University of Leoben, Austria
Petroleum Engineering

Dissertation

Simulation of Fluid-Structure Interaction using OpenFOAM®: Filtration Processes in Deformable Media

Marianne Mataln, M.Sc.

ICE Strömungsforschung GmbH, Austria
&
Department of Mineral Resources and Petroleum Engineering

1st Advisor:
A.o. Univ. Prof. DI Dr. techn. Wilhelm Brandstätter
Department of Mineral Resources and Petroleum Engineering
University of Leoben, Austria

2nd Advisor:
Univ.-Prof. DI Dr.mont. Thomas Antretter
Institute for Mechanical Engineering
University of Leoben, Austria

October 2010

I declare in lieu of oath, that I wrote this thesis and performed the associated research myself, using only literature cited in this volume.

Abstract

In filtration processes it is necessary to consider the interaction of the fluid with the solid parts and the effect of particles carried in the fluid and accumulated on the solid. Traditionally, for investigation of the driving parameters, such as particle deposition and material influence, destructive tests are used. In order to provide accurate repeatable ambient requirements, simulations offer an attractive alternative. While other related publications deal with the large particle deposition model [2-5], this thesis focussed on the development of a solver to model fibre deformation effects. Pressure and traction forces, induced by fluid motion consequently lead to deformations of the solid part. According to this multi-physical problem, it is necessary to couple the differential equations of fluid motion, namely the Navier-Stokes equations and structural mechanical equations, Hooke's law, for the solid region. For their numerical discretisation only one single computational mesh is used. This grid is changing with time and hence is recalculated at each time step to adjust to the deformation in order to conserve geometric consistency. The derived algorithm was summarized by one single solver and realised with the help of the Open Source, C++ based, computational fluid dynamics tool box OpenFOAM®. It was thoroughly validated by plausibility checks and available experimental data. With this, a strong tool for studying fluid-structure interaction phenomena on microscopic scale was developed. It was successfully applied on realistic, from CT-scans reconstructed fiber materials. Further on it was combined with the Lagrangian particle model. This provides the possibility of simultaneous simulation of all relevant physical phenomena by one single finite volume solver based on OpenFOAM®. Experiments showed a non linear behaviour of pressure drop in dependency of flow rates for soft filter materials. With the help of the newly developed filtration solver it was possible to prove this observation. Further on the particle deposition behaviour for different filter materials was investigated. New insights were gained, which underlined the high influence of the deformation of filter material on the overall filtration process. The final aim of the project is to design a filtration tool for the development and optimization of new high performance filter materials without need for performing time consuming and expensive experimental work.

Kurzfassung

In technischen Filtrationsprozessen sind vor allem zwei Effekte von grundlegender Bedeutung: die Wechselwirkung zwischen dem Fluid mit der Faserstruktur des Filters und die Ablagerung der, mit der Flüssigkeit transportierten Schmutzpartikel. Traditionell werden zur Untersuchung von Filtrationskenngrößen, wie der Filtereffizienz, aufwendige, destruktive Tests angewandt. Um die Zerstörung der Faserstruktur jedoch zu vermeiden und exakt reproduzierbare Bedingungen zu schaffen, bieten sich Simulationen als eine sehr gute Alternative an. Während sich andere, verwandte Publikationen mit der Entwicklung eines Schmutzpartikel-Ablagerungskonzeptes befassen [2-5], bezieht sich diese Arbeit auf die Entwicklung eines Simulationsprogrammes zur Modellierung von Deformationseffekten. Die Deformation der Filterfasern wird infolge der, auf die Oberfläche der Struktur wirkenden strömungsmechanischen Druck- und Scherkräfte ausgelöst. Im Gegenzug erfolgt eine Änderung des Strömungsfeldes durch die Bewegung des Festkörpers. Zur Modellierung dieses Verhaltens ist es erforderlich die strömungsmechanischen Differentialgleichungen des Fluides, die Navier-Stokes Gleichungen und jene der Strukturmechanik des Festkörpers, also das Hook'sche Gesetz, zu koppeln. Um die geometrische Konsistenz zu wahren, wird zur räumlichen Disketisierung ein zeitabhängiges Berechnungsgitter verwendet, welches in jedem Zeitschritt an die Deformation des Festkörpers angepasst wird. Der entwickelte Algorithmus ist in einem einzigen Simulationsprogramm auf Basis der Open Source Simulationstoolbox OpenFOAM® realisiert. Anhand von Plausibilitätskontrollen und verfügbaren, experimentellen Daten wird die entwickelte Software gründlich validiert. Auf diese Weise entsteht ein verlässliches Werkzeug zur Simulation der Fluid-Struktur Interaktion im mikroskopischen Bereich, welches erfolgreich an realistischen, aus CT-Scans rekonstruierten Fasermaterialen angewendet wird. Darüber hinaus wird eine Koppelung mit dem separat entwickelten Langrangen Schmutzpartikelmodell verwirklicht. Diese Verknüpfung ermöglicht somit die gleichzeitige Simulation aller relevanten physikalischen Phänomene in einem einzigen Finite Volumen Simulationsprogramm auf Basis von OpenFOAM®. In Experimenten kann ein nichtlineares Verhalten des Druckabfalles über die Dicke des Faserelements in Abhängigkeit der Durchflussraten beobachtet werden. Eine weitere, durch die Deformation beeinflusste, charakteristische Filtereigenschaft ist die Filtereffizienz, also die Fähigkeit Schmutzpartikel verschiedener Klassierungen aus dem Fluid abzuscheiden. Das neuentwickelte Simulationswerkzeug zur Filtrationsanalyse ermöglicht eine Untersuchung dieser Beobachtungen. In dieser Arbeit werden viele neuartige Erkenntnisse präsentiert, die den hohen Grad des Einflusses der Deformation des Filtermaterials auf den gesamten Filtrationsprozess unterstreichen. Die Zielsetzung dieser Arbeit ist das Design eines Filtrationswerkzeuges zur Entwicklung und Optimierung von neuen Hochleistungsfiltermaterialien ohne die Notwendigkeit von zeitraubenden und kostenintensiven Experimenten.

Acknowledgement

The studies underlying this thesis have been performed at the ICE Strömungsforschung GmbH in cooperation with MAHLE Filtersysteme Austria GmbH.

I would like to thank Prof. W. Brandstätter for giving me the opportunity to work on this thesis and the freedom to develop myself during my studies.

My thanks go to all my colleagues at the ICE Strömungsforschung GmbH for their help, especially to DI Bernhard Gschaider, to whom I owe a great deal of thanks for his persistent support in every programming "case of emergency".

Special thanks are due to Dr. Gernot Boiger who was a good friend and colleague during the whole project. I have a lot of respect for his technical skills and I very much appreciated our numerous conversations.

I wish to express my gratitude towards Prof. Antretter and Prof. Oberaigner from the Institute for Mechanics for the confidence they placed in me and giving me the opportunity to experience the life of a teaching assistant.

Further on, my thanks go to Malcolm Werchota for his help and his talent to provide me with a lot of exciting challenges.

Finally I would like to thank my family for all their support during my whole life, for their persistent help and all the wonderful time we had and still have together.

Table of Contents

1 Introduction

Filtration processes are important and used in a wide variety of industries. Examples therefore include the automotive industry, sewage filter systems, or even facilities present in our household, such as the coffee machine. This thesis will deal with oil filters, which are in widespread use within the automotive industry, Figure 1.1. Motor oil is used to lubricate the components of the engine. Additional functions include reducing wear, cooling the engine and protecting it from corrosion. Acting by itself, the oil would soon become saturated with dirt particles, such as dust that is ingested into the engine, as well as abrasive metal parts. This will consequently lead to wear of internal parts, reduced protection from corrosion and in the worst case to engine failure. At this point oil filters come into play, as only clean oil and proper flow can guarantee continuous excellent engine performance. Oil filters consist of a strong steel box, that can withstand high oil pressure (up to 6 bar), an anti-drain back valve, that shall create minimum backpressure, a pressure relief valve, that doesn't leak and a strong filter media element. Microscopically seen, this media consists of a maze of different fibres, which will capture most types of dirt particles. The surface of the filter media is folded to extend its size in order to trap a substantial amount of particles and to stabilize the overall filter media. As time goes by, dirt particles will clog the filter, restricting oil flow and hence leading to oil starvation. As a consequence less supply of oil will damage and in the worst case destroy the engine. In order to extend lifetime and reach higher efficiencies of filters, it is sensible to improve these in every possible form. Generally these improvements require experimental runs, which tend to be time consuming and very costly. In the majority of cases those experiments are carried out with destructive tests. In that case filters are cut or burned to gain knowledge about the investigated quantities. Yet these tests cannot be repeated in order to obtain other characteristics of the filter or to prove a different outcome. Another disadvantage of experiments is the difficulty to investigate filter material in detail, i.e. to find out, which influence the deposition of particles or the deformation of fibres would have had on the overall filter characteristics. Therefore CFD simulations can offer an attractive alternative to experiments. Apart from the cost saving fact, every quantity of a filter can

independently be investigated, displayed and processed for further usage. The virtual tests can be repeated as often as necessary with exactly the same initial situation and properties. CFD simulations on microscopic scale enable the creation and testing of new filter materials without extensive experimental runs.

Figure 1.1: Detailed view of an oil filter

In modelling of such a filter material, first it is required to investigate the main governing factors of the filtration process. One of those is that the flow of the fluid through this filter induces a pressure distribution on the fibres, which leads to a deformation of the solid parts. This deformation results in a different assembly of the fibres, which will have significant impact on the other factors, as well as on the overall material. Another effect results from the fact that the fluid is carrying abrasive particles resulting from engine wear. The efficiency is measured on the ability of the filter material to absorb different particle sizes. It is essential to understand the underlying driving factors for further improvement of filter materials.

Due to the complexity of the issue, a change in fibre morphology (i.e. pore size diameter) cannot be linearly linked to i.e. filter efficiency, because it influences the whole hydrodynamic situation. This thesis represents an extensive attempt to create a tool which can increase the understanding of filter effects and dynamic parameter dependencies by means of computational engineering and simulation technology. A detailed, deterministic calculation model, which simulates the most important filtration effects on a microscopic level, has been created. Figure 1.2 sketches out the basic concept behind this novel scheme. [2]

In a first step, computer-tomography (CT) scans are conducted on "real life" filter fibre samples. The CT output data is compiled in stacks of two dimensional (2D) gray scale images of the fibre. Then the data is read in, digitalized, and processed to a full 3D reconstruction of the microscopic filter element. The 3D object is automatically meshed by a structured grid generator, in order for the geometry to be utilized as boundary framework for oncoming CFD calculations. This is where the result of the main development task comes into play: A CFD tool, designed and programmed in order to resolve the dynamic filtration situation for a user definable set of process variables, within the reconstructed fibre element. Produced simulation results can then be used to estimate the performance and suitability of the tested medium. [2]

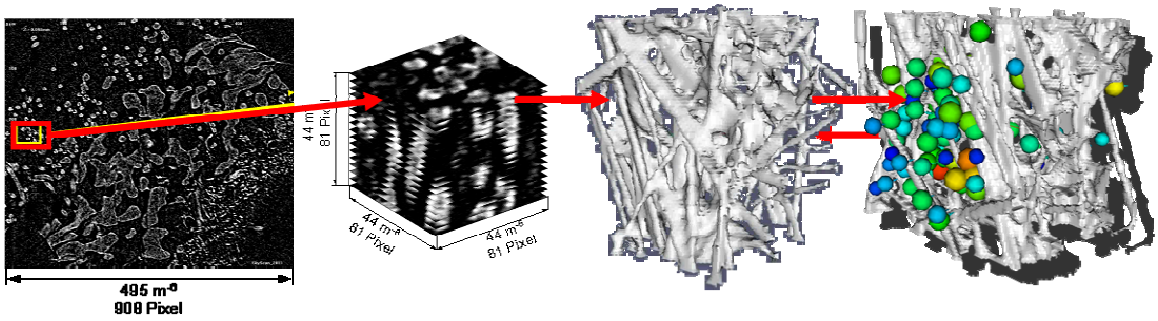


Figure 1.2: Sketch of the principle simulation concept. CT scans yield stacks of 2D grey scale images (left), which are transferred into 3D reconstructions of the fibre (middle). The 3D images are meshed and provide the geometry for the CFD filtration solver to be created (right).

Constant checks for result plausibility and validation have to be integral parts of any serious CFD development effort. In order to qualitatively and quantitatively validate the results, an extensive experimental set up has been created and a semi-empirical validation scheme has been devised. Figure 1.3 gives an overview of the underlying development scheme which links the experimental- and the simulation side. The application of this method leads to a continuous adjustment and improvement of the CFD model, according to the equivalent, experimental results. [2]

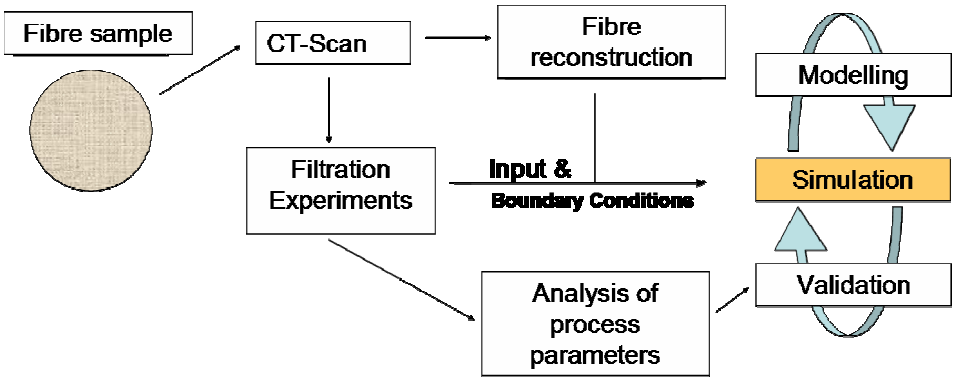


Figure 1.3: Experimental- and CFD development

From the beginning the development project was divided into four major working areas, as seen in Figure 1.4 [2]

(1) Digital Fibre Reconstruction (DFR) from CT information, as well as the suitable meshing of the 3D data.

(2) Creation of a Fluid Structure Interaction (FSI) tool in order to handle the fibre deformation effects under the influence of fluid flow.

(3) Development of a detailed, dirt particle- and deposition model, capable of simulating spherical- and non-spherical dirt particle behaviour in- and outside of the micro scale fibre vicinity.

(4) Validation of simulation results. Devise of an appropriate, experimental set up to verify solver functionality and to provide additional insight into filter fibre behaviour and characteristics.

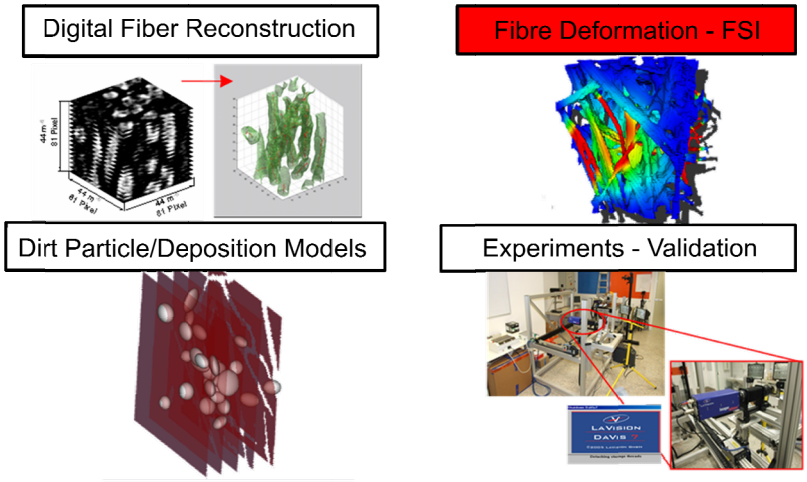


Figure 1.4: Overview of the four major areas of development behind the filtration solver project. The development of a suitable fluid-structure interaction solver for fibre deformation is at the focus of this thesis.

While other, related publications [2-5] extensively deal with the development of a detailed, dirt particle- and deposition model and the DFR utility, this thesis will only briefly discuss those two subjects. This thesis will be focussing on the development of the fluid-structure interaction part. The fully optimized and implemented solver constitutes a strong tool for the investigations on how the characteristics of the oil filter will change with the deformation of the fibres due to oil flow. Further on in chapter 5 the conducted verification process will be discussed and displayed in detail.

2 Overview

2.1 Fibre Material and Filter Types

Early designs of oil filters comprise steel wool, wire meshes and metal screens. Over the years disposal filters have become popular and with that, cellulose and papers were used. Cellulose is a natural material and presents a random irregular field of fibres to the oil.

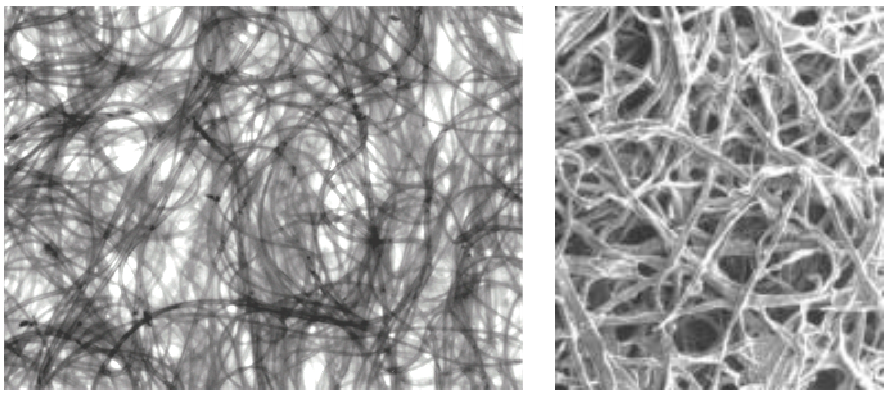


Figure 2.1: Detailed view of fibres from a filter media

In general, engine oil filter material has to yield the following features to maximize service intervals:

- Largest possible surface to trap as many particles as possible
- Least possible restriction to oil flow
- Handle as much contamination loading as possible, before need to be replaced
- Light weight, in order not to affect performance of the vehicle

It is vital to find the ideal balance of dirt trapping efficiency and dirt holding capacity. Only through this, excellent engine protection will be guaranteed.
For better quality filters, cellulose is blended with cotton or microscopic synthetic fibres. Synthetic media have smaller passages that trap more particles, nonetheless showing less restriction to the oil flow, since the fibres are thinner. Figure 2.2 illustrates the difference.

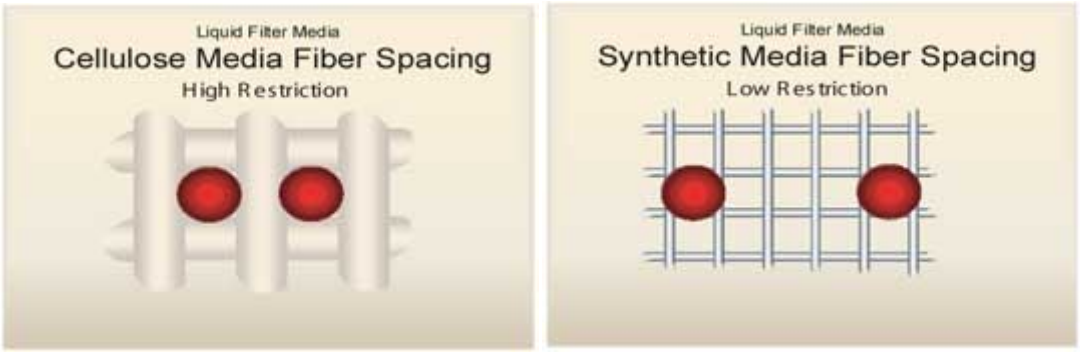


Figure 2.2: Difference between cellulose and synthetic filter media

Another way of enhancing filter material performance is to use the thinnest possible fibres. As a matter of fact, the thinner the fibres, made from a predefined amount of material are, the longer these will be and hence possessing a larger effective surface area. They can trap a larger amount of particles and additionally show much less restriction to the oil flow. This can also be realised by using synthetic fibres. The advantage of those is the possibility of specific manufacturing of the fibres regarding shape and thickness.

A special type of filter is the so-called “depth” filter. It has a passage size gradient, which means that it traps different size of particles at different spots in the media. The deeper the element, the smaller the passages will be and the smaller the particles which are trapped. Its main advantage is that it will hold more particles before blocking and needing to be replaced.

High end oil filters use fibre glass or extremely fine metal mesh. This will enhance the stability of the media.

2.2 Digital Fibre Reconstruction [2]

The ability to realistically model micro scale filtration processes in filter fibre materials is in large part based upon the realistic reconstruction of micro scale filter fibre geometries. Within the context of the development effort behind this work, a sophisticated method to digitally recreate real geometries was applied. In a first step, computer tomography (CT) scans are conducted on the fibre material to be

investigated. The data yielded by the CT scans are stacks of 2D grey scale images seen in Figure 2.3 (left).

MatLab® [6] based reconstruction algorithms have been programmed in order to process the CT data. The picture stacks can be uploaded and the individual slices are then analyzed. Local picture areas of higher grey scale intensities are recognized as fibre regions, which can be clearly distinguished against the low-intensity background. Identified fibre slices are then quantified, their pixel area is calculated and their local centres and radii are determined. By applying a skeleton algorithm [7] the centres of consecutive pictures are interconnected to constitute the basic, local fibre framework. By applying the calculated radius information attached to each centre point, the actual fibre structure is recreated as a 3D digital data matrix. This can be visualized as seen in Figure 2.3 (right).

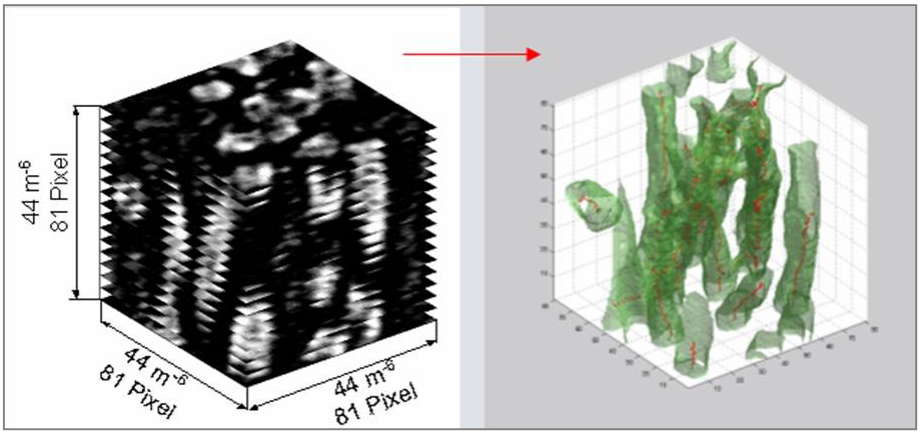


Figure 2.3: Fibre reconstruction and digitalization by MatLab® utilities. Stacks of grey scale images (left) out of CT scans are transferred to fully digitalized data matrices (right).

In a next step the digital data is automatically discretised into a structured, hexahedral grid mesh with a user definable cell-spacing-to-pixel ratio. This means that, if the CT scan resolution can be kept constant, a uniform spatial discretisation rate for any filter fibre simulation can be guaranteed. Thereby one of two modes of spatial resolution can be chosen: Either the finer mode, which features a spatial resolution of 1.6µm/Pixel or the coarser mode, which features a resolution of 3.2µm/Pixel. The reconstruction utility yields perfectly interfaced grids, of both the

fluid and the solid region of the fibre sample. Figure 2.4 presents an exemplary, structured, micro scale fibre grid mesh.

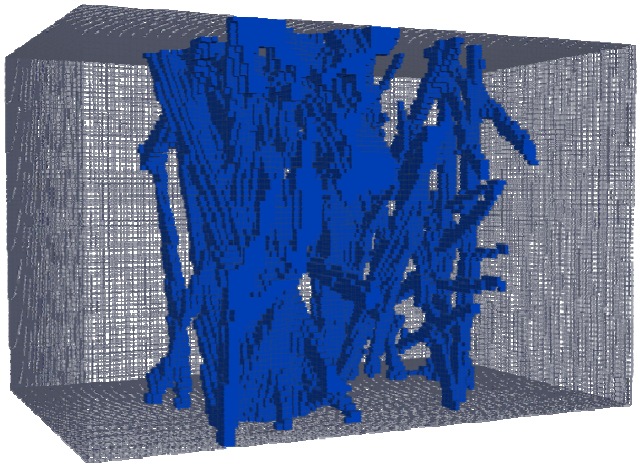


Figure 2.4: Filter fibre sample discretised into a structured fluid- and solid hexahedral grid mesh. Dimensions: 200µm x 200µm x 300µm. Number of cells: ~6.0 x 10^5

The prepared, structured grid meshes serve as geometry boundary conditions for the simulator to be developed.

2.3 Particle Deposition Model

The second part of the project was to develop a tool for simulation of large dirt particles. Low density clouds of dirt particles, ranging from 5 to 50 microns in diameter, occur in the oil stream. They are classified as "large particles", which means that their size is much larger than the size of a grid cell in the simulation grid. Those large particles contain a certain mass and momentum and interact with the flow field. This is in contrast to simulation of "small particles", where usually the temporal and spatial evolution via the solution of Eulerian conservation equations is sought. The difference is shown in Figure 2.5.

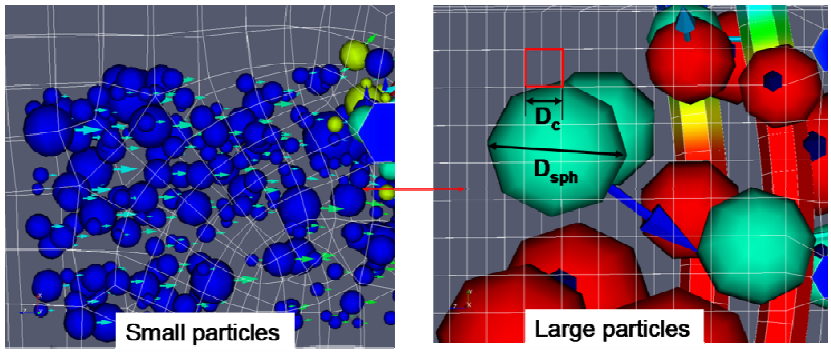


Figure 2.5: Particle simulation with small particles (left) and large particles (right)

The main effects governing particle motion are wall interactions, interactions with filter material and the collision between themselves and the plugging effect. The latter effect implies that with time, more and more particles find themselves entangled in the fibre structure and slowly accumulate there, thus increasing the effective solid surface and causing plugging of the flow. Even though those effects occur at a microscopic level, they will change macroscopic filter characteristics, such as porosity, pressure drop and overall filter efficiency. In order to simulate the encountered phenomena, not only is the implementation of a detailed particle model necessary, but further the calculation of particle force interactions is essential. Therefore a discrete phase Lagrange model was developed, which takes into account the two-way coupling between the fluid and the particles. The particles are related to Lagrange objects. This implies that they are regarded as individual objects. Their behaviour, such as velocity, acceleration and position, is totally dependent on surrounding fluid flow conditions such as flow velocity and flow pressure.

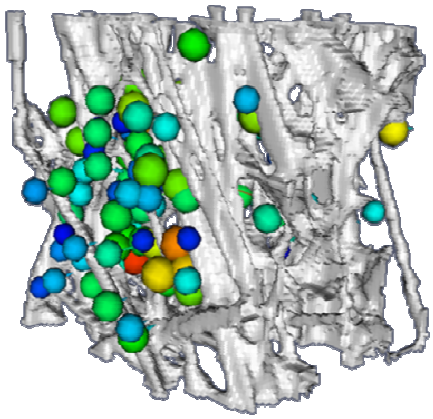


Figure 2.6: Particle cloud in a digitally reconstructed fibre geometry. Large numbers of particle-particle impact events occur. [2]

The model also enables the simulation of various shapes of dirt particles. They can be of regular spherical shape, shown in Figure 2.6, and also of nonspherical, presented in Figure 2.7. For developing a realistic dirt particle model it is vital to regard their morphologies. For example, a spherical particle model would significantly underestimate fluid skin friction and form drag forces. Other relevant implications are that elongated, non-spherical particles show the tendency to align themselves in the fluid stream thus increasing the possibility of slipping through a pore, whereas discs, with high rotational relaxation times, more likely block pores.

