldsymbol{Q} \cdot \boldsymbol{D} \cdot \boldsymbol{Q}^{-1}\right)^{-1} = \boldsymbol{F} \cdot \boldsymbol{Q} \cdot \boldsymbol{D}^{-1} \cdot \boldsymbol{Q}^\intercal \tag{6.12}$$

where

$$\boldsymbol{D} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \quad \text{and} \quad \boldsymbol{Q} = \begin{bmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 \end{bmatrix} \tag{6.13}$$

Applying this rotation on the principal directions in equation (6.11) and calculating the new directions for an applied shear strain $\Gamma = 0.3$ results in

$$\boldsymbol{v}_1 = \begin{bmatrix} 0.758 \\ 0.653 \end{bmatrix} \qquad \boldsymbol{v}_2 = \begin{bmatrix} -0.653 \\ 0.758 \end{bmatrix} \tag{6.14}$$

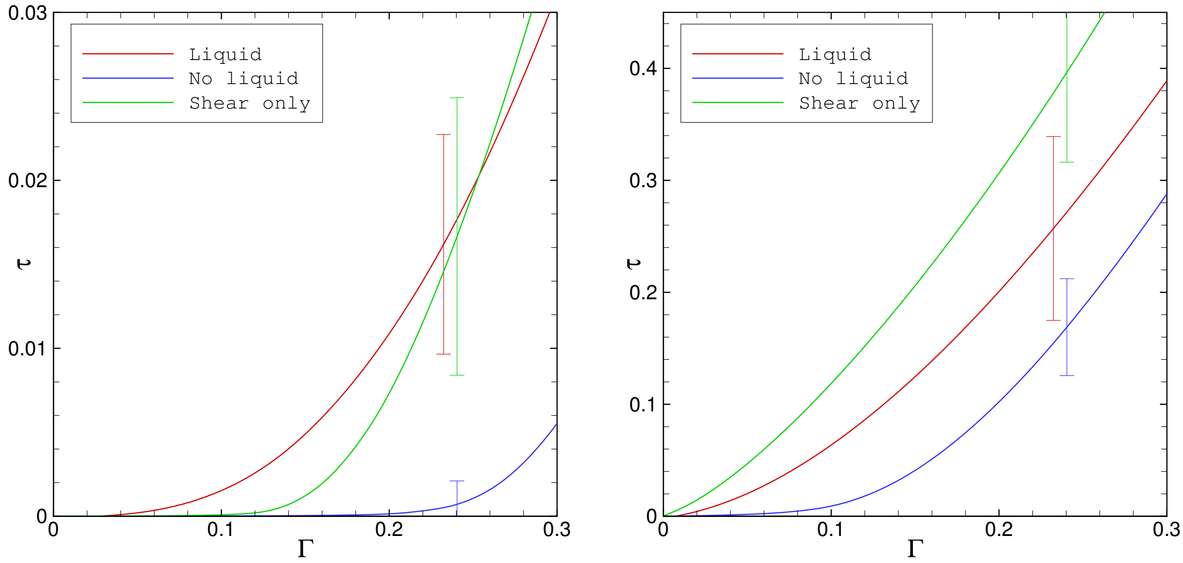


Figure 6.1: Shear response for liquid and non-liquid cell networks, both having been pre-strained in compression to $\varepsilon_2 = -0.1$ and a reference network, which has been sheared only. The left and right image shows the response for low ($\bar{\rho} = 13$) and high ($\bar{\rho} = 38$) density networks, respectively.

