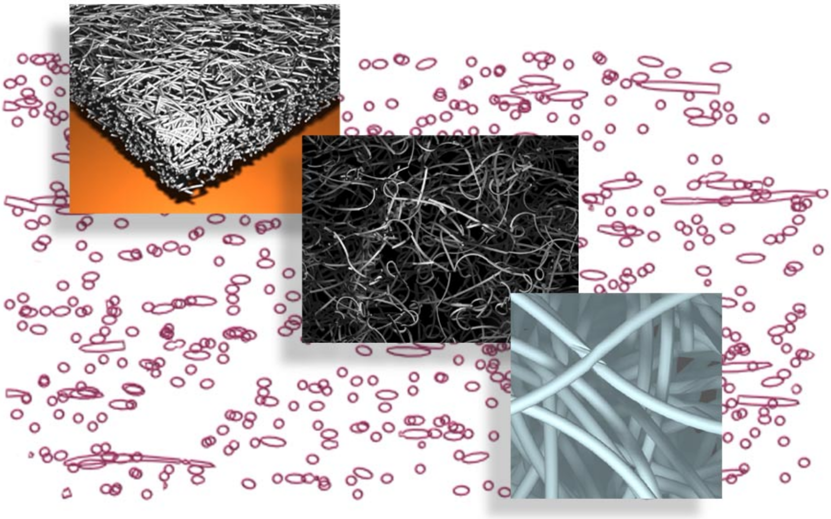


MODELLING AND VISUALIZATION OF THE GEOMETRY OF FIBRE MATERIALS

NIKLAS EDLIND

Structural Mechanics

Master's Dissertation

Structural Mechanics

ISRN LUTVDG/TVSM--03/5117--SE (1-80)
ISSN 0281-6679

MODELLING AND VISUALIZATION OF THE GEOMETRY OF FIBRE MATERIALS

Master's Dissertation by

Niklas Edlind

Supervisors:

Susanne Heyden, Jonas Lindemann
and Per Johan Gustafsson,
Div. of Structural Mechanics

Printed by KFS I Lund AB, Lund, Sweden, June 2003.

For information, address:
Division of Structural Mechanics, LTH, Lund University, Box 118, SE-221 00 Lund, Sweden.
Homepage: http://www.byggmek.lth.se

Acknowledgments

This master thesis work was carried from the summer of 2002 through to the winter of 2003 at the Division of Structural Mechanics, Lund Institute of Technology.

I would like to thank my supervisors, Dr Susanne Heyden and Professor Per Johan Gustafsson for their immense patience and support during the whole time that this thesis was written. Their guidance and inspiration has been an invaluable help.

Thanks are also due to Jonas Lindemann, who has, with great enthusiasm, helped me with the visualisation parts of this thesis.

Lund, May, 2003

Niklas Edlind

Abstract

A geometry model for three dimensional fibre networks is proposed. The geometry model is intended to be a part of a system, where data from a computer tomography of a fibre material can be fitted to the geometry model and from there, be evaluated according to key parameters such as fibre length, fibre orientation angles and radii of curvature. The model is also intended to give the geometry foundation for a FE-model. The model uses several linked circle arcs to describe a fibre, which accommodates for modelling varying degrees of out of plane curl and kinks. A program has been developed that, from distributions of the aforementioned key parameters, generates a periodic network according to the model which can be visualised by the output of two different file formats, vrml and a specialised format for the fibre network viewing program, FibreScope. Comparisons have been made between microscope photographs of several different fibre materials, including paper and cellulose fibre fluff, and visualisations of generated networks showing versatility in the model to describe a wide variety of fibre material types.

Keywords: fibre network, cellulose fibres, 3D model, visualisation, micro structure, geometry

Contents

1 Introduction

1.1 Background

Never before has the importance of developing the right material for every use been as great as now. Fibre materials are found everywhere around us. From the fabrics we wear, paper we write on or use as packages to the insulation in our walls. Traditionally, only organic fibres such as wood or wool have been used, but technical progress has allowed us to create fibres out of materials such as plastic, carbon or glass, which has widened the uses for fibre materials.

As with any other material, we want to know what properties we can expect from it before we put it to use. Aspects that are of interest are for example chemical, thermodynamic, mechanical and geometrical properties, the last being of interest in this work. One characteristic of fibre materials is the complexity of its microstructure. Therefore there is a need for geometry models that can as closely as possible describe the material in order to predict the behaviour of a fibre material in various situations.

Conditions for developing reliable micro-mechanical material models have lately improved significantly. Technical advances in the field of computer tomography (CT), i.e. three dimensional x-ray scans, make it possible to scan a fibre network in a large enough resolution to be able to make out individual fibres and their orientation. This opens for a possibility to transfer this data to a geometry model. Another important factor in the realization of a material model is that computer capacity has increased drastically allowing us to handle the vast amounts of data that will be needed.

1.2 Objectives

The objective of this thesis is to develop a general approach to the modelling of the geometry of fibre networks that can describe most naturally found 3-dimensional geometric constellations. A schematic diagram of the problem is presented in figure 1.1. The shadowed boxes symbolize the parts that are covered in this study. Further work is needed on the rest of the parts to complete the system.

1. CT-scan

   This work is carried out bearing in mind that it will be possible to receive data from a CT-scan. From the CT-scan, a 3-dimensional matrix of the density of the tested material is acquired. From this information the positions of the fibres can be reconstructed. This technique is under development. The reconstructed geometry can then be fitted into the geometry model. This is further discussed in chapter 3.2.

2. Network Representation

   A fibre material can be characterized by the distributions of a few key parameters such as fibre length, curl, cross-section properties and orientation. By

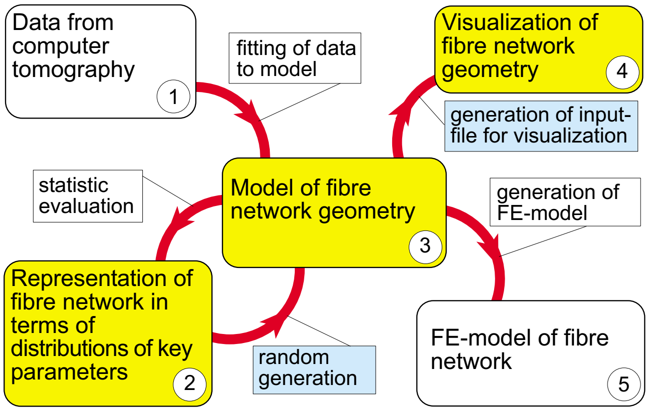


Figure 1.1: *Components in a general modelling concept for fibre materials. Shadowed boxes are treated in this work.*

doing a CT-scan of a piece of material and analysing the extracted model, all parameters of interest can be found. Knowing the distributions of these parameters opens several interesting options.

- By statistic evaluation of a scanned network, it can be reproduced as randomly generated networks with the same average properties as the real network.
- Different scanned networks can be compared with each other on a parameter level.
- Purely hypothetical networks can be generated.

3. Geometry Model

   The backbone of the system, a 3-dimensional geometry model which can either be randomly generated from the distributions in (2) or acquired from a CT-scan (1). It can be processed into an FE-model (5) and/or visualized (4). The basis of the model is that each individual fibre is represented as a series of linked circle arcs. Modelling a fibre in this manner gives us the possibility to describe out of plane curl and kinks. If it is a generated network, the individual fibres are placed into a unit-cube of a periodic network. An in-depth description on the fibre and network models are given in chapters 4 and 5, respectively.

4. Visualization

   When dealing with a complex structure such as a fibre network it is of great help if it can be visualized in three dimensions, giving more understanding of the structure. Visualization of deformation or contact points can be done after an FE-analysis. Also, during the development of the model visualizing is needed as a tool in the validating process. In this work, the programs FibreScope [9, 10] and 3ds max™ [1] are used for visualization.

5. Finite element model

   Differential equations are used to model many physical phenomena. Usually, the problems are far too complicated to solve using analytical methods. The finite element method (FEM) is an effective method for obtaining numerical approximate solutions to general differential equations.

   To make an FE-model from the geometry model (3) several steps are needed. Firstly, parallel to the geometry information, material properties need to be defined. Depending on what type of elements will be used the required parameters may vary. As an example, for a network made of beam elements we would need Young's modulus (E), shear modulus (G), fracture related properties and in some cases, dependency on external influences such as temperature or humidity have to be taken into account. If the size and shape of the cross-section is known, parameters linked to this, such as area (A), moments of inertia (I_x, I_y) and torsional stiffness (K_v) can be computed. Dealing with fibre networks, and fibre fluff in particular, the weak points of the structure are the bonds between fibres. In order to complete the FE-model these need to be identified and modelled. Also, topological information on fibres connected on either side of a periodic network cell must be retained. When this is done, unconnected fibres must either be forced to connect or removed, or else the system will become unsolvable.

1.3 Previous Work

Much work has been done in the modelling of fibre materials, the greater part focusing on planar geometry such as paper. Hamlen [3] and Kallmes & Corte [7] developed and analysed models for two-dimensional structures. Studies on three-dimensional planar network models, where fibres are allowed to stretch out-of-plane to account for the interwoven geometry of paper have been done by KCL-PAKKA [14] and Wang & Shaler [19]. A full three-dimensional model, which is required for analysing fibre fluff materials, and which this thesis is a further development of, has been proposed by Heyden [5]. In this model, every fibre is represented by a single circle arc and placed in a bounding box for a periodic network. The advantage of the arc model is that it is relatively easy to work with mathematically at the same time as it can describe the natural curl of fibres.

For viewing purposes, Lindemann [9, 10] has developed FibreScope, a visualization program specialized in fibre networks. Exported files from this model will be aimed at this program.

2 Fibre Materials

2.1 General Remarks

What makes fibre materials special is their geometric structure. They are built up of a complex network of arranged or randomly oriented fibres. A single fibre has load bearing capacity almost only in tension, so the stability of a composite material structure depends on the matrix material. In the case of low density fibre materials, which this thesis focuses on, the contact points are decisive. A variety of different fibre types are used to make materials of fibre network character. The demands on the network model depend on what type of fibre is used.

2.2 Material

The extent of the demands on the network model depends strongly on what type of material we are trying to describe. As the goal is to be able to describe all 3-dimensionally built fibre structures in a satisfactory way, we must look at the different behaviours we will expect.

In terms of geometry, we can coarsely divide fibre materials into two groups: natural and man made fibres, see comparison in figure 2.1.

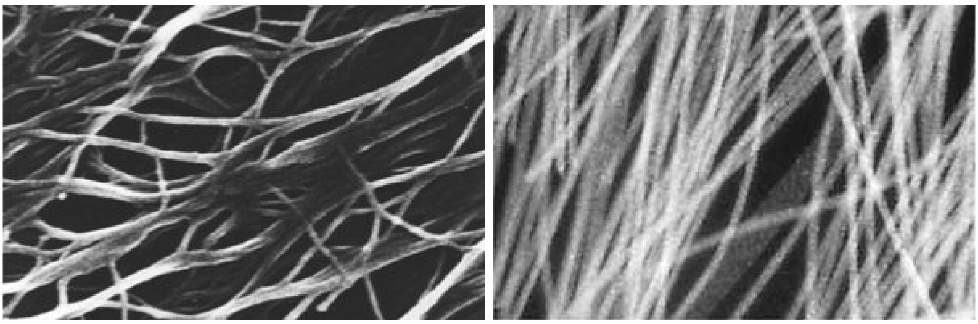


Figure 2.1: *Microscope photographs of wood (left) and rayon fibres (right)[13].*

Fibres extracted from natural materials often show an erratic shape with high curvature and many sharp turns along the length of the fibre called kinks. A cellulose fibre has a nearly quadratic, hollow cross-section when it is in the living wood. When dried, it collapses into a more rectangular, flat box shape. The size and mechanical properties of the fibre depends on the species of wood it is taken from.

Man made fibres such as glass-fibre and polymeric fibres display a quite different behaviour. They are often smooth and flowing in the length of the fibre. The cross-section shape is often regular and constant along the fibre length.

As well as setting up demands on the modelling of the fibre, the choice of material will have an influence on the modelling of inter-fibre bonds, a subject not covered in this study.

2.3 Manufacturing Process

This thesis focuses on three-dimensional low density networks. This definition includes a range of different materials, both man-made and natural fibre materials. This means that a variety of different production and extraction methods are used, which is the cause of the differences in geometric properties.

In the extraction of natural fibres [2], several steps are needed that change the shape and properties of the fibres. Although different methods are used for every fibre source, the general extraction steps include chemical treatment, mechanical separation of unwanted material and drying, which each affect the geometry of the fibres. Therefore, natural fibres often display traces of these treatments in the form of high curvature and kinks.

Man-made materials are not exposed to the same harsh treatment. They are mostly produced by melting the desired material and pressing the molten mass through small holes in a plate after which the extruded fibres are solidified through cooling. This process produces relatively straight fibres with little variation in curvature and cross-section shape. Of course, handling after production can also affect the geometry of the fibres.

3 Computer Tomography

Given the goal to make an as realistic model as possible from any fibre material, we need to know certain characteristic parameters of the network, such as density, fibre length, curvature, orientation etc. The geometry of a fibre, i.e. length, width and shape factor, which is a measure of the curl of a fibre, can be determined for free fibres, i.e. fibres not yet incorporated in a network or removed from a network, by methods like the STFI FiberMaster [17]. This method uses advanced image analysis of a fibre suspension passing a video camera to determine the parameters of interest. However, this method has a few limitations. It can not analyse the fibre geometry in its original network environment, which is needed to create a correct model of the material. Also, the parameters are determined through analysis of a two-dimensional projection, which is fine for long, planar fibres, but results will be misleading for fibres with high curvature in three dimensions. A solution to this could be the use of an industrial CT-scan (Computed Tomography), also called CAT-scan (Computer Aided Tomography), which produces a 3-dimensional image of the subject. This could then be analysed to acquire the network geometry needed.

Figure 3.1: *A Computer Aided Tomography scanner [6].*

3.1 Description

Industrial tomography is analogous to medical tomography. Gamma or X-rays are passed through the tested object, higher doses than in medical tomography are often used as the attenuation coefficients of observed objects usually are higher than in the human body. Radiation detectors mounted on the opposite side from the radiation source register the remaining radiation after passing through the object. From this data, by mathematical reconstruction, a picture of the attenuation (proportional to the density) can be created. All of the objects that the x-ray passes through overlap on the image, making it hard to isolate different elements. A CT scan works around this limitation by capturing only one very narrow slice of the object at a time. These slices can be viewed two-dimensionally or added back together to create a three-dimensional image of a structure.

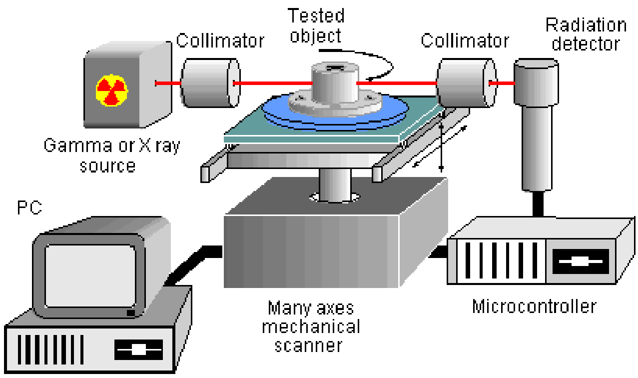


Figure 3.2: *Schematic diagram over a CT-scanner [6].*

The CT scanner moves around the object on a circular gantry passing x-ray beams and taking thousands of pictures as it rotates.

3.2 Reconstruction of Network Geometry

According to [4], the data that the CT-scanner outputs is a 3-dimensional density map of the test piece, where the density in a point of the matrix is given as a value between 0 and the specified density-depth.

The problem of reconstruction involves isolating the voxels, i.e. three dimensional pixels, that belong to one fibre. Having done this, the isolated fibre can be approximated by fitting of a fibre model to the extents of the real fibre.

4 Fibre Model

Chapters 4 and 5 deal with the contents of the box ”Random Generation” in figure 1.1, that is, generating a geometry model of a fibre network from distributions of key parameters. In the process of making a geometrical model of a low-density fibre material we first have to consider the geometry of a single fibre.

As discussed in chapter 2.2, the geometry of a fibre varies distinctly depending on the fibre material. We want to be able to model both the sharp kinks of fibres originating from wood and the smooth flowing of an industrially manufactured polymer fibre. These demands determine the modelling procedure. We want a model which is as general as possible to make it useful in a wide variety of cases.

In [5] the approach chosen was to model every single fibre as a circle arc. This approach approximates the natural curl of a fibre. However, a single fibre can in real life have a varying curvature and also out of plane curvature. This could be modelled by giving the fibre a higher order curvature. But a simpler approach, which also is the method used here, is to link several arcs with different orientation and radii together to approximate the shape of a natural fibre.

The benefits of this model is that each fibre segment is planar, even though the complete fibre displays a complex geometry. This simplifies the algebra used for generation and in computing points along the fibre. Also, using this method, it becomes quite easy to vary the complexity of the fibre geometry by simply changing the number of fibre segments used. Using several linked segments to model a fibre also opens the possibility of modelling kinks by aligning the segments so that the tangent is not continuous. Linking together segments this way also proves to be effective when generating controlled, specific fibre geometries, see figure 7.7.