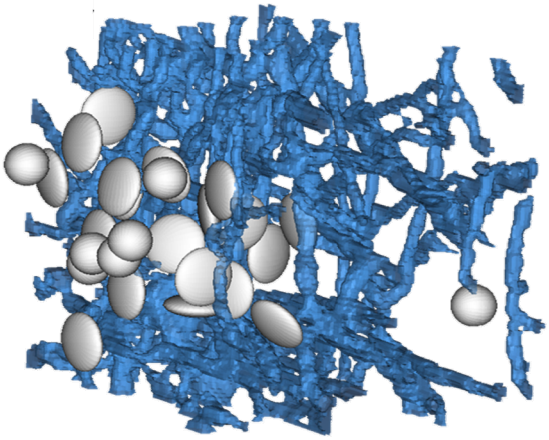


Figure 2.7: Nonspherical dirt particles entering a realistic digitally reconstructed fibre structure [2]

If those and many additional effects are not taken into account, the pressure drop, developing after some time, may be over- or underestimated and therefore only a nonspherical particle model will allow a realistic filter characterisation.

Very detailed descriptions regarding the background and development of large spherical and nonspherical particle models for filtration simulations are laid out in [2-5]

2.4 Programming Platform: OpenFOAM®

The entire CFD related software development behind this thesis was conducted within the framework of the Open Source CFD package OpenFOAM® (Open Source Field Operation and Manipulation) [8-10]. It is a CFD toolbox, which is applicable for a wide range of problems in continuum mechanics. It provides the user with

standard solvers, utilities and libraries, whereas libraries are repositories of function related software tools that can be accessed by solvers and utilities. All can be specifically selected according to the governing physics of the problem. The underlying programming language is the object oriented language C++ [11,12]. The syntax is similar to the notation of the differential equations being solved.

For example, the Navier-Stokes equations for incompressible flow read:

$$\frac{\partial(\rho u_i)}{\partial t} + \frac{\partial(\rho u_i u_j)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial^2(\eta \cdot u_i)}{\partial x_j{}^2} \qquad 2.1$$

Where u_i and u_j are the velocities along the coordinate directions x_i and x_j, ρ is the density, p is the pressure and η is the dynamic viscosity. [13]

This is implemented into the source code as following:

```
solve
(
        fvm::ddt(rho, U)
      + fvm::div(phi, U)
      - fvm::laplacian(eta, U)
        ==
      - fvc::grad(p)
);
```

where $\phi = \rho \cdot U$ (written as "phi" in the code)

The source code of the program has been made Open Source and thus is publicly available to anyone under the constraints of the GPL [14]. Therefore every aspect of the underlying source code can be altered as required by the user, which even permits the creation of whole new solvers, if necessary. It allows easy and direct implementation of new software modules at any point in the program. It is easy to automatise with the help of scripts and it extensively uses generic features (Templates). These several advantages make OpenFOAM® highly efficient and a very flexible tool, which in addition is free of license costs. The only disadvantage of OpenFOAM® is that it is not provided with a Graphical User Interface. This means that all inputs have to be provided by means of text files, which requires a higher effort for familiarising oneself with the software and a significant amount of prior knowledge of physics and programming techniques. Nevertheless OpenFOAM® is an

appropriate environment to create new solvers and hence model multiphysics problems such as the fluid-structure interaction found in oil filter applications.

Pre-and Postprocessing:

In this thesis the meshing was conducted via the commercial FLUENT® mesh generator GAMBIT® [15] or via self written meshing utilities. All results were post processed and visualized with the Open Source visualization tool ParaView [16] by Kitware®.

3 Fluid-Structure Interaction (FSI)

3.1 Introduction

In modelling of filter media a very important physical effect has to be considered concerning the interaction between the fluid and the fibre material. The fluid flow induces forces on the solid regions, which lead to deformation of the fibres and hence to changes of the overall material structure under working conditions. This induces severe modifications of the permeability of filters. The main challenge now is to simulate this situation with all the underlying physical phenomena considered.

Traditionally, for modelling fluid flow, CFD (computational fluid dynamics) codes are widely used, whereas for mechanical stresses CASA (computer aided stress analysis) codes are commonly employed. In the majority of cases, CFD codes are based on the so-called Finite Volume (FV) Method, where on the other hand CASA codes are mostly based on Finite Element (FE) principles. This procedure requires that two independent computer codes be used to calculate both, fluid flow and solid-stress simulation for the same system. Furthermore a third code is required for coupling and data management. This implies that all codes must be available and the user must have knowledge of all three of them. Another problem is that the CFD code must be converted into an appropriate format readable by a CASA program and vice versa. For both circumstances this conversion will cause a certain degree of loss in accuracy, especially as the two-way information exchange has to be undertaken iteratively, until convergence is achieved.

The solution is to combine the overall calculations in one single computer code. To enable the coupling, similarities between the fluid flow equations and solid stress equations have to be found. This will give the possibility to devise an algorithm, which will solve the solid-stress equations in one part of the field and the fluid-flow ones in another, yet realise both in one single computer code. The question now is which basic principle shall be used, the Finite Element (FE) method or the Finite Volume (FV) method.

The FE method provides an accurate and stable result for transient simulations of structural deformation. It is based on the variational principle [17] and uses pre-defined shape functions dependent on the topology of the element. It easily extends to higher order discretisation, produces large block matrices with usually high condition numbers and as a consequence relies on a direct solver. The discetisation is non conservative and hence does not guarantee to satisfy the conservation equations of the fluid flow. In case of non linear equations or discontinuous coefficients the FE method may show instabilities. Details can be found in [18,19]

The FV method can be considered to be a particular case of the FE method. For the FV method, the shape functions are regarded as piecewise linear and hence allows a conservative discretisation. Therefore it can handle complicated, coupled and non-linear differential equations, widely used in fluid flows [20]. The non-linearity is treated in an iterative way and creates diagonally dominant matrices well suited for iterative solvers. From this follows that the FV method also allows discontinuous solutions, such as they appear in local mesh refinement with discontinuous mesh line intersections. Additionally it has minor demands on the quality of the computational mesh.

The decisive factor was the available environment for the development of a new solver. Most of the conventional simulation programs do not allow access to the underlying computer code. OpenFOAM® being an Open Source program permits a direct access to every part and equation implemented in the program and additionally allows for modification. Due to these advantages, OpenFOAM® is the best suited environment for development of a new solver for fluid-structure interaction. It is based on the FV method, which additionally shows the best behaviour for simulation of filtration processes.

3.2 Constraints and Specifications

For simulation of oil filters not all factors, which govern fluid flow and solid deformation, have to be considered. A few simplifications can be made in order to reduce the time and effort required for CFD simulations.

3.2.1 Fluid Region

The simulation is carried out at a microscopic scale with the intention of statistically averaging results for several regions in order to retrieve macroscopic information in later phases of development. Due to the size of the geometry, which is in the scale of a few microns, the Reynolds numbers are very low and range between 0.5 and 5. In that case the flow can be classified as creeping flow and turbulence effects do not have to be taken into account. Temperature also has little influence on the deformation of the filter material and the existing particles and therefore the model can be assumed to be isothermal. There is only oil to be considered as a single fluid and hence single phase flow assumptions are valid as well. Generally not very high pressures are occurring during the filtration process and thus changes of the oil density are not significant. From this follows, that the flow can be assumed to be incompressible.

3.2.2 Solid Region

The effect of compression of the overall filter material is more important than the deflection of a single fibre. Hence it is satisfactory to regard only small deformations of fibres, where the longitudinal movement u is much less than the deflection w. Therefore u can be neglected.

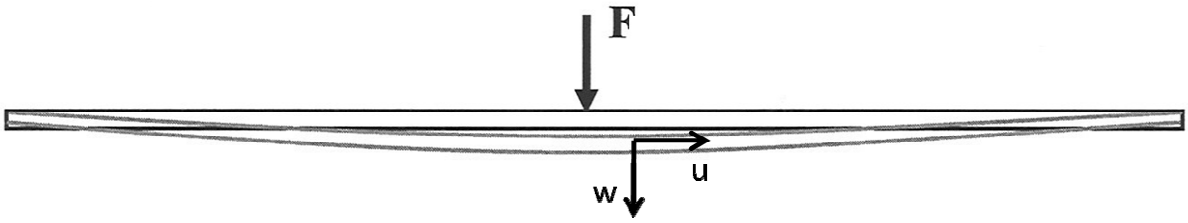


Figure 3.1: Deformation of a single fibre due to an arbitrary force F

It that case, the deformation behaviour is linear, which means that the change of load of a certain factor results in a deformation of the same factor. Therefore the stresses are proportional to strains according to Hooke's law. Additionally the material can be considered as elastic and hence, there only are deflections and no plastic deformations of the fibre itself. Finally no initial stresses will be taken into account.

3.3 Governing Equations

3.3.1 Governing Equations for Fluid Flow

The basis for deriving equations for fluid flow is mass and momentum conservation. Newton's second law states that the rate of change of momentum of a fluid particle is equal to the sum of all forces acting on the particle. All of the following equations are presented in Einstein's notation.

The rate of change of momentum in a unit cell is given by:

$$\rho \frac{du_i}{dt} = \rho \frac{\partial u_i}{\partial t} + \rho u_j \frac{\partial u_i}{\partial x_j} \qquad 3.1$$

where u_i and u_j are the fluid flow in coordinate directions x_i and ρ the density. $\frac{du_i}{dt}$ is called substantial derivative, $\frac{\partial u_i}{\partial t}$ is the partial derivative and $u_j \frac{\partial u_i}{\partial x_j}$ is the convective term [13,21-23].

There are two types of forces acting on a differential fluid unit cell:

- Surface forces:
  - Pressure forces (the negative sign means it is pointing outwards of the cell volume):

    $$-\frac{\partial p}{\partial x_i} \qquad 3.2$$

  - Viscous forces evolve due to stresses applied to the control fluid volume:

    $$\frac{\partial \tau_{ij}}{\partial x_j} \qquad 3.3$$

- Body forces:
  - Source term:

    $$\rho g_i \qquad 3.4$$

where p represents the pressure on the fluid cell, g_i the gravity vector and τ_{ij} the normal and shear stresses.

Adding up these contributions, we derive the following:

$$\rho \frac{\partial u_i}{\partial t} + \rho u_j \frac{\partial u_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} + \rho g_i \qquad 3.5$$

For Newtonian fluids it is valid that there is a linear relationship between stress and strain rate. The proportionality factor is constant at moderate temperatures and is called the dynamic viscosity η of the fluid. For one dimensional flow one obtains:

$$\tau = \eta \cdot \frac{\partial u_i}{\partial x_j} \qquad 3.6$$

Stokes extended Newton's idea from a simple 1D flow to a multidimensional flow leading to the Stokes' relations:

$$\tau_{ij} = \eta \cdot \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) \qquad 3.7$$

where δ_{ij} is called the Kronecker symbol and is defined as

$$\delta_{ij} = \begin{cases} 1 & für \quad i = j \\ 0 & für \quad i \neq j \end{cases} \qquad 3.8$$

By inserting Eqn. 3.7 into 3.5 we derive

$$\rho \frac{du_i}{dt} = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \cdot \left(\eta \cdot \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \delta_{ij} \frac{\partial u_k}{\partial x_k} \right) \right) + \rho g_i \qquad 3.9$$

After rearranging the second term on the right hand side by using Schwarz' theorem regarding the permutability of derivatives [24] and applying Eqn. 3.1 we derive the fundamental Navier-Stokes equations for conservation of momentum:

$$\rho \frac{\partial u_i}{\partial t} + \rho u_j \frac{\partial u_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \eta \cdot \frac{\partial^2 u_i}{\partial x_j{}^2} + \eta \cdot \left(1 - \frac{2}{3} \delta_{ij} \right) \cdot \frac{\partial u_k}{\partial x_k} + \rho g_i \qquad 3.10$$

In oil filtration processes the flow is regarded as incompressible, which means that $\delta u_k / \delta x_k = 0$. Further on, due to simulation on microscopic scale, the influence of the gravity is negligible.

Hence, the Navier-Stokes equations for incompressible flow read as follows:

$$\rho \frac{\partial u_i}{\partial t} + \rho u_j \frac{\partial u_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \eta \cdot \frac{\partial^2 u_i}{\partial x_j{}^2} \qquad 3.11$$

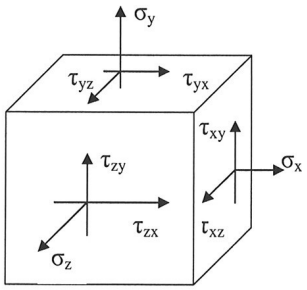
3.3.2 Governing Equations for Solid Deformation

The mathematical model describing small deformation of solids is based on the three dimensional stress distributions. From Newton's balance equation of momentum follows that

$$\rho \frac{\partial^2 u_i}{\partial t^2} = \frac{\partial}{\partial x_j}\left(\sigma_{ij}\right) + X_{bi} \qquad 3.12$$

where for i = j σ_{ij} represent the normal stresses, for i ≠ j $\sigma_{ij} = \tau_{ij}$ represent the shear stresses, u_i the displacement vector and X_{bi} are body forces.

This is based on the Cauchy theory, which says that the state of stress at a certain point in a body is completely defined by the nine components (six independent) σ_{ij} of a symmetric second order Cartesian tensor called the Cauchy stress tensor.



$$\underline{\underline{\sigma}} = \begin{pmatrix} \sigma_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_{zz} \end{pmatrix}$$

Figure 3.2: Three dimensional stresses on a solid element (left) with the Cauchy stress tensor (right)

For elastic deformation the relation between stress and strain in case of an isotropic homogenous structure is linear.

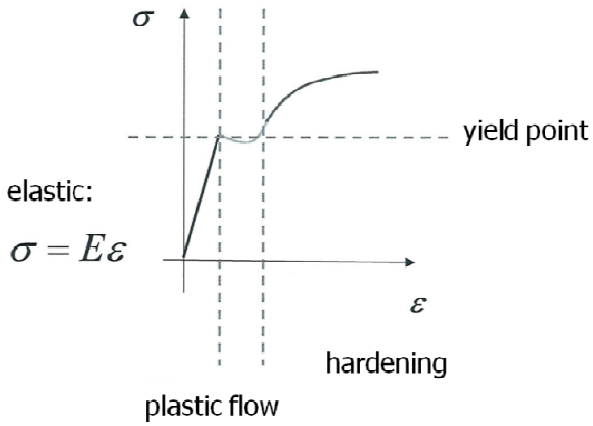


Figure 3.3: Relation between stress and strain acting on a solid body

For the elastic domain the generalized Hooke's law is valid:

$$\sigma_{ij} = E_{ijkl} \cdot \varepsilon_{kl} \tag{3.13}$$

where σ_{ij} is the stress distribution, E_{ijkl} is the elasticity tensor and ε_{kl} is the elastic strain distribution. In the special case of an isotropic material, the symmetry of the fourth order elasticity tensor can be taken advantage of. The tensor can be reduced to be dependent on only two constants.

In component notation the constitutive law can be written as

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{13} \\ \sigma_{23} \end{bmatrix} = \begin{bmatrix} 2\mu+\lambda & \lambda & \lambda & & & \\ \lambda & 2\mu+\lambda & \lambda & & 0 & \\ \lambda & \lambda & 2\mu+\lambda & & & \\ & & & 2\mu & & \\ & 0 & & & 2\mu & \\ & & & & & 2\mu \end{bmatrix} \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \varepsilon_{12} \\ \varepsilon_{13} \\ \varepsilon_{23} \end{bmatrix} \tag{3.14}$$

whereas ε_{ij} is defined by

$$\varepsilon_{ij} = \frac{1}{2} \cdot \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \tag{3.15}$$

λ and μ are called the Lamé constants and have the following relationship

$$\mu = \frac{E}{2 \cdot (1+\nu)} \tag{3.16}$$

$$\lambda = \frac{\nu \cdot E}{(1+\nu)(1-2\nu)} \tag{3.17}$$

where E is the Young's modulus and ν is called the Poisson number. Additionally μ = G is called the shear modulus.

After combining the equations (3.13 - 3.17), we derive Lamé-Navier's equation for solid displacement:

$$\rho \frac{\partial^2 u_i}{\partial t^2} = G \cdot \frac{\partial^2 u_i}{\partial x_j{}^2} + G \cdot \left(1 + \frac{2\nu}{1-2\nu}\right) \frac{\partial^2 u_j}{\partial x_i \partial x_j} + X_{bi} \tag{3.18}$$

3.4 A Single Algorithm

The interfacing between two different codes is rendered unnecessary, when it is possible to develop a single computer code, which can then solve the solid-stress equations and displacements in one part of the field and other one for fluid-flow, i.e. fluid velocity in another. If a closer look is taken at the governing equations for solid-stress and fluid flow, derived in chapter 3.3, similarities are detected. Therefore it is possible to couple the two equation systems (Eqn. 3.11 and Eqn. 3.18) and develop one single computer code, as shown below.

The main difference is the pressure gradient term of the Navier-Stokes equations, Eqn. 3.11, which is absent in the displacement equations for solids. This term can be seen as a body force from external sources represented by the term X_b in the solid displacement equations.

General Navier Stokes' Equations (3.10):

$$\rho \frac{du_i}{dt} = -\frac{\partial p}{\partial x_i} + \eta \cdot \frac{\partial^2 u_i}{\partial x_j^{\,2}} + \eta \cdot \left(1 - \frac{2}{3}\delta_{ij}\right) \cdot \frac{\partial u_k}{\partial x_k}$$

Lamé-Navier's Equations (3.18):

$$\rho \frac{\partial^2 u_i}{\partial t^2} = G \cdot \frac{\partial^2 u_i}{\partial x_j^{\,2}} + G \cdot \left(1 + \frac{2\nu}{1-2\nu}\right) \frac{\partial^2 u_j}{\partial x_i \partial x_j} + X_{bi}$$

This way a coupling between the fluid flow and solid-stress equation systems can be directly achieved. The pressure field of the fluid region can be mapped as a boundary condition on the solid region and hence added as an external force to the term X_b in Eqn. 3.17. It induces deformations in the solid regions, i.e. the fibre material.

The forces acting on the fibre material can be decomposed into pressure based (normal) and a shear stress based (tangential) forces, as shown in Figure 3.4.

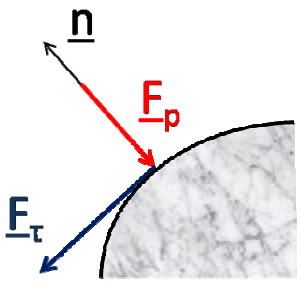


Figure 3.4: Sketch of the forces acting on a surface by the flow of a fluid

In tensor notation, the pressure force F_{pi} acting on an interface A given its normal vector n_i is defined as

$$F_{pi} = \int_A p \cdot n_i \cdot dA \qquad 3.19$$

In addition to the pressure force F_{pi} the flow of the fluid with the free stream velocity u_∞ also induces shear forces $F_{\tau i}$ on the solid region. These forces are caused by velocity gradients normal to the solid surface and also have a major influence on the deformation of a solid:

$$F_{\tau i} = -\eta \cdot \int_A \partial_i u_j n_j dA \qquad 3.20$$

where η is the dynamic viscosity and $\partial_i u_j$ is the velocity gradient field. It is apparent that using only the normal pressure field as a boundary condition on the deformation of the solid part is not sufficient to achieve physically plausible results.

Hence the two vectors can be summarized and written in differential form, which is

$$\frac{\partial}{\partial A}\left(F_{pi} + F_{\tau i}\right) = p \cdot n_i - \eta \cdot \partial_i u_j n_j \qquad 3.21$$

Furthermore:

$$\tilde{X}_b = \partial_i \left(\frac{d}{dA}\left(F_{pi} + F_{\tau i}\right)\right) = \left(\partial_i p - \eta \Delta u_i\right) \cdot n_i \qquad 3.22$$

From this we derive the source term, which is in tensor notation

$$X_{bk} = \frac{\partial}{\partial n_k} \tilde{X}_b = \partial_k p - \eta \cdot \Delta u_k \qquad 3.23$$

All equations and information can be summarized in one single algorithm. It is displayed in Figure 3.5.

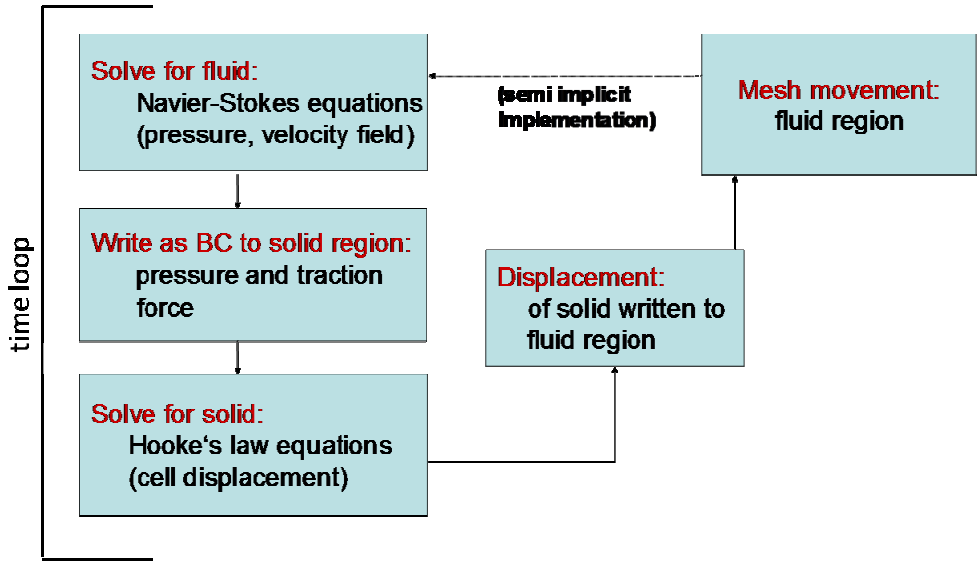


Figure 3.5: Algorithm to model Fluid/Structure interaction

There are two regions defined, one for the fluid and one for the structure. First the Navier-Stokes equations are solved in the fluid region. From this the pressure and velocity field is derived. At the interface between the solid and the fluid region, the pressure and traction increment (δp and δt) are transferred as boundary conditions from the fluid side to the solid side (seen in Figure 3.6) and the corresponding source term is calculated as derived in Eqn. 3.23.

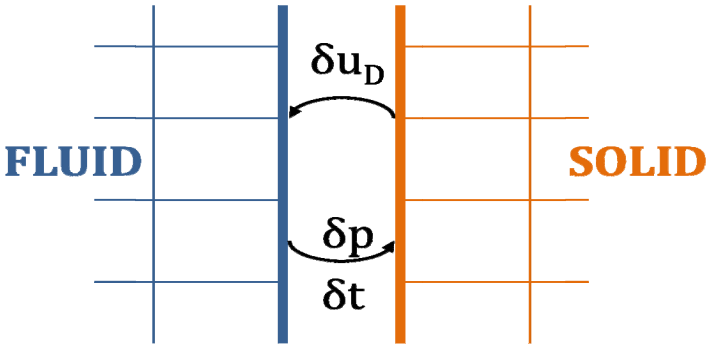


Figure 3.6: Transfer of coupling data

In the next step, the Lamé-Navier equations are solved for the solid region. The resulting displacement increment (δu_D) of the solid is passed back to the fluid region. The derived algorithm is implemented and integrated in an explicit way. Essential for the successful application of the algorithm is an efficient method to compensate for the deforming geometry of the solid region, i.e. a moving computational mesh strategy for the fluid region. This will pass the information of the solid displacement back to the fluid region. This topic will be discussed in the following chapter.

3.5 Discretisation

There are hardly any analytical solutions for the derived equations in chapter 3.4. Therefore those equations have to be solved with numerical methods. In this case, the Finite Volume method is applied. The overall domain of interest is decomposed into a finite number of control volumes (CV). In each cell the conservation equations, as they are for mass, momentum and energy are valid. The purpose of discretisation is to transform the partial differential equations into a corresponding set of algebraic equations, which can be solved easily [25-27].

The procedure is separated into two parts: spatial discretisation of the computational domain and equation discretisation.

3.5.1 Spatial Discretisation of the Computational Domain

This part comprises discretisation of time and space. For space discretisation the overall spatial domain is subdivided into a number of discrete polyhedral control volumes (CV), which fill the overall volume and do not overlap. Each CV is bounded by a set of flat faces and each face has only one neighbouring CV. The cell faces of the mesh are divided into internal faces, which delimit one CV from the neighbouring one and boundary faces, which constitute the boundary field of the overall domain.

In contrast to the Finite Element Method, in finite volume techniques the topology of the volume elements is not important as there are no topology dependent shape functions necessary. Hence different cells of general polyhedral shapes with a variable number of neighbours can be used, as displayed in Figure 3.7.

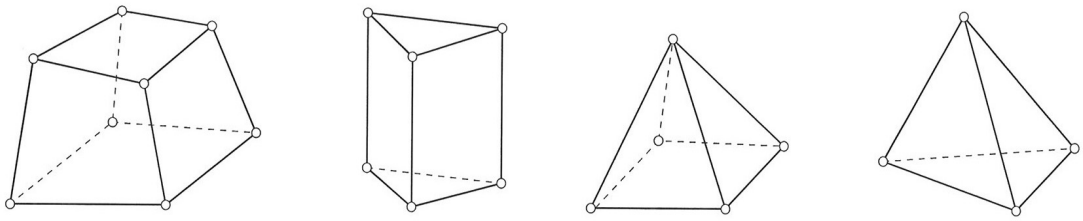


Figure 3.7: Typical 3D finite volume elements

This creates an arbitrarily unstructured mesh on which the governing equations are subsequently solved.

For time discretisation the time interval is split into a finite number of time-steps. This means that the time derivative d/dt of the conservation equations is discretised with the help of those discrete time steps. For more detailed information on those topics see [26-28]

3.5.2 Equation Discretisation

The equation discretisation produces a numerical description of the computational domain. In FV methods the basis for discretisation is the integral form of the equations, i.e. the Navier-Stokes' equations for momentum conservation (Eqn. 3.11):

$$\int_{CV} \frac{\partial(\rho u_i)}{\partial t} dV + \int_{CV} \rho \cdot u_j \frac{\partial u_i}{\partial x_j} dV = \int_{CV} \frac{\partial}{\partial x_j} \left(\eta \cdot \frac{\partial u_i}{\partial x_j} \right) dV + \int_{CV} S_\phi dV \qquad 3.24$$

where $S\phi$ represents an arbitrary source term.

Gauss divergence theorem claims that changes inside a certain volume are equal to the fluxes over the boundary surfaces:

$$\int_{CV} \frac{\partial u_i}{\partial x_j} dV = \oint_A u_i \cdot n_i \cdot dA \qquad 3.25$$

From this, the change of a quantity can be calculated by considering the fluxes over the boundary surface of the CV.

Applying Eqn. 3.25 we obtain the integral form of the transport equation:

$$\frac{\partial}{\partial t} \int_{CV} \rho u_i dV + \int_A \left(\rho \cdot u_i u_j \right) \cdot n_i dA = \int_A \eta \cdot \frac{\partial u_i}{\partial x_j} n_i dA + \int_{CV} S_\phi dV \qquad 3.26$$

From this point on the equations are solved in a segregated manner. That means that each component is treated separately.

After discretisation of the equations a system of linear algebraic equations is obtained. It describes the change of a system over time. This set of equations has to be solved in an iterative manner. After each iteration step, the linear equation system is calculated again on the basis of the previous iteration. This will be done until a convergence criterion is reached. A very important constraint is that the algebraic equations are solved in a way that the overall integral balances, i.e. the mass and momentum conservation, are fulfilled. This is called "conservative" discretisation. For more detailed information on discretisation of equations see [26,27].