6.2 Pre-strain results

To understand the effect of a pre-strain on a network, three types of simulations are performed. Two networks, one with and one without a liquid cell, are pre-strained with a compressive strain of 10% in exactly the same way as done in section 5.3, followed by shearing. In the process of shearing, the network boundaries are fixed, otherwise the network will accomodate for the applied pre-strain by deforming to its original square shape. A third network is sheared without any pre-strain applied on it, and serves as a reference network for the two pre-strained networks. Results are shown in figure 6.1, where the left and right image show the shear stress response for low density ($\bar{\rho} = 13$) and high density ($\bar{\rho} = 38$) networks respectively. To make sure that the DODF and SOF will have a statistical meaning, the average of at least 40 different random realizations is taken. Note that under pre-strain, the liquid cell network will stretch in the $\boldsymbol{e}_1$ direction to conserve the cell volume, while the non-liquid cell network does not stretch. For both densities, the non-liquid cell network has a slower stiffening response than the reference network. The liquid cell network however does stiffen earlier than the reference network for a low density, but this is not the case for high density networks. In the end, the prestrained liquid cell network has a lower shear stiffness than the reference network for both densities. To understand why these differences in network behaviour occurs, the quantities SOF and DODF and the network geometries are being studied.

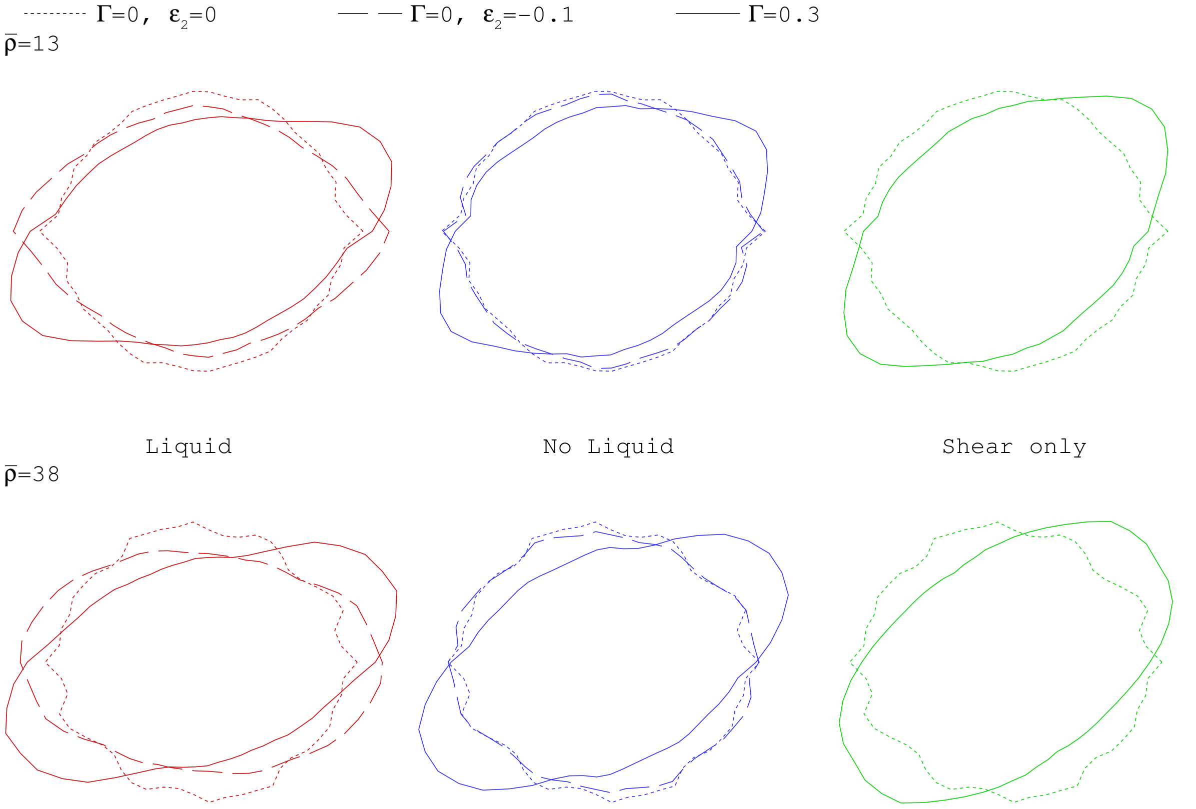


Figure 6.2: The discrete orientation distribution function of all three networks for both low and high densities.