4.1 Fibre segment

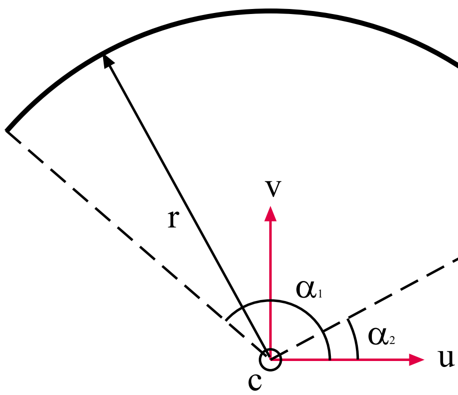


Figure 4.1: *Parameters defining a fibre segment.*

The basic unit of a fibre is the fibre segment, which is defined as a single circle

Table 4.1: *Input data and their descriptions.*

Parameter	*Description*	*Comment*
l_{seg}	length of fibre segment	independent
r_{seg}	radius of curvature of fibre segment	independent
θ	orientation angle	relative to previous segment
θ_{k1}	kink angle 1	relative to previous segment
θ_{k2}	kink angle 2	relative to previous segment
$\mathbf{t}_{ec,prev}$	unit vector pointing from the end point to the centre	belongs to previous segment
$\mathbf{n}_{prev}$	segment plane normal	belongs to previous segment
$\mathbf{d}_{e,prev}$	ending tangent vector	belongs to previous segment
$\mathbf{p}_{e,prev}$	segment end point	belongs to previous segment

arc. To define this we need several parameters, as seen in figure 4.1: centre point of circle $\mathbf{c}$, radius r_{seg}, unit base vectors $\mathbf{u}$ and $\mathbf{v}$, and the angles between the $\mathbf{u}$ vector and the start and end point of the arc, α_1 and α_2, respectively. Note that these parameters only apply to the geometrical model of the fibre segment. For a FE-model a number of additional properties are needed. Also, for visualizing purposes, parameters describing the cross-section of the fibre segment may be needed.

This choice of parameters is made because of the simplicity of the mathematics and programming needed and the ease with which the points of the fibre segment can be determined by use of equation 4.1.

$$\mathbf{x} = \mathbf{c} + r \cdot \cos\alpha \cdot \mathbf{u} + r \cdot \sin\alpha \cdot \mathbf{v} \qquad \alpha_2 \leq \alpha \leq \alpha_1 \tag{4.1}$$

The input for generating a fibre segment are quite different, however. Firstly, it is more natural to specify a fibre segment's length l_{seg} and radius of curvature, r_{seg} than vectors and angles. As we wish to be able to join together several fibre segments into a complete fibre we also need input data specifying the position of the fibre segment relative to the previous segment. These data are θ, defining the relative orientation of the fibre planes, kink angles θ_{k1} and θ_{k2} as well as data on the position and orientation of the previous fibre segment, see table 4.1.

The first five input parameters in table 4.1 are for every fibre segment determined from the given distributions mentioned in step (2) in figure 1.1. The rest of the parameters derive from the previously generated segment. The previous ending point is needed to ensure that the following segment is connected in that point, while the vectors serve as a reference when using the relative angles θ, θ_{k1} and θ_{k2}. If the segment is the first of the fibre, no previous segment exists and values of these are given arbitrarily. This has no effect on the end result as the final orientation and placement of the fibre is determined at a later point. Figure 4.2 shows the definition of the aforementioned parameters.

The orientation of a fibre segment in relation to its neighbouring segments is described by the angle θ. This is defined as the angle between the segments own plane and the previous segment plane. To be able to control in which direction a fibre is curled, θ is defined between 0 and 2π, positive direction being from the previous plane to the current, positive rotation around the common starting/ending vector $\mathbf{d}$ in the case that there is no kink. The orientation parameter has a strong influence on the final shape of the fibre. The distribution of θ controls both the planarity of the fibre as well as its extension. Figure 7.7 shows a variety of shapes controlled by θ.

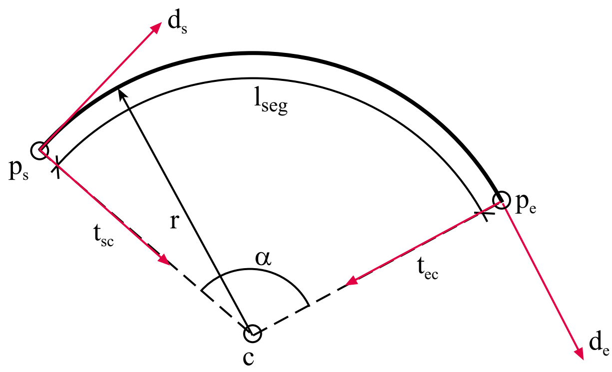


Figure 4.2: *Definition of fibre segment parameters used during the generating process.*

As will be seen later in this chapter, a large part in the process of generating fibre segments is rotating vectors in space around a specified axis. To rotate a vector (x_1, x_2, x_3) ϕ radians around an axis defined by a unit vector (a_1, a_2, a_3) the following equation is used.

$$\mathbf{x}' = \mathbf{R}(\mathbf{a}, \phi) \cdot \mathbf{x} \tag{4.2}$$

where $\mathbf{R}$ and $\mathbf{A}$ are given by

$$\mathbf{R}(\mathbf{a}, \phi) = \mathbf{I} - \sin\phi \cdot \mathbf{A} + (1 - \cos\phi) \cdot \mathbf{A}^2 \tag{4.3}$$

$$\mathbf{A} = \begin{bmatrix} 0 & -a_3 & -a_2 \\ -a_3 & 0 & a_1 \\ a_2 & -a_1 & 0 \end{bmatrix} \tag{4.4}$$

From the nine input parameters listed in table 4.1 the fibre segment parameters defined in figure 4.1 can be calculated. A graphical overview of the calculations is shown in figure 4.4. In the figure, solid boxed parameters refer to those that define the fibre segment while parameters that are marked with dashed boxes are used in the generation of the following segment. The numbers specified in each calculation step refer to the equation in which the calculation is carried out.
First, the opening angle of the circle arc is calculated.

$$\alpha = l_{seg}/r \tag{4.5}$$

The starting point of the currently generated segment is always identical to the ending point of the previous segment.

$$\mathbf{p}_s = \mathbf{p}_{e,prev} \tag{4.6}$$

The starting direction, $\mathbf{d}_s$, however, is not always the same as the previous ending direction. Adjustments have to be made first according to a possible kink. Two angles θ_{k1} and θ_{k2} are needed to describe an arbitrary kink. The angle θ_{k1} is defined as the angle between the projection of $\mathbf{d}_s$ onto the previous segment plane and the vector $\mathbf{d}_{e,prev}$ of the previous segment while θ_{k2} is the angle between $\mathbf{d}_s$ and the previous segment plane. The following steps are shown in figure 4.3. Two temporary vectors, $\mathbf{t}'$ and $\mathbf{d}'$ are created by rotating $\mathbf{t}_{ec,prev}$ and $\mathbf{d}_{e,prev}$ θ_{k1} around the previous segment plane normal.

$$\mathbf{t}' = \mathbf{R}(\mathbf{n}_{prev}, \theta_{k1}) \cdot \mathbf{t}_{ec,prev} \tag{4.7}$$

$$\mathbf{d}' = \mathbf{R}(\mathbf{n}_{prev}, \theta_{k1}) \cdot \mathbf{d}_{e,prev} \tag{4.8}$$

Now, rotating $\mathbf{d}'$ θ_{k2} radians around $\mathbf{t}'$ will give us the starting direction of the new segment.

$$\mathbf{d}_s = \mathbf{R}(\mathbf{t}', \theta_{k2}) \cdot \mathbf{d}' \tag{4.9}$$

When the starting direction $\mathbf{d}_s$ has been calculated the plane of the new segment is determined by rotating $\mathbf{t}'$ an angle θ around $\mathbf{d}_s$. This gives $\mathbf{t}_{sc}$.

$$\mathbf{t}_{sc} = \mathbf{R}(\mathbf{d}_s, \theta) \cdot \mathbf{t}' \tag{4.10}$$

If $\theta_{k2} = 0$, an angle $\theta = 0$ implies that the fibre segment lies in the same plane as the previous segment. We now have two unit vectors in the fibre segment plane, which means that the normal to this plane can be calculated.

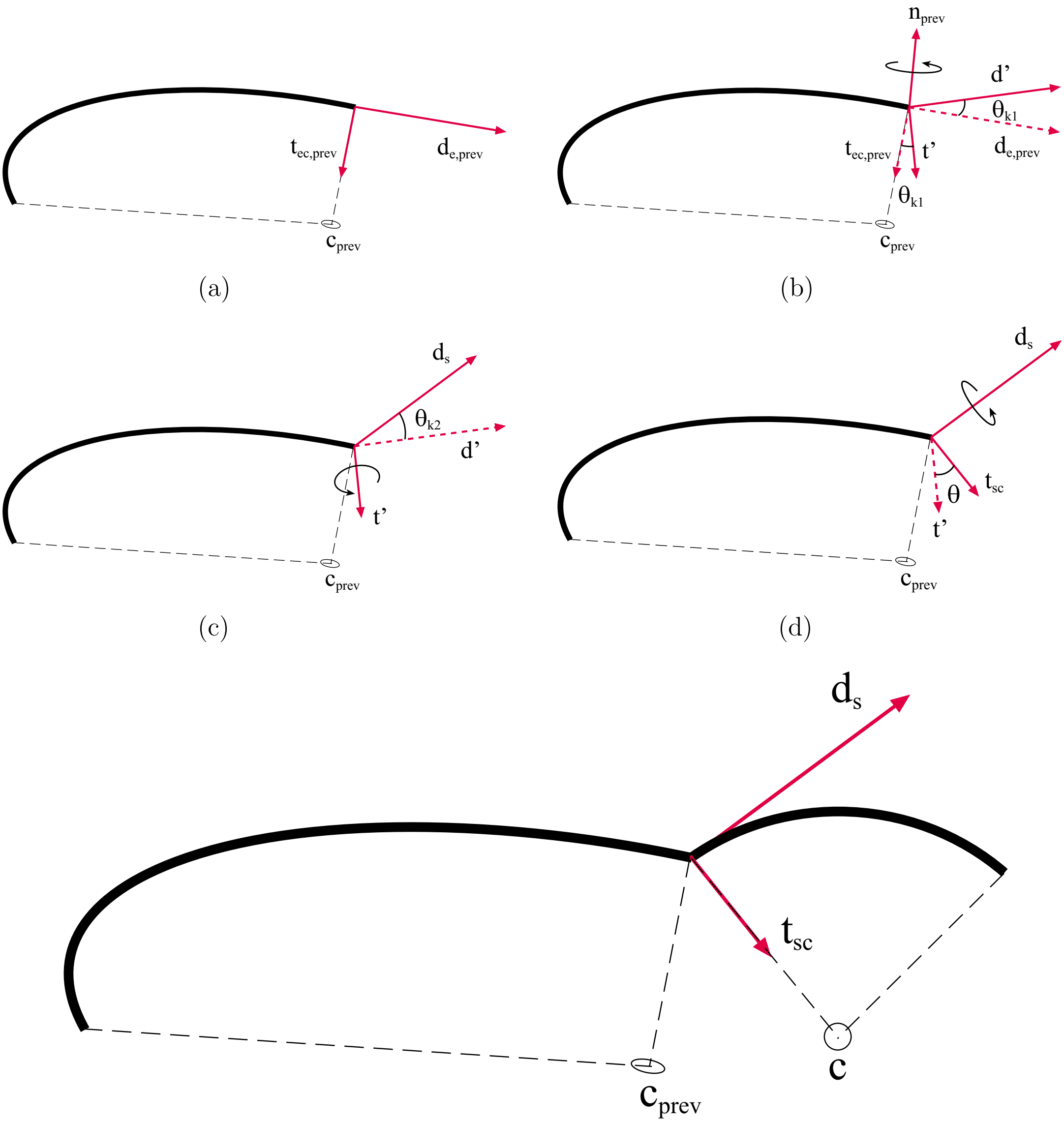
$$\mathbf{n} = \mathbf{t}_{sc} \times \mathbf{d}_s \tag{4.11}$$

Knowing the fibre plane normal, we can now calculate the base vectors $\mathbf{u}$ and $\mathbf{v}$ for the fibre segment plane. The vectors $\mathbf{u}$ and $\mathbf{v}$ are chosen so that they, together with $\mathbf{n}$ make an orthonormal basis. Further, $\mathbf{u}$ is chosen so that its projection onto the xy-plane is parallel with the x-axis. We have

$$\mathbf{n} \cdot \mathbf{u} = 0 \tag{4.12}$$

$\mathbf{u}$–$\mathbf{u}\cdot$(0,0,1)(0,0,1) must be of the form (a,0,0), a arbitrary. This yields

$$\mathbf{u} = \frac{\begin{bmatrix}1 & 0 & \frac{-n_x}{n_z}\end{bmatrix}}{\sqrt{1 + (\frac{n_x}{n_z})^2}} \tag{4.13}$$



(e)

Figure 4.3: *Definition and usage of the orientation angle θ and the kink angles θ_{k1} and θ_{k2}.*

unless the normal is in the xy-plane. In this case $\mathbf{u}$ is chosen as

$$\mathbf{u} = [n_y \quad -n_x \quad 0] \tag{4.14}$$

The last unit vector of the orthonormal basis, $\mathbf{v}$ is simply the cross-product of $\mathbf{n}$ and $\mathbf{u}$.

$$\mathbf{v} = \mathbf{n} \times \mathbf{u} \tag{4.15}$$

Also, knowing $\mathbf{t}_{sc}$, we can derive the centre point of the segment.

$$\mathbf{c} = \mathbf{p}_s + r \cdot \mathbf{t}_{sc} \tag{4.16}$$

Rotating $\mathbf{t}_{sc}$ around the segment normal α radians gives us a unit vector pointing from the end point in the direction of the centre point.

$$\mathbf{t}_{ec} = \mathbf{R}(\mathbf{n}, \alpha) \cdot \mathbf{t}_{sc} \tag{4.17}$$

The same rotation of $\mathbf{d}_s$ returns the outgoing direction $\mathbf{d}_e$.

$$\mathbf{d}_e = \mathbf{R}(\mathbf{n}, \alpha) \cdot \mathbf{d}_s \tag{4.18}$$

and thus, we have the end point as

$$\mathbf{p}_e = \mathbf{c} - r \cdot \mathbf{t}_{ec} \tag{4.19}$$

Finally, the opening angles α_1 and α_2 are calculated as the angles between $\mathbf{u}$ and $\mathbf{t}_{sc}$ and $\mathbf{t}_{ec}$, respectively.

$$\alpha_1 = \pi - \arccos(\mathbf{t_{sc}} \cdot \mathbf{u}^T) \tag{4.20}$$

and

$$\alpha_2 = \pi - \arccos(\mathbf{t_{ec}} \cdot \mathbf{u}^T) \tag{4.21}$$

We now have all the desired parameters that were defined in figure 4.1.

4.2 Fibre

The complete fibre consists of one or more fibre segments. They will either be joined together so that they are continuous in tangent in the connecting joint or kinked, i.e. discontinuous in tangent. The input data for generating a fibre are: number of segments, n_{seg} and vectors of length n_{seg} containing the parameters in table 4.1 for all the segments that make up the fibre.

The most important characteristics of the individual fibre are fibre length and curl. The length of a fibre is easy to control. With segment length as an input data for the fibre segments, the total fibre length is simply split between its segments. Distributions of fibre length needs to be handled in a slightly different way, which is further discussed in chapter 5.3.

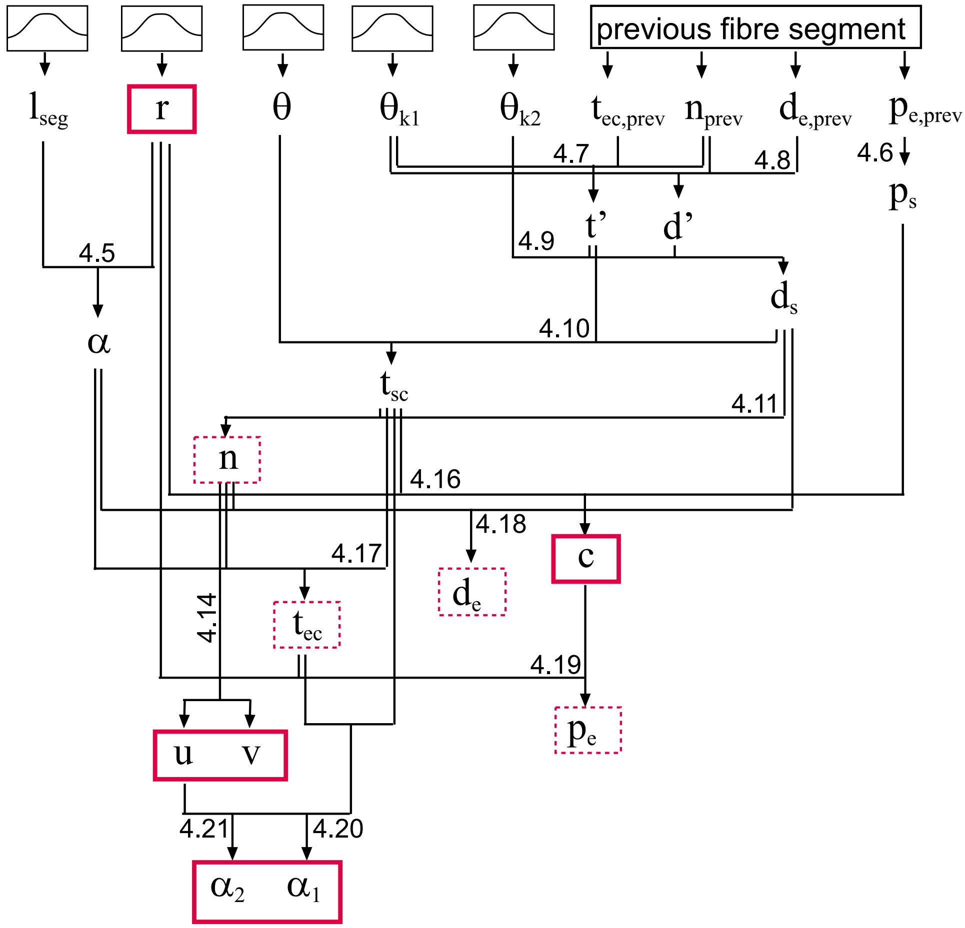


Figure 4.4: *Flowchart showing the calculations of the parameters defining a fibre segment from the input parameters. Solid boxes denote segment-defining parameters, dashed boxes, parameters needed for the generation of the next segment.*

Curl is here defined by the relationship between a fibre's length and its maximum extent, i.e. the diameter of the smallest sphere that the fibre can fit into, see figure 4.5 and equation 4.22.

$$C = \frac{l}{d} - 1 \tag{4.22}$$

If a fibre is straight the curl is zero, otherwise the curl increases with the curvature. The curl value of a fibre depends on segment-to-segment orientation θ, θ_{k1} and θ_{k2}, segment opening angles α, number of segments and ratio between the segments radii. Figure 4.6 shows the curl value of a two-segment fibre as a function of α and θ, where α and r_{seg} are the same for both segments. Figure 4.7 shows the influence

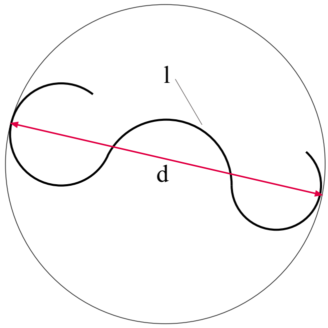


Figure 4.5: *Parameters used in the definition of curl.*

of the number of fibre segments on curl.