3.5.3 Moving the Computational Mesh

Due to fluid flow the fibres of the oil filter material are displaced. Hence for physically plausible results the topology of the fluid mesh is adjusted to the moving solid object Recalculation at each time step is important to model the flow field changes and conserve geometric consistency. Both regions, fluid and solid, are having a separate mesh. For handling both regions a combined Lagrangian and Eulerian grid method is utilized. For the fluid region a Lagrangian grid is employed. This mesh is adjusted to the moving boundary which is updated every time step of the transient simulation. Furthermore the conservation values, like mass, momentum and energy are transported with the moving grid. For the solid region a Eulerian mesh is used. This means that the grid is fixed in time and space. The displacement information is still communicated to the fluid mesh leading to a deformation of the fluid mesh. As the solid mesh is not moved, for post processing it is adjusted to the fluid mesh by the "warp" function of ParaView [16].

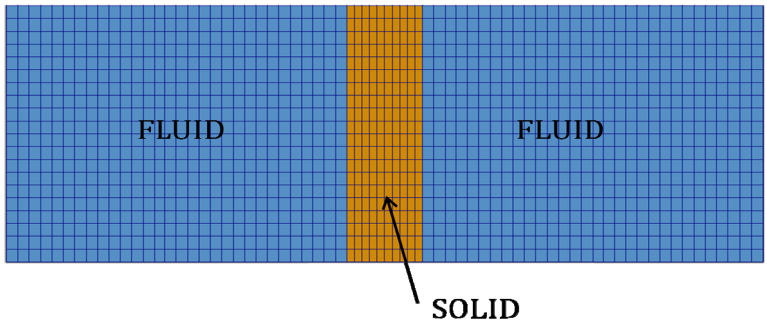


Figure 3.8: Fluid (blue) and solid (orange) regions posses their own independent mesh

The important constraint is that at the interface the vertices and faces have to be equal and match congruent with the opposite region.

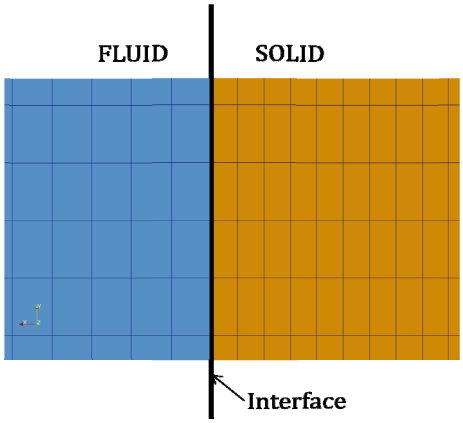


Figure 3.9: Number of faces and vertices on both sides of the interface has to be the same and congruent

The moving mesh stays topologically the same, no new cells will be created or existing ones destroyed. In OpenFOAM® there are two types of mesh manipulation approaches available. For the case of no topology changes and motion of interval mesh points it is called "dynamicMotionSolverFvMesh" [29,30]. Further on in order to solve the mesh motion equation for calculating the mesh points an appropriate solver is necessary.

The two available solvers are:

- *displacementLaplacian:* for this solver the final displacement of the mesh components is needed as well as the mesh displacement of the internal field. It is based on the Laplacian diffusivity and the cell displacement.
- *velocityLaplacian:* this solver deals with the boundary velocities instead of the final motions. It is used if the magnitude of the maximum displacement is small compared to the size of the overall domain. It is based on the Laplacian diffusivity and the cell motion velocity.

For a continuous mesh movement and small displacement values the velocityLaplacian solver is the most convenient and hence used for all simulations in this work.

The mesh motion consists of four steps:

- The moving interface is detected as declared by the user.
- Points and faces are detected on the moving interface.
- Displacement of solid is mapped to the fluid region
- the fluid mesh is moved

The distortion of the mesh results in topology changes of each single cell volume. For simulation accuracy a good quality mesh is vital. Hence to keep the topology as consistent as possible over time, a propagation of the deformation values of single cells to the overall cell-collective is carried out. This will ensure that not only a single cell has to take all the deformation. It is dispersed in the overall collective. The movement of the boundary is predetermined by the boundary conditions calculated from fluid flow. The movement inside the fluid mesh and hence the coordinates of its grid points is controlled by a diffusion mechanism. It is based on the Laplace equation:

$$\nabla \cdot \left(D_f \cdot \nabla \underline{u} \right) = 0 \qquad 3.27$$

where $\underline{u}$ represents the grid propagation velocity and D_f the diffusion coefficient.

Therefore the propagation of the deformation is ruled by the diffusion coefficient D_f. The simplest way of improving mesh quality is by introducing variable diffusivity. There is a variety of coefficients available in OpenFOAM® [10,29,30]

Distance-based methods: A number of boundary patches are selected by the user. The diffusion field D_f is a function of cell centre distance L to the nearest selected boundary.

- Linear inverse Distance: $D_f = 1/L$
- Quadratic inverseDistance: $D_f = 1/L^2$
- Exponential: $D_f = 1/e^{(L)}$

Quality-based methods: Here the diffusion field D_f is a function of a cell quality measure:

- Mean cell non-orthogonality
- Mean cell skewness

- Mixed, which is a combination of mean cell non-orthogonality and mean cell skewness.

All of the available diffusion coefficients were tested on a 2D mesh with an arbitrary square moving through the computational mesh.

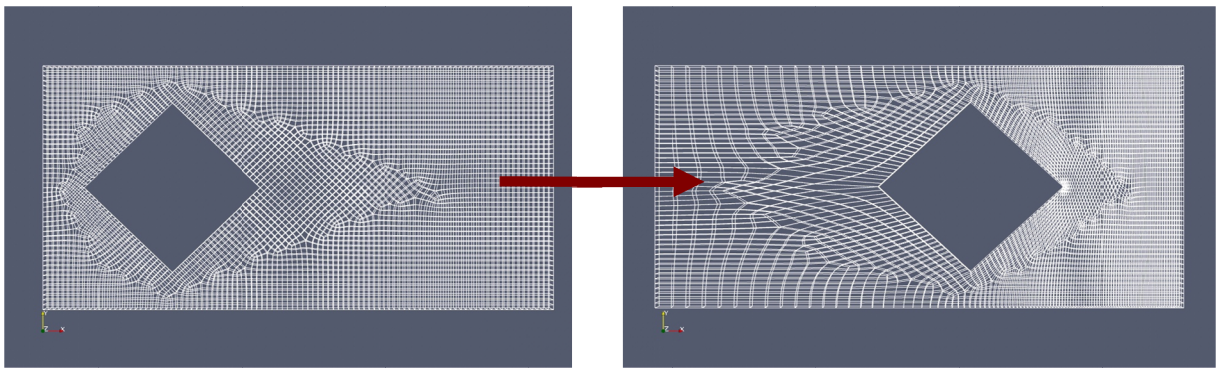


Figure 3.10: A square moving from left to right through a 2D grid

In Figure 3.10 on the left hand side the starting point of the moving square is shown. On the right hand side it is already in motion. It is moved to the maximum possible displacement, where the simulation is aborted due to heavily distorted grid cells. At the tip of the square, an area of interest is developing. In Figure 3.11. it is marked with a red circle.

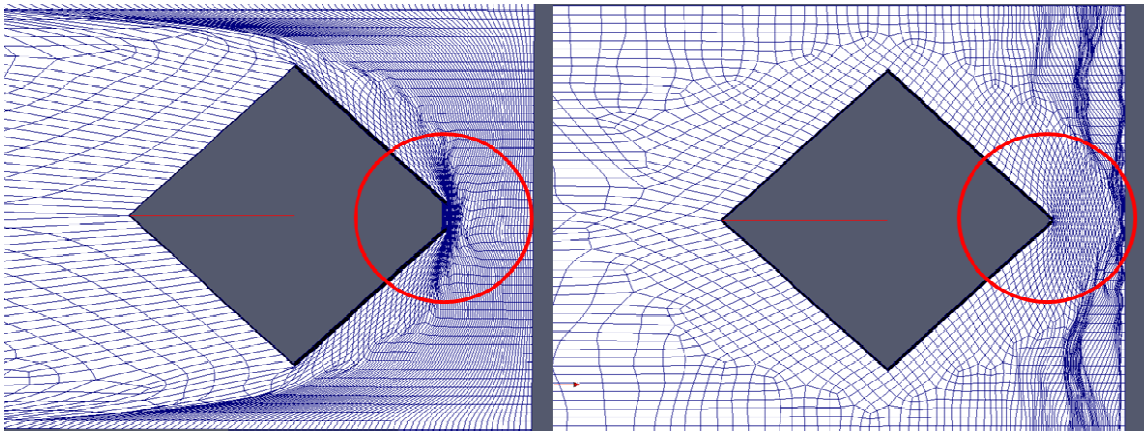


Figure 3.11: Both methods not conserve the overall geometric consistency of the mesh (left hand side: "directional" diffusion coefficient, right hand side: "faceOrthogonality" diffusion coefficient)

On the left hand side a quality based diffusion coefficient is used. It is called "directional" and takes the face skewness into account. It can be observed that the

cells at the tip are heavily distorted and even create negative cell volumes. This will quickly lead to termination of the simulation. The diffusion coefficient, used on the right hand side, takes the face orthogonality into account. It shows that in imminent vicinity of the tip, the cell volumes are staying in good condition, whereas the grid in other parts of the fluid volume is distorted. Again, this diffusion mechanism is not the best choice. After testing all diffusion coefficients provided by OpenFOAM®, a particular one showed excellent mesh conservation over time.

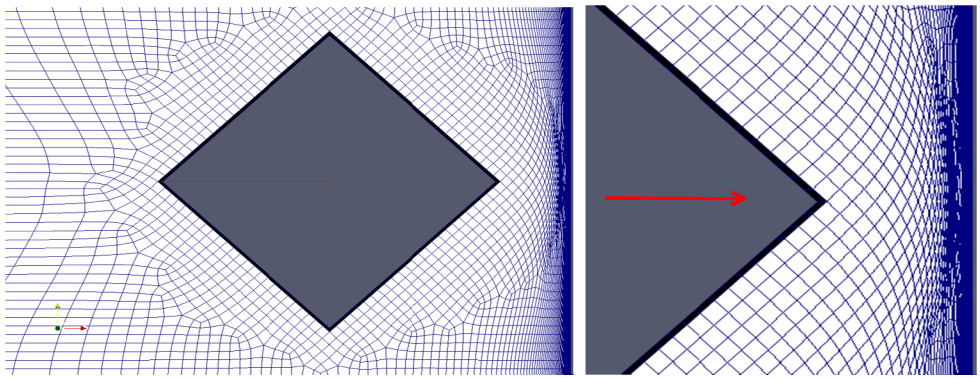


Figure 3.12: Favourable diffusion mechanism: "quadratic inverseDistance"

Contrary to the coefficients used in Figure 3.11, in Figure 3.12 a distance-based method is used. It is called "quadratic inverseDistance" coefficient regarding the fluid-solid interface. This diffusion mechanism was used for all of the following simulations.

From this investigation follows that it is vital to investigate the underlying diffusion mechanism to guarantee the conservation of the overall mesh quality during the whole simulation process.

3.6 Structure of an OpenFOAM® Case for FSI Simulation

In OpenFOAM® the data in and output is handled over text files, called dictionaries. This chapter gives an overview of the necessary dictionaries and input parameters for a successful FSI simulation. Every case consists of three main directories. In each of them there are two sub folders, one for the fluid and one for the solid region. The overall structure of a typical case for FSI simulation is shown in Figure 3.13.

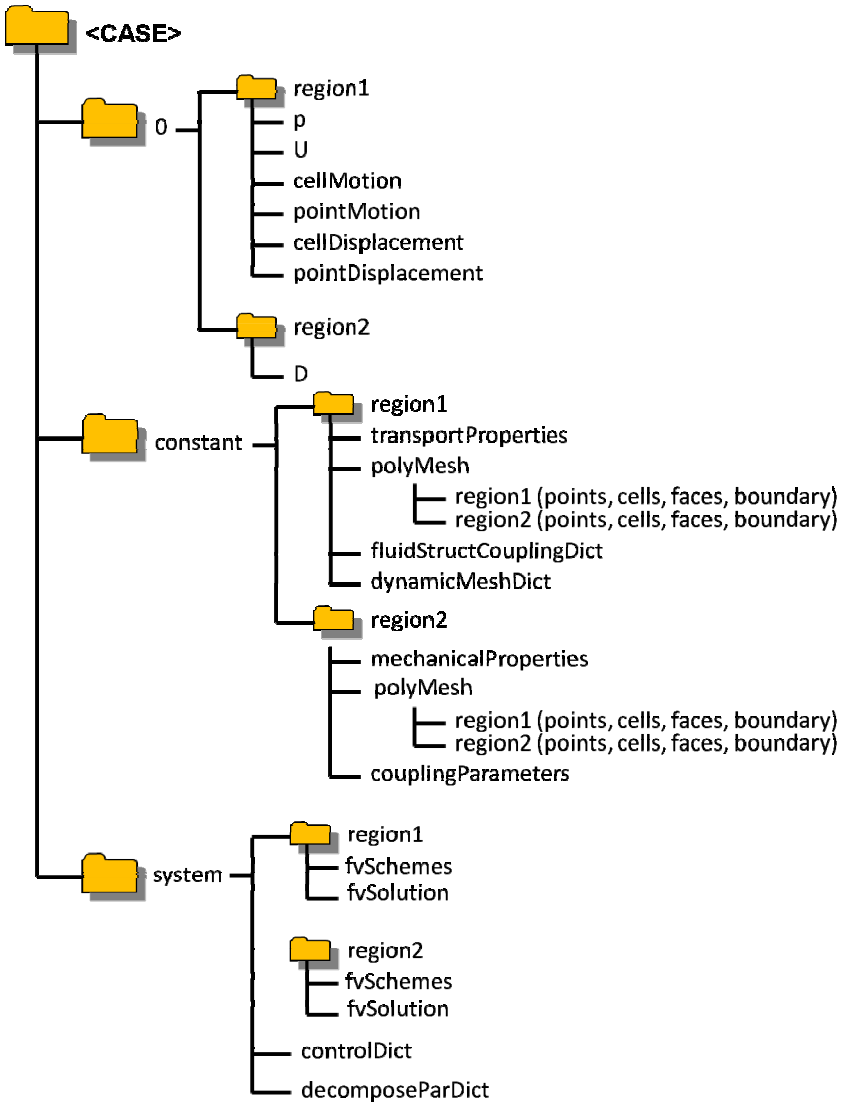


Figure 3.13: Structure of a regular OpenFOAM® case with two regions for FSI simulation

1. ***"0":*** This folder contains all the information about the initial conditions at time step 0.

   For the fluid region (region1):

   1.) *p:* for initial conditions of the pressure (see chapter 3.7.1)

   2.) *U:* for initial conditions of the velocity (see chapter 3.7.1)

   3.) *cellMotionU/pointMotionU:* for mesh movement (necessary for mesh motion solver: velocityLaplacian)

   4.) *cellDisplacement/pointDisplacement:* for mesh movement (for mesh motion solver: displacementLaplacian); not used in this work

   For the solid region (region2):

   1.) *D:* initial boundary conditions for the displacement (see chapter 3.7.1)

2. ***"constant":*** This directory comprises information about material properties, the computational mesh and moving mesh parameters.

   For the fluid region (region1):

   1.) *polyMesh:* This folder contains a full description of the fluid mesh, like point coordinates, cells and boundaries. The text files are not supposed to be changed, as it would lead to severe inconsistencies and failures during simulation;

   2.) *transportProperties:* This dictionary is used for adjusting fluid properties like the fluid density and the kinematic viscosity.

   3.) *fluidStructCouplingDict:* This newly developed dictionary handles additional parameters like the ones for the collision concept. It gives the opportunity to define the fraction of the distance between fibres at which the collision concept shall start to work and to adjust the minimal distance for freezing of fibres. These topics will be discussed in detail in chapter 3.8. Further on it contains a switch to either turn the structure deformation on or off in case only a fluid flow solution is wanted by the user. This is also valid for the collision concept.

   4.) *dynamicMeshDict:* This dictionary is responsible for all parameters concerning the mesh movement. It gives the user the opportunity to choose the diffusion mechanism, the necessary library and solver used for mesh movement. The necessary library for mesh movement without topology changes is called "libDynamicFvMesh.so" and is used for all simulations done in this work.

   For the solid region (region2):

   1.) *polyMesh:* This folder contains a full description of the solid mesh. Again, the text files for points, cells and boundaries shall not be altered.

   2.) *mechanicalProperties:* In this text file it is possible to adjust the properties of the solid, like the Young's modulus, the Poisson ratio and the density of the material and if necessary neglect or consider plane stress.

3.) *couplingParameters:* The user can define which boundary patch will be the coupling interface between the solid and the fluid region. It must have the same name defined in each region in the boundary file of the polyMesh folder, which is predefined in the pre-processing of the mesh. With the parameter meshMotionTime the start time for mesh motion can be defined. Before this time point, only the fluid flow is calculated for stabilization of the fluid solution. The parameter motionRelaxation-iTime is used for relaxation of the solid displacement to support the convergence of the solution.

3. ***"system":*** In this folder all the parameters for the solution procedure are handled.

1.) *controlDict:* Here all control parameters are set, for example time step size, start/end time of the overall simulation and the write accuracy.

2.) *fvSchemes* and *fvSolution:* Those dictionaries are used to set discretisation schemes, equation solvers, tolerances and all other algorithm controls for the run separately for each domain, region1 and region2.

3.) decomposeParDict: In this dictionary, the method for domain decomposition and its necessary coefficients for running the simulation in parallel are defined. More details on this topic will be discussed in chapter 4.1.1.

The initial boundary field dictionaries for p and U are shown in chapter 3.7. A detailed description and further specifications of the system folder, with all available settings for discretisation schemes and solution controls are listed in the OpenFOAM® user's and programmer's guide [29,30]. The following figures show a selection of input dictionaries important for FSI simulation.

cellMotionU

```
dimensions      [0 1 -1 0 0 0 0];

internalField   uniform (0 0 0);

boundaryField
{
    Wall
    {
        type            fixedValue;
        value           uniform (0 0 0);
    }
    Fiber
    {
        type            cellMotion;
        value           uniform (0 0 0);
    }
    Outlet
    {
        type            fixedValue;
        value           uniform (0 0 0);
    }
    Inlet
    {
        type            fixedValue;
        value           uniform (0 0 0);
    }
}
```

pointMotionU

```
dimensions      [0 1 -1 0 0 0 0];

internalField   uniform (0 0 0);

boundaryField
{
    Wall
    {
        type            zeroGradient;
    }
    Fiber
    {
        type            fixedValue;
        value           uniform (0 0 0);
    }
    Outlet
    {
        type            fixedValue;
        value           uniform (0 0 0);
    }
    Inlet
    {
        type            fixedValue;
        value           uniform (0 0 0);
    }
}
```

Figure 3.14: Dictionaries for mesh movement

transportProperties

```
nu              nu [0 2 -1 0 0 0 0]    1.76e-4;

rhof            rhof [1 -3 0 0 0 0 0] 800;
```

mechanicalProperties

```
rho             rho [1 -3 0 0 0 0 0]   1854;

nu              nu [0 0 0 0 0 0 0]     0.3;

E               E [1 -1 -2 0 0 0 0]    1e9;

planeStress     yes;
```

Figure 3.15: Dictionaries for handling domain properties: transportProperties for the fluid domain and mechanicalProperties for the solid domain

fluidStructCouplingDict

```
// 1 with Structure deformation
// 0 without Structure deformation

fluidStructCoupling   1;

// 1 with collision of fibers
// 0 without collision of fibers

collision     1;

// for collision:
moveLimit       moveLimit [0 1 0 0 0 0 0] 1e-10;
fraction        0.85;
```

dynamicMeshDict

```
twoDMotion      no;

dynamicFvMesh   dynamicMotionSolverFvMesh;

motionSolverLibs ("libfvMotionSolvers.so");

solver          velocityLaplacian;

diffusivity     quadratic inverseDistance 1(Fiber);
```

Figure 3.16: Dictionaries for mesh movement (dynamicMeshDict) and auxiliary input parameters (fluidStructCouplingDict)

```
couplingParameters

fluidSide   Fiber;
solidSide   Fiber;

startMeshMotion time     [0 0 1 0 0 0 0]   0.01;
motionRelaxation iTime   [0 0 -1 0 0 0 0]  2;
```

Figure 3.17: Dictionary for general coupling parameters

3.7 Application to a Simple Filter Model

All components of the algorithm were assembled. The result was a transient fluid-structure interaction solver for incompressible, laminar flow and elastic solid displacement. In order to test the derived and programmed algorithm a simple case was set up. For the sake of simplicity this test case consisted of a single fibre situated in a flow channel at right angle to the fluid flow direction.

Material parameters and dimensions of the model:

Fluid:	Viscosity:	1.76×10^{-4} m²/s
	Density:	800 kg/m³
	Length of flow channel L:	0.7 m
	Cross section A:	0.16 m²
Fiber:	Young's modulus:	10^5 N/m²
	Length of the fibre:	0.4 m
	Diameter of the cross section:	0.07 m

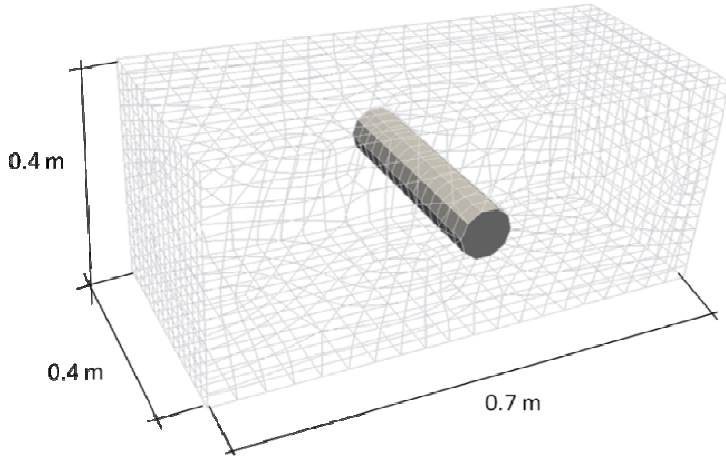


Figure 3.18: Dimensioned fluid domain with a single fibre at initial state

3.7.1 Boundary conditions

There are two base types of boundaries used:

1. patch: this is the basic type, which contains no geometric or topology information about the mesh and hence is a generic patch. Here it is used for inlet and outlet.
2. wall: this is a type of patch including all the information about geometry and topology of the defined patch. It is used for the surface of the fibres and the outer boundaries of the fluid region, besides inlet and outlet.

The conditions applied on the patches are [30]:

1.) *fixedValue:* a Dirichlet boundary condition, where a certain value ϕ is specified;

2.) *fixedGradient:* a Neumann boundary condition, where the normal gradient of the value ϕ is specified;

3.) *zeroGradient:* a special type of the fixedGradient condition; the normal gradient of the value ϕ is set to zero;

4.) *slip:* for simulation of frictionless flow parallel to the patch; if the value ϕ is a scalar, it is set to zeroGradient, if the value ϕ is a vector the normal component and the normal gradient of the tangential component are set to zero.

For this case the boundary conditions from the "0" dictionary reads as follows:

Pressure

```
dimensions      [0 2 -2 0 0 0 0];

internalField   uniform 1;

boundaryField
{
    Inlet
    {
        type            zeroGradient;
    }

    Outlet
    {
        type            fixedValue;
        value           uniform 0;
    }

    Wall
    {
        type            zeroGradient;
    }

    Fiber
    {
        type            zeroGradient;
    }

}
```

Velocity

```
dimensions      [0 1 -1 0 0 0 0];

internalField   uniform (0 0 -0.3);

boundaryField
{
   Inlet
    {
        type            fixedValue;
        value           uniform (0 0 -0.3);
    }

   Outlet
    {
        type            zeroGradient;
    }

   Wall
    {
        type            fixedValue;
        value           uniform (0 0 0);
    }

    Fiber
    {
        type            fixedValue;
        value           uniform (0 0 0);
    }

}
```

Figure 3.19: Boundary field for the fluid region

Displacement

```
dimensions      [0 1 0 0 0 0 0];

internalField   uniform (0 0 0);

boundaryField
{
    Fiber
    {
        type            tractionDisplacement;
        traction        uniform ( 0 0 0 );
        pressure        uniform 1;
        value           uniform (0 0 0);
    }

    Wall
      {
        type            fixedValue;
        value           uniform (0 0 0);
      }
}
```

Figure 3.20: Boundary field for the solid region

In the solid region, for the patch "Fibre" a special boundary condition is used. It is called "tractionDisplacement" and handles the application of pressure and traction forces on the solid side (see Figure 3.20).

3.7.2 Results

The results of applying the newly developed solver to a single fibre are presented in this section. Figure 3.21 shows the deformed fibre. Red marked regions are regions with high deformation values. Blue marked areas are regions with zero displacement,

as in this case are the end points of the fibre, which are kept at fixed positions. In between the displacement is gradual.

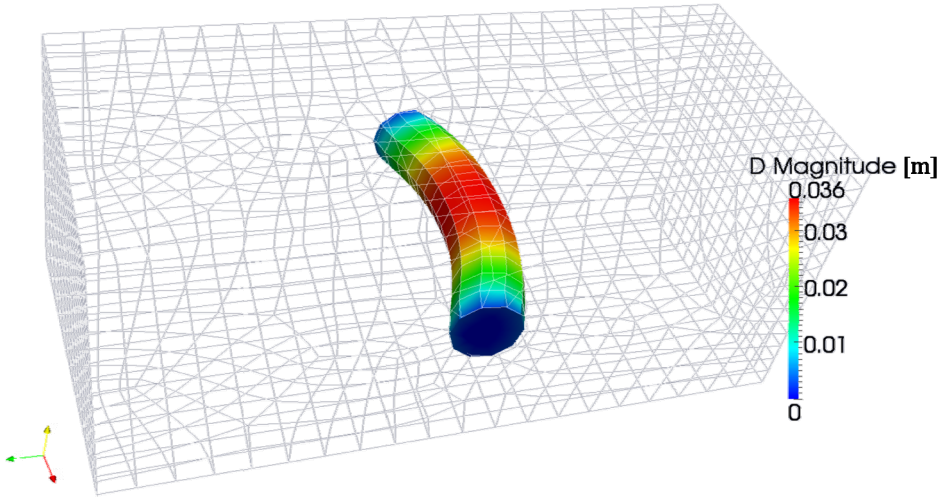


Figure 3.21: A single fibre deforming under fluid flow

This first application shows very plausible results. In order to verify it, a simple analytical model was set up. In Figure 3.22 the deflection (displacement) along the length of the fibre is plotted for the results achieved by simulation with the newly developed FSI solver and the analytical model.

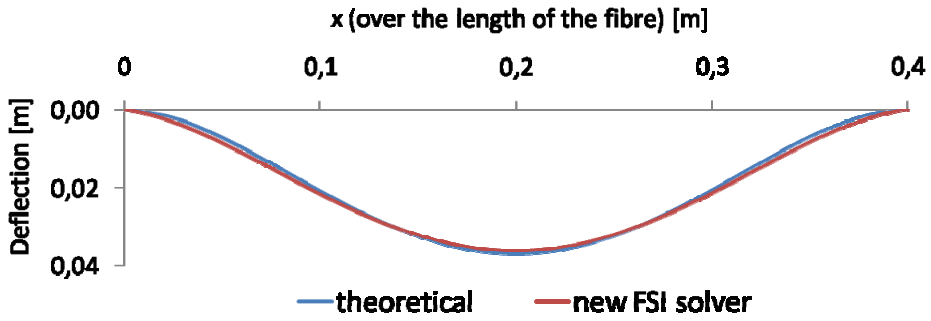


Figure 3.22: Comparison of the deflection (displacement) results achieved by simulation and the ones calculated by the analytical model

It can be observed that the two curves match very well. Especially in the middle of the fibre. The maximum difference appears close to the fibre endpoints.

For further testing, the case was extended to six fibres with two different diameters (0.04 and 0.06 m) of the cross section. The same model parameters and dimensions as in the previous case were used.