The discrete orientation distribution functions for both low and high densities are plotted in figure 6.2. The initial filament orientation is rather randomly distributed, as seen from the circular shape of the distribution function. This means that there are approximately an equal number of filaments in all orientation intervals. During shearing, filaments rotate towards the shear direction resulting in more filaments becoming axially loaded, thus giving the network its increased stiffness. The main difference between the two pre-strained networks can be found just after the pre-strain has been applied. Filaments in the liquid cell network orient themselves in the $\boldsymbol{e}_1$ direction, because the network stretches in that direction to conserve its volume. These filaments form bundles as well, which gives the network a higher stiffness than it initially had. For low densities, this explains the initial higher shear stiffness of the liquid cell network compared to the reference network seen in Fig. 6.1, but it does not explain the response for high densities. Moreover, the liquid cell initially may have a higher shear stiffness than the reference network, it does not stay this way. Since the filaments have already formed bundles under network pre-strain, the liquid cell network has less freedom than the reference network

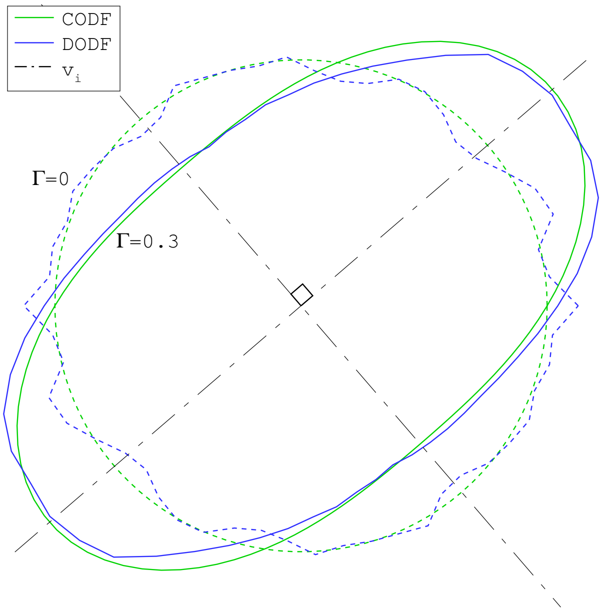


Figure 6.3: The CODF *compared with the* DODF *for both* $\Gamma = 0$ *(dotted) and* $\Gamma = 0.3$ *(solid) for a network with density* $\bar{\rho} = 38$ *under simple shear.*

to reorient filaments in the shear direction when a shear strain is applied. This finally results in the liquid cell having a lower stiffness than the reference network without prestrain. For low densities, this can be seen by the crossing of the stress–strain curves in figure 6.1. At high densities, the reference network stiffens immediately due to the many crosslinks in the network. But now, the liquid cell has a lower stiffness, because the filaments have become horizontally oriented, and not in the shear direction. Figure 6.2 also shows that the non-liquid cell does not have any significant re-orientations during pre-straining. Because no reorientations takes place it will have the same stiffness as the reference network in the end. However, the slower response cannot be explained by the DODF.

Figure 6.3 shows both the chain orientation distribution function for the affine model as well as the discrete orientation distribution function calculated in the simulation. The dotted lines show the inital distribution, while the solid lines are the orientation distributions when the network is under an applied shear strain of $\Gamma = 0.3$. No pre-strain has been applied and the networks have density $\bar{\rho} = 38$. It is quite clear that the DODF is quite similar to the CODF. Although there are some small differences between the two, the statistical spread in the DODF makes it unclear if there really is a difference between the two distributions or that it is only related to the statistics of the DODF. The conclusion that can be drawn is that the filament orientation in the simulations show a similar distribution as the affine model for this shear strain and network density. However, it might be possible that the orientation distribution function