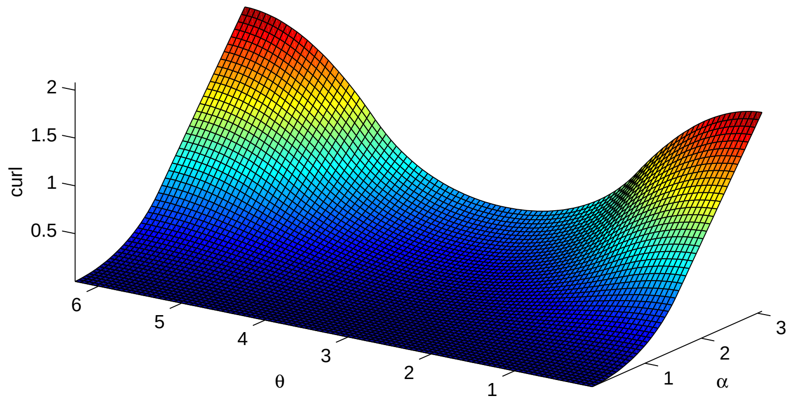


Figure 4.6: *Fibre curl as a function of α and θ for a two-segment fibre with identical radii and α and no kink.*

It would be desirable to be able to generate a fibre based solely on a specified fibre curl C. However, due to the complexity of curl dependencies when more than one fibre segment is used as seen in the numerical example above, it can not be used as an input parameter, since there is no unique way of generating a fibre with a certain curl value. For single-segment fibres curl depends only on α and thus is easily controllable. When using more than one segment, one can use the trial-and-error method to generate a number of typical fibres with a certain curl or, alternatively, simply see the curl as an output data.

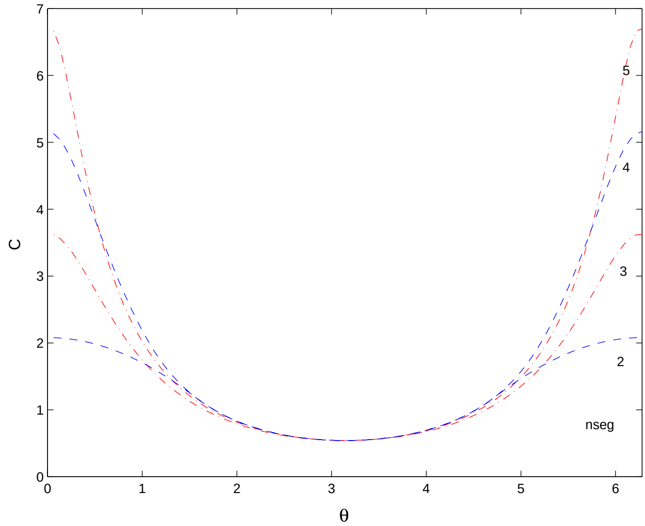


Figure 4.7: *Curl as a function of θ with $\alpha = \pi$. The influence of the number of segments used is shown.*

When the fibre is complete it must be prepared for insertion into the network. To control the placement of the fibre, a centre point and orientation in local coordinates needs to be defined. The fibre is shifted so that the end points centre around the local origin (step 2 in figure 4.8 and rotated so that the end points lie on the local x-axis c.f. figure 4.8, step 3.

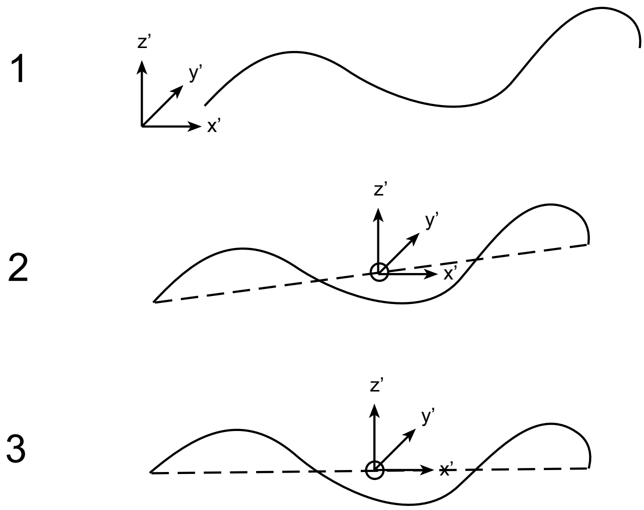


Figure 4.8: *Normalization of generated fibre.*

5 Network Model

The top level of the generation process is the network generation. While the fibre and segment-level generations work with specific input data, the network model deals with distributions of variables. Input data for a fibre is picked from these distributions and a fibre is generated. It is then placed into the network according to orientation and position picked from distributions of these. The fibre is then trimmed and modified to comply with the specified dimensions of the periodic network. This process is continued until the desired network density is reached.

Table 5.1: *Table of input parameters.*

Input parameter	*Description*
L_x, L_y, L_z	Unit cell dimensions in x, y and z directions
ρ	Network density. Total fibre length per unit volume
$f(\phi_1)$, $f(\phi_2)$	Fibre orientation. Distributions of two angles
$f(l_{fib})$	Distribution of fibre length
n_{seg}	Number of segments per fibre
$f(r)$	Distribution of fibre segment radii
$f(\theta)$	Distribution of relative fibre segment orientation
$f(\theta_{k1})$, $f(\theta_{k2})$	Distribution of kink angles
P_{kink}	Probability of kink at a segment-to-segment contact point

5.1 Fibre placement

When placing the fibres into the network model, the centre point between the end-points of the fibre is positioned randomly into the unit cell. No consideration is taken to fibres already in the network. This implies that two or more fibres can take up the same space. Adjusting every fibre individually according to the presence of other fibres would, however, be a difficult and time-consuming task. Also, in low-density three-dimensional networks, the expected low occurrence of this duality would not be a very big problem.

A desired fibre orientation vector is created by rotating a unit vector pointing in the direction of the x-axis first by the angle ϕ_1 around the y-axis, then ϕ_2 around the z-axis, the angles defined as in figure 5.1. By choosing the distributions of these angles a preferred fibre orientation can be modelled. By choosing ϕ_2 as a constant value, the fibres will be placed so that a certain plane is favoured.

As described in chapter 4.2, the generated fibre is, prior to insertion, oriented along the local x-axis and centred around the local origin. By vector multiplication of the desired orientation and the local orientation, i.e. the x-axis, a vector perpendicular to these is obtained to rotate the fibre around into position. Figure 5.2 shows the placement routine. The centre points are transposed according to the desired position of the fibre (step 1). In step 2, a rotation matrix R is created according to

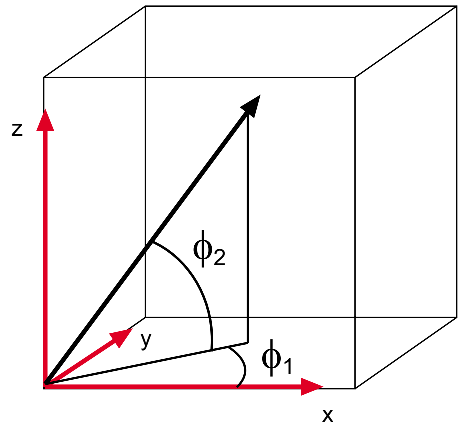


Figure 5.1: *Definition of the fibre orientation angles* ϕ_1 *and* ϕ_2*.*

equation 4.3 after which the centre point and base vectors of every fibre segment are rotated to the desired orientation. A random rotation is given to the fibre around its own axis in step 3. As this orientation angle is not specified, this is done to avoid similarity.

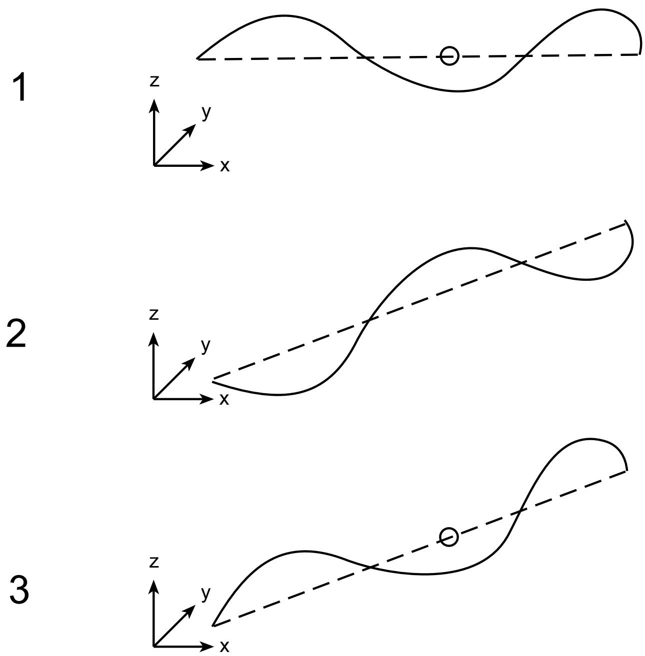


Figure 5.2: *Placement of a fibre into the network.*

5.2 Periodic Networks

When dealing with large fibre networks, during generation of geometry and especially the analysis of an FE-model, computing time will become a limiting factor. A way of minimizing the size of the computed network is to model a relatively small periodic cube of fibres, where each fibre is trimmed to fit into the box and the trimmings are replaced so that the original fibre will be complete and unbroken when several identical cubes are placed around the first one. Examples of this are shown in figures 5.3 and 5.4.

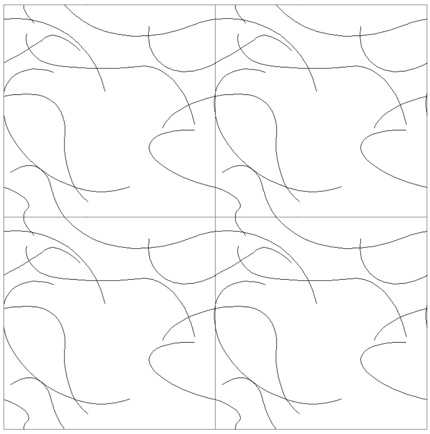


Figure 5.3: *Example of a periodic network.*

To achieve this, every fibre has to be checked along its entire length if it is partially outside the defined unit cube. Also, if the fibre part checked is further away than another unit cubes' length from the perimeter, it has to be trimmed once more.

The trimming procedure is done segment-wise and works by redefining the opening angles α_1 and α_2 and the centre point of the arc. First the fibre segment is checked using equation 4.1 to see if and what planes it passes, and thus is to be trimmed at. The angle α at which the segment crosses the plane at is calculated whereby a new segment is created for each plane crossed. For example the segment in figure 5.4 has its original opening angles α_1 and α_2. After controlling which boxes it passes through, it is determined that control of crossing points needs to be done along the planes $y = L_y$, $x = L_x$ and $x = 2L_x$. Control of these planes gives us the angles β_1 through β_4 where the cuts will take place. The original fibre segment will be replaced by five new segments identical to the original save for the modifications given in the table below.

When the network is generated as a periodic cube, the size of the unit cell will become an important parameter which affects the properties of the network. For computational reasons, the cell should be as small as possible. However, if the cell is

Table 5.2: *Modifications to the created fibre segments during trimming.*

Fibre segment	α_1	α_2	c_x	c_y
1	—	β_1	—	—
2	β_1	β_2	$c_{x,orig} - L_x$	—
3	β_2	β_3	$c_{x,orig} - L_x$	$c_{y,orig} - L_y$
4	β_3	β_4	$c_{x,orig} - 2L_x$	$c_{y,orig} - L_y$
5	β_4	—	$c_{x,orig} - 2L_x$	—

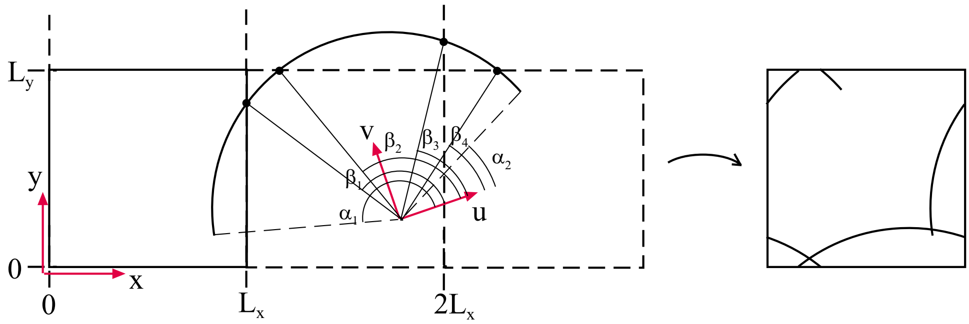


Figure 5.4: *Example of the cutting procedure on a fibre segment.*

too small in relation to fibre length it may affect the results. Heyden [5] found that, for straight fibres, the relationship between cell size and fibre length L/l_{fib} should be no less than 1 to 1.2, for calculation of elastic stiffness, using periodic boundary conditions.

5.3 Distributions

The distribution function used for distributed parameters is the beta distribution. It is used due to the its versatility and to the fact that it is defined within specified limits, unlike the normal distribution which can give negative values. This would not work on distributions of the length parameters l_{fib} and r_{seg}. The distribution is given by the four parameters a, b, q, and r, where a and b denote the interval of the distribution and q and r determine the shape [12].

The beta distribution is given as

$$f(X) = \frac{1}{\beta(q,r)} \cdot \frac{(X-a)^{q-1}(b-X)^{r-1}}{(b-a)^{q+r-1}} \qquad \begin{cases} a \leq X \leq b; \\ 0 < q; \\ 0 < r; \end{cases} \tag{5.1}$$

Where the Beta function $\beta(q,r)$ is given by

$$\beta(q,r) = \int_0^1 x^{q-1}(1-x)^{r-1}dx \tag{5.2}$$

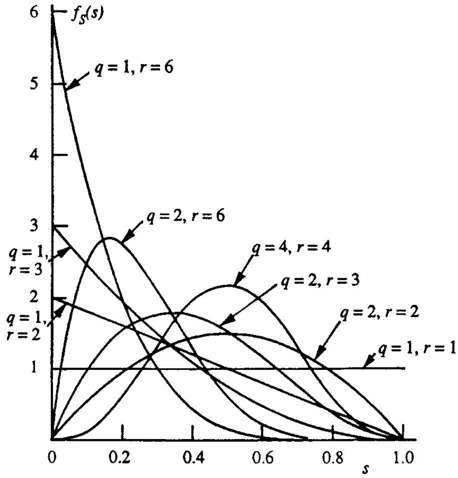


Figure 5.5: *Standard beta distribution probability density function for different parameters.*

If $a = 0$ and $b = 1$, the standard beta distribution $f_S(s)$ is obtained. As seen in figure 5.5, the shape of the distribution can vary from a rectangular distribution to different forms of symmetrical and asymmetrical shapes. Using $q = r$ gives a symmetrical distribution.

When a distribution for fibre length is given, it is advantageous to translate this to a distribution of segment length so as the segments of a fibre are not simply a division of a fibre length into n_{seg} equal parts. To do this, we must see to that the mean and standard deviation of the fibre length distribution is preserved. When adding distributions the following rules apply:
for mean μ

$$\mu_{tot} = \mu_1 + \mu_2 + \mu_3 + \ldots \tag{5.3}$$

and standard deviation

$$\sigma_{tot} = \sqrt{\sigma_1^2 + \sigma_2^2 + \sigma_3^2 + \ldots} \tag{5.4}$$

The fibre segment length distributions are chosen to be the same, which leaves

$$\mu_{seg} = \frac{\mu_{fib}}{n_{seg}} \tag{5.5}$$

and

$$\sigma_{seg}^2 = \frac{\sigma_{fib}^2}{n_{seg}} \tag{5.6}$$

The mean and variance of the beta distribution are given by

$$\mu_X = a + \frac{q(b-a)}{q+r} \tag{5.7}$$

and

$$\sigma_X^2 = \frac{qr(b-a)^2}{(q+r)^2(q+r+1)} \tag{5.8}$$

Using only symmetric distributions ($q = r$) and with $a_{seg} = a_{fib}/n_{seg}$ and $b_{seg} = b_{fib}/n_{seg}$ equations 5.6 and 5.8 gives

$$q_{seg} = r_{seg} = \frac{2q_{fib} - n_{seg} + 1}{2n_{seg}} \tag{5.9}$$

This equation solves our problem of segment length distribution but also imposes limitations on the distribution of fibre length. For the summation of the segment length distributions to work, q_{seg} must per definition be larger than zero. This implies that

$$q_{fib} > \frac{n_{seg} - 1}{2} \tag{5.10}$$

Using this rule will give the fibre length distribution combined from the segment distributions the correct mean value and standard deviation, however, numerical tests have showed that for the shape of the distribution to resemble the one specified, q_{seg} should be equal to or larger than one, which gives a stricter condition to be fulfilled:

$$q_{fib} > \frac{3n_{seg} - 1}{2} \tag{5.11}$$

Even when 5.11 is fulfilled one may not expect the shape of the distribution for the total fibre length to exactly match that of a beta distribution.