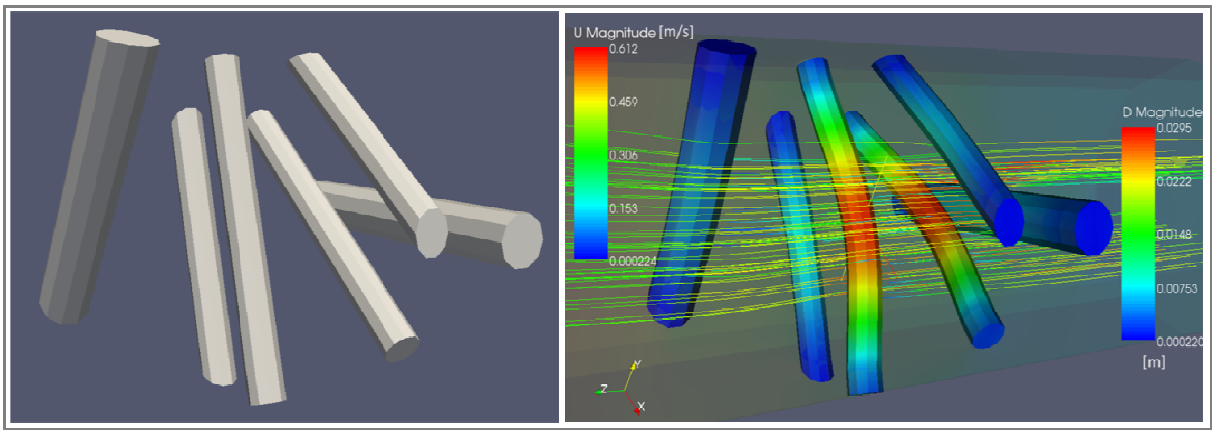


Figure 3.23: Model of six fibres with different thickness bending in the flow of fluid

As it can be seen in Figure 3.23 the fibres are deformed due to the flow of the fluid in dependency on their thickness. Fibres with larger diameters are barely deformed. In the middle of the flow channel, the flow velocities of the fluid are at the maximum. Therefore fibres with small diameters and additionally placed in the middle of the flow channel are deformed the most. Hence not only the diameter, but also the spatial location of the fibre influences displacement.

The flow of the fluid is deviated by the presence of the fibres, which result in regions of low or even zero velocities. Corresponding to this the absolute fluid flow velocity around the fibres is displayed in Figure 3.24.

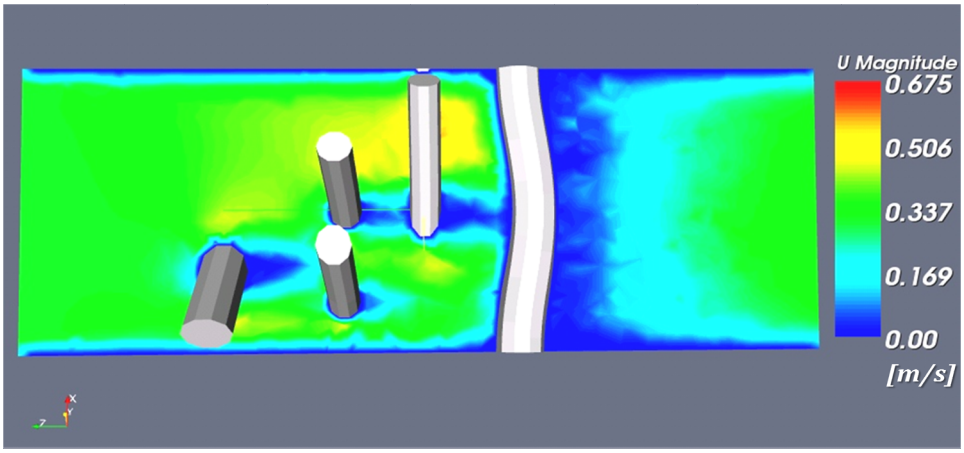


Figure 3.24: Velocity magnitude within surrounding fluid in a cut plane

In this figure the regions with almost zero velocity are marked blue. As expected, they can be found in the downstream regions of the fibres. This is also confirmed by the velocity vector field, shown in the immediate vicinity of the fibres, which is depicted in Figure 3.25.

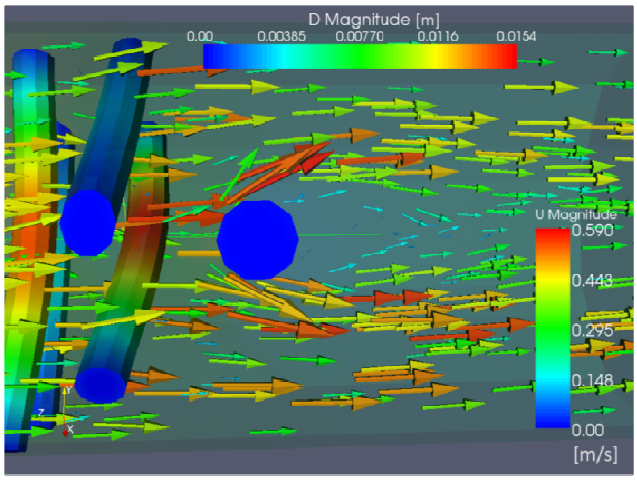


Figure 3.25: Velocity field of fluid is deviated around fibres

Therefore, thin fibres in the downstream flow field of thicker fibres will also tend to zero deformation, as there is no force from the fluid flow acting on the fibres.

The forces transferred from the fluid to the solid are pressure and traction forces, as discussed in chapter 3.4. To verify their implementation into the solver, these can be displayed. In Figure 3.26 on the left hand side the traction force variation is displayed, whereas on the right hand side the pressure distribution is illustrated.

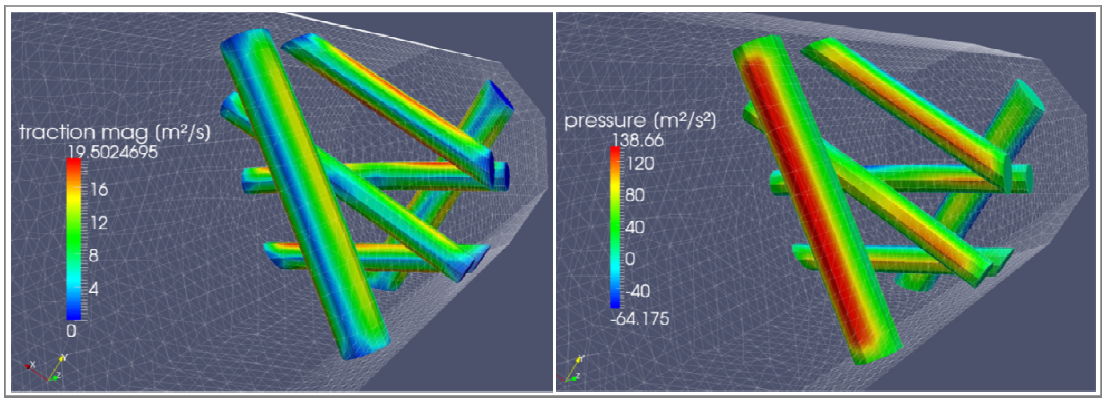


Figure 3.26: Traction and pressure induced by the fluid

It is evident that pressure and traction forces mapped on the fibres show physically plausible results. The pressure is high on the front side of the fibres whereas the traction forces show high values on the sides. Therefore their model implementation represents the expected tendencies correctly.

The model can now be extended to any number of artificial fibres. Figure 3.27 shows a case with several fibres, randomly distributed and of varying diameters being deformed due to fluid flow.

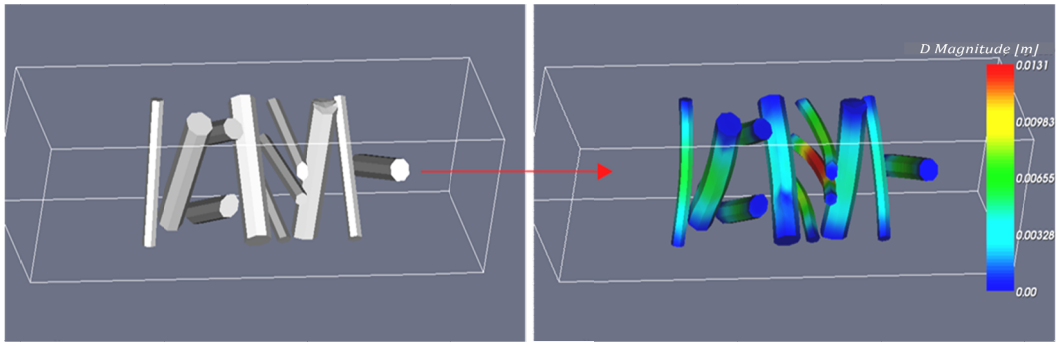


Figure 3.27: Randomly distributed fibres in a flow channel

3.8 Further Improvement

3.8.1 New Boundary Conditions

Until now the ends of the fibres in the simulations have been fixed. This does not reflect reality. Since just a microscopic piece of the filter is modelled, fibres do continue outside of the modelling domain. Hence the fibres should have the possibility of gliding through the domain. As there are two regions, one for solid and one for fluid, for both a new boundary condition has to be applied.

At first the boundary condition for the flow velocity at the walls of the flow channel is changed from the fixed value of zero to slip condition. The slip condition is provided by OpenFOAM® and therefore it only needs to be changed inside the "0" folder of the case for the boundary condition of the walls for the fluid.

```
dimensions      [0 1 -1 0 0 0];

internalField   uniform (0 0 -0.01);

boundaryField
{

    Outlet
    {
        type            zeroGradient;
    }

    Inlet
    {
        type            fixedValue;
        value           uniform (0 0 -0.01);
    }

    Wall
    {
        type            slip;
    }
    Fiber
    {
        type            fixedValue;
        value           uniform (0 0 0);
    }

}
```

Figure 3.28: Dictionary for the fluid region with the new slip boundary condition

To compare those two, the fluid flow velocity distribution of the cross section is displayed. In this case it is cut through the first fiber in the flow channel, as can be seen in Figure 3.29.

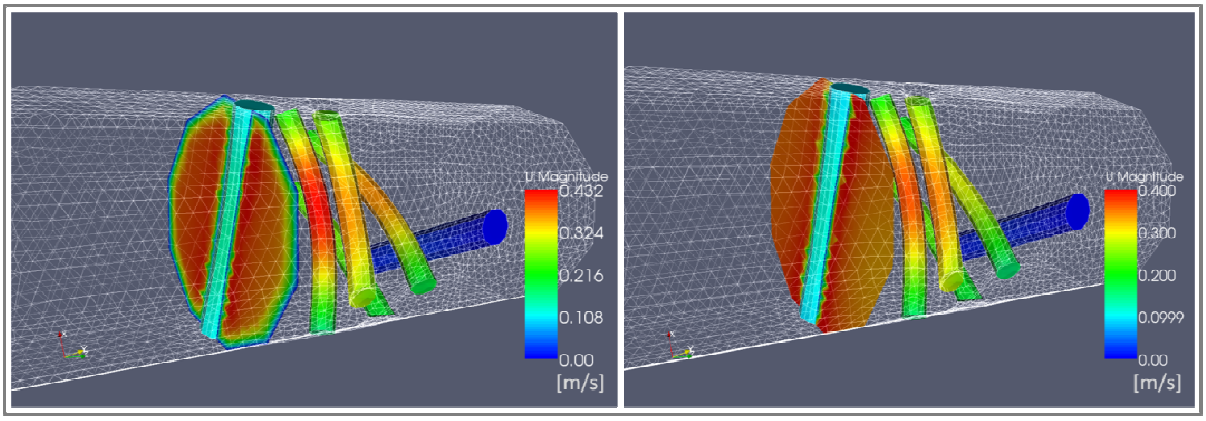


Figure 3.29: Fluid flow velocity distribution in the cross section of the flow channel. Boundary condition on the wall: Left hand side: fixedValue 0, right hand side: slip

The difference of those boundary conditions can clearly be observed by zooming out the area of interest, as displayed in Figure 3.30. The gap shows the place of the fibre, which is crossing the area.

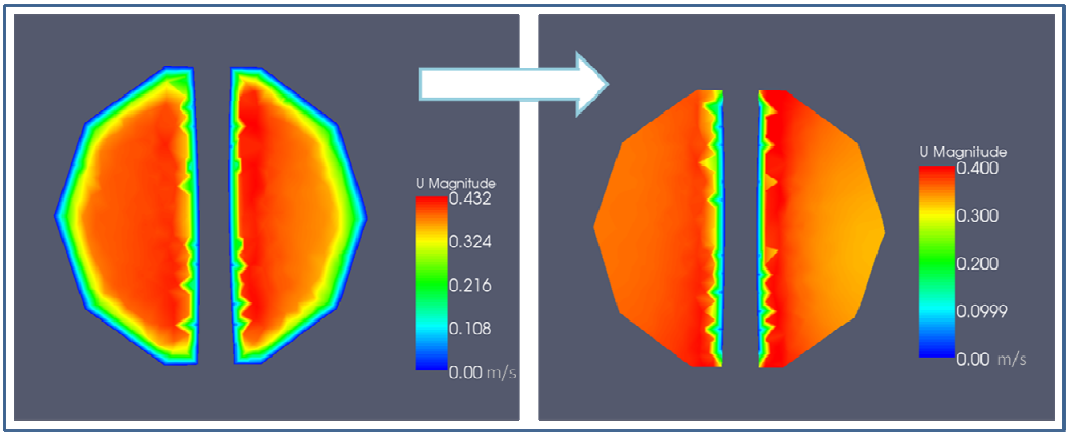


Figure 3.30: Change of boundary condition at the wall for fluid flow from fixedValue = 0 to slip condition

On the left hand side the previous used boundary condition is shown. It refers to a fixed value of zero for the flow velocity on the outer wall of the channel, marked in blue. On the right hand side the slip condition is presented. "Slip" means frictionless flow, which implies that the normal component of the velocity vector and the normal gradient of the tangential component are set to zero.

Further on, in order to mimic reality in a better way a new boundary condition for the solid region was implemented, in detail for the end points of the fibres, which are at contact to the walls of the flow channel. The new condition means that the fibres are not fixed anymore but are now able to slide on their ends, as the fibres continue over the walls of the simulated domain. Again, this boundary condition was developed especially for this thesis as it is not provided by OpenFOAM®. The idea behind this new boundary condition is sketched in Figure 3.31.

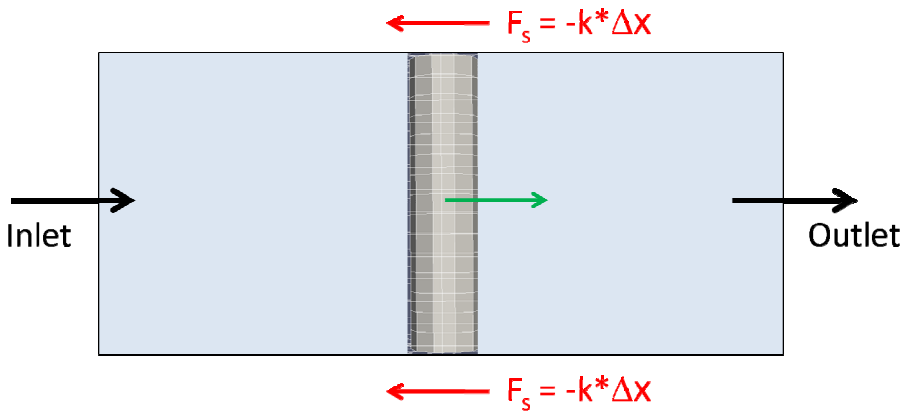


Figure 3.31: Sketch of the new boundary condition for the end points of the fibres

Each fibre can slide along the boundaries of the computational domain until a force retains it. This spring force F_s, which acts reversely on its motion, is defined as

$$F_s = -k \cdot \Delta x \qquad 3.28$$

where k is the spring constant, which is set to an empirical value, and Δx is the sliding length distance of the fibre along the boundary. The dictionary entry for this new boundary condition for the solid region is shown in the following figure.

```
dimensions      [0 1 0 0 0 0 0];

internalField   uniform (0 0 0);

boundaryField
{
    Fiber
    {
        type            tractionDisplacement;
        traction        uniform ( 0 0 0 );
        pressure        uniform 1;
        value           uniform (0 0 0);
    }

    Wall
      {
        type          springBC;
        value         uniform (0 0 0);
        ks            3e2;
      }
}
```

Figure 3.32: Dictionary entry for the solid region with new boundary condition

At the beginning of the movement, the fibre can slide unhindered along the wall of the flow channel. This implies that the force increases with distance and eventually hinders the fibre to move further. After some time, equilibrium develops between the fluid forces acting on the fibre and the spring force. When this point is reached, the fibre end points are regarded as fixed and hence cannot move further, whereas in the middle of the fibre it can keep on moving and therefore will bend under the force due to fluid flow. This behaviour is shown in the following figure.

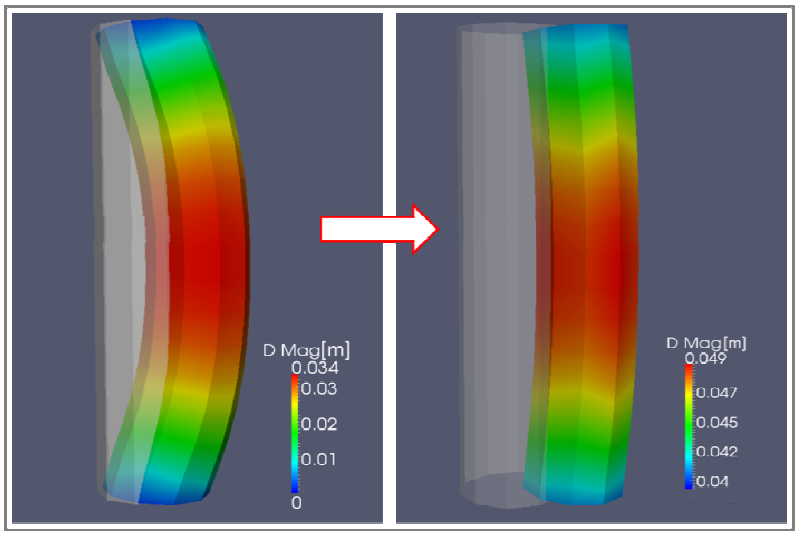


Figure 3.33: Single fibre with fixed boundary condition of the fibre end points on the left hand side and spring boundary condition on the right side.

A clear difference can also be observed in the next Figure. The same geometry as displayed in Figure 3.23 was used.

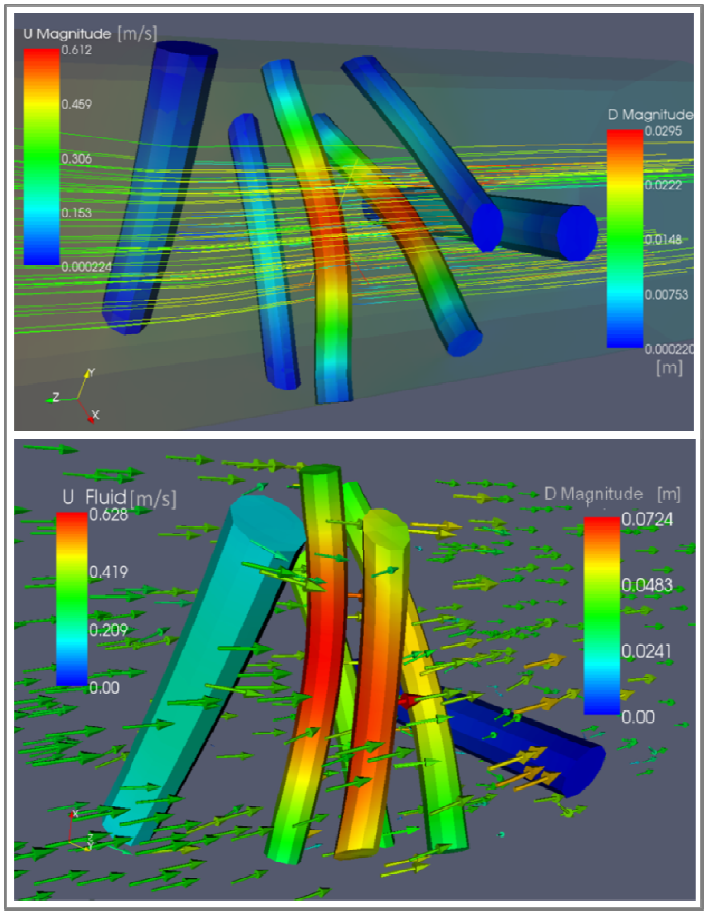


Figure 3.34: Difference between fixed boundary condition in the upper figure and spring boundary condition in the lower figure.

The difference is quite evident. In case of fixed end points, the fibres can only bend in the middle, whereas with the newly developed spring boundary condition, the fibres can be pushed together and additionally bend under the flow of the fluid. This simulates the compression of the filter material in a more realistic manner. It can be said that the implementation of this new boundary condition is considered to be the next step for the application of the complete model to real, from CT-scans reconstructed filter material geometries.

During the implementation of this new boundary condition another problem occurred. After the equilibrium has been reached, the fibres start to oscillate in the fluid stream, as plotted in Figure 3.35. The onset of this phenomenon affected the computational stability and resulted in catastrophic program failures.

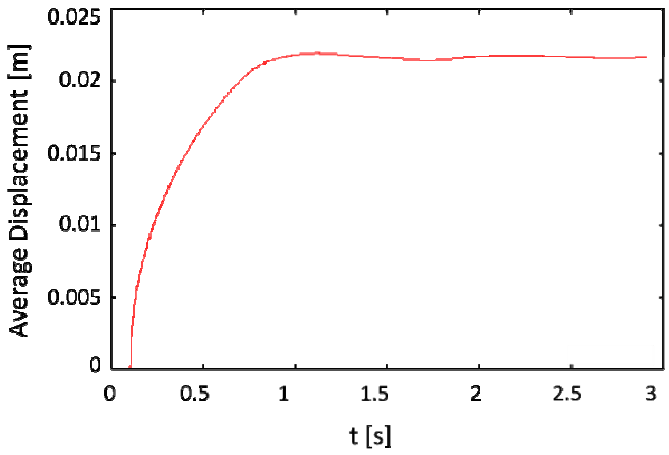


Figure 3.35: Oscillation of a fibre after establishing equilibrium

To avoid this, an improved procedure was designed, where the fibre movement is frozen, when the equilibrium condition is reached for this particular fibre. In Figure 3.36, the displacement of a single fibre is plotted over time. It can be observed that the displacement reaches equilibrium where no further displacement is applied on the fibre. Therefore at this point, which is 0.82 s in this special case, the fibre is frozen and only the equations for fluid flow are solved.

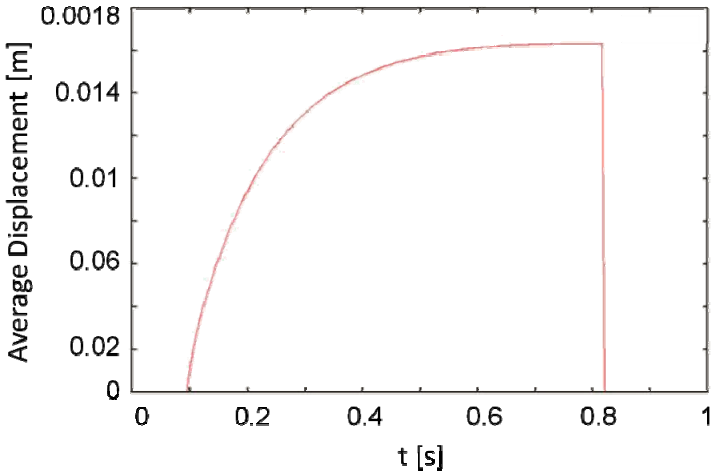


Figure 3.36: Displacement of a single fibre, stopped at t = 0.82s

It has to be taken into account that the fibre shall not be at a halt until the end of the simulation. During the flow of the fluid, the pressure induced forces can change. Thus the motion of fibres is not frozen for the rest of the simulation. It is possible for them to start moving again if the pressure field changes. This is realised by checking the pressure boundary field at the surface of the fibre at each single time step and compare it with the previous one. If it rises above a certain percentage over the old value of the time step before, the fibre is released and movement can be resumed. The plot in Figure 3.37 shows the continuing of the motion of the fibre four times after the first freezing.

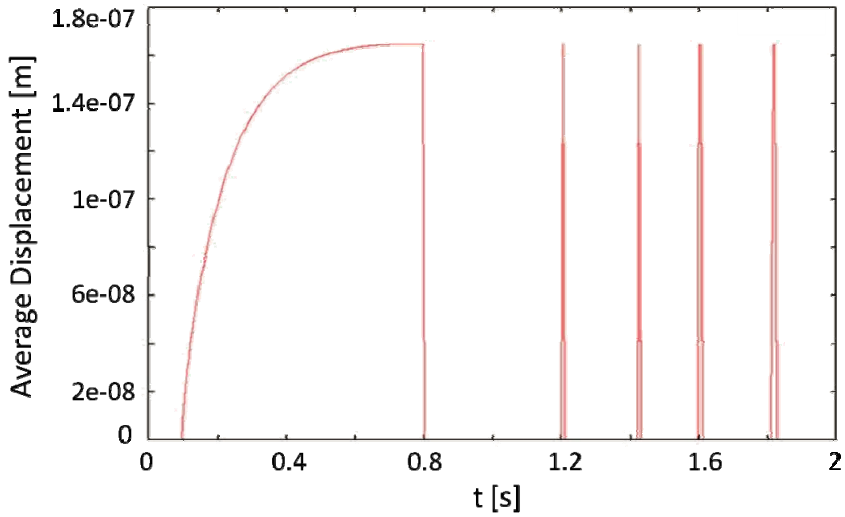


Figure 3.37: Movement of fibre, frozen at time point t = 0.82s, continuing at four other time points

The main advantages of this implementation are that it does not only stabilize the simulation but also saves quite a lot of computational time, as the equations for solid displacement are not solved anymore, when equilibrium is reached. In case of changed flow conditions it is possible to continue movement of the fibres.

3.8.2 Collision Algorithm

When modelling deforming fibres, there is one fundamental constraint to consider. The boundaries of the fibres are not allowed to touch each other under all circumstances. Otherwise it would cause squeezing of the cells of the fluid mesh in between and in the worst case lead to zero or even negative cell volumes. If so, the quality of the mesh is not acceptable anymore. It will lead to severe errors and consequently to termination of the simulation. Therefore it was vital to develop a new concept modelling this behaviour and preventing collision. The main task is to guarantee a conservation of the mesh quality by keeping the fibres at a certain minimal distance and thus the cells alive.

The basis for developing the new collision concept is a special tool based on Matlab® [31] and programmed at ICE Strömungsforschung GmbH [32]. It uniquely marks the geometry of each single fibre in order to detect and distinguish the collision partners. The result can be seen in Figure 3.38.

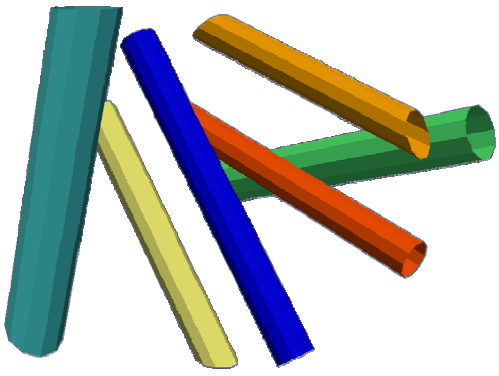


Figure 3.38: Geometry with marked fibres

The idea for the new concept:

1.) Detect the nearest neighbour fibre

2.) Check the smallest distance to the neighbour

3.) If distance is smaller than a predefined critical value:

➡ Apply a pressure force on the nearest neighbouring face element on both collision partners

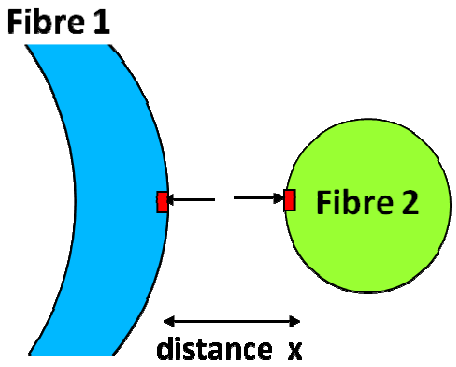


Figure 3.39: A sketch of the collision concept shown for two crossing fibres, the red square shows the nearest neighbouring surface elements

It turned out that for the pressure to be applied a quadratic equation is the best choice.