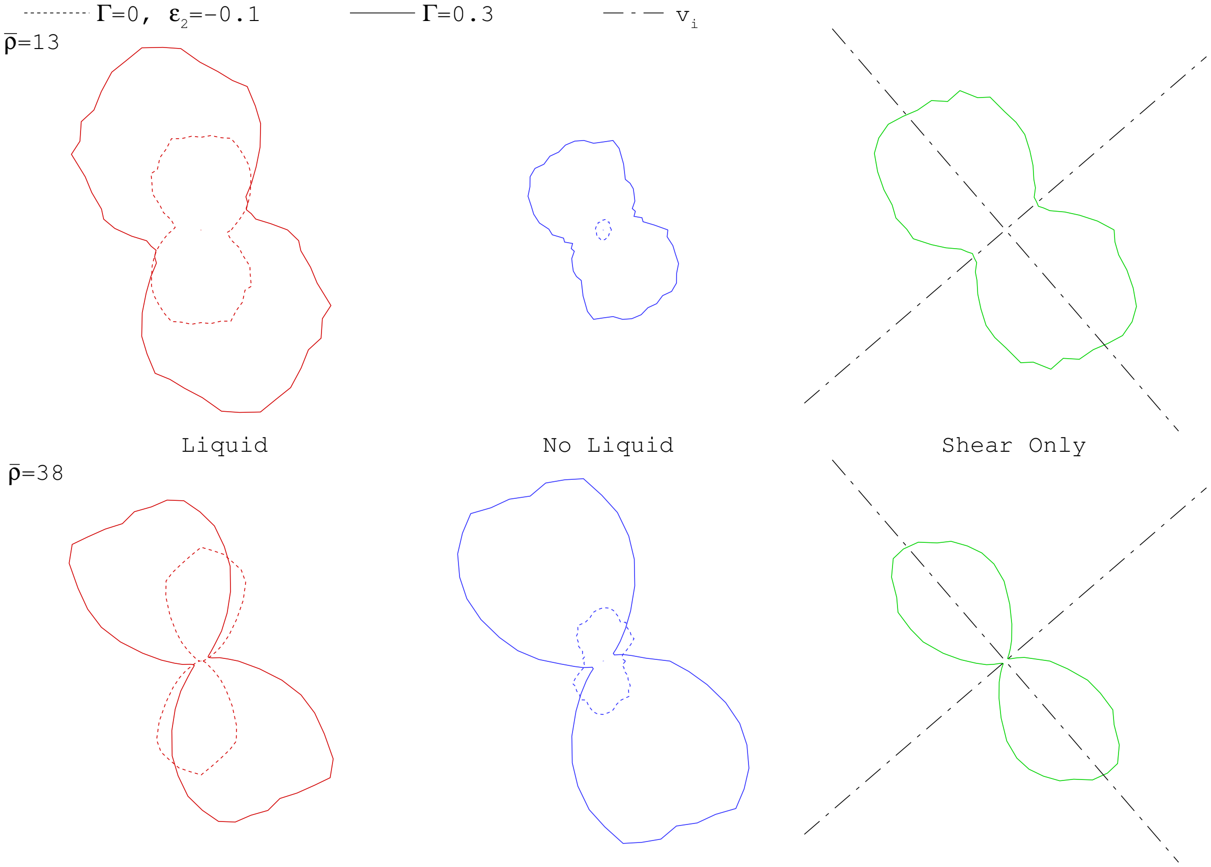


Figure 6.4: The slack orientation function of all three networks for both low and high densities. The plots are scaled properly, meaning that the area of the propellor-shaped curve is the total amount of slack in the network.

differs significant in low density networks for (very) small shear strains, but this has not been investigated yet.

The slack orientation function is plotted in figure 6.4, again for both densities. Initially, there is no slack, because the network is generated with straight filaments only. Under pre-strain, slack is generated in both the liquid and non-liquid cell networks. Most slack is generated in the $\boldsymbol{e}_2$ direction, which is the direction of the compression. Because stiffening occurs in the liquid cell network, but not in the non-liquid cell network, one might have expected that the amount of slack in the non-liquid cell network is greater than in the liquid cell network, but the opposite is true. Figure 6.5 shows the network geometries after pre-strain, but before the shear strain is applied. The boxes are a magnification of a small area of network, where an undulated filament (approximately at an angle of 45°) can be seen. In the non-liquid cell network this filament has a slack of about 10%, which is the same amount of compressive strain applied. In the liquid cell network the filament is much more undulated, because due to orientations, bundles of filaments