As for the distribution of the network orientation angles ϕ_1 and ϕ_2, if an isotropic is desired, special attention has to be paid to these. To achieve this, ϕ_1 should be distributed as a cosine function, while ϕ_2 is defined as a rectangular distribution between 0 and 2π. The beta distribution comes in handy in this case also, as it can very closely approximate the cosine shape using $q = r \approx 3.4$.

5.4 Input combinations

As mentioned earlier, it is conceivable to implement several different varieties of input/output combinations using the proposed fibre model. The network can be governed by specifying fibre density, i.e. length of fibre per volume unit, or total fibre length. To save computing time, one might also prefer to generate just a few fibres, which are then reused and placed in the network several times. The different generated fibres can then be chosen based on a percentage belonging to the fibre, representing the part of the network consisting of this fibre type.

Two different approaches have been looked at in this work. The input data specified in table 5.1 is aimed at a program generating random fibres according to those distributions specified. The other method uses a "library" of fibres as input data, which are then picked and placed into the network. This method might be preferred when a certain appearance of the fibres, which can not be quantified in the distributions of the previous method are desired.

6 Computer Implementation

6.1 Overview

Programming has been done to implement the creation of periodic network cubes of fibres from statistic distributions. All programming has been done in MATLAB® [18], a program for numerical computation. The MATLAB® developing environment has been chosen for various reasons; it is advantageous to have a platform independent program so as to not have to reprogram for different operative systems, it is a simple and effective environment for testing, and also, earlier work [5] has been done in this environment. The most important programs can be found in appendix A of this report.

Table 6.1: *Descriptions of the functions developed for creating three dimensional periodic fibre networks.*

Function name	*Description*
`gennetbeta.m`	Main program. Generates a periodic fibre network based on beta distributions of various parameters.
`gennetlib.m`	Main program. Generates a periodic fibre network based on specific fibre types.
`genfib3.m`	Generates a single fibre.
`genfibseg.m`	Generates a single fibre segment.
`modfibseg.m`	Modifies a placed fibre segment to fit in the periodic network.
`curl.m`	Calculates the curl and length of one fibre.
`place.m`	Places a fibre into the periodic network.
`rot3daxl.m`	Rotates a vector around a specified axis.
`asort3d.m`	Sorts crossing angles from α_1 to α_2
`cross_c_p3d.m`	Returns crossing point(s) between segment circle and specified plane.
`cuttingplanes.m`	Returns planes that the fibre crosses.
`isonarc.m`	Checks if crossing point from `cross_c_p3d` is on the fibre segment.
`fibre2vrml.m`	Generates a vrml-file of the network.
`fibre2scope.m`	Generates a nef-file of the network for viewing in FibreScope.
`plotfibre.m`	Test plot of fibre.
`plotbox.m`	Test plot of bounding box.
`plotnet.m`	Test plot of fibre network.

The program is built up of several specialized functions shown in table 6.1 which can be used individually or as a complete main program as `gennetbeta.m` or `gennetlib.m`. The generating functions on segment and fibre-level use exact input of all parameters, while distributions are handled on the network-level. The reason for this is, apart from the benefit of having a well-structured program is that, for example, `genfib3.m` can be used stand-alone to generate a specific fibre, where the exact parameters for all segments are specified without having to use different functions. Also, other input combinations than those used in `gennet.m` might be interesting to use. In that case, only the main program needs to be altered, as the versatility of the subfunctions allow for usage in other variants.

A few visualizing functions are included in the list above. They have been used in the verifying process during programming. They are very time-consuming when plotting normal sized networks, but are quite effective for viewing the appearance of single fibres or small networks.

A simplified schematic diagram of `gennetbeta.m` is shown in figure 6.1. The logical structure of `gennetlib.m` is much the same.

6.2 Data Storage

The segment-defining parameters introduced in chapter 4.1 are stored in a vector `segdat` which `genfibseg.m` returns.

$$segdat = [r \quad c_x \quad c_y \quad c_z \quad \alpha_1 \quad \alpha_2 \quad u_x \quad u_y \quad u_z \quad v_x \quad v_y \quad v_z] \tag{6.1}$$

During generation of a fibre, `segdat` for each segment is added to a matrix `fibdat` containing the data of all the included segments.

$$size_{fibdat} = n_{fibresegments} \cdot 12 \tag{6.2}$$

Finally, `fibdat` of all the fibres in the network are added to a three-dimensional matrix `netdat` containing all fibre segments of all the fibres in the network. As clipping of the fibres occur to fit them into the periodic network, new fibres are formed from the original. These clipped fibres are not necessarily made up of the specified number of segments. If this is the case, the remaining parts of the fibre matrix, where the following segments normally would be found are filled with zeros.

6.3 Performance

The following tests have been done on an AMD 1700+ processor with 524 MB of RAM. Table 6.3 shows a summary of one execution of one of the main programs `gennetbeta.m`. The generated network had a density of $\rho = 0.01$ in a cube of side length 100 and consisted of three-segment fibres. The total time for the execution was 13.23 seconds and the number of fibres generated was 200, which can be seen in the number of calls to the fibre-generating function `genfib3.m`. The time column shows the number of seconds that have been spent inside a function, that is, also counting the time of included functions. Self-time depicts the time of a function without included functions. The summary of the self-time percentage does not accumulate to 100%. This is because functions that are built in to MATLAB®,

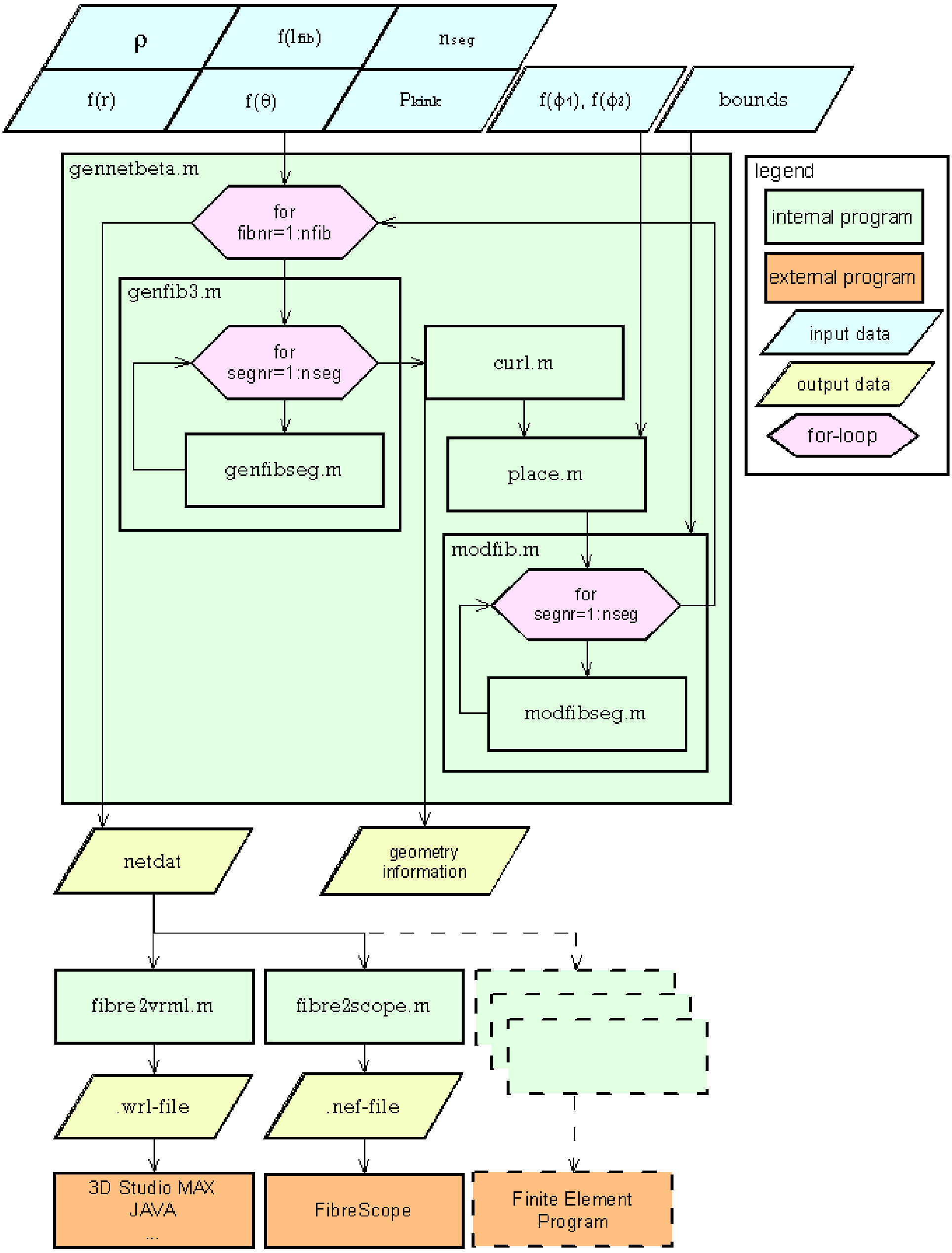


Figure 6.1: *Flowchart of the program structure.*

Table 6.2: *Table of developed functions used in gennetbeta and their performance.*

Name	*Time (s)*	*(%)*	*Calls*	*Time/call (s)*	*Self time (s)*	*(%)*
`gennetbeta.m`	13.23	100.00%	1	13.23400	0.08	0.60%
`genfib3`	9	68.00%	200	0.04498	0.17	1.30%
`curl.m`	7.95	60.10%	200	0.03975	7.95	60.10%
`modfib.m`	1.25	9.40%	200	0.00625	0.02	0.10%
`cuttingplanes.m`	1.11	8.40%	600	0.00185	1.11	8.40%
`genfibseg.m`	0.66	4.90%	600	0.00109	0.11	0.80%
`rot3daxl.m`	0.56	4.30%	7600	0.00007	0.56	4.30%
`place.m`	0.22	1.60%	200	0.00109	0.04	0.30%
`modfibseg.m`	0.13	1.00%	600	0.00021	0.08	0.60%
`isonarc.m`	0.03	0.20%	298	0.00011	0.03	0.20%
`cross_c_p3d.m`	0.02	0.10%	149	0.00011	0.02	0.10%
`asort3d.m`	0	0.00%	600	0.00000	0	0.00%
$\sum$	—	—	—	—	10.17	76.8%

for example the function that picks random numbers from the beta distribution, are not included.

It is clear that one of the most computational intensive functions is the `curl.m` command, which uses more than 60% of the total time. This percentage also increases with the number of fibre segments used. This is because the function compares a specified number of points along the fibre with all other points to obtain the largest inter-fibre distance. The curl can be calculated with a fewer number of points per segment to make it faster, though this also increases the error in the result. There may be more effective algorithms for the curl computation, but this has not been looked into. Instead, to save time, if the curl is not needed, `gennetbeta.m` can also be executed without curl calculation, which makes it subsequently faster.

Table 6.3 shows the equivalent summary of one execution of `gennetlib.m`. Keeping in mind that curl is not calculated, the program is still considerably faster than its counterpart. This is due to the fact that no fibres are generated in the program, but picked from the library of fibres specified in the input data.

A comparison of computing time dependency on the number of fibre segments and network density has also been done. Figure 6.2 shows that the relationship between the network density and execution-time is linear, which implies that the same counts for the number of fibres. For higher densities, though, the effect of the number of segments on time has an increasing tendency.

Table 6.3: *Table of developed functions used in gennetlib and their performance.*

Name	*Time (s)*	*(%)*	*Calls*	*Time/call (s)*	*Self time (s)*	*(%)*
`gennetlib`	2.984	100.00%	1	2.984	0.063	2.10%
`modfib`	1.093	36.60%	200	0.005465	0.079	2.60%
`cuttingplanes`	0.828	27.70%	600	0.00138	0.828	27.70%
`place`	0.501	16.80%	200	0.002505	0.249	8.30%
`rot3daxl`	0.301	10.10%	4000	0.00007525	0.301	10.10%
`modfibseg`	0.186	6.20%	600	0.00031	0.092	3.10%
`asort3d`	0.064	2.10%	600	0.000106667	0.064	2.10%
`isonarc`	0.03	1.00%	192	0.00015625	0.03	1.00%
`cross_c_p3d`	0	0.00%	96	0	0	0.00%
$\sum$	—	—	—	—	1.706	57.00%

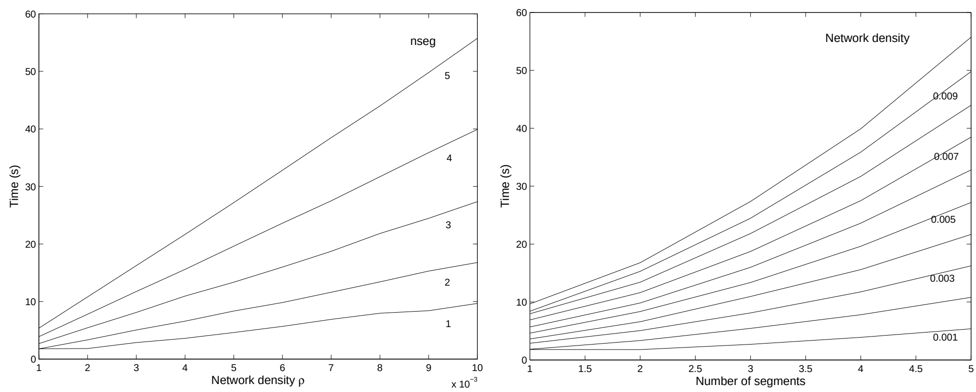


Figure 6.2: *Comparison of the effects of varying number of segments and network density on execution-time of the main program.*

7 Applications

This chapter demonstrates the versatility of the developed fibre network model by showing various examples. It is also an excellent way of showing how the input parameters affect the final shape of the fibre or network.

7.1 Visualization software

For visualizing the generated networks, mainly two programs have been used, FibreScope and 3ds max™ . However, the geometry can be imported into almost any 3D modelling program by the file format vrml c.f. figure 6.1.

VRML (Virtual Reality Modelling Language)[20] is a file format with the extension .wrl describing three dimensional content. Once developed as an ISO-standard (VRML97) to facilitate for 3d on the world wide web, complying to a set of requirements: platform independence, extensibility and the ability to work over low-bandwidth connections. The awaited explosion of virtual worlds on the net failed to come, but the format is widespread as an interface between competing formats as most programs can import and export it. This is also the main argument for choosing vrml as an export option in this project.

3ds max™ [1], or 3D Studio MAX, is a widely used program for three dimensional modelling, animation and rendering. In this work, it has been used to import vrml files for final rendering. Most of the figures showing generated networks are rendered with 3ds max™ . The only retouching applied to the generated vrml models are material application, where a bitmap picture is applied to the surface of the model, and lighting, which simulates highlights and shadows from a light source.

FibreScope is a program developed by Lindemann [9, 10] specifically for viewing fibre networks. Unlike 3ds max™ which is used to produce rendered pictures and animations, FibreScope can in real-time rotate and manoeuvre through the network for easy examination. Figure 7.1 shows the FibreScope environment. There is also an option to view fibre networks in 3D with special goggles.

7.2 Examples of single fibres

On segment-level, all that can be varied is the opening angle and radius of curvature. Setting a very large radius in proportion to opening angle will approximate straight fibre segments. As is shown in figure 7.2, the model can handle this well. The "cuts" in the straight fibre segment are due to problems in the importing of the vrml-file when large radii are used. The geometry in the model is not affected by this.