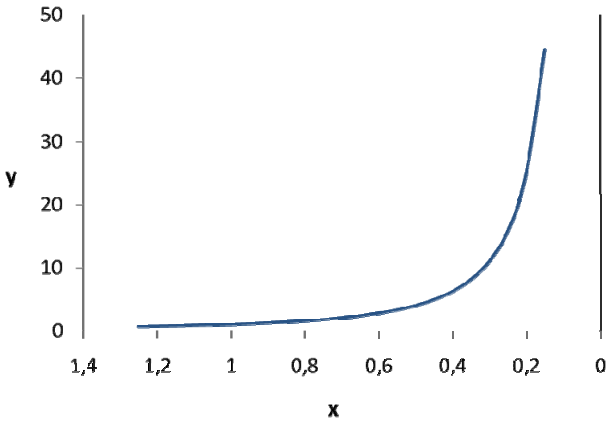


Figure 3.40: Function for additionally applied pressure

At the point where the two fibres are set active for the collision algorithm, there is already initial fluid pressure p_0 acting on the surface of the fibres. Including this pressure and the function plotted in Figure 3.40, the following equation was developed.

$$p(x) = p_0 \cdot \left(1 + \frac{1}{x^2}\right) \qquad 3.29$$

where p is the pressure force to be applied, p_0 is the fluid pressure acting on the surface and x is the distance between the nearest collision partners.

As the distance is inversely proportional to the pressure force, pressure increases quadratically with decreasing distance. Hence the two collision partners can draw nearer but will never touch as the applied pressure force will go to infinity at very small distances. This concept is very plausible and therefore implemented into the developed FSI Solver.

As a next step this concept was tested on a pair of fibres. To ensure possible collision two different diameters were used: a thin fibre at the front and a thick fibre behind. Further on, they were placed by crossing each other and starting from a close distance as displayed in Figure 3.41.

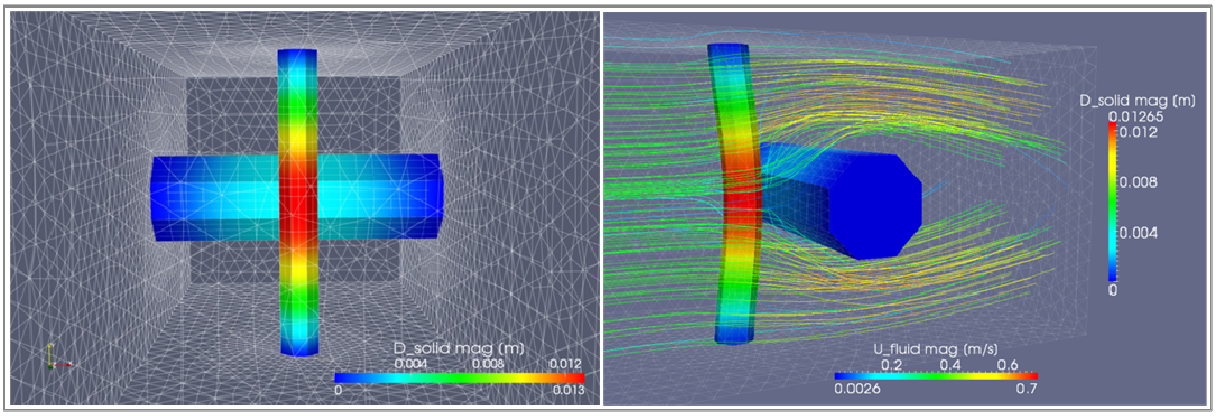


Figure 3.41: Two fibres with different cross section diameters are crossing and deforming under the flow of a fluid.

Applying the developed FSI solver with integration of the new collision concept on the designed geometry has proven reasonable results. This is demonstrated in the following figure (Figure 3.42).

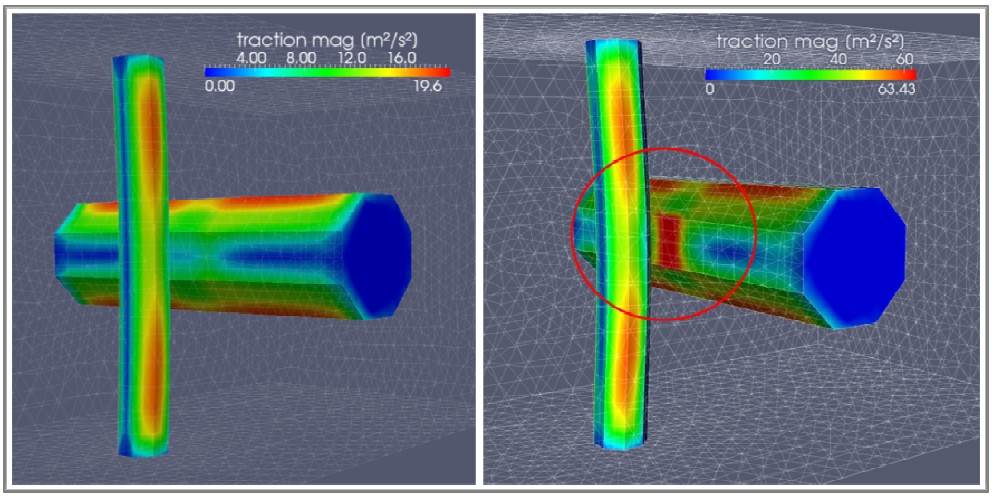


Figure 3.42: Red circle shows the area of the additional applied pressure force on the fibre

The figure on left hand side of Figure 3.42 shows the test geometry with small deformation. Red marked areas are, in contrast to the blue marked ones, areas of high pressure and traction forces. On the second figure of Figure 3.42 the fibre in front is already deformed as much as necessary to fall below the predefined critical value of minimal distance. The area of risk is marked with a red circle. It can be observed that the additional pressure force emerges at the correct position, which is the place of the nearest neighbouring surface elements. It can be identified by the red colour appearing in the circled area. As there the flow regime is regarded as incompressible, the pressure and traction forces are normalised by the density. This results in the unit [m^2/s^2] for the discussed forces.

Further on for a double-ended balance it is important to apply the pressure force on both collision partners. Certainly this is also realized in the solver and can also be observed on the fibres.

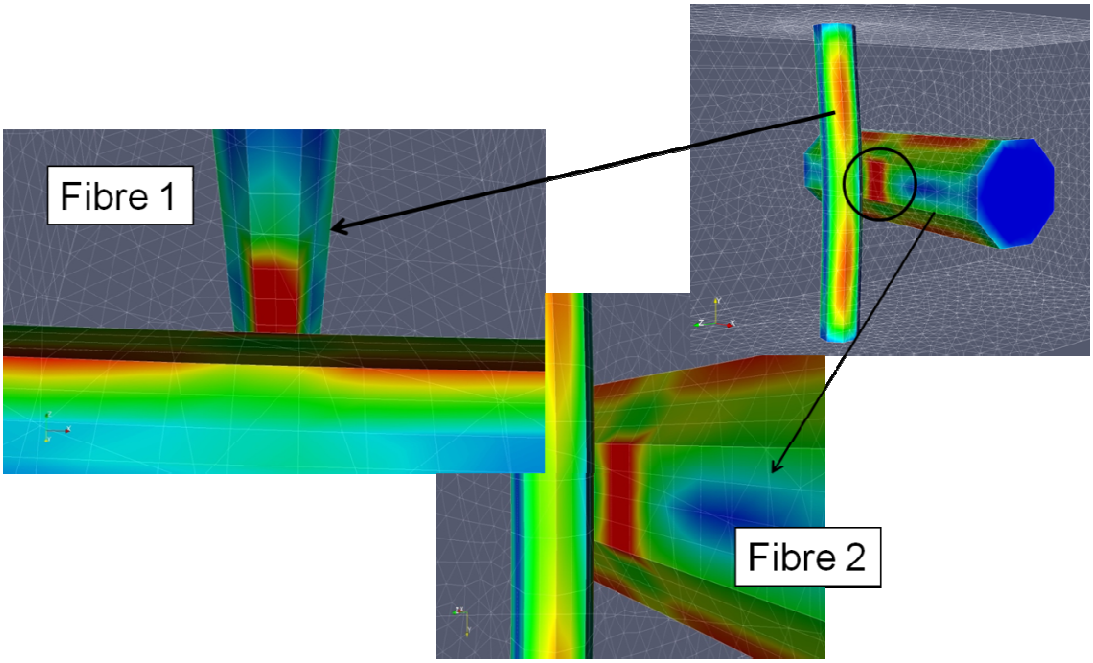


Figure 3.43: Pressure force applied on the surface of both collision partners

It is essential that only the collision partners are affected by the collision concept. The remaining fibres, which do not fall below the critical value, deform undisturbed with no additional force applied. Over time, new pairs of collision partners will arise supplied with the collision concept. The geometry, shown in Figure 3.44 offers the possibility to illustrate this time delay, i.e. the difference between contact time of the two individual fibres. The previous model is extended by adding a second thin fibre parallel to the existing one with an upstream offset. This can be seen in Figure 3.44.

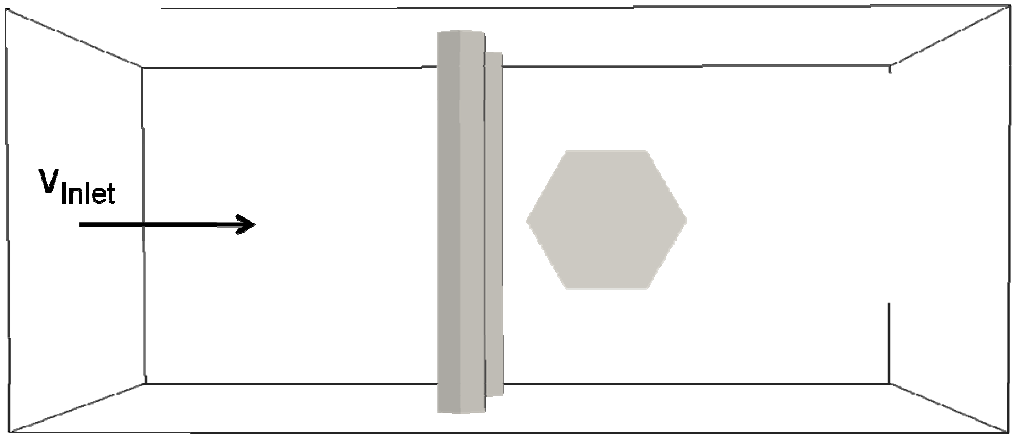


Figure 3.44: Initial state of two parallel thin fibres in front of a thick fibre

In Figure 3.45 the delay of applying the additional pressure can be observed. The fibre on the left hand side is closer to the thick fibre. Therefore it will be the first one falling below the critical distance. At a certain point in time, the additional pressure force on this fibre will be higher than the one on the neighbouring fibre.

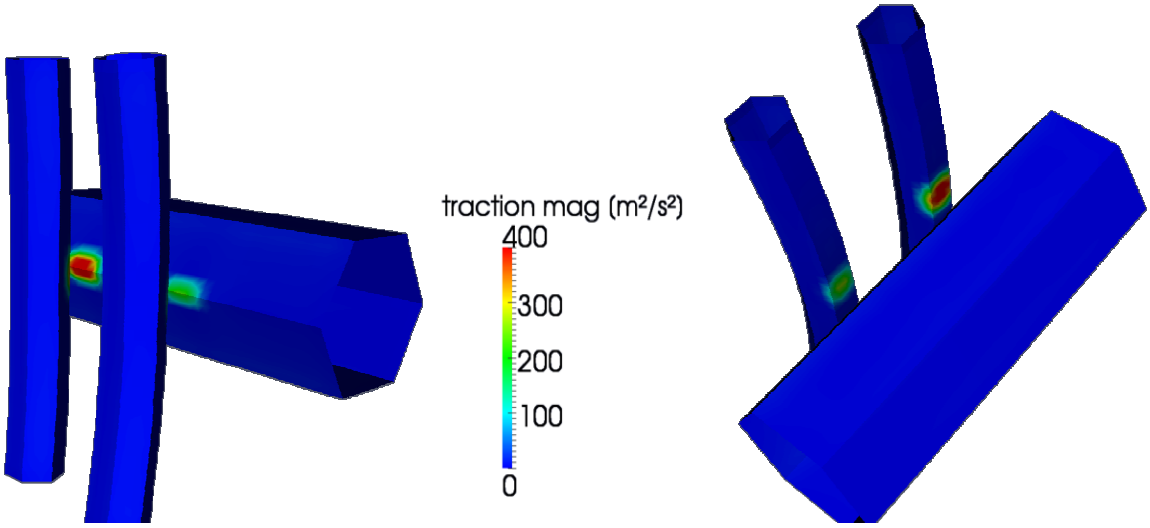


Figure 3.45: Additional applied pressure force depending on the initial distance

3.8.3 Combination

By applying the novel spring boundary condition in combination with the collision concept, a reasonable realistic behaviour of the fibres can be modelled. It enables the possibility to simulate bending of one fibre over another one. This phenomenon implies that fibres with higher mobility like thin or loose fibres can wrap around obstacles like less displaced fibres with a higher cross sectional diameter or fixed fibres. For testing purposes the geometry of the two crossing fibres from the previous chapter has been used. The result of this improvement is depicted in the following figure.

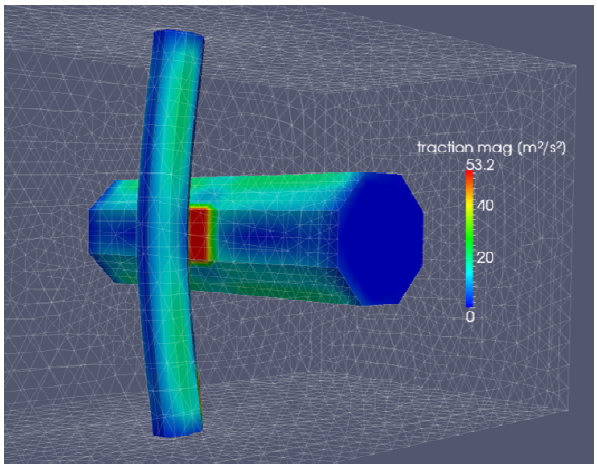


Figure 3.46: Two crossing fibres after applying the novel spring boundary condition and the collision concept

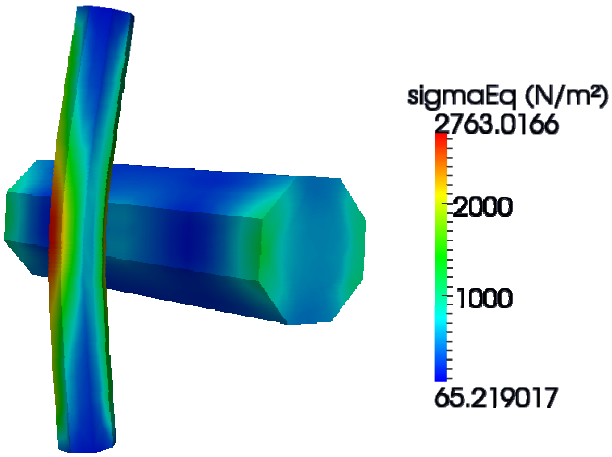


Figure 3.47: Stress distribution on the solid surface

It can clearly be observed that the simulation describes the reality well. The stress distribution on the solid side, shown in Figure 3.47, can be regarded as plausible.

A further realistic behaviour can be observed in the next test case. Two fibres are crossing each other at a right angle and are placed close to each other. The main characteristic of this geometry is the overturn of the second fibre in opposite direction of the fluid flow. This permits gliding of the first fibre along the second fibre, which is illustrated in the following figure.

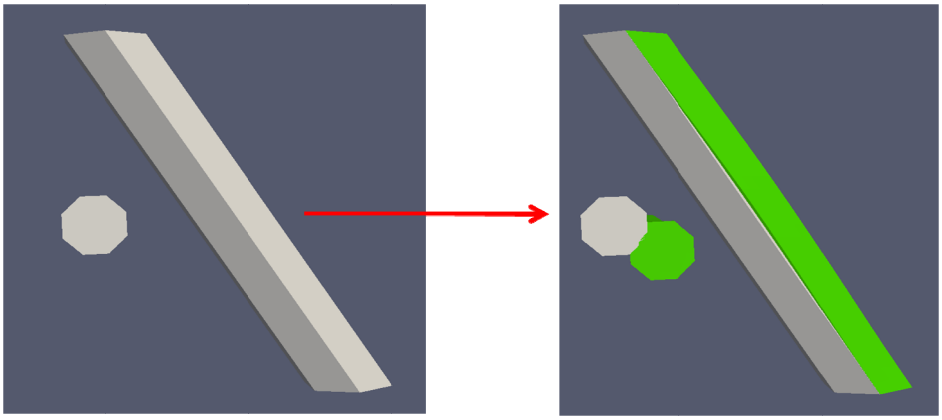


Figure 3.48: Collision concept enables gliding of one fibre along the second one. The grey colour refers to the initial state, green colour to the displaced state.

On the left hand side the initial, undeformed state is shown. On the right hand side, the difference between the initial, coloured in white, and the deformed state, marked with green colour, is illustrated. For simplicity and a clear outcome the backward leaning fibre is fixed. The front fibre can slide along the obstacle and moves as expected along the negative slope. The gliding along the walls of the fluid domain is realised by the spring boundary condition, which retains the fibre at a certain point. The contribution of the collision concept to the results is shown in Figure 3.49.

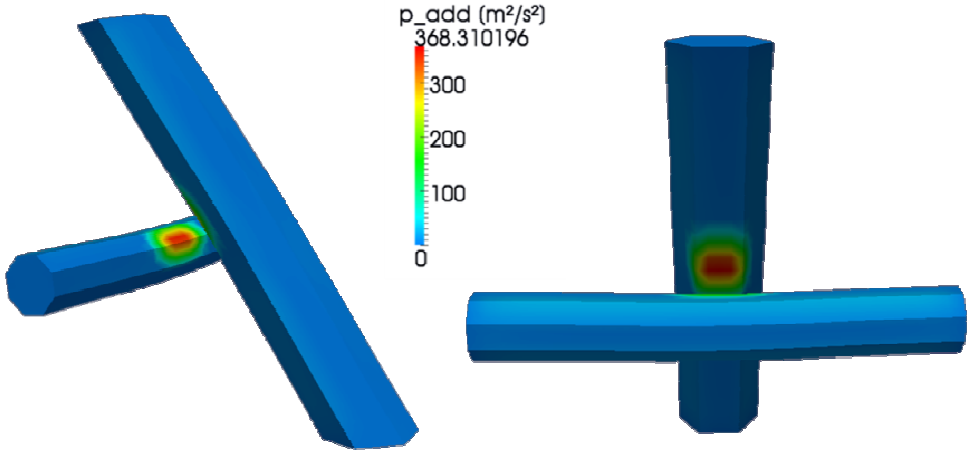


Figure 3.49: Additionally applied pressure force due to collision prevention

Further verification was carried out by using the previously designed case, where two parallel thin fibres are crossing a third, immobile fibre.

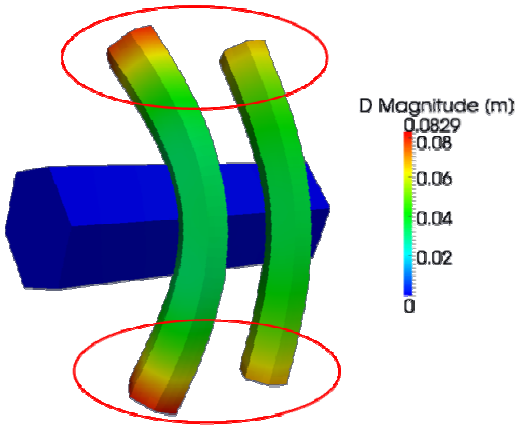


Figure 3.50: Bending of two thin fibres around a thick fibre with marked areas of failure risks.

It can be observed that the wrapping of the fibres is modelled reasonable. At the end points of the fibres, critical areas are detected. In Figure 3.50 those areas are marked with red circles. In these regions high distortion of the fluid cells takes place. The next figure shows a view onto the fluid region from outside with the fibre end sticking out.

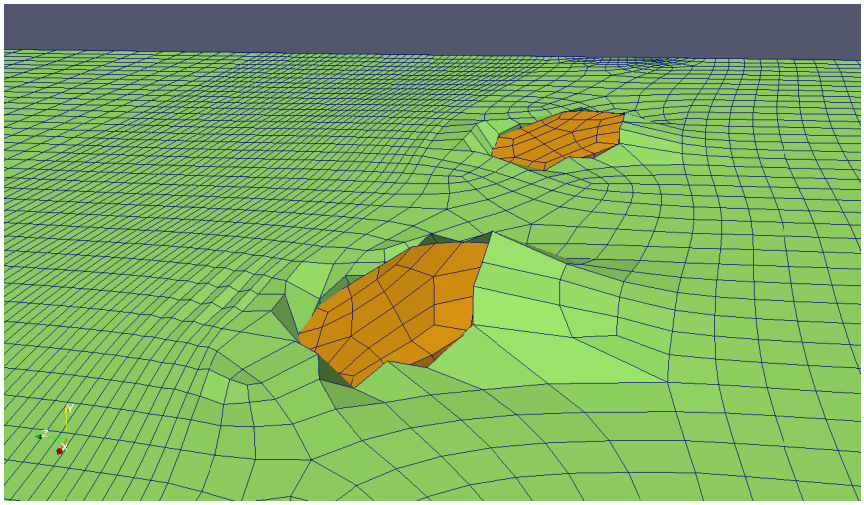


Figure 3.51: View onto the fluid region from outside showing high mesh distortion due to first approach of spring boundary condition

The mesh is highly distorted, which consequently will lead to huge errors in the fluid flow solution and termination of the simulation. To prevent such behaviour the spring boundary condition was adjusted to this problem. The fibre has to stay clipped at the walls in order to avoid tilting of the end points. This can be achieved by keeping the tangent aligned to the walls of the fluid domain. With that, the fibre end points will

glide along the walls, like on a guide rail. This special implementation will guarantee the stability of the mesh during the whole simulation.

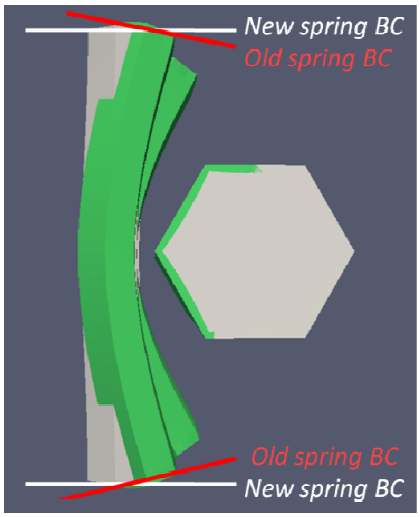


Figure 3.52: Demonstration of the adjusted spring boundary condition, where the tangent stays aligned to the walls of the fluid domain

The following figure (Figure 3.53) clearly shows the difference between the old and the new approach.

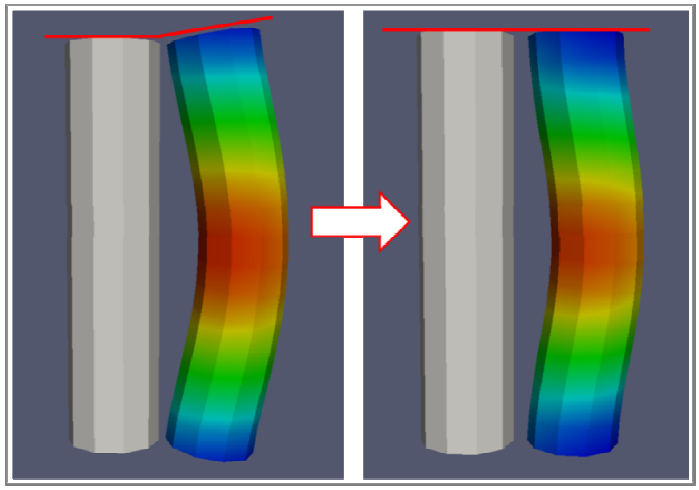


Figure 3.53: Comparison between old and new approach of spring boundary condition. The grey marked fibre shows the initial state

Another advantage of the adjusted boundary condition is the fact that it models the fibre end points in a much more realistic way. As only a microscopic piece of the overall filter material is used, the fibres continue outside of the domain. They continuously extend over the walls with no buckling. To provide it, a symmetric boundary condition, where the change-over tangent at the walls stays the same, has

to be applied. In Figure 3.52 it is demonstrated, that only the new adjusted boundary condition can assure this.

The next case shows three fibres placed in a row within the flow channel. The initial state is shown in Figure 3.54.

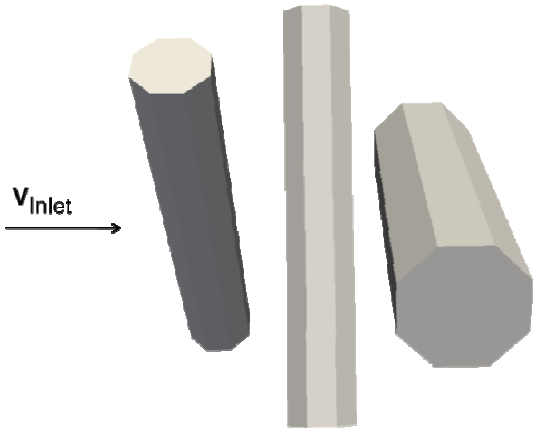


Figure 3.54: Initial state of new test case with three fibres in a row

On this geometry all the new developed and implemented features were tested. At first, the collision concept was combined with the fixed end point boundary condition.

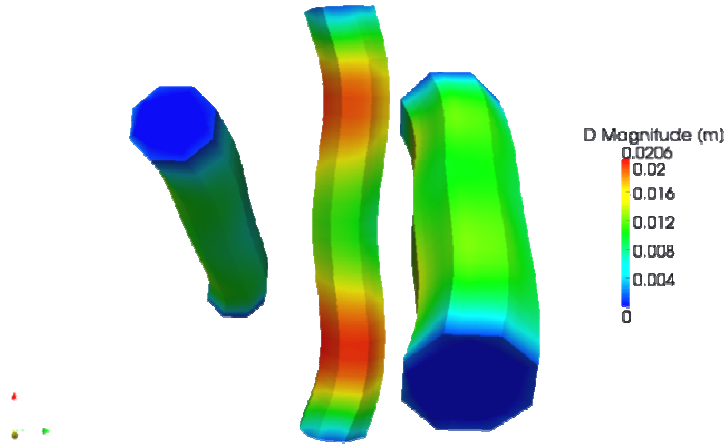


Figure 3.55: Test geometry with enabled collision concept and fixed fibre end point boundary condition

According to Figure 3.55 the deformation of the fibres shows very plausible results. The collision concept allows a wrapping of the fibres. The second fibre is kept at a distance in the middle, whereas the rest can still keep on moving. The only non realistic behaviour can be observed at the fibre end points. They seem very

unnaturally fixed. At this point the new spring boundary condition comes in handy. The following figure shows the result.

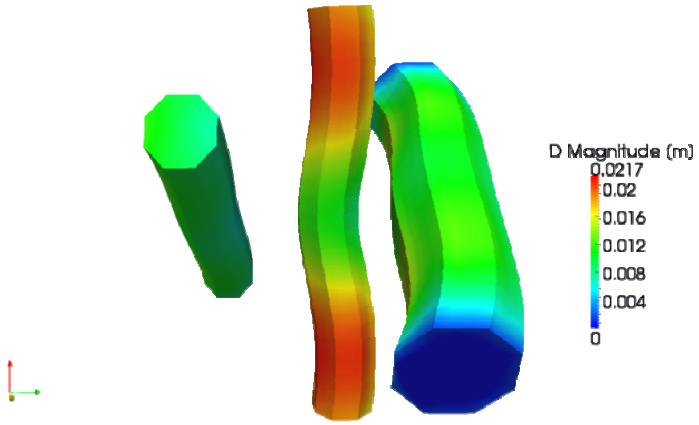


Figure 3.56: Test geometry with enabled collision concept and the new spring boundary condition.