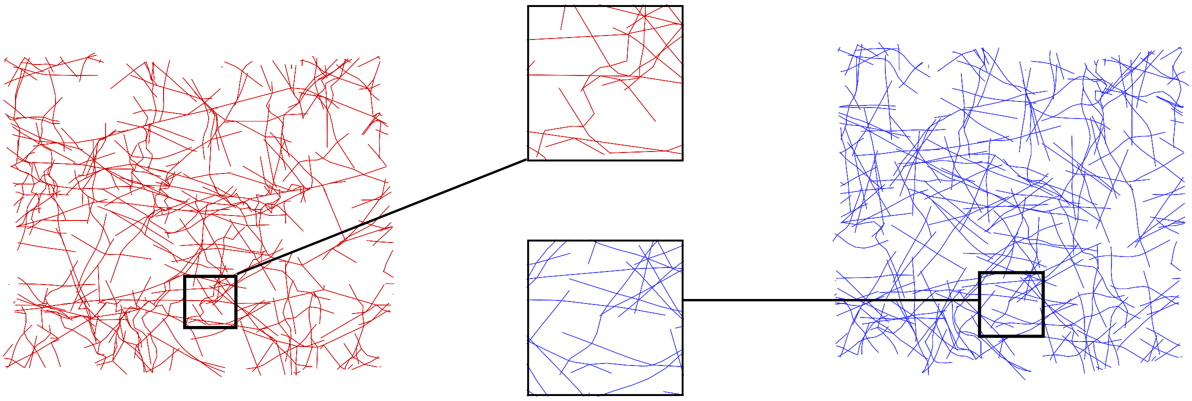


Figure 6.5: Geometries of the two pre-strained networks with low density just before a shear strain is applied. The liquid cell network and non-liquid cell network is plotted left and right respectively. The boxes are magnifications of undulated filaments in the networks.

are formed with highly undulated filaments in between. Because no reorientations occurs in the non-liquid cell network, the network deformes uniformly (affine), resulting in the compressive strain transfered directly into the filaments. The slack in the liquid-cell network therefore is much larger.

For low density networks, there is a significant amount of slack in the direction where filaments are axially loaded. However, the network is still in the bending–tension transition phase; hence not all of the slack in that direction has been pulled out yet, but this will happen eventually when the applied shear strain is increased further. High density networks do not have any slack in this direction, because the network is already past the transition region for $\Gamma = 0.3$. Stiffening in high density networks occurs earlier and therefore, the transition region can be found for lower strains.

In contrast to the reference network which starts with straight filaments, the non-liquid cell network under pre-strain does have slack. When a shear strain is applied, this slack needs to be pulled out of the network first, before it can stiffen. Because the orientation distributions of the filaments are equal for both networks, the slower stiffening response of the non-liquid cell network can be entirely attributed to the slack, for both high and low densities.

In figures 6.3 and 6.4, two dash-dotted lines have been drawn with index v_i. These are the principal directions as calculated in equation (6.14) for $\Gamma = 0.3$, which make an angle with the horizontal axis of 41° and −49° and intersect each other perpendicular. The DODF has the most and least filaments oriented in the same direction as the 41° respectively −49° principal axis, which was to be expected as it already has been shown that the DODF shows affine behaviour. For the SOF, similar conclusions can be drawn. As can be seen in figure 6.4 (especially for high densities), the least and most slack lie in the same direction as the two principal axis.

Concluding this chapter, a compressive pre-strain followed by a shear strain will give a different response compared to a network which is sheared only. It has been shown that in a liquid cell network reorientations already take place in the pre-strain phase, resulting in an initial higher stiffness for low density networks, but evolving towards a lower network stiffness than a network that is not pre-compressed. No re-orienations take place in a non-liquid cell network, making the slack in the network responsible for the slower stiffening response. It has also been shown that the orientation distribution in the simulations of higher density networks is similar to that in the affine model.

CHAPTER 7

Conclusions and recommendations

This chapter will give a summary of the conclusions that have been drawn in this thesis. Furthermore, a few recommendations are made on what kind of future research could be done with the two-dimensional network model.