Fibre-level parameters leave more options to be explored and parameters to be varied. Figure 7.7 shows the effects on otherwise identical parameter set-ups when

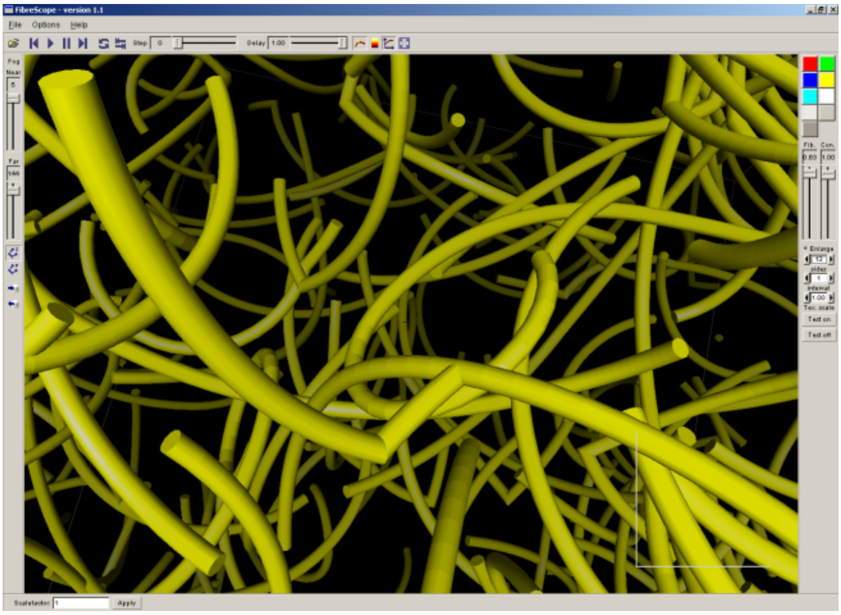


Figure 7.1: *Screenshot of FibreScope with generated network.*

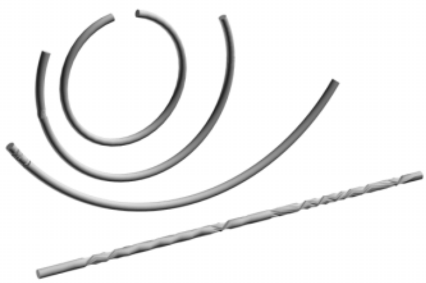


Figure 7.2: *Examples of fibre segment shapes.*

the segment orientation angle θ is set to different values. Opening angles of all segments are π and the number of segments per fibre is five. The angle θ is defined in figure 4.3 as a relative angle between two fibre segments. For $\theta = 0$ successively shorter radii were used to avoid the fibre curling into itself. The fibres on the edges of the figure are three dimensional renderings of generated fibres using the specified θ angle and its projection on a plane. The centre diagram shows how θ affects the fibre shape when it is constant over all fibre segments. If θ is close to zero, the fibre will curl inwards to a C- or spiral shape while an angle closer to π will curl the segments away from each other, resulting in the S-shape shown. These angles will also produce fibres in one plane. Angles between these values will give the fibre an out of plane shape which is a cross between the spiral and S-shapes. Special cases are the angles $\pi/2$ and $3\pi/2$ where the fibre in profile will take on the form of a flight of stairs, the step-shape. If the value of θ is between 0 and π the fibre will form a right handed helix, that is, the twist of the fibre will be the same as the thread of a screw. Between π and 2π a left handed helix will be formed. This example illustrates the importance of θ in deciding the shape of the fibre and also gives a hint to the hypothetical fibres which can be generated using the model.

An example of kinked fibres can be seen in figure 7.4. The left hand figure displays how fibres with 100% kink probability look. To the right are fibres which are generated with a parameter set-up to resemble cellulose fibre fluff. The kink probability is set to 60%.

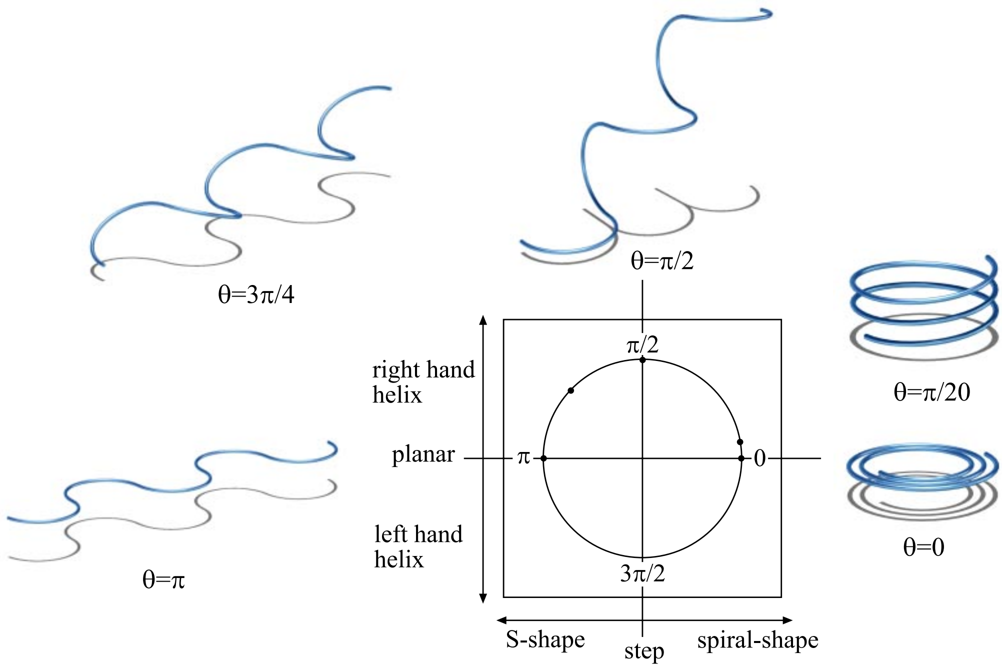


Figure 7.3: *Examples of the influence of the orientation angle θ. For the fibre examples, shadows are also showed.*

7.3 Network Examples

The most interesting experiments can be done on network-level. Adjusting the input parameter distributions opens many possibilities to shape the generated network in the desired way. An example of this, figure 7.6, shows the effects of changing the position of the distribution of θ. The left column shows a θ-distribution centred

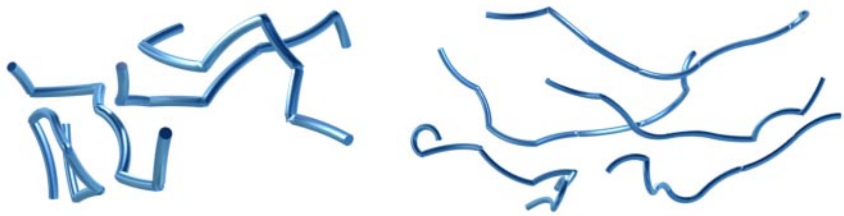


Figure 7.4: *Example fibres with kink probability P_k set to 1 (left) and, to the right, fibres resembling cellulose fibre fluff ($P_k = 0.6$). In both cases, rectangular distributions between $-\pi/2$ and $\pi/2$ are used for the kink angles.*

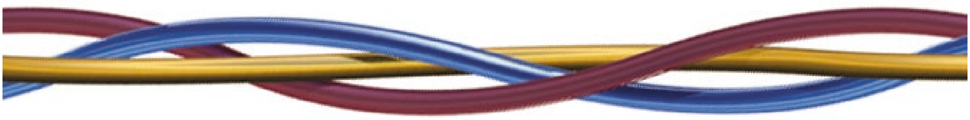
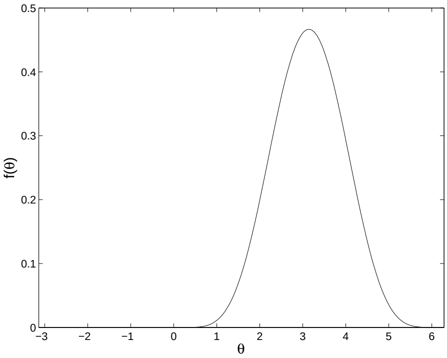


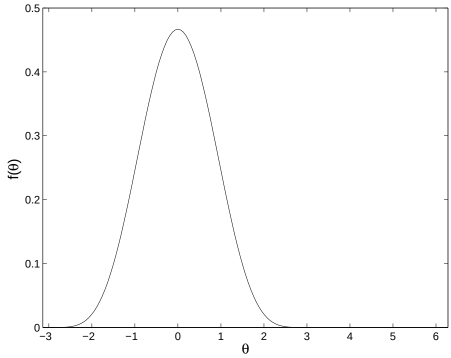
Figure 7.5: *An example of a generated thread made up of three fibres.*

around π, favouring the S-shape while the right column distribution, with the same q-value, is centred around 0 which will produce more spiral shapes. Two networks were generated using identical set-ups, save θ and the renderings of them can be seen under the distributions. It appears as if the fibres on the left-hand picture are slightly straighter, which we would expect from the generation set-ups. Analysing the curl of each fibre in the two networks, it is clear that the right-hand side shows a significantly higher curl value than its counterpart. Although these results are precisely what was expected from the definition of θ, it shows in a concrete way the actual impact on generated network geometry of changes in input parameter distributions.

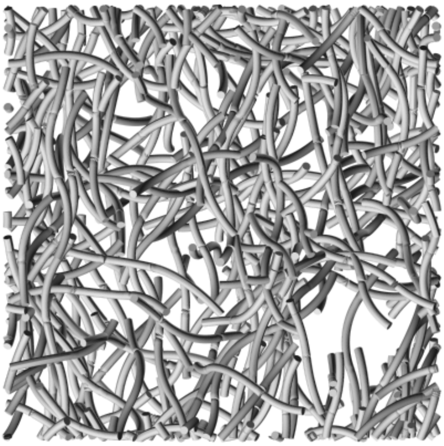
The fibre orientation angles ϕ_1 and ϕ_2 are important parameters as they strongly affect the appearance and also the mechanical properties of a fibre network. As discussed in chapter 5.3, if an isotropic network is wanted, one of the angles should be a rectangular distribution and the other distributed as a cosine shape. Figures 7.7(a) and (b) show a rendering of a network with these settings. A more paper-like appearance with the fibres lying in planar layers can be modelled by setting one angle as a constant (in this case 0) while the other is rectangularly distributed. Figures 7.7(c) and (d) shows an example of this. Figures 7.7(e) and (f) show an extreme, where both angles are constant (here both are set to 0). This is not a very natural appearance, however, it clearly shows that modelling a preferred fibre



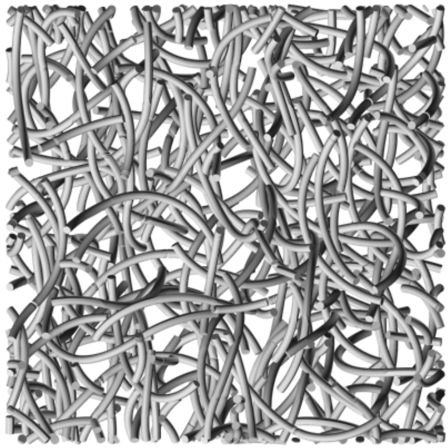
(a) *Distribution of θ, centred around π.*



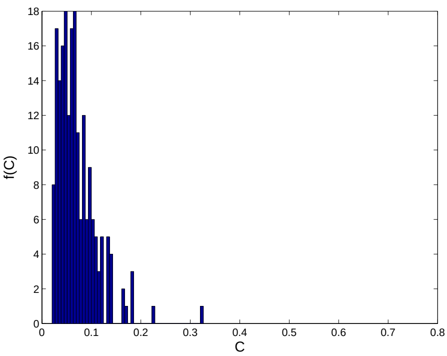
(b) *Distribution of θ, centred around 0*



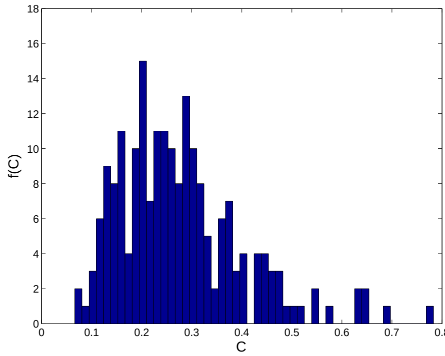
(c) *Rendered network with distribution (a).*



(d) *Rendered network with distribution (b).*



(e) *Distribution of curl in network (c).*



(f) *Distribution of curl in network (d).*

Figure 7.6: *Example of the influence of segment-level parameter θ on network curl distribution.*

orientation which can be found in for example paper, is possible.

So far, all networks have been generated using `gennetbeta.m`. If designed fibres are wanted in a network, `gennetlib.m` should be used instead. Figure 7.8 shows a network generated using this. Two different fibre types are used, an almost full circle, and a three segment pure S-shape. For clarity the different fibre types have been given different colours.

7.4 Networks similar to real materials

To evaluate the developed geometry model photographs of real fibre materials were used to try and duplicate the appearance with a generated network. A number of different types of fibre materials have been looked at. A number of figures following show different fibre materials along with attempts to create generated networks resembling the real materials on the right. It should be taken into account when studying these pictures that, since the system described in figure 1.1 is incomplete, no knowledge of actual parameter distributions in the actual materials exists. Thus the parameters used when generating the networks are simply guesses from studying the photographs of the materials. Deeper knowledge of the materials would no doubt give better resemblance.

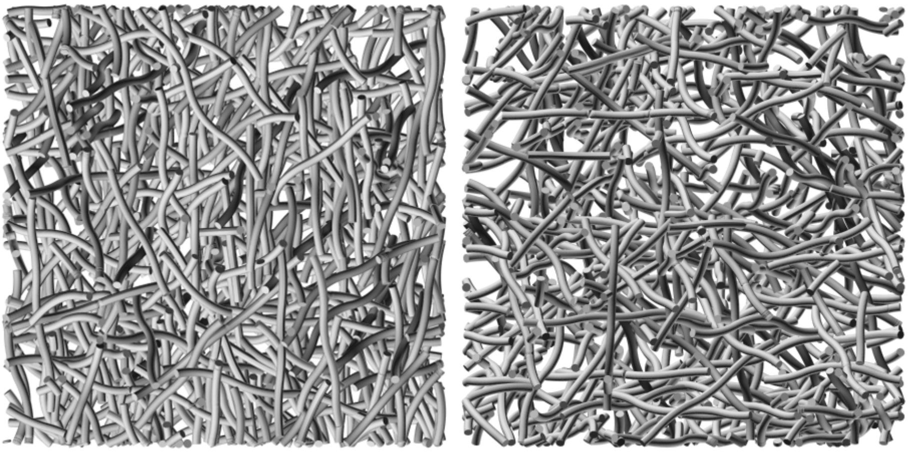
Although the coating applied to the paper in figure 7.9 hides the structure of the fibres on the surface of the paper, the generated network displays a dense paper-like network where the the fibres lie in planar layers. It must be stressed though, especially for dense networks such as this, that the developed model does not take into account fibres occupying the same space. The photograph also shows that the fibres are not circular in cross section, instead a more elongated, flattened shape is displayed. This also points to a weakness of the model, which assumes only circular cross sections.

Figure 7.10 shows a highly magnified photograph of carded polyamide-6 fibres and a generated network. The smooth and seemingly circular shape of the fibres makes it quite suitable to model. The appearance of the generated network also has a striking resemblance to the original.

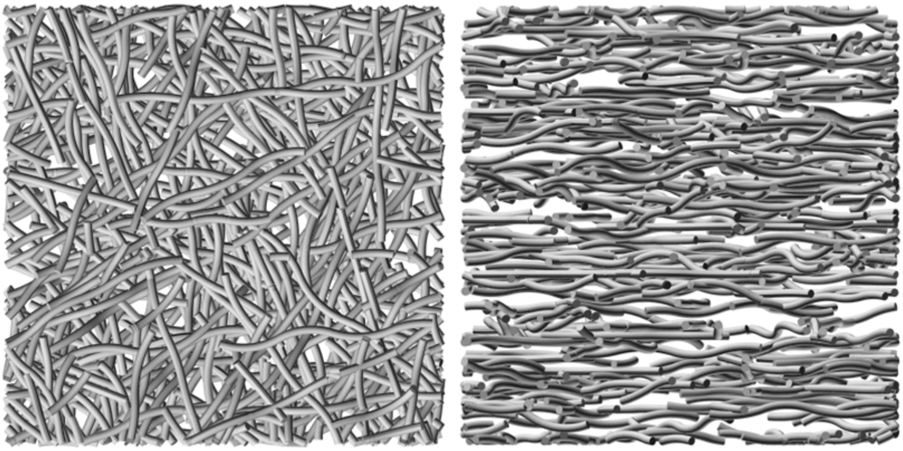
Figure 7.11 shows a close-up of an acoustic board made of wood wool fixated by cement. The structure of the material resembles that of paper products in its layered build up, except on a larger scale. Also in this case, the resemblance of the generated network is good.

So far, we have only looked at surface structures and extreme close-ups of fibre materials when evaluating the model. A photograph of a thin slice of three-layered paper can be seen in figure 7.4. The bottom picture is a section of generated network produced in 3ds max™ . At a quick glance, resemblance is fair, however, the comparison shows that fibre cross section shape plays a large role in the appearance of the material. There is also a large variation in cross section shape. The varying density of the three different layers in the paper is not modelled in the generated network.

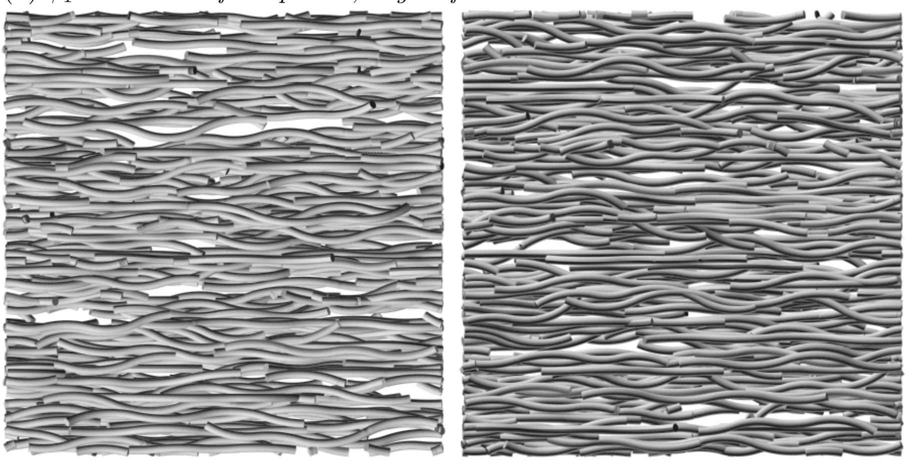
Finally, the complicated geometry of fibre fluff was reproduced. With fibres as



(a) *Isotropic network. Left: top view, Right: front view.*



(b) $\phi_1 = const.$ *Left: top view, Right: front view.*



(c) $\phi_1 = \phi_2 = const.$ *Left: top view, Right: front view.*

Figure 7.7: *Example networks with varying input for the fibre orientation angles* ϕ_1 *and* ϕ_2.