Especially the collision concept can be studied in a very good way (see Figure 3.58).

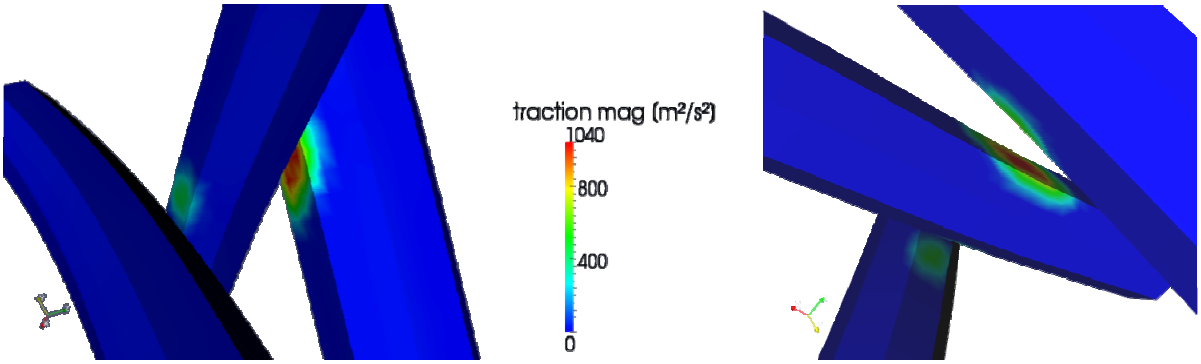


Figure 3.57: Validation of the collision concept with the help of a geometry with three fibres in a row

Regarding the colour of the additionally applied pressure, it can be observed, that the second pair in the row is closer to each other than the pair in front. It is evident that there is a higher pressure applied, which is depicted in red colour and thus the collision progress is already lasting over a longer period of time. The fibre in the middle experiences areas of high pressure on both sides. This provides a realistic dynamic situation, where at each time step a new equilibrium is tried to be established by adjusting the whole system to the present fluid flow situation and conditions.

The spring boundary condition and the collision concept, separately and in combination, offer a very powerful tool for the simulation of the deformation of filter fibre material. It allows a dynamic description of the displacement of the fibres. Therefore the compression of the material is modelled in a realistic manner.

3.8.4 Investigation of Different Branching Types

In realistic fibre modelling there are not only single straight fibres present, but also different types of branching fibres. A variety of possibilities was investigated to ensure the applicability of the newly developed solver. Especially the collision concept was verified. The first test case is a set up of two straight fibres, connected via a third orthogonally positioned fibre of the same diameter. A fluid stream from the left hand side is provided. The following figure shows the initial state.

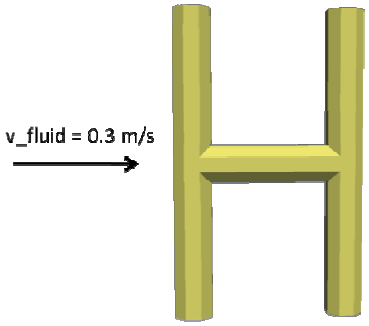


Figure 3.58: Initial state of two straight fibres, connected via a third orthogonal fibre

The results of this test case can be observed in Figure 3.59. It shows the displacement of the fibre combination due to to fluid flow.

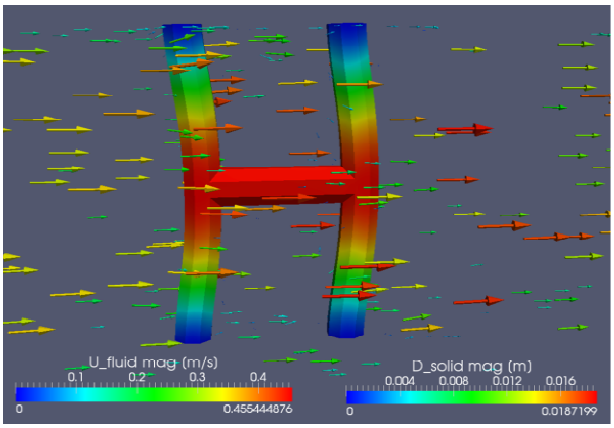


Figure 3.59: Displacement of two fibres connected with a bridge fibre

To verify the correct implementation of the collision concept the traction force, which contains the additional pressure, is mapped on the resulting geometry. This can be observed in the following figure.

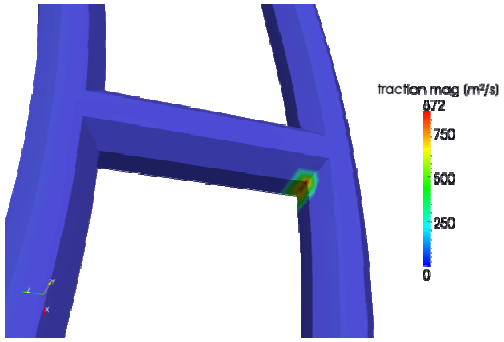


Figure 3.60: Additional traction force due to the collision concept

The collision can be observed at the intersection between the bridge and the second fibre. It prevents the fibres from collision and additionally from squeezing of the edge cells. Further on other geometry set ups were tested. The deformation results of two characteristic fibre combinations are shown in the next figure. It can be observed that also the crossing of 2 fibres, placed in two ways into the flow channel can be simulated in a realistic manner. Therefore the modelling of realistic fibre geometries is rendered possible.

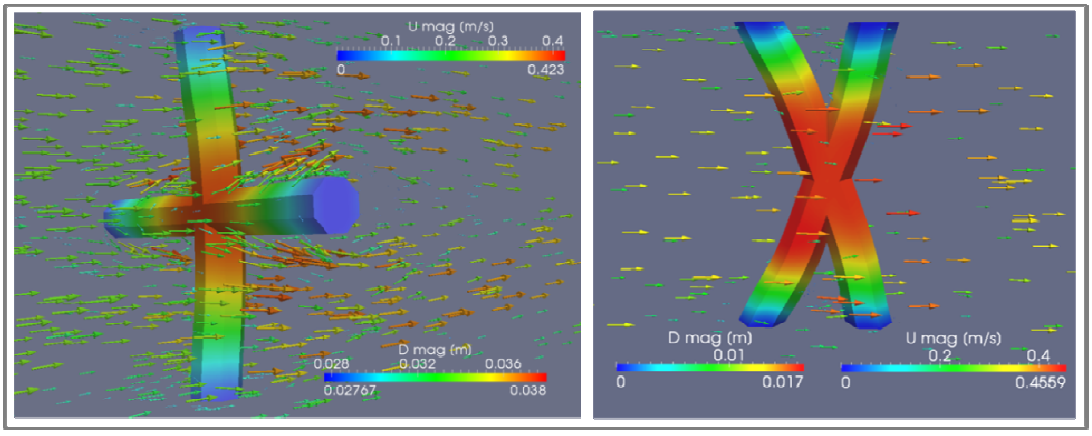


Figure 3.61: Results of deformation of two crossing fibres placed in two different manners in the flow channel

Again it should be stressed that only small displacements are modelled. Modelling of large displacements will cause severe mesh problems due to squeezed cells at the junction points of the crossing fibres.

4 Application to Digitally Reconstructed Fibres

A fully functional FSI-solver with additional, novel boundary conditions was developed. Testing on different types of fibres has proven that it can be used for simulation of realistic, digitally reconstructed fibre material. Until now it is not possible to simulate a complete engine oil filter on the microscopic scale. For a high resolution it would lead to several billion cells or even much more, impossible to simulate with the computer resources down to the present day. Therefore representative sections of a filter material of interest are taken. From that global material parameters, such as permeability and true surface/volume evolution are gained. The integral results of these simulations can be used for analysis of complete filter systems, where, on a global level, porous materials are treated as homogenous regions.

At first, a section of the overall material was chosen. A characteristic CT-scan picture of the filter material called Ahlstrom A55 is shown in Figure 4.1.

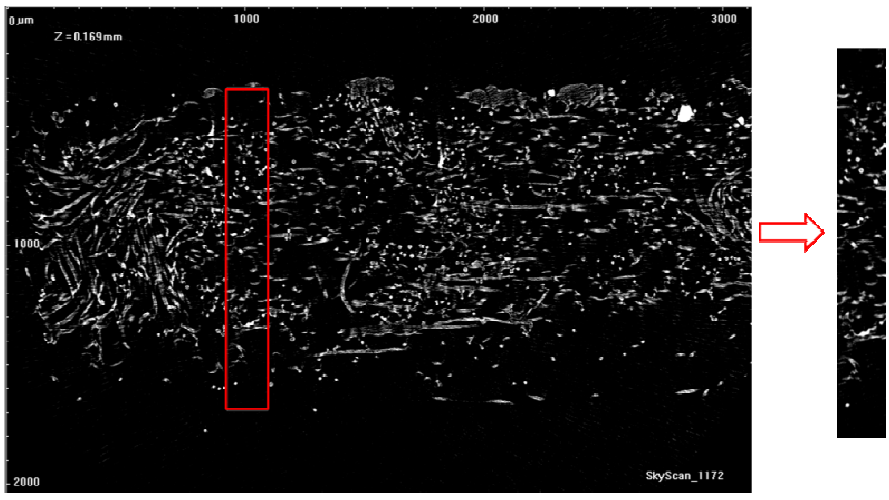


Figure 4.1: CT-scan of a real fibre material Ahlstrom A55

Further on, the red marked section was reconstructed and transformed into a voxel based computational grid readable by OpenFOAM®. This was carried out by a Matlab® based algorithm developed at ICE Strömungsforschung GmbH.

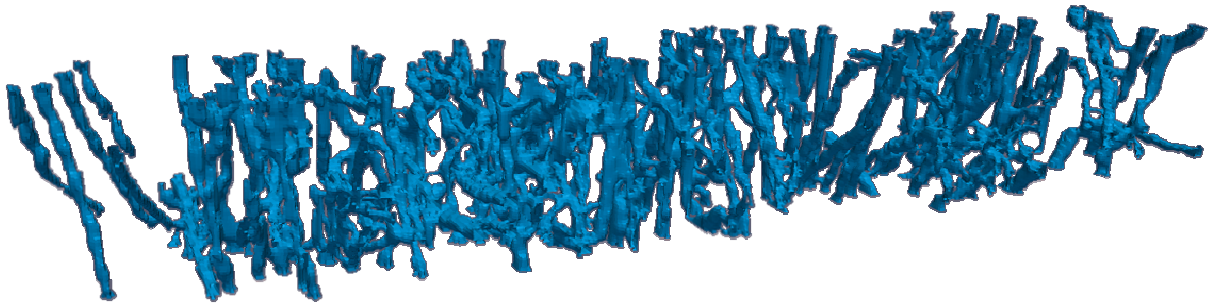


Figure 4.2: Reconstructed filter material Ahlstrom A55 (170x170x1000µm), transformed into a voxel based grid

This geometry consists of around 2 million cells. A simulation of this geometry on a single processor would take several weeks up to months. Hence it is vital to find ways and means to reduce the simulation run time. Two main options are presented in the following chapter.

4.1 Computational Optimisation

Until now, only small, artificially constructed geometries were used, where the cell number ranges between 4 000 and 200 000 cells. In reconstructed realistic fibre material, the cell number of the computational mesh is a lot higher. For smaller cut-out geometries it starts with around 600 000 cells but can also reach more than two million cells. Therefore, it is vital to optimize the simulation process in order to decrease the time for performing a run.

4.1.1 Parallelisation

With the growing number of mesh cells, the request on computational performance is increasing exponentially. One way to deal with that is to distribute the simulation on different processors and let them work simultaneously. This is done by splitting up the spatial domain and assigning each part to one processor of a parallel computer.

The process of parallel computation involves:

- Pre-processing: Decomposition of mesh and fields
- Running the solver in parallel
- Post-processing: Reconstruction of the mesh and field data

The decomposition of the spatial domain is performed by using the so-called decomposePar utility provided by OpenFOAM®. The aim is to break up the domain with minimal effort and reaching a fairly economic solution. All the parameters are handled via the decomposeParDict, found in the system folder of the case. The method used is called "metis" and requires no geometric input from the user. It attempts to minimise the number of inter-processor boundaries. Further information of other decomposition methods and their usage in OpenFOAM® can be found in the OpenFOAM® user guide [10].

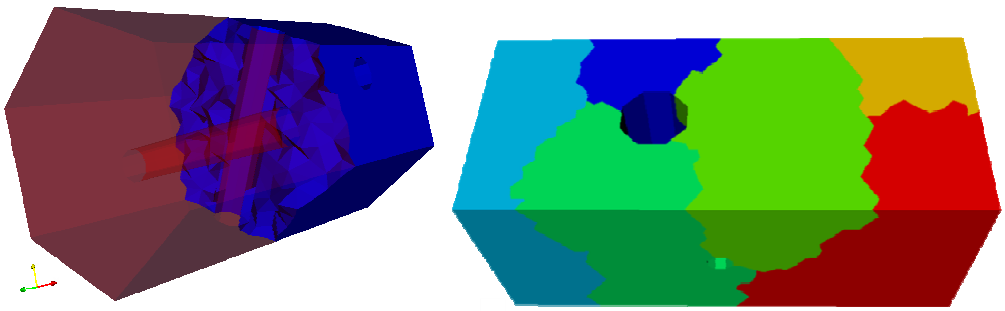


Figure 4.3: Examples for decomposition of the spatial domain. On the left hand side split in two sub domains, on the right hand side in six sub domains.

To run the solver in parallel, the underlying source code has to be parallelised. Originally the code consists of a sequence of instructions, which are executed one after the other. When splitting up the domain, the solver shall be able to exchange and process information over inter-processor boundaries. OpenFOAM® handles this issue with the public domain OpenMPI implementation of the standard message passing interface (MPI) [33]. Therefore all the applications, equations and numerical schemes are already parallelised and can be directly used. For more details on parallel and high performance computing and programming, information can be found in [34,35]. Therefore the main work on the newly developed solver was to adjust the new boundary conditions, especially the collision concept, which checks the minimal distance between all fibres to a multiprocessor application.

For testing the performance of the parallelised solver a piece of realistic reconstructed fibre geometry was used. The physical dimension of the fluid domain is

40x40x200 µm. The overall mesh consists of 185 000 cells, which includes the fluid and the solid mesh. The solid region is shown in Figure 4.4.

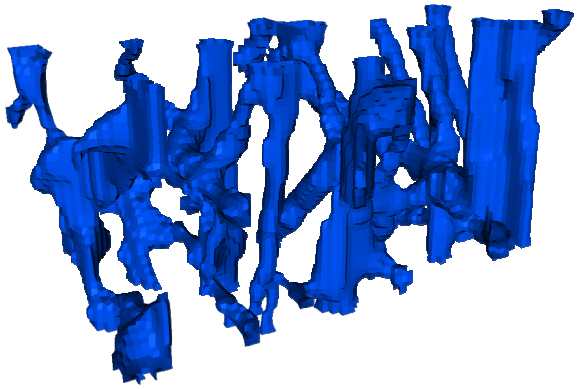


Figure 4.4: Reconstructed fibre material for testing of parallelisation (120x120x150µm)

The available computer resource for running simulations is a LINUX cluster of 7 nodes, each equipped with two quad core processors with a clock rate of 2,3 GHz and a memory of 16 GB. A Gigabit Ethernet works as interconnect.

Three cases were set up:

1.) Single domain for single processor run

2.) Decomposition of the domain into two sub domains for a two processor run

3.) Decomposition of the domain into four sub domains for a four processor run

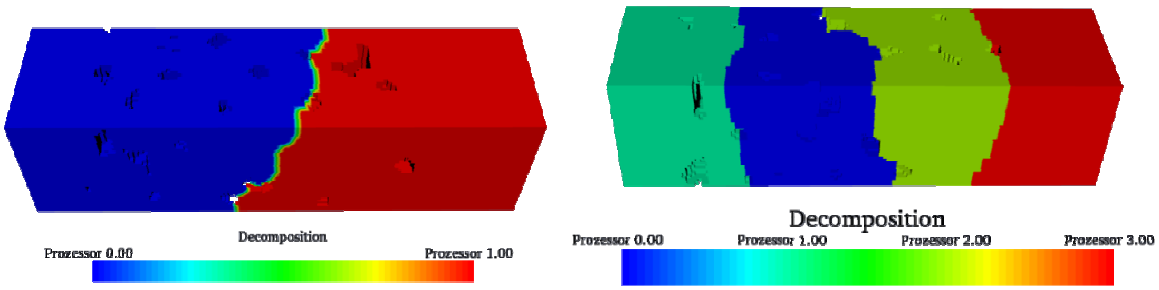


Figure 4.5: Decomposition of real fibre geometry in two and four sub domains

For evaluation of the parallelisation performance a factor called speed-up is used. It is defined by the ratio of the time needed for a sequential single processor run and the time needed for a parallel processor run.

$$S(p) = \frac{T(1)}{T(p)} \qquad 4.1$$

where S(p) is the speed-up for a p processor run, T(1) the time used for a single processor run and T(p) the time used for a p processors run. The value of the speed-up is normally in between 1 and p. In exceptional cases it can be more than p, which is called super linear speed-up. It comes from the fact that multiprocessors can use their resources in a more efficient way, such as reduction of RAM access time. When executed sequentially, the working set of a problem may exceed the cache size of a single processor, whereas parallel processors with shared memory computer architectures can provide more memory/cache. [36]

On this basis, the efficiency can be derived:

$$E(p) = \frac{T(1)}{p \cdot T(p)} = \frac{S(p)}{p} \qquad 4.2$$

where E(p) is the efficiency for a p processor run and S(p) the speed-up for a p processors run. The value for the efficiency is in the range of 0 and 1.

In reality the speed-up is lower than the ideal one. The reasons are:

- It is not possible to fully parallelise an algorithm. Sometimes processors are out of work and have to wait for the others to complete their task
- Additional time is needed for communication between parallel working processors

Therefore the factor speed-up defines the performance of the parallelisation compared to the loss of time in communication and synchronisation.

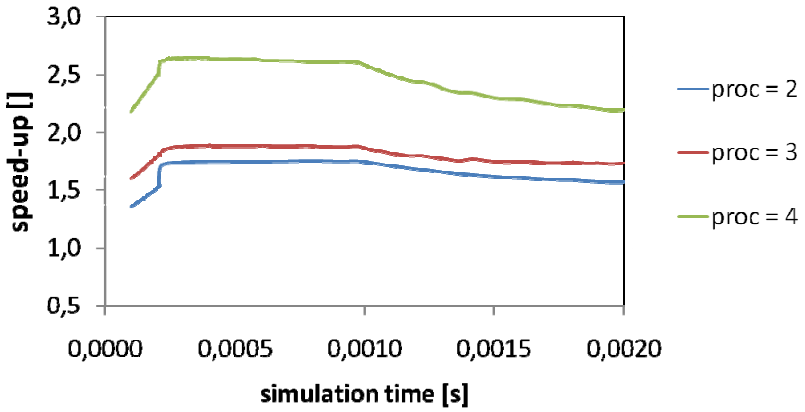


Figure 4.6: Speed-up for parallelisation of the FSI algorithm tested on the real fibre geometry shown in Figure 4.4

In general, with more and more processors used, the speed-up is decreasing due to more communication time needed. The efficiency for this test case is shown in the following figure.

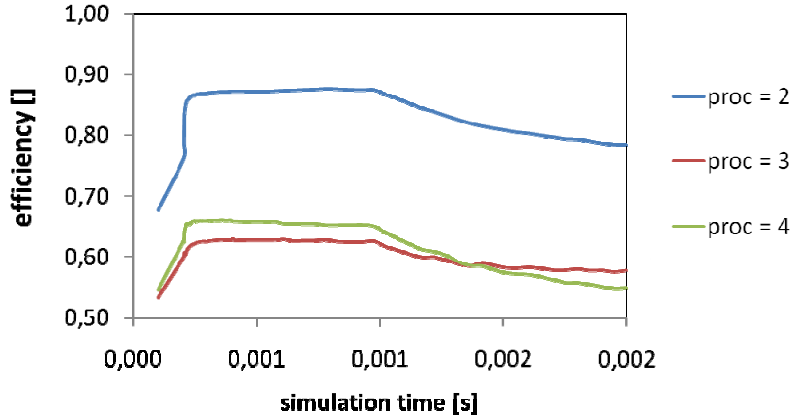


Figure 4.7: Efficiency of parallelisation

Due to convergence issues the simulations are started without FSI. Only later, at time step 0,001s the FSI part is switched on. This can be observed in the latter plots as the speed-up and efficiency are decreasing from this time point on. In this specific test case it is not efficient to use more than 4 processors because the efficiency for four processors in comparison to three is not increasing in a favourable way. Especially in case of fluid-structure interaction, the efficiency for a four processor run is even lower than for a three processor run. This comes from the fact that the serial part of the FSI code is very high, because not all sections of the code can be parallelised. As expected the structure solving part costs a lot of time, especially the

collision concept, which checks the distance between the fibres at each time step and hence over processor boundaries. However, the parallelisation is still regarded as a big improvement. It permits the simulation of realistic fibre geometries within reasonable simulation time.

4.1.2 Restart

Due to the long wall clock time, necessary for one simulation run, it is necessary to implement the possibility of restarting the simulation at any wanted time step.

There are several incidents requiring a restart option:

- *Save simulation in case of machine failure:* there is always the possibility for the system to fail, either due to power loss or simple hardware failure of individual nodes or processors. In order to avoid starting again from the beginning it is practicable to continue the simulation from the latest saved time step on and hence economize run time.
- *Change parameters of simulation:* during a long simulation run it can be necessary to change parameters, without starting the overall run from time point 0.

The main challenge in making the solver restartable is the issue of moving the computational mesh. The points at initial state and all the special parameters for the collision concept are not saved over time. Hence a new dictionary called "restart" is introduced. When starting a simulation, it creates itself automatically inside the case using mechanics provided by OpenFOAM®. It saves the state of the simulation and allows the run to be restarted. It was tested on a simple geometry, shown in Figure 3.21. The simulation was interrupted at a certain time point. Then it was restarted at an earlier time point in order to reach an interval of overlapping of the solutions. The accuracy of the solution for the magnitude of the maximum appearing velocity before and after the restart can be observed in Figure 4.8.

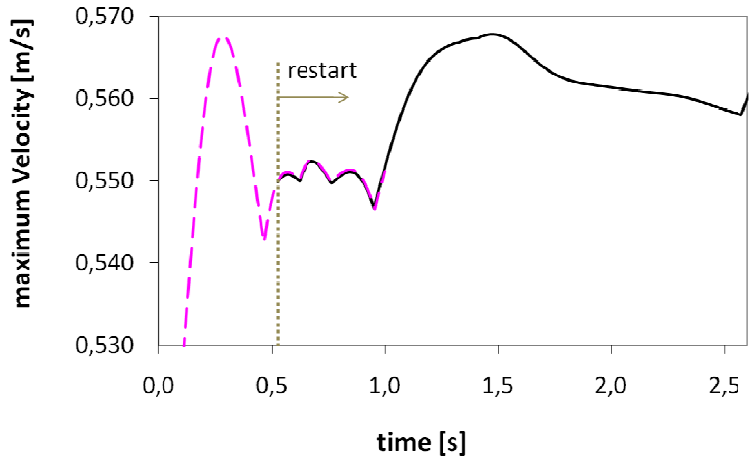


Figure 4.8: Comparison of the magnitude of the maximum velocity before and after restart, demonstrated with an overlapping interval; Restart time: t = 0.51s

It can be seen that the solutions of both analyses in the overlapping interval agree very well. In order to reach a more accurate match, the points and parameters shall be written out with at least double the amount of decimal numbers. The disadvantage of writing out more decimal numbers is the increasing computational time and also consumption of data storage. It depends on the needs of the user and hence can be specified in the controlDict of the system folder.

Also for the maximum pressure solution, a very good result agreement was achieved, shown in Figure 4.9.

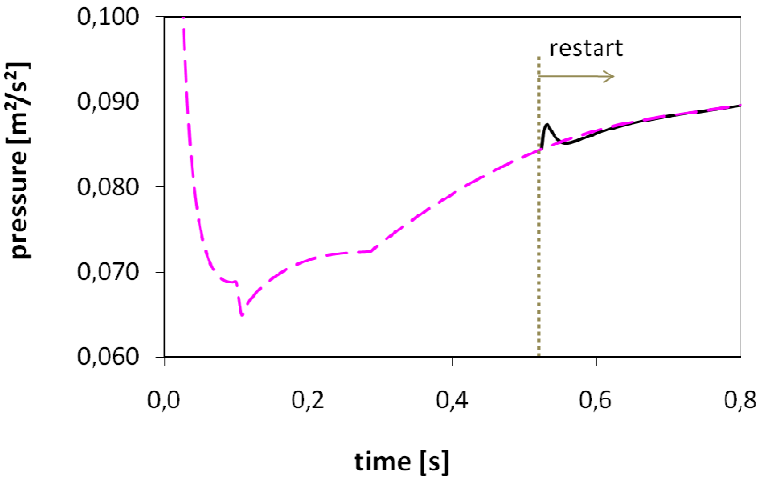


Figure 4.9: Comparison of maximum pressure before/after restart; restart at t = 0.51s

The small pressure rise at time point 0.5s can be explained by the fact that when restarting the simulation the pressure is in transient mode and has to readjust itself to the saved steady state fluid flow conditions. This phenomenon was also observed in simulations with commercial software such as FLUENT® and in stock OpenFOAM® solvers. After a short period of time, the pressure solution converges to the original solution. Therefore this effect can be neglected. The implementation of this restart option was a significant step, as especially in simulation of reconstructed real fibre geometries, the possibility of restarting the simulation at any saved time step is very important as it will save a lot of time in case of unexpected incidents.

4.2 Results

4.2.1 Fibre Deformation

At this point the fully optimized solver was ready to use on realistic reconstructed fibre material. The first type of filter material investigated is called FULDA A43. The following figure shows a CT scan of this material.

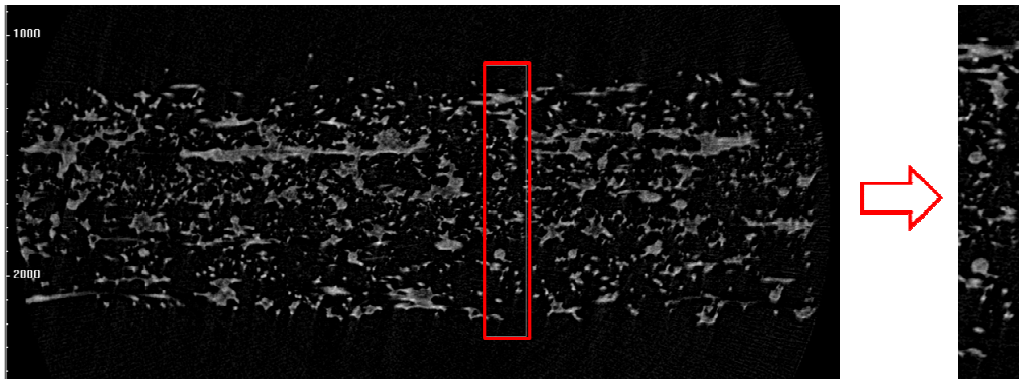


Figure 4.10: Computer tomography scans of FULDA A43.

From this scan, the geometry was digitally reconstructed and is displayed in the subsequent figure (Figure 4.11).

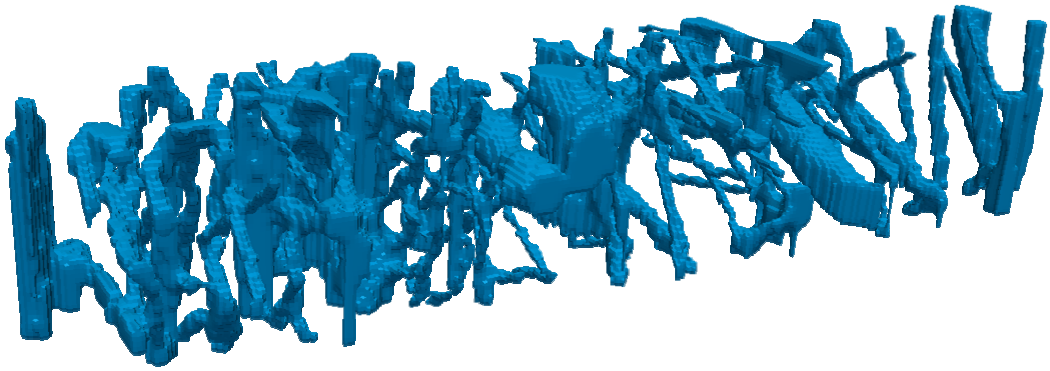


Figure 4.11: Reconstructed filter material Fulda A43 (170x170x1000µm)

This fibre material consists of different types of fibres. They have different cross sectional diameters and additionally considerably vary in shape. Most of them are cylindrical, whereas others are flat. This can be observed in the following figure, which shows a representative small cut-out of the red marked section.