7.1 Conclusions

- No significant difference has been found in the network response under simple shear between the unit cell model used in [9] and the RVE model.
- After the introduction of a liquid cell in the network model, the shear stress response is independent of how the boundaries are prescribed. However, this is not true for the normal stresses. When the boundaries are fixed, an external tensile force is needed to keep thse boundaries at their place, because the network wants to compress. But when the boundaries are not fixed, the liquid cell prevents the network from compressing and no external force is needed.
- For compression and tension simulations, the shape of the cell is determined by the liquid only. The network itself, however, does show the formation of filament bundles, resulting in stiffening of the network.
- A pre-strained non-liquid cell network exhibits slower stiffening than a network that has been sheared only, since slack is generated in the network during pre-straining and this

has to be pulled out under shear before stiffening can take place. A pre-strained liquid cell network has a final shear stiffness that is lower than that of a network that has only been sheared. However, for low density networks, reorientations of filaments occur under pre-strain, resulting in an initially higher stiffness than the networks without pre-strain.

- It has been shown that the filament orientation in a network under simple shear is distributed similar to that in the affine model. Although it has not directly been shown that the same holds for the slack orientation, the principal directions as well as the integration boundaries indicate that the slack orientation function shows affine behaviour.

7.2 Recommendations

- The normal stress induced by shear according to the affine model is compressive, while the simulations give a tensile stress. Why does this opposite behaviour occur? Is it the non-affinity of the network, or are there assumption in the affine model that do not work out correctly?

- Only the influence of a compressive pre-strain has been investigated, but what will be the influence of a tensile pre-strain?

- All the simulations in this thesis have been performed with initially straight filaments only, i.e. a persistance length of $l_p \to \infty$. It has been shown that for simple shear that undulated filaments ($l_p = 1$) only delay the stiffening of the network [9], but does the same hold for compression, tension or pre-strain simulations as done in this thesis?

- It has been shown that for high filament densities or relatively high shear strains, the orientation distribution function is near affine behaviour, but this might not be the case in low density networks for small strains. It is worth investigating this further in view of claims made in [2].

- Another quantity that can be used to characterize a network is the amount of filaments that are being stretched. This was done by using the definition for the SOF in equation (6.5), but by redefining equation (6.4) to

$$s_\alpha = \left(\frac{r}{l_c}\right)^n \tag{7.1}$$

where n is a large number. Due to the lack of time, no satisfactory results have been found yet.

Bibliography

[1] J.F. Besseling. *Stijfheid en sterkte 2 - Toepassingen.* Oosthoek Scheltema and Holkema, 1975.

[2] D.A. Head, A.J. Levine, and F.C. MacKintosh. Distinct regimes of elastic response and deformation modes of cross-linked cytoskeletal and semiflexible polymer networks. *Phys. Rev. E.*, 68(061907), 2003.

[3] P.A. Janmey and D.A. Weitz. Dealing with mechanics: mechanisms of force transduction in cells. *TRENDS in Biochemical Sciences*, 29, July 2004.

[4] T. Koeman. Formation and deformation of cytoskeletal networks. Master's thesis, University of Groningen, 2005.

[5] H. Lodish, A. Berk, S.L. Zipursky, P. Matsudaire, D. Baltimore, and J. Darnell. *Molecular Cell Biology.* W.H. Freeman and Company, fourth edition, 2000. isbn:0716731363.

[6] Daryl L. Logan. *A first course in the finite element method using Algor.* Brooks/Cole, second edition, 2001.

[7] F.C. MacKintosh and P.A. Janmey. Actin gels. *Solid State and Materials Science*, 2:350–357, 1997.

[8] F.C. MacKintosh, J. Käs, and P.A. Janmey. Elasticity of semiflexible biopolymer networks. *arXiv:cond-mat*, 1(9507093), 1995.

[9] P.R. Onck, T. Koeman, T. van Dillen, and E. van der Giessen. Alternative explanation of stiffening in cross-linked semiflexible networks. *Phys. Rev. Lett.*, 95(178102), 2005.

[10] J.V. Small, K. Rottner, P. Hahne, and K.I. Anderson. Visualising the actin cytoskeleton. *Microscopy research and technique*, 47:3–47, 1999.