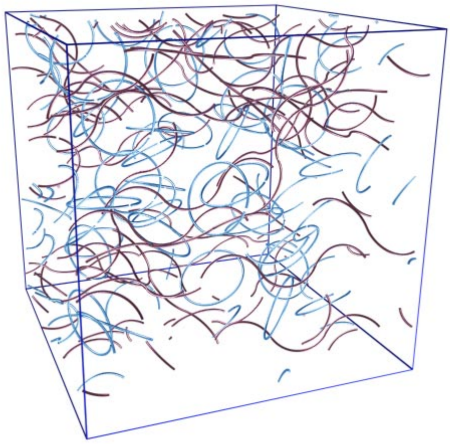


Figure 7.8: *Example of network generated with `gennetlib.m`.*

long and crooked as those in figure 7.4 and with many kinks, the model must be made with many fibre segments. In the accompanying modelled network, seven segments were used per fibre and also a somewhat different distribution of segment radii. In the beta function, using a q-value of 0.7 gives a distribution that favours values of r_{seg} close to the specified distribution limits. This was used to resemble the appearance of the real fibres where often long straight parts are followed by extremely curled areas on the same fibre. The middle picture shows a generated network with a circular cross-section. This gives a satisfactory likeness to the real material, however, to show the effects of having a more representative cross-section shape, a network was visualised using rectangular cross-section. It must be kept in mind though, that the orientation of the cross section cannot be controlled in the model. The likeness to the real material is substantially improved.

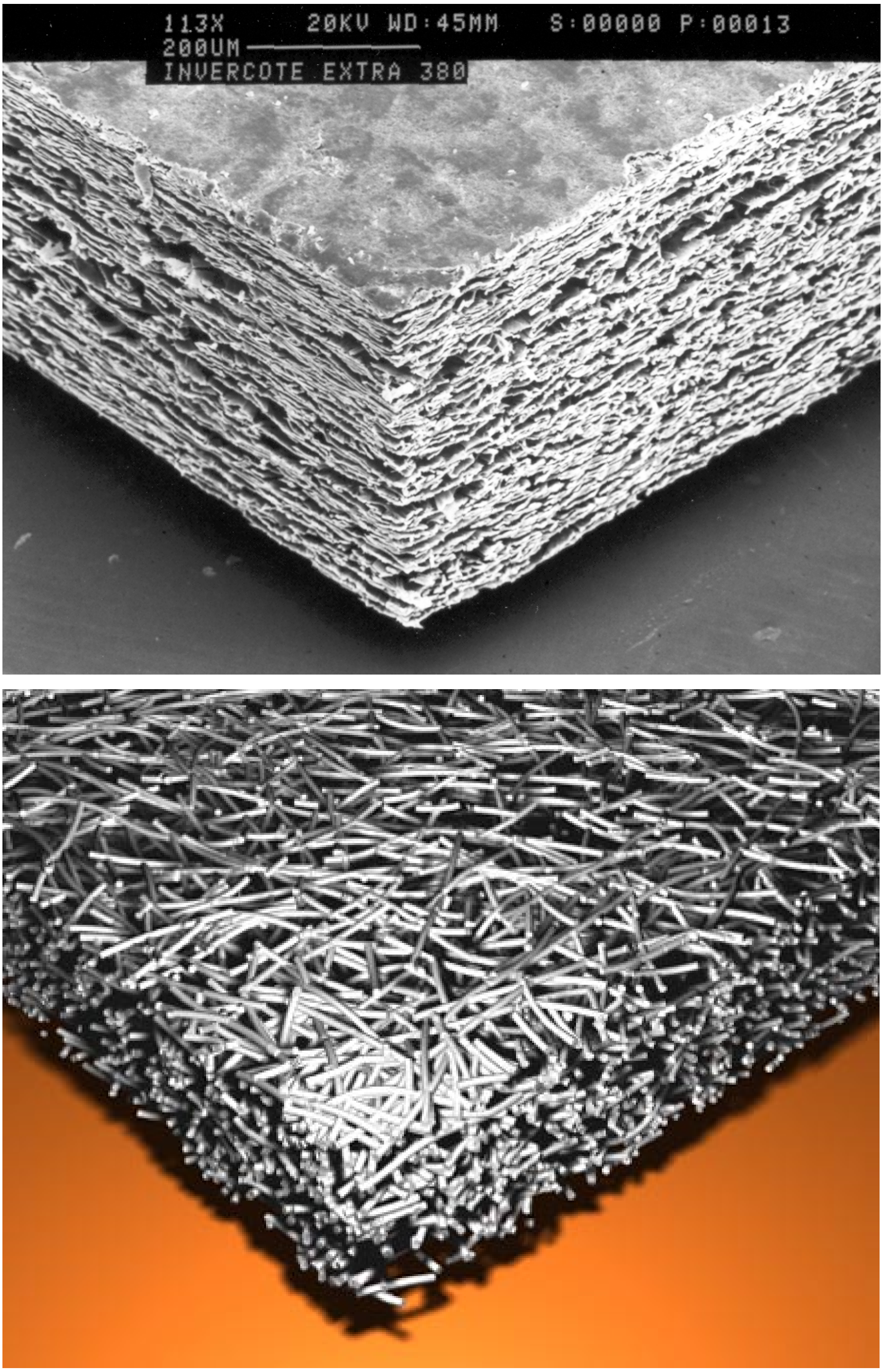


Figure 7.9: *Coated paper seen from an angle (top) and generated network (bottom).*

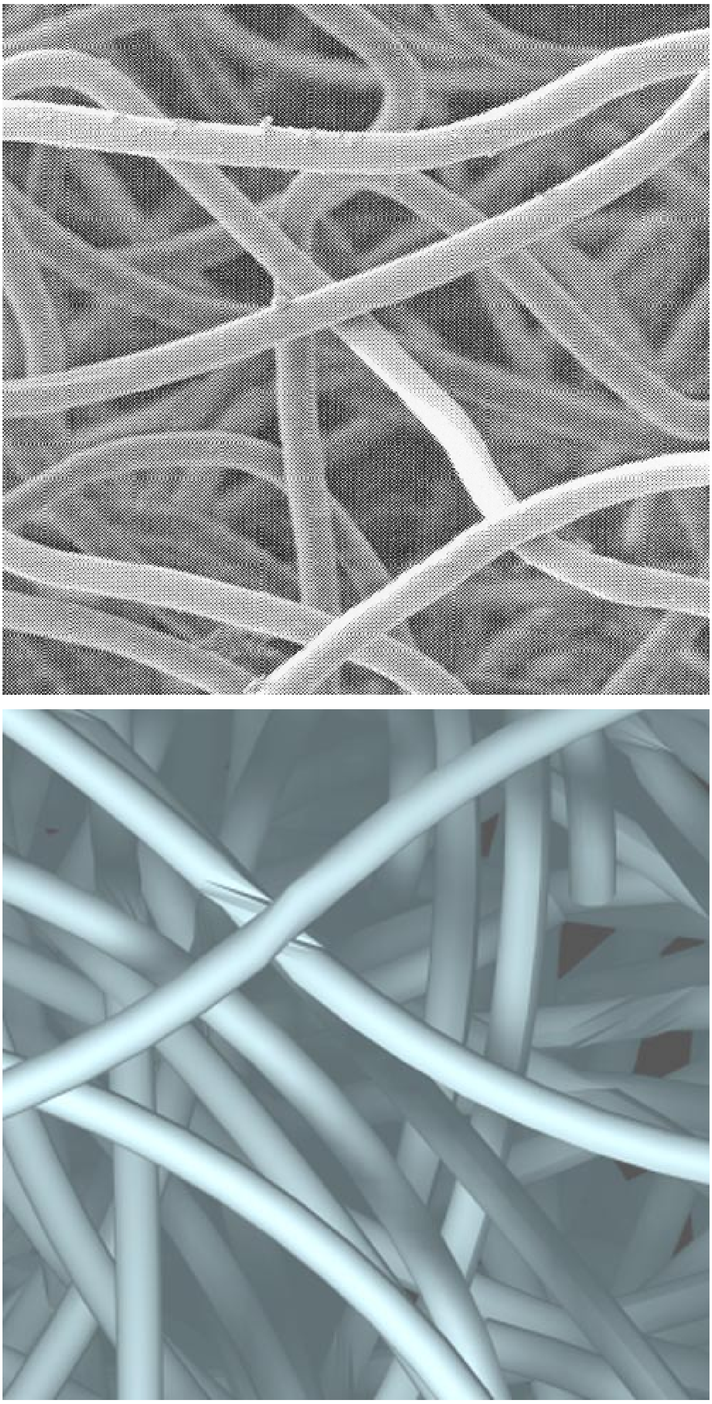


Figure 7.10: *Carded polyamide-6 fibres, diameter 50μm (top) and generated network (bottom).*

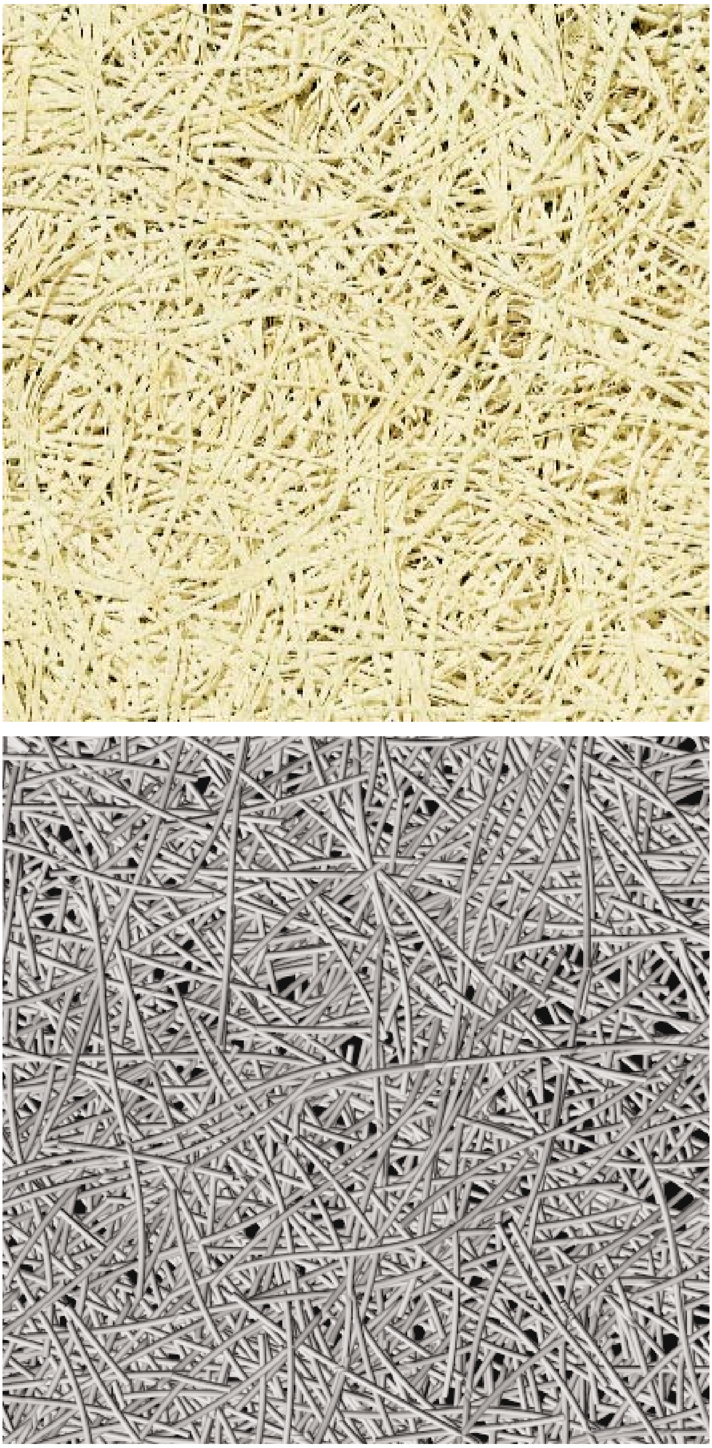


Figure 7.11: *Wood wool acoustic board "T-Akustik Diskret" (top) and generated network (bottom).*

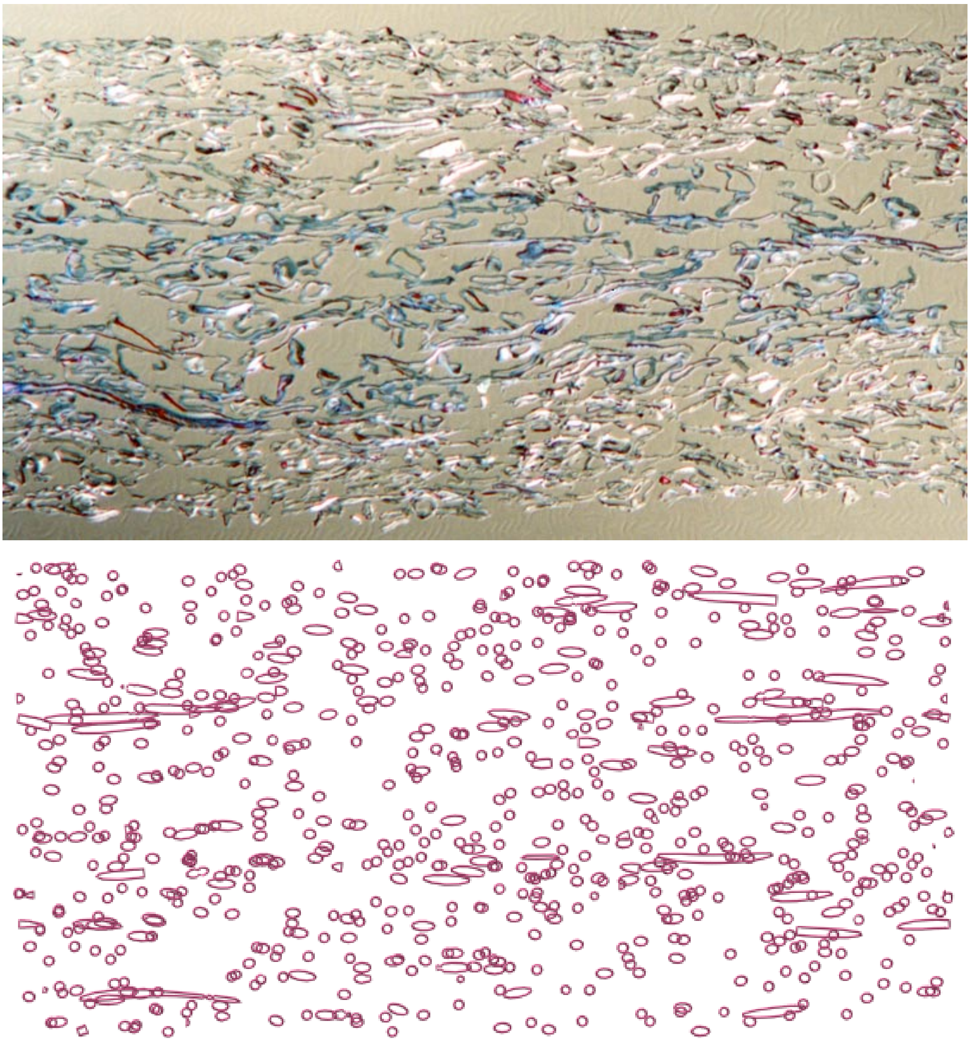


Figure 7.12: *Photograph of a section slice of a three-layered paper seen from the side (top) and a section created from a generated network (bottom).*

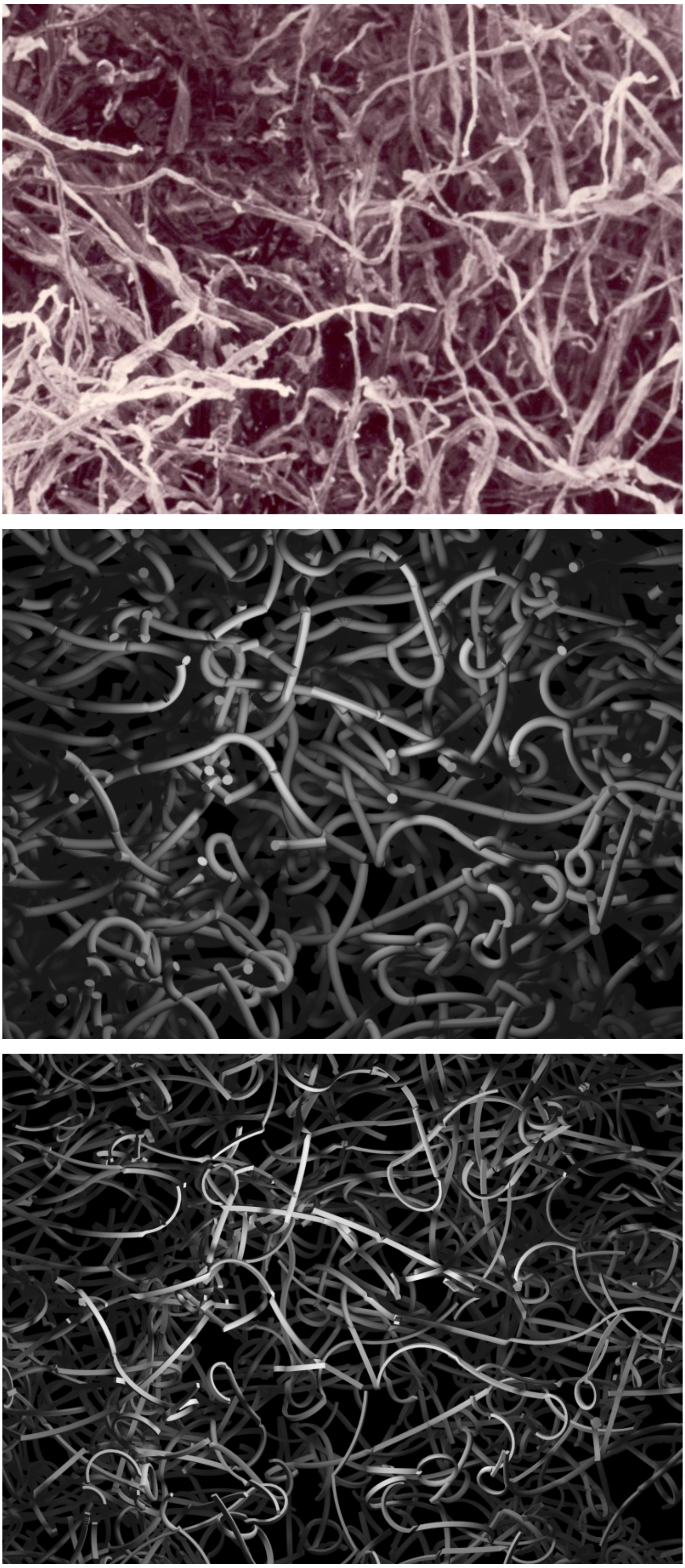


Figure 7.13: *Microscope photograph of cellulose fibre fluff (top) and generated net-*

8 Conclusions

A geometry model for describing any fibre material has been proposed and implemented according to the goals specified in figure 1.1. Although further work needs to be done to complete the system, the applications section of this work has shown that the geometry part of the system works well. It can model most fibres in a satisfactory way and the likeness of the visualized networks to existing materials is good though it lacks cross-section handling.