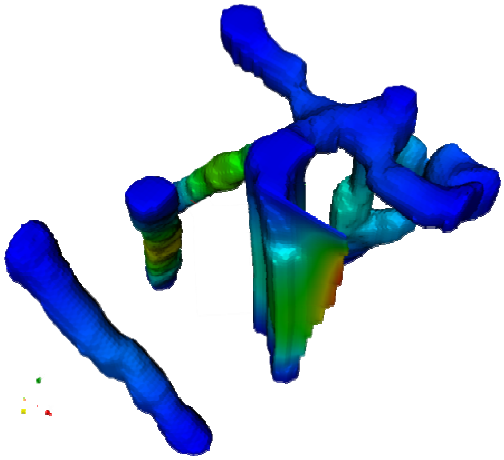


Figure 4.12: Different shapes appearing in a real fibre geometry (120x120x80µm)

The flow field around such a fibre composite is depicted in the following figure:

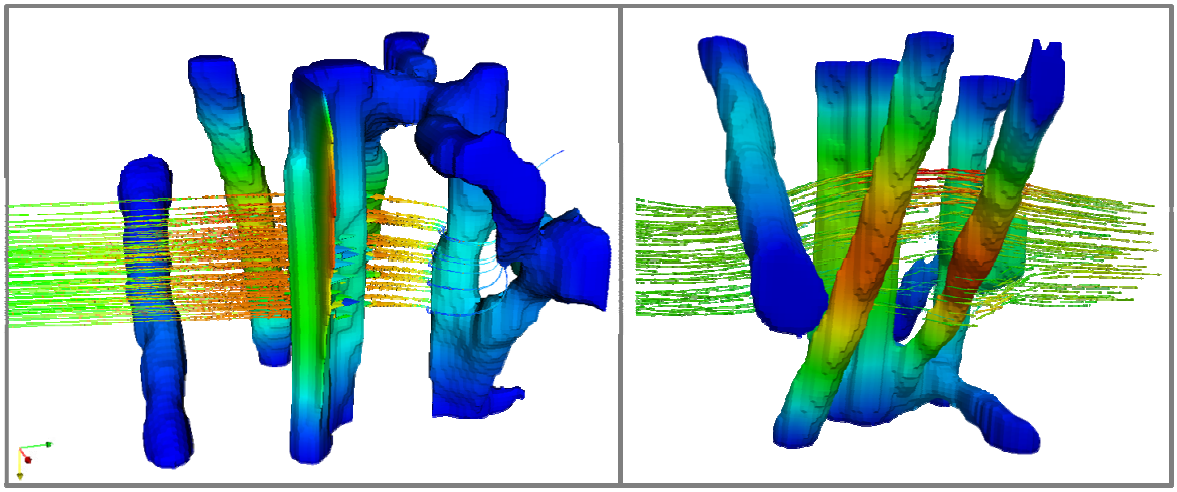


Figure 4.13: Flow field around a characteristic fibre composite shown from two different views

On the left hand side the geometry is shown from the side of the thick fibre composite. It can be observed that the streamlines of the fluid flow field show a very plausible behaviour. On the right hand side it is displayed from the opposite side, where loose fibres are present. The flow field is deviated due to the solid parts and furthermore deforms the fibres.

The geometry used for testing parallelisation is a larger cut out of Figure 4.11. The newly developed and optimized FSI solver was applied and resulted in the following displacement.

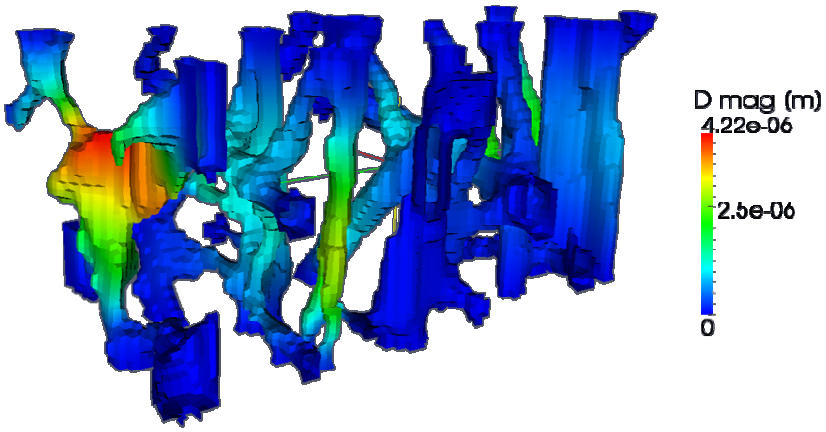


Figure 4.14: Displacement of a larger cut out of Fulda A43 (120x120x300)

The overall geometry consists of a high difference in cross section diameter of the fibres. Some are very thin compared to the rest and hence bend more under the fluid flow, whereas denser parts with a lot thicker fibres are more rigid. The front fibre section is displaced more than the rest of the fibres as it has a bigger contact surface and the fluid forces are higher at the beginning of the flow channel.

Another section of Fulda A43 is the following geometry (see Figure 4.15). It is shorter in length but consists of a representative maze of fibres, which comprises regions of thinner fibres but also connected clusters.

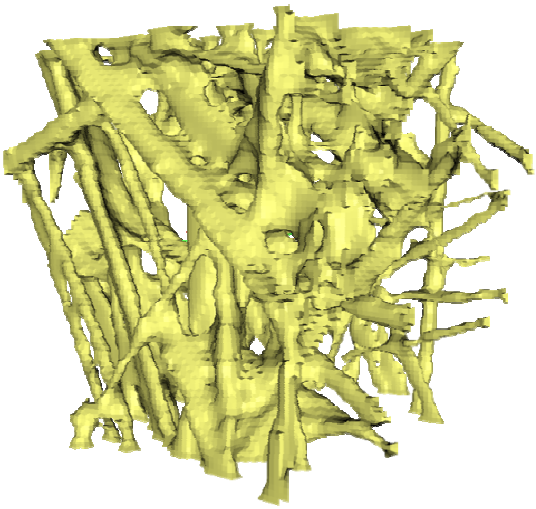


Figure 4.15: Initial state of another section of Fulda A43 (300x300x200µm)

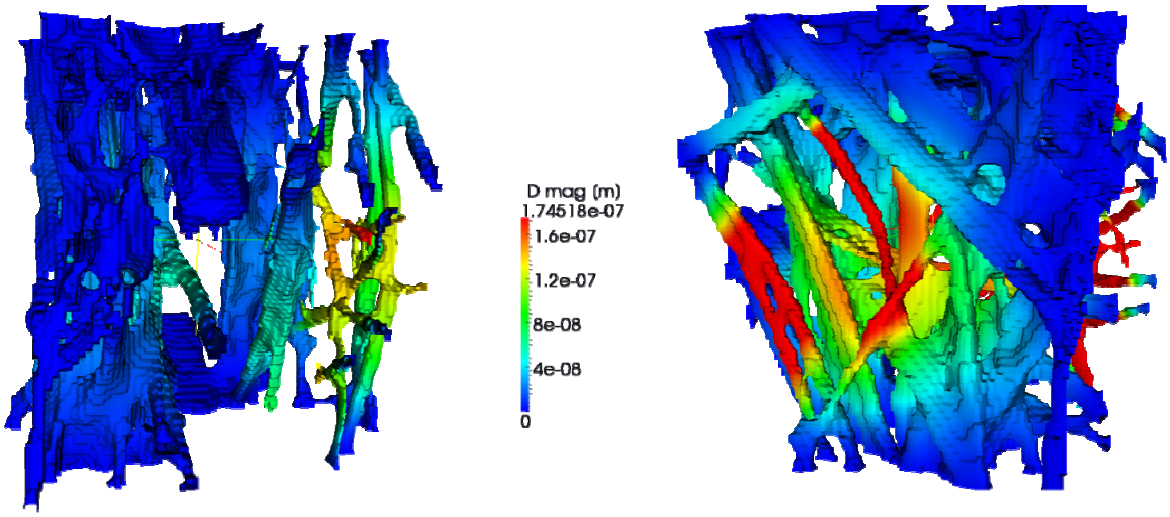


Figure 4.16: Displacement results for Fulda A43 (300x300x200µm) in two different views

It can be observed that the lattice of thin fibres at the back of the fibre material is displaced in a high manner, whereas the composite of connected interwoven fibre material in the middle section is more rigid. This is shown on the left hand side of Figure 4.16. On the right hand side, the geometry is displayed from a different view, where the front fibre deformation is presented. As there are thinner fibres present, higher values of displacement are monitored.

The filter material Ahlstrom A55 is of another type. The CT scan and reconstruction was already shown in Figure 4.1 and
Figure 4.2. The main difference to Fulda A43 is that it consists of a higher amount of similar, cylindrically shaped fibres, mainly loose, at some points interconnected with small short bridges. The deformation of a representative section of the reconstructed geometry is demonstrated in the following figure.

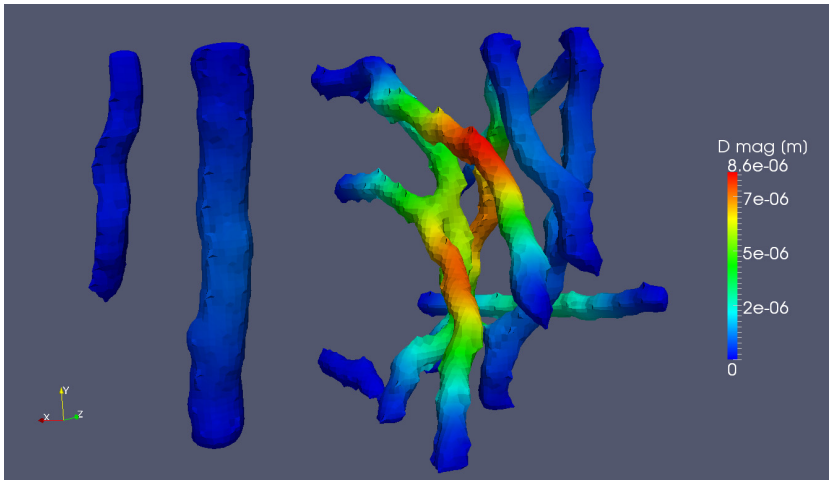


Figure 4.17: Representative section of Ahlstrom A55 (170x170x300)

There are more stand-alone, thinner fibres, which are displaced at a higher amount and further on result in a higher compaction of the overall filter material. As this short geometry is not very representative for the overall filter material a longer section was investigated.

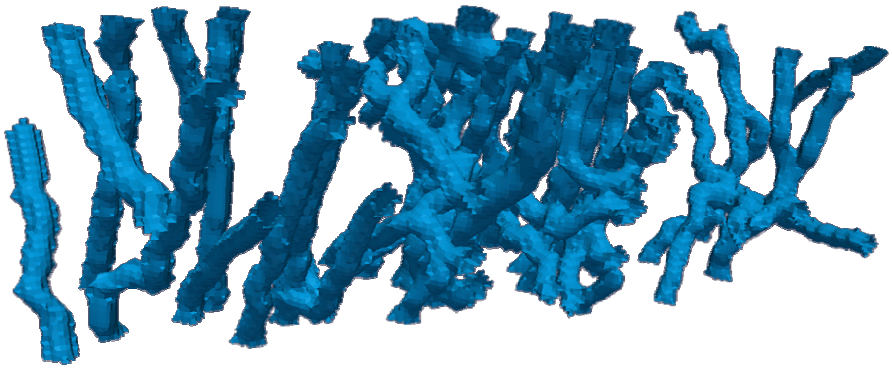


Figure 4.18: Initial state of a longer section of Ahlstrom A55 (170x170x700µm)

Applying the newly developed FSI solver plausible results were achieved. Especially the fibres in front, where the fluid pressure is at maximum, are deformed the most. In the middle part, a cluster of connected fibres is present. This results in less deformation and less compression. In the last part, the forces induce again higher deformation, whereas fibres in the middle of the flow channel are more displaced due to higher flow velocities. The resulting geometry is depicted in the following figure.

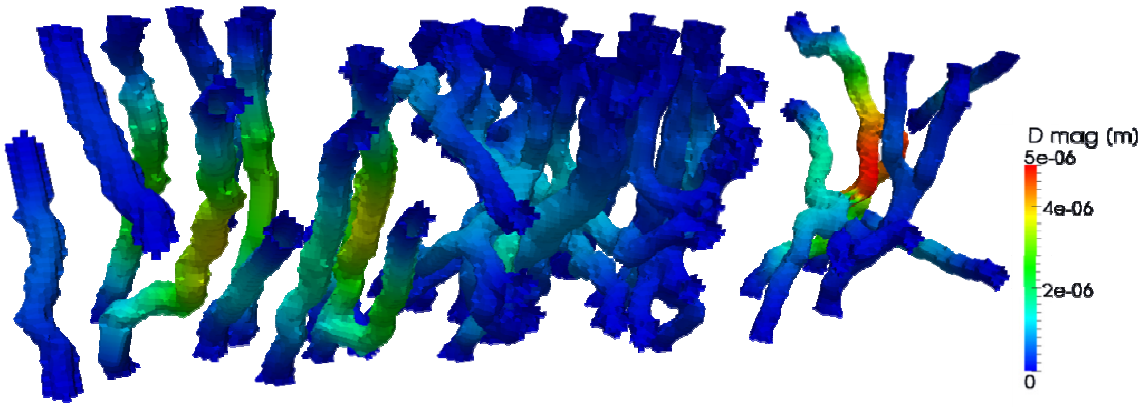


Figure 4.19: Results of deformation of a longer section of Ahlstrom A55

By viewing the geometry from the side, the resulting compaction and compression can be observed (Figure 4.20).

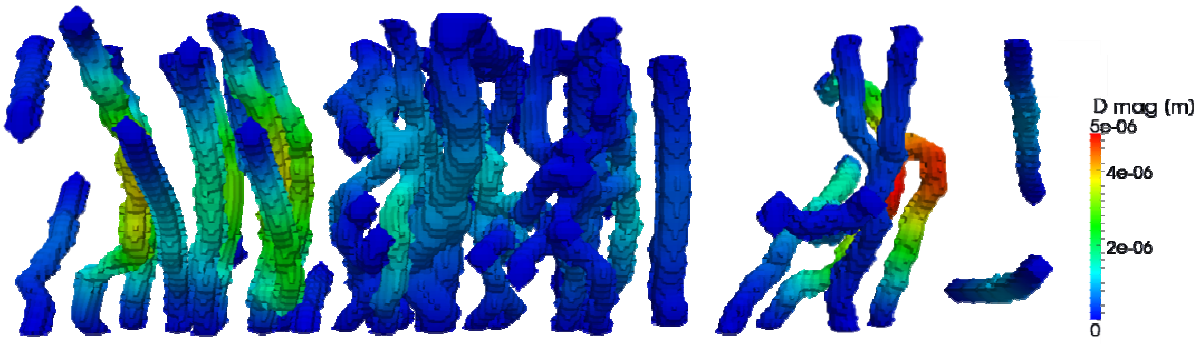


Figure 4.20: Side view of compaction of Ahlstrom A55

Especially in the front part, the predominant compression is evident. The influence of this behaviour on the overall filtration properties for different filter materials will be discussed in detail in chapter 5.

4.2.2 Combination with Dirt Particles

The aim of the second part of the overall project was the development of a Lagrangian dirt particle model. The particles are considered to be large particles. It means that they are at least the size of a grid cell and can even span several hundred cells. They contain a mass and a momentum. Therefore the discrete phase Lagrangian model was used, where a two-way coupling between fluid and particles is possible by considering the region inside a particle having a porosity of 0. Further on the velocity, acceleration and positions are totally dependent on the surrounding fluid flow conditions. Several other factors also influence particle motion. Particle-fibre interaction describes the situation when a particle collides with an obstacle, i.e. the fibres. The surface characteristics of the fibre will determine if the particle sticks onto it or is carried along with the fluid flow. Further on the collision between particles will also influence their path and has to be considered in the model. A detailed description of the overall programming and algorithm development can be taken from [2,5].

The next step was to combine the two parts, which are the fully optimized and tested FSI solver on the one hand, and the Lagrangian dirt particle solver on the other hand. This was possible due to the object oriented nature of OpenFOAM®. The

two modules were connected and tested on different geometries. At first, artificially designed fibres were used. The result is shown in the following figure (Figure 4.21).

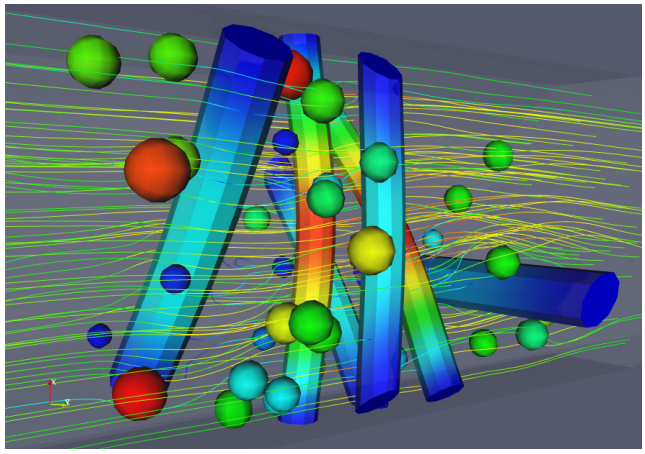


Figure 4.21: Combination of FSI and Lagrangian particle solver applied to an artificially designed geometry

Spherical dirt particles were injected into a fluid stream and transported through a geometry, which is deforming under the fluid flow forces. The next figure (Figure 4.22) shows an artificial geometry with different spherical dirt particles, which highly vary in diameter.

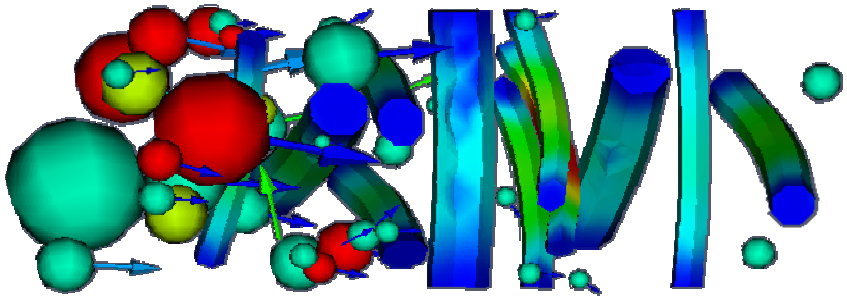


Figure 4.22: Different diameters of injected particles

The larger particles are obviously entangled in between the fibres and therefore block the space between them. This so called plugging effect occurs due to the approach of considering particles having a porosity of 0. It can be observed in the following figure (Figure Figure 4.23), where a view is taken along the flow of the fluid.

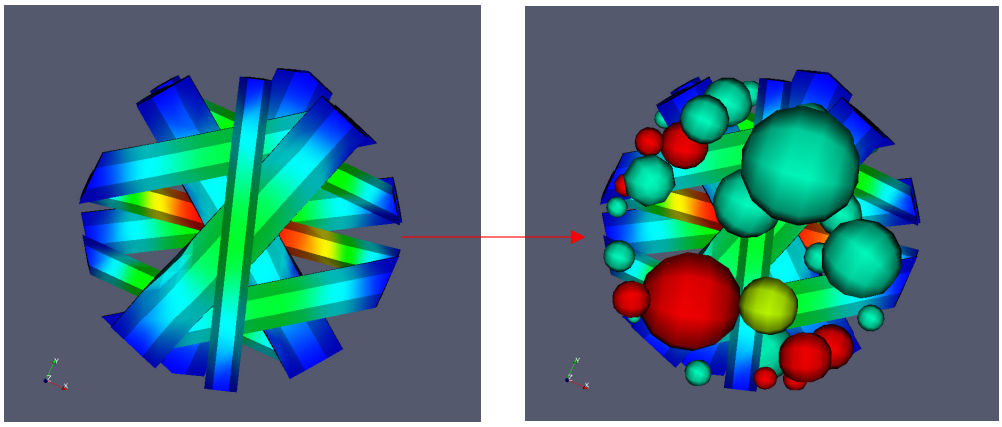


Figure 4.23: On the left hand side the empty filter is shown. On the right hand side particles of different sizes are blocking the pores.

More and more particles will accumulate on the fibres and hence increase the effective solid surface. It deviates and impedes the fluid flow. As a consequence the efficiency is increased, but the pressure drop also rises, which makes it necessary to change the filter in a very short time.

As a next step spherical particles were injected into the real, reconstructed fibre geometry of Fulda A43, which was already shown in Figure 4.15.

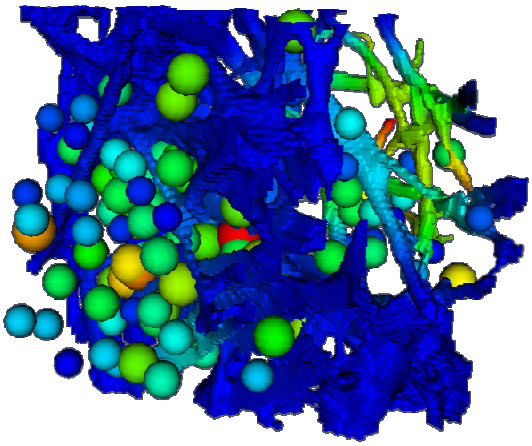


Figure 4.24: Real fibre geometry, solid displacement combined with injection of large particles [2,5]

More particles are entangled in the denser parts of filter fibres, whereas some particles are small enough to pass through the geometry. Especially the deformation of the solid part has an influence on the particle deposition behaviour. This can be observed in Figure 4.24 in the back region of the geometry where a lattice of thinner

fibres is present. The displacement of the solid makes it easier for particles to pass through the material at that position.

Further on, also the combination of fluid-structure interaction and the particle model applied on a longer section of A55 showed plausible results.

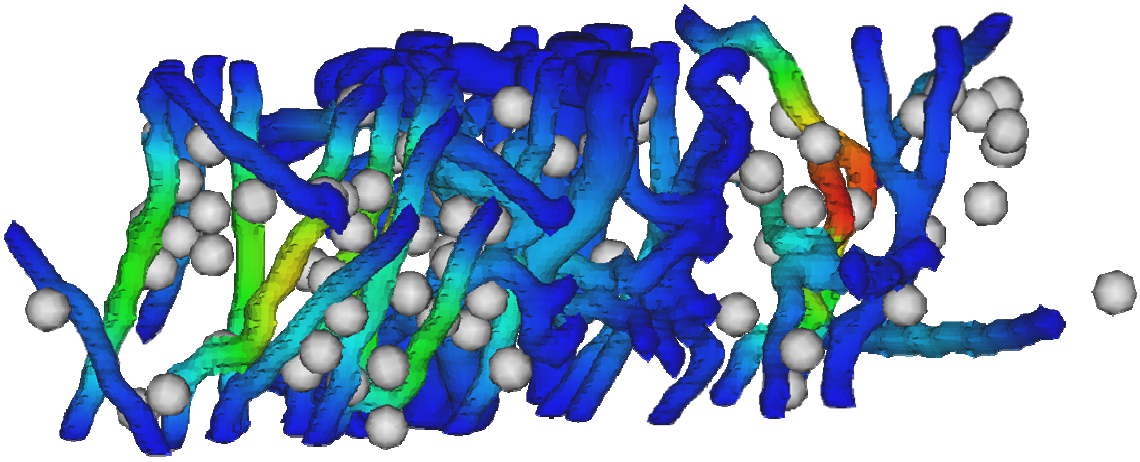


Figure 4.25: Deformation of Ahlstrom A55 in combination with injected particles

In reality dirt particles are generally not of a smooth spherical shape. There are many different shapes possible such as rods or disks. As the deposition behaviour of non-spherical particles highly differs from the spherical ones, the overall filter efficiency could be under- or over estimated, when using only the spherical model. This fact made it necessary to develop a non-spherical dirt particle model. All the necessary programming and underlying development is described in detail in [2-4]. Results can be seen in Figure 4.26.

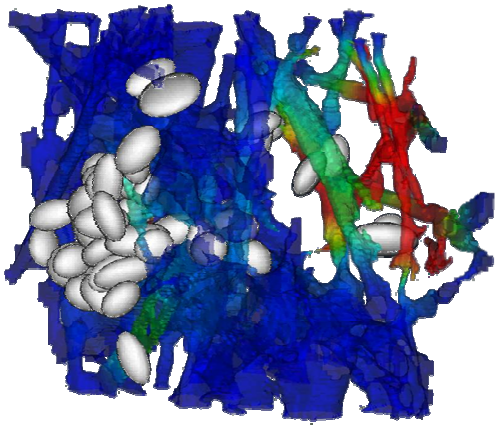


Figure 4.26: Dense cloud of non-spherical particles (30µm≤Dsph≤50µm) in a realistic deforming filter fibre geometry [2]

5 Influence of FSI on Overall Filter Efficiency

The permeability of a filter material is dependent on the pore size distribution. The compression of material due to structure deformation leads to a change of pore size. This fact highly influences the particle deposition behaviour and hence affects the overall permeability of the filter.

In order to prove this observation an artificially constructed geometry was set up. It consists of a selection of fibres with different cross section diameters, arranged in lattice form. The initial state is depicted on the left hand side of Figure 5.1 whereas the deformation results are shown on the right hand side.

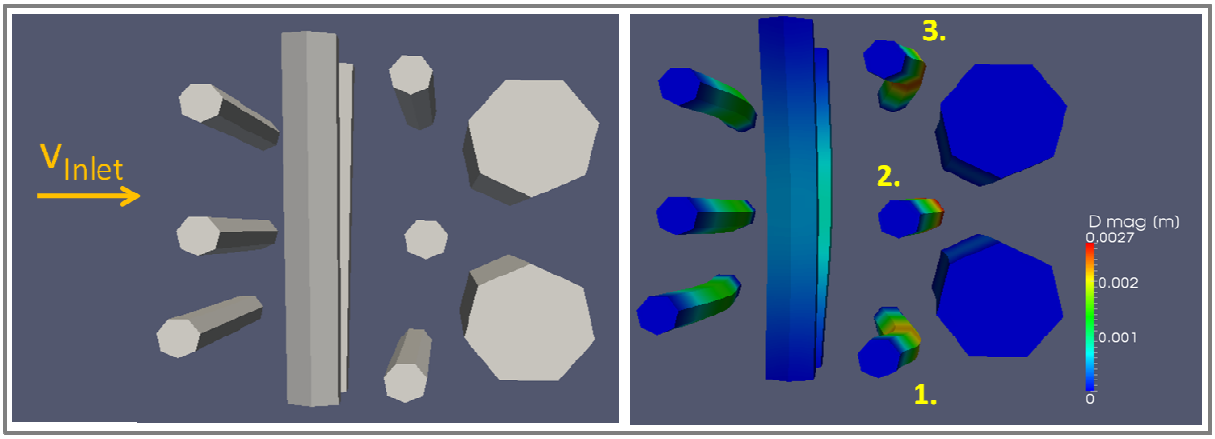


Figure 5.1: Initial state of the geometry used for permeability tests and the resulting deformation after applying the new FSI algorithm

At first the boundary condition of the fibre end points is set to fixedValue of zero in order to observe the pure deformation effect. By regarding the screen shot of the deformation in Figure 5.1 it is evident that especially the three small fibres in the middle of the geometry play a very important role. They are marked with the numbers 1,2 and 3. Due to deformation, fibre nr. 2 is blocking the space between the two thick, nearly immobile fibres, whereas fibres nr. 1 and 3 are obstructing the passage between the wall and the two thick fibres. It is obvious that a massive change of fibre arrangement takes place. It is displayed from two different views in Figure 5.2.