[11] C. Storm, J.J. Pastore, F.C. MacKintosh, T.C. Lubensky, and P.A. Janmey. Nonlinear elasticity in biological cells. *Nature*, 435:191–194, May 2005.

[12] T. van Dillen, P.R. Onck, T. Koeman, and E. van der Giessen. Stiffening in cross-linked semi-flexible biopolymer networks under simple shear. *not yet published*, 2006.

[13] N. Wang and D.E. Ingber. Probing transmembrane mechanical coupling and cytomechanics using magnetic twisting cytometry. *Biochemical cell biology*, 73:327–335, 1995.

[14] J. Wilhelm and E. Frey. Elasticity of stiff polymer networks. *Phys. Rev. Lett.*, 91(108103), 2003.

[15] H. Xie. Shear deformation of cytoskeletal actin networks. Master's thesis, University of Groningen, 2004.

APPENDIX A

Fortran code changes

This appendix will give a short overview of the major changes that have been applied to the Fortran code of *cytoskeleton2D*. This may be useful for the group members that want to continue on this subject or it might be used in the three dimensional network model [4].

actinfilament.f: The double array *icontie* is generated in this file. This is one of the most important changes in the program. The initial variable had only two columns, but it is expanded to three columns, with the third used for coupling with the new network stretching parameters. It is also important to realize which node is placed in which column. Previously this was not taken into account, because the degrees of freedom were the same (eq (3.2)), but now it is important that the left and bottom boundary nodes are placed in the first column and the corresponding right and bottom boundary nodes in the second. The network variables U_1, U_2 and U_{12} are placed at the end of the displacement vector at locations *neqs-2*, *neqs-1* and *neqs* respectively, where 'neqs' is the number of equations. Because the number of equations are not yet known, dummy variables are used to indicate coupling with the boundary nodes.

```
! left (4) boundary
icontie(1, ntie + 1) = ndof * (nnod + 1 − 1) + 1
icontie(2, ntie + 1) = ndof * (nnod + 2 − 1) + 1
icontie(3, ntie + 1) = −1
icontie(1, ntie + 2) = ndof * (nnod + 1 − 1) + 2
icontie(2, ntie + 2) = ndof * (nnod + 2 − 1) + 2
icontie(3, ntie + 2) = 0
icontie(1, ntie + 3) = ndof * (nnod + 1 − 1) + 3
icontie(2, ntie + 3) = ndof * (nnod + 2 − 1) + 3
icontie(3, ntie + 3) = 0
! right (2) boundary
```

```
icontie(1, ntie + 1) = ndof * (nnod + 2 - 1) + 1
icontie(2, ntie + 1) = ndof * (nnod + 1 - 1) + 1
icontie(3, ntie + 1) = -1
icontie(1, ntie + 2) = ndof * (nnod + 2 - 1) + 2
icontie(2, ntie + 2) = ndof * (nnod + 1 - 1) + 2
icontie(3, ntie + 2) = 0
icontie(1, ntie + 3) = ndof * (nnod + 2 - 1) + 3
icontie(2, ntie + 3) = ndof * (nnod + 1 - 1) + 3
icontie(3, ntie + 3) = 0
! bottom (1) boundary
icontie(1, ntie + 1) = ndof * (nnod + 1 - 1) + 1
icontie(2, ntie + 1) = ndof * (nnod + 2 - 1) + 1
icontie(3, ntie + 1) =  -3
icontie(1, ntie + 2) = ndof * (nnod + 1 - 1) + 2
icontie(2, ntie + 2) = ndof * (nnod + 2 - 1) + 2
icontie(3, ntie + 2) =  -2
icontie(1, ntie + 3) = ndof * (nnod + 1 - 1) + 3
icontie(2, ntie + 3) = ndof * (nnod + 2 - 1) + 3
icontie(3, ntie + 3) = 0
! top (3) boundary
icontie(1, ntie + 1) = ndof * (nnod + 2 - 1) + 1
icontie(2, ntie + 1) = ndof * (nnod + 1 - 1) + 1
icontie(3, ntie + 1) =  -3
icontie(1, ntie + 2) = ndof * (nnod + 2 - 1) + 2
icontie(2, ntie + 2) = ndof * (nnod + 1 - 1) + 2
icontie(3, ntie + 2) =  -2
icontie(1, ntie + 3) = ndof * (nnod + 2 - 1) + 3
icontie(2, ntie + 3) = ndof * (nnod + 1 - 1) + 3
icontie(3, ntie + 3) = 0
```