8.1 Further work

As mentioned in the introduction, the work done in this thesis is thought to form the foundation of a larger system where finite element models can be made from the original geometry of a fibre material. To complete this, further work must be done. A connection between a three dimensional computer tomography and the geometry model must be established. Also, routines for generating the finite element model are needed before the system is finished.

As was shown in chapter 7.4 the fibre cross-section shape and its orientation has an evident effect on network appearance, an effect which also will have consequences for the future finite element model realism. Therefore, it it of interest to look at the implementation of the handling of these variables in future work. For further realism, allowing for a varying cross-section along the length of a fibre might be of interest.

References

[1] Discreet
www.discreet.com/products/3dsmax/ 2003-01-16

[2] e4s, Textiles Online
www.e4s.org.uk/textilesonline/content/6library/report3/fibre_production.htm
2003-01-16

[3] Hamlen, R.C. *Paper Structure, mechanics, and permeability: Computer-aided modelling.* Ph.D. thesis, University of Minnesotam USA, 1991.

[4] Herman, G. *Image Reconstruction From Projections – The Fundamentals of Computerized Tomography* Computer Science and Applied Mathematics, Academic Press, 1980.

[5] Heyden, S. *Network Modelling for the Evaluation of Mechanical Properties of Cellulose Fibre Fluff.* Lund University, Division of Structural Mechanics, Lund, Sweden, TVSM-1011 2000.

[6] ICPE
www.icpe.ro/lab/comp-ndt/tomo/ 2003-01-16

[7] Kallmes, O. and Corte, H. *The Structure of Paper, I. The Statistical Geometry of an Ideal Two Dimensional Fiber Network.*Tappi, Vol. 43, pp. 737-752, 1960.

[8] Krenk, S. *A General Format for Curved and Non-Homogenous Beam Elements* Computers & Structures, 50(4), pp. 449-454, 1994.

[9] Lindemann, J. and Dahlblom,O. *Real-time visualisation of fibre networks*, The Visual Computer, 2002, Volume 18, p20-28, Springer Verlag AG

[10] Lindemann, J. *Programming and Visualisation Techniques in Finite Element Software*, Licentiate Thesis, TVSM-3050, Division of Structural Mechanics, Lund University, 2001.

[11] Mayfield Clinic, University of Chicago
www.mayfieldclinic.com/PE/PE-CT.htm 2003-01-16

[12] Melchers, R.E. *Structural Reliability Analysis and Prediction.* Wiley, Second Edition, 1999.

[13] Molecular expressions
http://micro.magnet.fsu.edu/optics/intelplay/gallery/pages/rheinberg/fibers.html
2003-01-16

[14] Niskanen, K.J. *Strength and Fracture of Paper.* KCL Paper Science Centre, Finland, 1993.

[15] Olsson, P. *Visualisering av deformations- och brottförlopp i nätverksmaterial* Lund University, Division of Structural Mechanics, Lund, Sweden, TVSM-5086

[16] Ottosen N. & Peterson H. *Introduction to the Finite Element Method* University of Lund, Sweden

[17] STFI
www.stfi.se/documents/contract/pulp/fibermart.htm 2003-01-16

[18] The MathWorks
http://mathworks.com 2003-01-16

[19] Wang, H. and Shaler, S.M. *Computer-Simulated Three-Dimensional Microstructure of Wood Fibre Composite Materials* Journal of Pulp and Paper Science: Vol 24 no. 10 October 1998.

[20] Web 3D Consortium
www.web3d.org 2003-01-16

Appendix A

gennetbeta.m

```
function [netdat,rho,curl,length]=...
    gennetbeta(bounds,rho,fphi1,fphi2,flfib,nseg,fr,ftheta,fthetak1,fthetak2,pk)

% Generates periodic fibre network
%
%  [netdat,rho,curl,length]=...
%   gennetbeta(bounds,rho,fphi1,fphi2,flfib,nseg,fr,ftheta,pk)
%  [netdat,rho]=gennetbeta(bounds,rho,fphi1,fphi2,flfib,nseg,fr,ftheta,pk)
%
% INPUT:
% bounds    - network boundingbox widths [Lx Ly Lz]
% rho       - network density (fibrelength/volume)
% nseg      - number of segments used per fibre
% pk        - probability of kink between segments
%
% Distribution inputs, for beta distribution [a b q] where
% a and b are lower resp. upper limits and r=q (symmetric)
%
% fphi1     - distribution of fibreorientation angle phi1
% fphi2     - distribution of fibreorientation angle phi2
% flfib     - distribution of fibre length
% fr        - distribution of segment radii
% ftheta    - distribution of segment orientation angle
% fthetak1  - distribution of kink angle 1
% fthetak2  - distribution of kink angle 2
%
% OUTPUT:
% netdat    - 3d matrix containing radius of curvature,
%             centre point, alfa1, alfa2, u and v for all
%             fibres dim: [nfib nseg 12]
% rho       - network density (fibrelength/volume)
% curl      - vector containing curl of all fibres
% length    - vector containing length of all fibres

nfib=floor(rho*prod(bounds)/(flfib(1)/2));
n=1;
flseg=[flfib(1)/nseg flfib(2)/nseg (2*flfib(3)-nseg+1)/(2*nseg)];
h=waitbar(0,'Generating Network. Please wait...');
for fibnr=1:nfib
    waitbar(fibnr/nfib);
```

```
    for segnr=1:nseg
        R(segnr)=betarnd(fr(3),fr(3))*(fr(2)-fr(1))+fr(1);
        THETA(segnr)=betarnd(ftheta(3),ftheta(3))*(ftheta(2)-ftheta(1))+ftheta(1);
        lseg=betarnd(flseg(3),flseg(3))*(flseg(2)-flseg(1))+flseg(1);
        ALFA(segnr)=lseg/R(segnr);
        if rand<pk
            THETAK1(segnr)=betarnd(fthetak1(3),fthetak1(3))...
                *(fthetak1(2)-fthetak1(1))+fthetak1(1);
            THETAK2(segnr)=betarnd(fthetak2(3),fthetak2(3))...
                *(fthetak2(2)-fthetak2(1))+fthetak2(1);
        else
            THETAK1(segnr)=0;
            THETAK2(segnr)=0;
        end
    end
    if nargout==4
        [fibdat,curl(fibnr),length(fibnr)]=genfib3(nseg,R,THETA,ALFA,THETAK1,THETAK
    else
        [fibdat]=genfib3(nseg,R,THETA,ALFA,THETAK1,THETAK2);
    end
    mp=[rand*bounds(1) rand*bounds(2) rand*bounds(3)];
    phi1=betarnd(fphi1(3),fphi1(3))*(fphi1(2)-fphi1(1))+fphi1(1);
    phi2=betarnd(fphi2(3),fphi2(3))*(fphi2(2)-fphi2(1))+fphi2(1);
    fibdir=rot3daxl([0 1 0],phi1,[1 0 0]);
    fibdir=rot3daxl([0 0 1],phi2,fibdir);
    [fibdat]=place(fibdat,fibdir,mp);
    [fibdatmod]=modfib(fibdat,bounds);
    for i=1:size(fibdatmod,1)
        netdat(n,:,:)=fibdatmod(i,:,:);
        n=n+1;
    end
end
if nargout==4
    rho=sum(length)/prod(bounds);
end
close(h);
```

gennetlib.m

```
function [netdat]=gennetlib(bounds,nfib,fiblib,fibpart,fphi1,fphi2)
```

```
% Generates periodic fibre network
%
%  [netdat]=gennetlib(bounds,nfib,fiblib,fibpart,fphi1,fphi2)
%
% INPUT:
% bounds    - network boundingbox widths [Lx Ly Lz]
% nfib      - number of fibres
% fiblib    - library of fibres to be placed into the network
%             dim: [number of fibretypes, 12]
% fibpart   - part of network using fibretype (index)
%             sum(fibpart)=1
%
% Distribution inputs, for beta distribution [a b q] where
% a and b are lower resp. upper limits and r=q (symmetric)
%
% fphi1     - distribution of fibreorientation angle phi1
% fphi2     - distribution of fibreorientation angle phi2
%
% OUTPUT:
% netdat    - 3d matrix containing radius of curvature,
%             centre point, alfa1, alfa2, u and v for all
%             fibres dim: [nfib nseg 12]

fib=0;
n=1;
h=waitbar(0,'Generating Network. Please wait...');
for fibtypenr=1:size(fiblib,1)
    for fibnr=1:round(fibpart(fibtypenr)*nfib)
        fib=fib+1;
        waitbar(fib/nfib);
        mp=[rand*bounds(1) rand*bounds(2) rand*bounds(3)];
        phi1=betarnd(fphi1(2),fphi1(2))*fphi1(1);
        phi2=betarnd(fphi2(2),fphi2(2))*fphi2(1);
        fibdir=rot3daxl([0 1 0],phi1,[1 0 0]);
        fibdir=rot3daxl([0 0 1],phi2,fibdir);
        [fibdat]=place(squeeze(fiblib(fibtypenr,:,:)),fibdir,mp);
        [fibdatmod]=modfib(fibdat,bounds);
        for i=1:size(fibdatmod,1)
            netdat(n,:,:)=fibdatmod(i,:,:);
            n=n+1;
        end
    end
end
```

```
% rho=sum(length)/prod(bounds);
close(h);
```

genfib3.m

```
function [fibdat,crl,l]=genfib3(nseg,rvect,thetavect,alfavect,thetak1vect,thetak2ve

% Subroutine used in gennetbeta and gennetlib
% Generates one fibre
%
% [fibdat,crl,l]=genfib3(nseg,rvect,thetavect,alfavect,kinkprob)
%
% INPUT:
% nseg      - number of segments
% rvect     - vector containing segment radii
% thetavect - vector containing segment orientation angles
% alfavect  - vector containing segment opening angles
% kinkprob  - probability of kink between segments
%
% OUTPUT:
% fibdat    - vector containing radius of curvature,
%             centre point, alfa1, alfa2, u and v for
%             all fibre segments
% crl       - fibre curl value
% l         - fibre length

fibdat=zeros(nseg,12);
outdir=[1 0 0]; n=[0 1 0];
l=0; ep=[0 0 0]; tec=[0 1 0];

%--- generering av fibern ---
for segnr=1:nseg
    [fibdat(segnr,:),outdir,ep,n,tec]=genfibseg(ep,rvect(segnr),alfavect(segnr)...
        ,thetavect(segnr),thetak1vect(segnr),thetak2vect(segnr),outdir,tec,n);
end

%--- fiberdata ---
if nargout==3
    [crl,l]=curl(fibdat,20);
```

```
end
spep=ep;

% %--- centrering av fibern ---
newc=ep/2;
for segnr=1:nseg
    fibdat(segnr,2:4)=fibdat(segnr,2:4)-newc;
end

%------------------------------

fibn=cross(spep,[1 0 0]);
fibn=fibn./(sqrt(fibn*fibn'));
spep=spep./sqrt(spep*spep');
alfa=acos([1 0 0]*spep');
for i=1:nseg
    r=fibdat(i,1);c=fibdat(i,2:4);alfa1=fibdat(i,5);alfa2=fibdat(i,6);u=fibdat(i,7

    %steg 3
    fibdat(i,2:4)=rot3daxl(fibn,alfa,c);
    fibdat(i,7:9)=rot3daxl(fibn,alfa,u+c);
    fibdat(i,10:12)=rot3daxl(fibn,alfa,v+c);
    fibdat(i,7:9)=fibdat(i,7:9)-fibdat(i,2:4);
    fibdat(i,10:12)=fibdat(i,10:12)-fibdat(i,2:4);
    if i==1
        sp=c+r*(cos(alfa1)*u+sin(alfa1)*v);
    end
end

%-----------------------------------------------------------------
```

genfibseg.m

```
function [fibdat,outdir,ep,n,tec]=...
    genfibseg(sp,r,beta,theta,thetak1,thetak2,indir,tec,nprev)

% Subroutine used in genfib3
% Generates one fibresegment
%
```

```
% [fibdat,outdir,ep,n,tec]=...
%    genfibseg(sp,r,beta,theta,thetak1,thetak2,indir,tec,nprev)
%
% INPUT:
% sp         - startingpoint of fibre segment
% r          - radius of curvature
% alfa       - angle between starting and ending tangents
% theta      - rotationangle of fibresegment
% indir      - startingtangent
%
% OUTPUT:
% fibdat     - vector containing radius of curvature,
%              centre point, alfa1, alfa2, u and v
% outdir     - endtangent
% ep         - endpoint
% n          - fibre segment normal vector
% tec        - vector pointing from endpoint to centrepoint

alfa=pi-beta;
b=r*beta;
x=2*r*sin(beta/2);
l=x/(2*sin(alfa/2));

indir=rot3daxl(nprev,thetak1,indir);
tsc=rot3daxl(nprev,thetak1,tec);
indir=rot3daxl(tsc,thetak2,indir);
tsc=rot3daxl(indir,theta,tsc);

n=cross(indir,tsc);
n=n/sqrt(sum(n*n'));

outdir=rot3daxl(n,pi/2,indir);
outdir=rot3daxl(n,pi/2-alfa,outdir);
outdir=outdir/(sqrt(outdir*outdir'));

cp=sp+l*indir;
ep=cp+l*outdir;

%BASE VECTORS IN THE FIBRE PLANE U V
%if n is in the xy-plane
if (n(3)<1/10000)
    u=[n(2) -n(1) 0];
    if (u(1)<0)
        u=-u;
```

```
    end
else
    u=(1+(n(1)^2/n(3)^2))^-0.5*[1 0 -n(1)/n(3)];
end
u=u/sqrt(sum(u*u'));
v=cross(n,u);
v=v/sqrt(sum(v*v'));

rv=cross(n,indir);
c=sp+rv*r;

tec=c-ep; tec=tec./sqrt(tec*tec');

%ANGLES ALFA1, ALFA2
ts=sp-c;
ts=ts/sqrt(ts*ts');
alfa1=acos(ts*u');
if acos(ts*v')>pi/2
    alfa1=2*pi-alfa1;
end
alfa2=alfa1+beta;
if (alfa1<0)
  alfa1=alfa1+2*pi;
end
if (alfa2>2*pi)
  alfa2=alfa2-2*pi;
end

fibdat=[r c alfa1 alfa2 u v];
```

place.m

```
function [fibdat]=place(fibdat,fibdir,mp)

% Places one fibre into a network
%
%  [fibdat]=place(fibdat,fibdir,mp)
%
% INPUT:
```

```
% fibdat    - vector containing radius of curvature,
%             centre point, alfa1, alfa2, u and v for the fibre
% fibdir    - specified orientation of the fibre in the network
% mp        - specified midpoint of fibre in network
%
% OUTPUT:
% fibdat    - modified fibre data

nseg=size(fibdat,1);
fibdir=fibdir/sqrt(fibdir*fibdir');
fibn=cross(fibdir,[1 0 0]);  %normal till planet som spänns upp av fiberns önskade
fibn=fibn/sqrt(fibn*fibn');
alfa=-acos(fibdir*[1 0 0]');
beta=rand*2*pi;

%steg 1
for i=1:nseg
r=fibdat(i,1);c=fibdat(i,2:4);alfa1=fibdat(i,5);alfa2=fibdat(i,6);u=fibdat(i,7:9);v

%steg 3
fibdat(i,2:4)=rot3daxl(fibn,alfa,c);
fibdat(i,7:9)=rot3daxl(fibn,alfa,u+c);
fibdat(i,10:12)=rot3daxl(fibn,alfa,v+c);
fibdat(i,7:9)=fibdat(i,7:9)-fibdat(i,2:4);
fibdat(i,10:12)=fibdat(i,10:12)-fibdat(i,2:4);

%Tranposing of ceter points
fibdat(i,2:4)=fibdat(i,2:4)+mp;

%Random rotation around own axis
fibdat(i,2:4)=rot3daxl(fibdir,beta,fibdat(i,2:4));
fibdat(i,7:9)=rot3daxl(fibdir,beta,fibdat(i,7:9));
fibdat(i,10:12)=rot3daxl(fibdir,beta,fibdat(i,10:12));
end
```