bansol2.f: In the unit cell model, dummy records were made in the variable *icon* out of *icontie* and then used in the subroutine *skyprp*. However, the variable *icontie* has changed in dimensions and this trick does not work anymore. Therefore a few lines of code are added to this subroutine. This requires the line "nelm = ncelm" in **iemx.f** to be placed before the call to skyprp instead of after.

```
do n = 1, ntie
  dof1 = icontie(1, n)
  dof2 = icontie(2, n)
  dof3 = icontie(3, n)

  call set_jdiag(jdiag, dof1, dof2)
  if (dof3 .gt. 0) then
    call set_jdiag(jdiag, dof1, dof3)
    call set_jdiag(jdiag, dof2, dof3)
  endif
enddo
```

initbc.f: Remove this subroutine as boundary node displacements are not prescribed anymore.

input.f: After all filaments are placed, the number of equations can be calculated. The new number of equations are the number of nodes times the degrees of freedom of a single node (3) and adding up 3 more, which are accounting for the new degrees of freedom of the network. After the number of equations is established, the dummy values in the *icontie* variable can be re-adjusted into the real values.

```
neqs = ndof * nnod + 3

do i = 1, ntie
  if (icontie(3, i) .eq. -1) icontie(3, i) = neqs - 2
  if (icontie(3, i) .eq. -2) icontie(3, i) = neqs - 1
  if (icontie(3, i) .eq. -3) icontie(3, i) = neqs
enddo
```

liquid.f: This is a complete new subroutine, which is called from **iemx.f** after tying has been performed. Alternatively, k (the bulk modulus of the liquid) can be placed in the input file as well.

```
subroutine liquid

include 'common.f'
real*8 RR(6,6), F(6), eq(6), frac, k
integer iv(6)

call pzero(RR, 36)
call pzero(F, 6)
call pzero(eq, 6)

k = 2.15D9
frac = 2 * nfil * length * radius / (lxcur * lycur)

RR(1,1) = (1 - frac) *  k * lycur / lxcur
RR(2,2) = (1 - frac) * k * lxcur / lycur
RR(1,2) = (1 - frac) * k
RR(2,1) = (1 - frac) * k

iv(1) = neqs - 2
iv(2) = neqs - 1

call skyasm(RR, F, eq, iv, 2)

return
end
```

setbc.f: One randomly chosen node in the network is fixed, otherwise the network has full translational freedom, which resultsin the equations not being solvable. Furthermore, a way to prescribe shear and tensile needs to be implemented.

```
! fix one node
inod = 25
ibc(ndof * (inod - 1) + 1) = 1
ibc(ndof * (inod - 1) + 2) = 1
bc(ndof * (inod - 1) + 1) = 0.0D0
bc(ndof * (inod - 1) + 2) = 0.0D0
```

tie.f: The call to tie_equation has changed a bit. Q is not zero anymore as used to be.

```
Q = C(icontie(3, i))
call tie_equation(dofj, dofi, icontie(3, i), Q, ktie, aorc)
```

tie_equation.f: This is the routine where the tying matrix is generated. The old matrix is completely removed and replaced by the one derived in section 4.1.

```
if (dofq .gt. 0) then
  RR(1, 3) = -ktie
  RR(2, 3) =  ktie
  RR(3, 1) =  RR(1, 3)
  RR(3, 2) =  RR(2, 3)
  RR(3, 3) =  ktie
endif

! fill iv
iv(1) = dofj
iv(2) = dofi
iv(3) = dofq

! Assemble force
F(1) = 0
F(2) = 0
F(3) = 0

! Assemble Stiffness matrix
if (iv(3) .gt. 0) then
  call skyasm(RR, F, EQCOR, iv, 3)
else
  call skyasm(RR, F, EQCOR, iv, 2)
endif
```