modfibseg.m

```
function [segdatmod]=modfibseg(segdat,cutplanes,cut,bounds)
```

```
% Subroutine used in modfib
% Modifies one fibre segment according to network boundaries
%
% [segdatmod]=modfibseg(segdat,cutplanes,cut,bounds)
%
% INPUT:
% segdat    - vector containing radius of curvature,
%             centre point, alfa1, alfa2, u and v
% cutplanes - matrix of planes to be controlled
%             example: plane x=100 [1 0 0 100]
% cut       - boolean vector: 1 if fibre is to be cut at actual plane
% bounds    - network boundingbox widths [Lx Ly Lz]
%
% OUTPUT:
% segdatmod - vector containing modified values for
%             radius of curvature, centre point,
%             alfa1, alfa2, u and v

ind=2;
alfavect(1)=segdat(5);
alfavect(2)=segdat(6);

if cut==1
    for plane=1:size(cutplanes,1)
        [x1,x2,found]=cross_c_p3d(segdat,cutplanes(plane,1),cutplanes(plane,2),cut
        if (found==1)
            [onarc,alfa]=isonarc(segdat,x1);
            if(onarc==1)
                ind=ind+1;
                alfavect(ind)=alfa;
            end
            [onarc,alfa]=isonarc(segdat,x2);
            if(onarc==1)
                ind=ind+1;
                alfavect(ind)=alfa;
            end
        end
    end
end

[alfavect]=asort3d(alfavect,ind);

for i=2:length(alfavect)
    shift=[0 0 0];
```

```
         if alfavect(i-1)>alfavect(i)
                alfac=alfavect(i-1)+.5*(2*pi-alfavect(i-1)+alfavect(i));
                if alfac>2*pi
                    alfac=alfac-2*pi;
                end
            else
                alfac=alfavect(i-1)+.5*(alfavect(i)-alfavect(i-1));
            end

    p=segdat(2:4)+segdat(1)*(cos(alfac)*segdat(7:9)+sin(alfac)*segdat(10:12));
    shift(1)=bounds(1)*floor(p(1)/bounds(1));
    shift(2)=bounds(2)*floor(p(2)/bounds(2));
    shift(3)=bounds(3)*floor(p(3)/bounds(3));

    segdatmod(i-1,:)=segdat;
    segdatmod(i-1,2:4)=segdatmod(i-1,2:4)-shift;
    segdatmod(i-1,5)=alfavect(i-1);
    segdatmod(i-1,6)=alfavect(i);
end
```

cross_c_p3d.m

```
% cross_c_p3d.m

function [x1,x2,found]=cross_c_p3d(fibdat,A,B,C,D)

% Function used by modfib3d in program fibre3d.
% Calculate crossings of a plane and a circle in space.
% Circle  x(theta)=c+r*cos(theta)*u+r*sin(theta)*v
% Plane   ax+by+cz+d=0
% u and v must be of length 1.
% Susanne Heyden   970911

r=fibdat(1); c=fibdat(2:4); u=fibdat(7:9); v=fibdat(10:12);

found=1;
x1=-1;
x2=-1;
```

```
c1=A*r*u(1)+B*r*u(2)+C*r*u(3);
c2=A*r*v(1)+B*r*v(2)+C*r*v(3);
c3=A*c(1)+B*c(2)+C*c(3)+D;
c4=sqrt(c1^2+c2^2);

%If plane and plane of circle are parallel
if (abs(c4)<(r/20))
 found=0;
 break
end

if (c2/c4>0)
  phi=asin(c1/c4);
elseif (c1/c4>0)
  phi=acos(c2/c4);
else
  phi=acos(c2/c4)-2*asin(c1/c4);
end

theta1=asin(-c3/c4)-phi;
theta2=pi-asin(-c3/c4)-phi;


%If no crossings
if (abs(imag(theta1))>0)
 found=0;
 break
end

%If circle is tangent to plane
if (abs(theta1-theta2)<0.01)
 found=0;
 break
end

x1=zeros(3,1);
x1(1)=c(1)+r*cos(theta1)*u(1)+r*sin(theta1)*v(1);
x1(2)=c(2)+r*cos(theta1)*u(2)+r*sin(theta1)*v(2);
x1(3)=c(3)+r*cos(theta1)*u(3)+r*sin(theta1)*v(3);
diff1=A*x1(1)+B*x1(2)+C*x1(3)+D;

x2=zeros(3,1);
x2(1)=c(1)+r*cos(theta2)*u(1)+r*sin(theta2)*v(1);
```

```
x2(2)=c(2)+r*cos(theta2)*u(2)+r*sin(theta2)*v(2);
x2(3)=c(3)+r*cos(theta2)*u(3)+r*sin(theta2)*v(3);
diff2=A*x2(1)+B*x2(2)+C*x2(3)+D;
```

fibre2vrml.m

```
function fibre2vrml(filename,fibdat,prec,t)

% Exports fibre network geometry to vrml file format
%
%  fibre2vrml(filename,fibdat,prec,t)
%
% INPUT:
% filename  - location and name of exported file
% fibdat    - vector containing radius of curvature,
%             centre point, alfa1, alfa2, u and v for the network
% prec      - number of straight lines to represent one fibre segment
% t         - diameter of the fibres circular ross-section

nfibs=size(fibdat,1);
narcs=size(fibdat,2);

%-----------------------------
%---- VRML post processing ----
%-----------------------------

options = [0]; % No global transform;

%-----------------------------------
%---- Create model for geometry ----
%-----------------------------------

fid=vrmlcreate(filename,options);

%---- Define geometry

vrmlbegin(fid,'Transform');
vrmlbeginarr(fid,'children');
%---- Define circular cross section
```

```
i=0;
for angle=0:pi/5:2*pi
    i=i+1;
    sec(i,1)=t*cos(angle);
    sec(i,2)=t*sin(angle);
end

for fib=1:nfibs
    n=0;
    for arc=1:narcs
        r=fibdat(fib,arc,1);
        c=fibdat(fib,arc,2:4);
        alfa1=fibdat(fib,arc,5);
        alfa2=fibdat(fib,arc,6);
        u=fibdat(fib,arc,7:9);
        v=fibdat(fib,arc,10:12);

        if r==0
            break
        end

        for i=1:prec
            n=n+1;
            if alfa1>alfa2
                a2=alfa1+i*(2*pi-alfa1+alfa2)/prec;
                if a2>2*pi
                    a2=a2-2*pi;
                end
            else
                a2=alfa1+i*(alfa2-alfa1)/prec;
            end

            points(n,:)=c+r*(cos(a2)*u+sin(a2)*v);
        end
    end

    vrmlextrusion(fid,points(1:n,:),sec);
end
vrmlendarr(fid);
vrmlend(fid);

vrmlclose(fid,options);
```

fibre2scope.m

```
function fibre2scope(filename,fibdat,prec,t)

% Exports fibre network geometry to nef file format
%
%  fibre2scope(filename,fibdat,prec,t)
%
% INPUT:
% filename  - location and name of exported file
% fibdat    - vector containing radius of curvature,
%             centre point, alfa1, alfa2, u and v for the network
% prec      - number of straight lines to represent one fibre segment
% t         - diameter of the fibres circular ross-section

nfibs=size(fibdat,1);
maxarcs=size(fibdat,2);

fid=fopen(filename,'w');
fprintf(fid,'%12f\n',t);

for fib=1:nfibs
    n=0;
    for arc=1:maxarcs
        r=fibdat(fib,arc,1);
        c=fibdat(fib,arc,2:4);
        alfa1=fibdat(fib,arc,5);
        alfa2=fibdat(fib,arc,6);
        u=fibdat(fib,arc,7:9);
        v=fibdat(fib,arc,10:12);

        if r==0
            arc=arc-1;
            break
        end

        for i=0:prec
            n=n+1;
            if alfa1>alfa2
                a2=alfa1+i*(2*pi-alfa1+alfa2)/prec;
                if a2>2*pi
                    a2=a2-2*pi;
                end
            else
```

```
                a2=alfa1+i*(alfa2-alfa1)/prec;
            end

            points(n,:)=c+r*(cos(a2)*u+sin(a2)*v);
        end
    end

    fprintf(fid,'%12f\n',arc*(prec+1));
    for j=1:n
        fprintf(fid,'%12.8f %12.8f %12.8f\n',points(j,:));
    end
end

fclose(fid);
```

cuttingplanes.m

```
function [planes,cut]=cuttingplanes(segdat,bounds)

% Returns the planes that a fibre segment passes through
%
%  [planes,cut]=cuttingplanes(segdat,bounds)
%
% INPUT:
% segdat    - vector containing radius of curvature,
%             centre point, alfa1, alfa2, u and v for the segment
% bounds    - vector containing unit cell dimensions in x, y, and z
%
% OUTPUT:
% planes    - passed planes, in the form [1 0 0 100] (meaning the plane
%             x=100)
% cut       - vector of same length as planes. value 1 if the plane is to
%             be cut, otherwise 0

r=segdat(1);
c=segdat(2:4);
alfa1=segdat(5);
alfa2=segdat(6);
u=segdat(7:9);
```

```
v=segdat(10:12);

seg=20;
index=0;
index2=0;
index3=0;
cut=1;

for i=0:seg
    if alfa1>alfa2
        alfa=alfa1+i*(2*pi-alfa1+alfa2)/seg;
        if alfa>2*pi
            alfa=alfa-2*pi;
        end
    else
        alfa=alfa1+i*(alfa2-alfa1)/seg;
    end
    p=c+r*(cos(alfa)*u+sin(alfa)*v);

    for j=1:3
        boxtemp(j)=floor(p(j)/bounds(j));
    end

    if i==0
        index=index+1;
        box(index,:)=boxtemp;
    elseif sum(box(index,:)~=boxtemp)~=0
        index=index+1;
        box(index,:)=boxtemp;
    end

    for j=1:index-1
        a=box(j,1:3)~=box(j+1,1:3);
        if a(1)==1
            index3=index3+1;
            planetemp(index3,:)=[1 0 0 max(box(j,1),box(j+1,1))*bounds(1)];
        end
        if a(2)==1
            index3=index3+1;
            planetemp(index3,:)=[0 1 0 max(box(j,2),box(j+1,2))*bounds(2)];
        end
        if a(3)==1
            index3=index3+1;
            planetemp(index3,:)=[0 0 1 max(box(j,3),box(j+1,3))*bounds(3)];
```

```
        end

        for j=1:index3
            if exist('planes','var')==0
                index2=index2+1;
                planes(index2,:)=planetemp(index3,:);
            else
                add=0;
                for s=1:index2
                    if sum(planes(s,:)==planetemp(index3,:))==4
                        add=1;
                    end
                end
                if add==0
                    index2=index2+1;
                    planes(index2,:)=planetemp(index3,:);
                end
            end
        end
    end
end
if exist('planes','var')==0
    planes=0;
    cut=0;
end
```

isonarc.m

```
% isonarc3d.m

function [onarc,alfa]=isonarc3d(fibdat,x)

% Function used by modfib3d.m in program fibre3d.
% Onarc is true if the point (xyz) which is on the circle is
% also on the circle arc defined by center x0, radius r, and
% situated between angles alfa1, alfa2. Alfa is the angle of
% point x.
```

```
r=fibdat(1); x0=fibdat(2:4)'; alfa1=fibdat(5); alfa2=fibdat(6);
u=fibdat(7:9)'; v=fibdat(10:12)';

cxv=sum((x-x0).*v);
alfa=acos(sum((x-x0).*u)/r);
if (sign(cxv)<0)
  alfa=2*pi-alfa;
end

if (alfa1<alfa2)
  if((alfa1<alfa)&(alfa<alfa2))
    onarc=1;
  else
    onarc=0;
  end
else
  if ((alfa>alfa1)|(alfa<alfa2))
    onarc=1;
  else
    onarc=0;
  end
end
```

plotfibre.m

```
function plotfibre(fibdat,color)

for s=1:min(size(fibdat))
    r=fibdat(s,1);
    c=fibdat(s,2:4);
    alfa1=fibdat(s,5);
    alfa2=fibdat(s,6);
    u=fibdat(s,7:9);
    v=fibdat(s,10:12);
    n=cross(u,v);

    seg=20;

    a1=alfa1;
```

```
    for i=1:seg
        if alfa1>alfa2
            a2=alfa1+i*(2*pi-alfa1+alfa2)/seg;
            if a2>2*pi
                a2=a2-2*pi;
            end
        else
            a2=alfa1+i*(alfa2-alfa1)/seg;
        end

        X1=c+r*(cos(a1)*u+sin(a1)*v);
        X2=c+r*(cos(a2)*u+sin(a2)*v);

        line([X1(1) X2(1)],[X1(2) X2(2)],[X1(3) X2(3)],'Color',color);

        a1=a2;
        pause(.001);
        hold on
        axis equal;
    end
end
```

plotbox.m

```
function plotbox(bounds,col)

line([0 bounds(1)],[0 0],[0 0],'Color',col);
line([0 0],[0 bounds(2)],[0 0],'Color',col);
line([0 0],[0 0],[0 bounds(3)],'Color',col);
line([bounds(1) bounds(1)],[0 bounds(2)],[bounds(3) bounds(3)],'Color',col);
line([bounds(1) bounds(1)],[0 0],[0 bounds(3)],'Color',col);
line([0 bounds(1)],[0 0],[bounds(3) bounds(3)],'Color',col);
line([0 bounds(1)],[bounds(2) bounds(2)],[bounds(3) bounds(3)],'Color',col);
line([0 0],[0 bounds(2)],[bounds(3) bounds(3)],'Color',col);
line([0 0],[bounds(2) bounds(2)],[0 bounds(3)],'Color',col);
line([0 bounds(1)],[bounds(2) bounds(2)],[0 0],'Color',col);
line([bounds(1) bounds(1)],[0 bounds(2)],[0 0],'Color',col);
```

```
line([bounds(1) bounds(1)],[bounds(2) bounds(2)],[0 bounds(3)],'Color',col);

axis equal
```

plotnet.m

```
function plotnet(fibdat,bounds)
figure
axis off
set(gcf,'Color','w')
plotbox(bounds,'r');
for i=1:size(fibdat,1)
    temp=squeeze(fibdat(i,:,:));
    plotfibre(temp,'b');
    title(num2str(i));
end
```

asort3d.m

```
% asort3d.m

function [vect]=asort3d(vect,n)

% Function used by modfib3d in program fibre3d.
% Sorts the angles in vect in their order along
% the circle arc.

alfa1=vect(1);
for i=1:n-1
  minv=1e10;
  for k=i:n
    if (vect(k)<minv)
      minv=vect(k);
      index=k;
    end
```

```
  end
  vect(index)=vect(i);
  vect(i)=minv;
end
while (abs(vect(1)-alfa1)>1e-5)
  a=vect(1);
  for i=1:n-1
    vect(i)=vect(i+1);
  end
  vect(n)=a;
end
```

rot3daxl.m

```
function [v]=rot3daxl(a,theta,u)

% Rotates the vector u theta radians around the axel a
%
%  [v]=rot3daxl(a,theta,u)
%
% INPUT:
% a         - axel of rotation
% theta     - angle of rotation
% u         - vector to rotate
%
% OUTPUT:
% v         - rotated vector

A=[0 a(3) -a(2);-a(3) 0 a(1);a(2) -a(1) 0];
R=eye(3)-sin(theta)*A+(1-cos(theta))*A^2;

% if nargin==3
    v=R*u2';
    v=v';
% elseif nargin==4
%     v2=R*u2';
%     v1=R*u1';
%     v=v2-v1;
%     v=v/sqrt(v'*v);
```

```
%     v=v';
% end
```

curl.m

```
function [curl,length]=curl(fibre,prec)

% Subroutine used in genfib3
% Calculates the curl of a fibre
%
% [curl,length]=curl(fibre,prec)
%
% INPUT:
% fibre     - vector containing radius of curvature,
%             centre point, alfa1, alfa2, u and v
% prec      - precision of calculation, specifies the number of points
%             along each segment controlled
%
% OUTPUT:
% curl      - curl value of fibre
% length    - length of fibre

nseg=size(fibre,1);
length=0;

for i=1:nseg
    alfa=fibre(i,6)-fibre(i,5);
    if alfa<0
        alfa=2*pi+alfa;
    end
    l=alfa*fibre(i,1);
    length=length+l;
end

n=1;
points(1,:)=fibre(1,2:4)+fibre(1,1)*(cos(fibre(1,5))*fibre(1,7:9)+sin(fibre(1,5))*
for i=1:nseg
    alfa1=fibre(i,5); alfa2=fibre(i,6);
    for j=1:prec
```

```
        n=n+1;
        if alfa1>alfa2
            a2=alfa1+j*(2*pi-alfa1+alfa2)/prec;
            if a2>2*pi
                a2=a2-2*pi;
            end
        else
            a2=alfa1+j*(alfa2-alfa1)/prec;
        end
        points(n,:)=fibre(i,2:4)+fibre(i,1)*(cos(a2)*fibre(i,7:9)+sin(a2)*fibre(i,
    end
end

lmax=0;
for i=1:n
    for j=i+1:n
        v=points(i,:)-points(j,:);
        l=sqrt(v*v');
        lmax=max(l,lmax);
    end
end

curl=length/lmax-1;
```