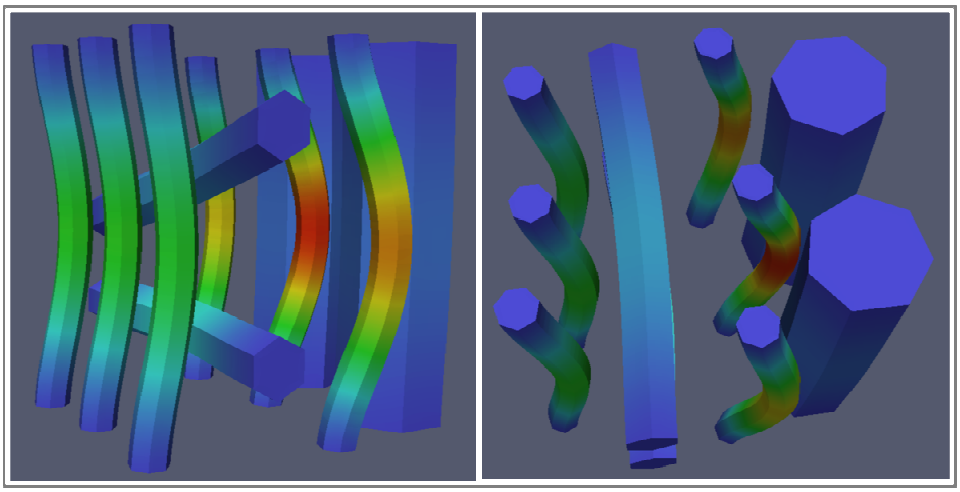


Figure 5.2: Deformation of test geometry shown from two different views

In this situation it will be hardly possible for injected particles to pass. Even smaller particles, which would go through in an undeformed geometry, will be entangled in the fibre structure. The effect is even more evident when applying the new spring boundary condition on the fibre end points. The initial state and the resulting deformation are shown in the following figure.

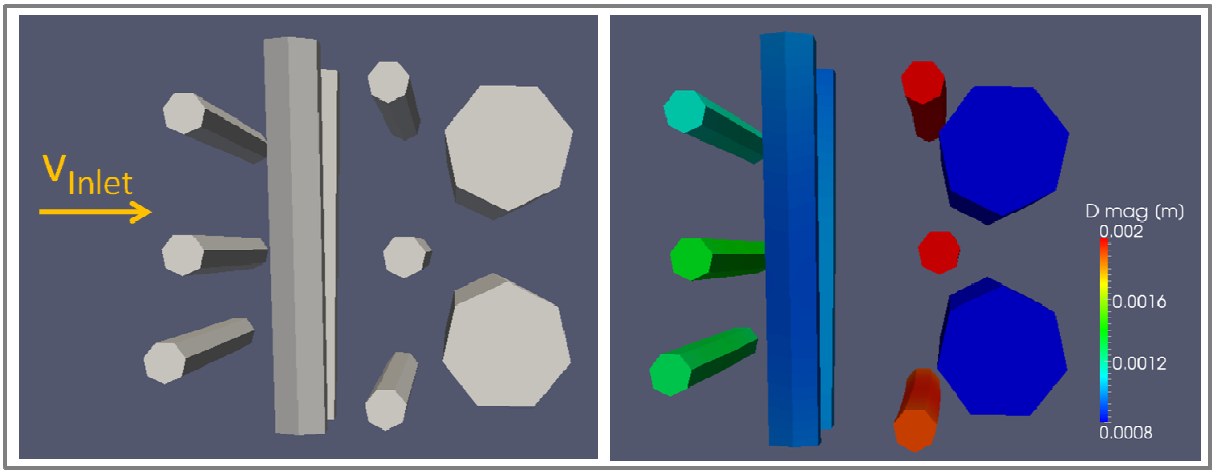


Figure 5.3: Comparison of initial state and deformed geometry; with spring boundary condition

As a consequence of applying the spring boundary condition, the full length of the fibre is placed into the free space. In the middle of the flow channel, the fluid forces are very high. Therefore fibre nr 2 has a higher displacement and hence is squeezed in between the two thick ones. This causes a high degree of blockage of the fluid flow. It increases the effective solid surface and consequently impedes the fluid flow.

Apparently injected particles would block the fluid stream sooner than in the undeformed geometry, where a lot more space is present in between the fibres. Additionally a higher amount of particles are entangled in the fibre structure. It would lead to miscalculations and maybe wrong predictions of the pressure drop over the filter material, if structure deformation was not taken into account.

The observation of geometry changes already proves the high influence of fluid-structure interaction on the overall permeability and particle deposition behaviour. It is obvious that softer material would lead to higher compression and increase the effect. Therefore objective and realistic filter efficiency can only be predicted when considering deformation of the fibres.

5.1 Experimental Setup

The influence of fluid-structure deformation on permeability and consequently on the pressure drop was also observed in experiments. At ICE Strömungsforschung GmbH an oil-fibre test facility to verify simulation results was designed. The facility has been planned and constructed according to proposals within ISO 4548-12 [37] concerning the set-up of fluid filter fibre multi-pass tests. Figure 5.4 shows a comparison of the underlying test-rig process plan, proposed by ISO 4548 and the derived, simplified version which corresponds to the oil-fibre test facility. The overall planning, measuring and evaluation was done by Boiger [2] and Reiss [38].

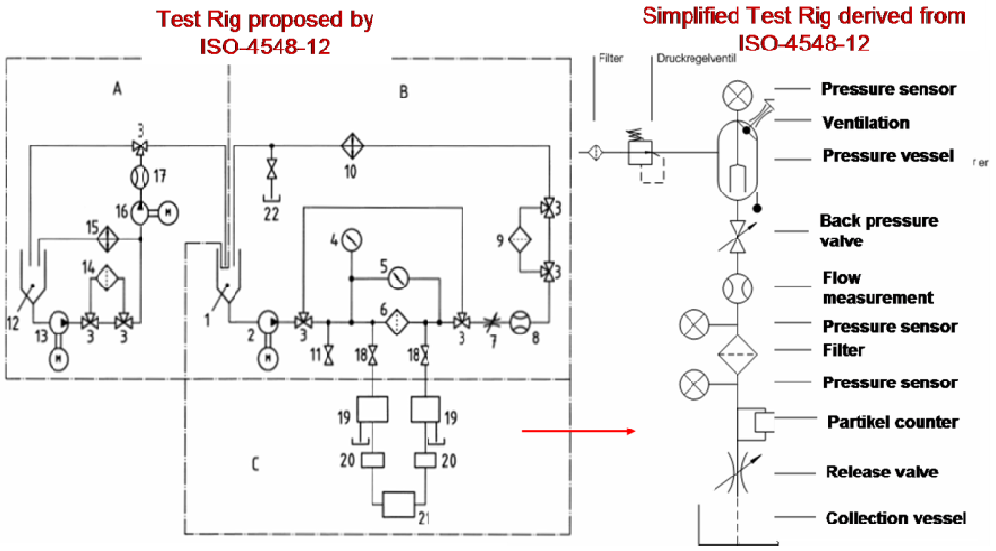


Figure 5.4: Process plan of designed oil-fibre test facility, proposed and simplified version [2]

Further on the planned facility was realized. The construction is shown in the following figure.

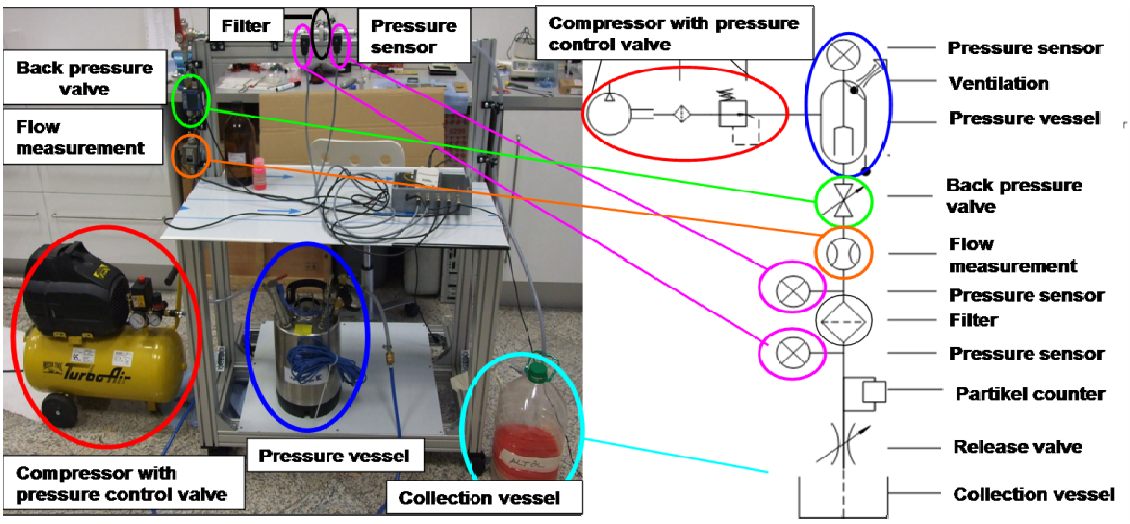


Figure 5.5: Constructed oil-fibre test facility [2]

The oil-fibre test facility is a fibre sample testing device designed to investigate the reaction of the material to an oil stream. While exposing any circular fibre sample of diameter D_S=2.5cm to controlled flow conditions of test-particle laden oil, the development of decisive process parameters, i.e. pressure drop Δp_S, volumetric flow rate v_S and cumulative oil volume V_S over the sample can be closely monitored, stored and electronically processed [2]. The overall measuring procedure and

principles are described in detail in [2,38]. The main task is to determine those parameters for several different filter materials and subsequently compare them to the simulation results.

5.2 Comparison of Results of Experiments and Simulation

5.2.1 Nonlinear Pressure Drop Behaviour

For laminar flow through a rigid, porous medium, Darcy's law for filtration is valid:

$$\Delta p_{darcy} = \frac{\eta \cdot \Delta x_f}{A_f \cdot k_f} \cdot \dot{V}_f \qquad 5.1$$

whereby Δp describes the pressure drop over the filter sample thickness Δx, η the dynamic viscosity, $\dot{V}_f$ the flow rate, k_f the permeability and A_f the cross section.

Therefore the flow rate is a linear function of the pressure drop for any specific filter material, given that its porosity remains constant. However, the conducted experiments with the oil-fibre test facility have shown a nonlinear relation of flow rate and pressure drop in particular for soft materials. The Reynolds numbers of the fluid flow ranges between 0.5 and 3. Therefore any turbulence effects can be definitely excluded for explaining this behaviour.

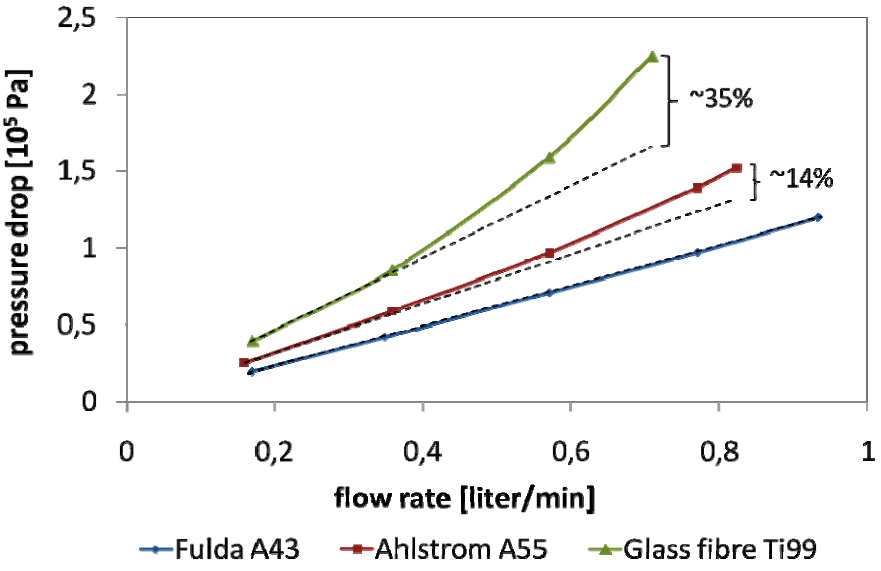


Figure 5.6: Results of experiments conducted on different fibre materials with the help of the oil-fibre test facility

It can be explained by the fact that during the flow of oil through the medium, pressure and shear forces deform each single fibre and compress the overall interwoven fibre structure. This compression reduces the pore diameter and leads to reduced permeability and consequently to higher pressure drops. This behaviour is directly related to the Young's modulus of the material.

The nature of the three filter media used for the experiments underline this explanation. Ti99 is a glass fibre fleece. It is very soft and deforms easily under fluid flow forces. In contrast, Fulda A43 has a very high Young's modulus. Additionally the fibre structure is different from Ahlstrom A55. This was already observed in chapter 4. Fulda A43 consists of more interconnected, thicker fibres and hence provides higher stability. On the contrary, Ahlstrom A55 consists of more unconnected, thin fibres, which are compressed more easily.

In order to investigate and verify the observed behaviour, the newly developed FSI solver was applied to the geometry shown in Figure 5.7.

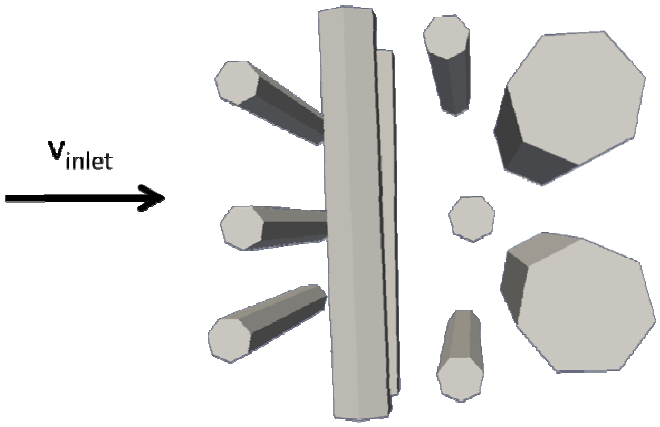


Figure 5.7: Test geometry for observing nonlinear pressure drop behaviour

This geometry turned out to be the ideal test geometry. It represents a realistic fibre maze in a reasonable way. The main advantage is that it requires much less time for one simulation run than a real reconstructed fibre geometry.

In order to reproduce experiments, simulation and experiments were conducted under the same conditions as they are:

Inlet velocity: u_f = 0.0067 m/s - 0.0204 m/s

Dynamic viscosity: $\eta_f = 1.93 \times 10^{-3}$ Pas

Fluid density: ρ_f = 800 kg/m³.

Inlet area: $A_{inlet} = 4{,}84 \times 10^{-4}$ m²

Two cases were set up, which only differ in the underlying Young's modulus. Each case was run with several different inlet velocities. There are no particles injected in order to investigate the pure influence of structure deformation on the pressure drop. The results of inlet flow rate versus pressure drop are plotted in the following figure (Figure 5.8).

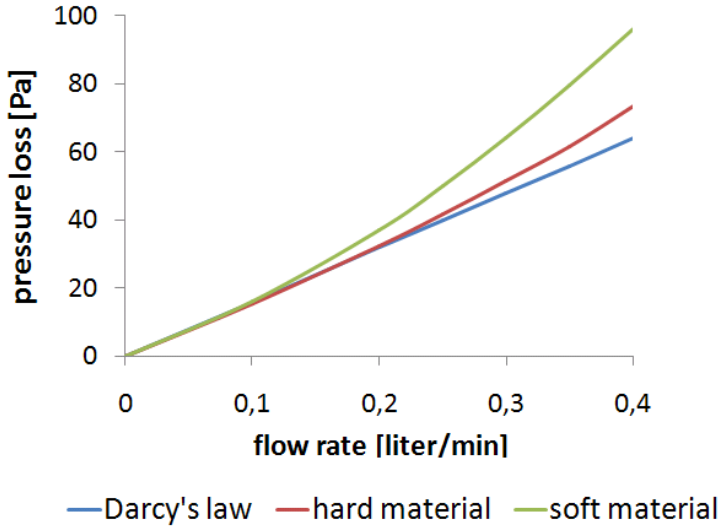


Figure 5.8: Simulation results of pressure drop in dependency of flow rate for different filter material

Apparently also in simulations this nonlinear behaviour can be observed. The blue line in Figure 5.8 represents the theoretical linear relationship due to Darcy's law. It is evident that the softer the material is the higher the nonlinearity becomes. This can be explained by its ability for higher compression at higher flow rates. The harder the material, the more this effect can be neglected and hence the pressure drop / flow rate relationship tends to be linear.

It can be seen that the pressure drop measured in experiments is much higher than the one resulting from the simulation. There are several reasons explaining it. First of all, the geometry used for simulation is artificially designed. It only consists of ten

characteristically located fibres, whereas the sample used for experiments is a whole interwoven maze of fibres, such as the cut-out shown in
Figure 4.2. Further on, to indicate the pure influence of the fluid-structure interaction, no particles where injected. Therefore the results shall be regarded as demonstration of the nonlinearity in general irrespective of qualitatively comparable numerical values.

The discovery of the non-linear behaviour has a great impact on the overall filtration theory. With this it is proven that it is necessary to consider fibre deformation in order to predict pressure drop with a much higher accuracy. It will be very significant for designing new filter materials in the future, because for filtration not only surface properties of fibres have to be investigated. It is even more vital to regard Young's modulus.

5.2.2 Filter Fibre Efficiency Curves

In this section, fibre deformation effects on resulting pressure drop without particle injection were investigated. As a next step, it was combined with particle laden flows in order to investigate the particle deposition behaviour and its dependency on fibre deformation. The filtration efficiency is a measure for the capability of a filter to remove particles of a certain size. This implies that a filter is challenged with particles of varying size and the percentage of captured particles of a certain size is monitored. A typical efficiency curve is shown in the following figure.

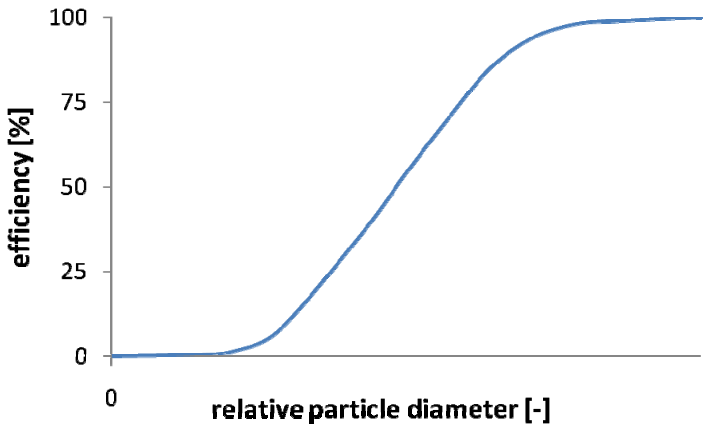


Figure 5.9: Typical filter efficiency curve of a filter material with Gauss distribution of theoretical dirt particle cloud

In order to investigate the influence of fibre material compression on the efficiency the same test geometry as used in chapter 5.2.1 was employed. To obtain characteristic efficiency curves the following procedure was carried out. For each particle size class i, a fixed amount of particles $n_{t,i}$ was injected during transient flow simulation within the fibre geometry. When kinetic particle energy in the system dropped to a steady state, the final result was evaluated by comparing the remaining particles $n_{r,i}$ to the total amount $n_{t,i}$:

$$E_{ip} = \frac{n_{r,i}}{n_{t,i}} \qquad 5.2$$

where E_{ip} is the particle deposition efficiency, $n_{r,i}$ the remaining particles and $n_{t,i}$ the total amount of particles injected of a specific particle class i.

Three main cases were set up with the previously used geometry, shown in Figure 5.7. For each case a different Young's modulus value was used and several particle classes were injected. In the following figure one exemplary case is shown, where a particle cloud is injected into the fluid stream through the chosen test geometry.

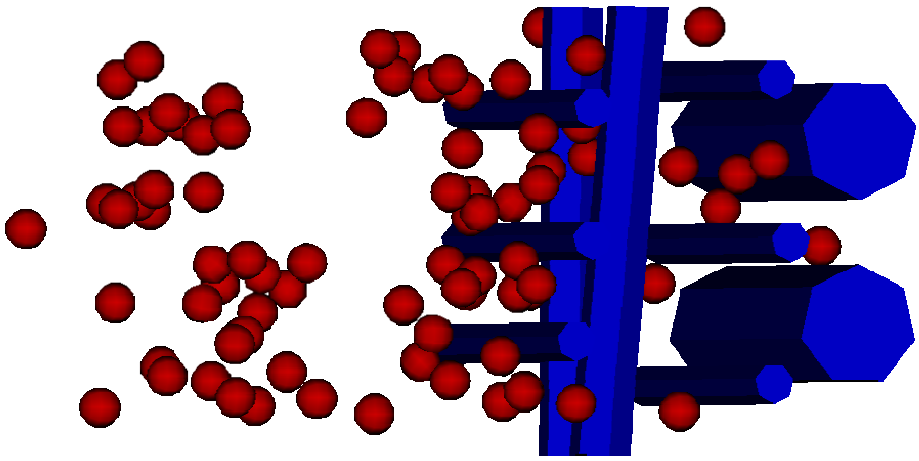


Figure 5.10: Particle cloud injected in the test geometry

For each particle class, the efficiency was measured. The fluid flow conditions, i.e. density and viscosity stayed the same as in the previous case and for each new one. The particle density was set to ρ_p=1500kg/m³. The results of the simulations show a very interesting behaviour. It is displayed in Figure 5.11.

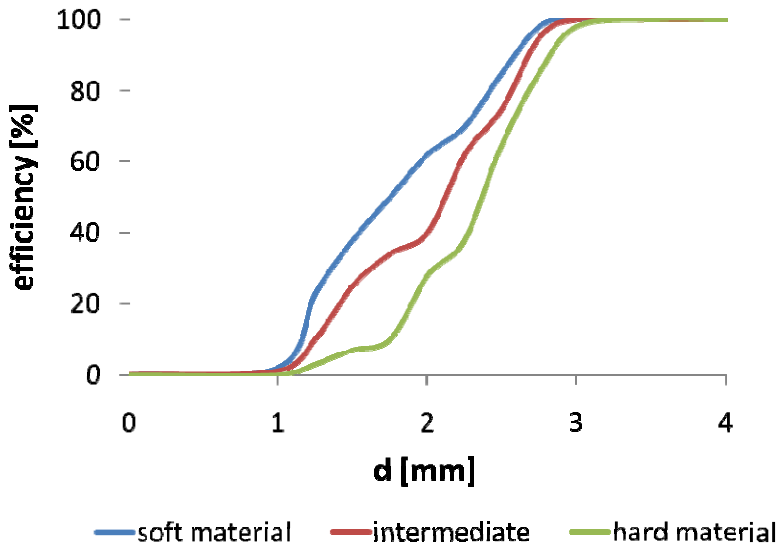


Figure 5.11: Filtration efficiency in % for three different materials and several particle classes

In this plot the nature of soft materials can be observed once again. Its higher compression rate strongly reduces the pore size. Higher portions of small particles are filtered out, which would have passed in case of stiffer material. Therefore the softer material increases the filter fibre efficiency more than a hard material would do. For observing the filtration behaviour of the geometry, a special area of interest was defined. It is displayed in the following figure with the help of a yellow square.

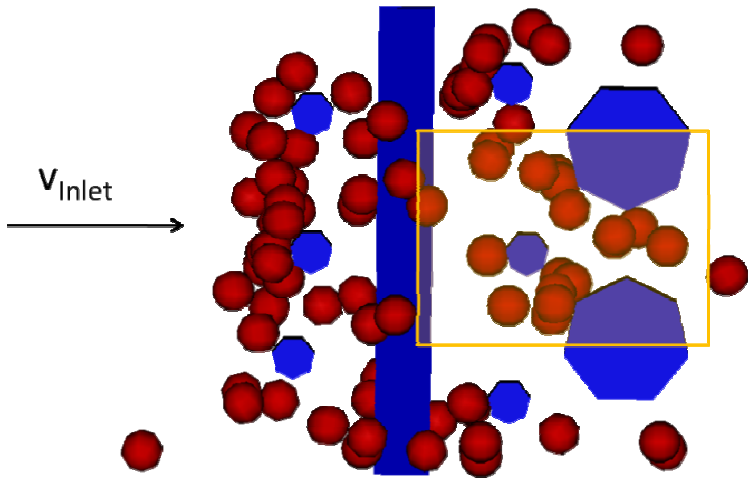


Figure 5.12: Special area of interest marked with a yellow square

This area was zoomed out for two different deformation grades. They are depicted in the following figure.

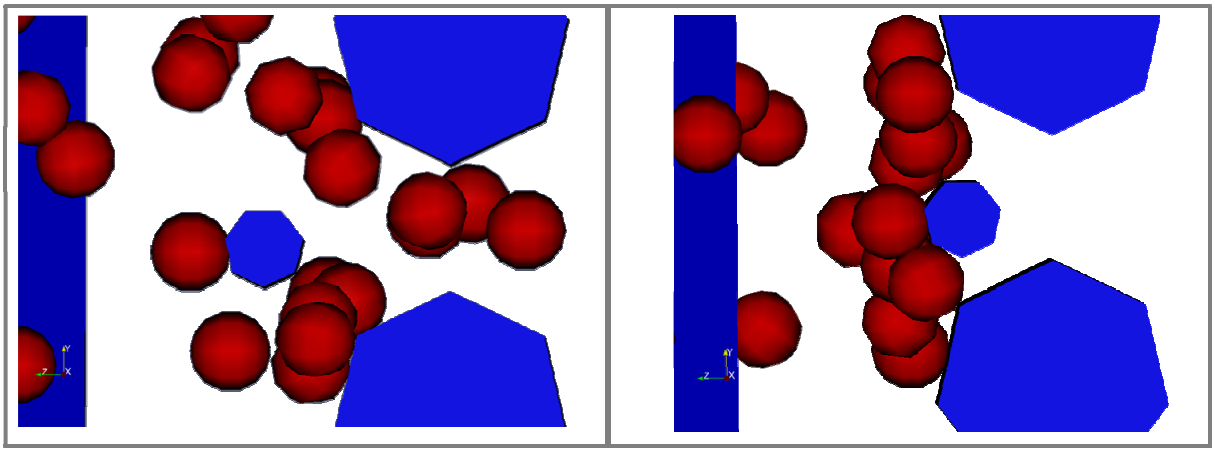


Figure 5.13: Particle deposition behaviour within the marked area for two different deformation grades

On the left hand side, the particle deposition behaviour in the undeformed state of the geometry can be observed. The particles are passing through the gap. On the right hand side, the same position is shown but in the deformed geometry. It can be seen, that due to the compression, the pore size shrinks. The particles cannot pass through and therefore are entangled in the filter fibres.

It is evident that the accuracy of pressure drop prediction and particle deposition behaviour in design of new filter material is highly dependent on the considered material properties. This also applies to the working conditions of the filter, e.g. exposure to humidity. At this point it opens a large field of investigation possibilities. The newly developed algorithm offers a very powerful tool.

6 The Graphical User Interface [2]

Figure 6.1 gives an overview of the entire workflow behind the overall simulation project. The whole procedure comprises three main phases:

Meshing and Pre-processing: CT-scan data is gathered from real-life fibre samples. A Digital Fibre Reconstruction utility digitalizes the CT-scan data, reconstructs a 3D image of the fibre structure and yields a structured grid mesh, suitable for OpenFOAM®. Then the user defines the physical inital and boundary conditions within the OpenFOAM® dictionaries.

Processing: The flow field is calculated either in combination with the fibre-deformation phenomena (based on the FSI solver) and/or in combination with depositing dirt particles (based on the presented particle solver above). At the end of each time step the results are streamed out as text files.

Post-processing: If necessary, the text file based data is processed by self-programmed Python® utilities [39] in order to extract information, such as filter fibre efficiency, particle penetration depth or kinetic particle energy. Compact, numerical results can thus be obtained. A conversion to the *VTK file format* [40] enables the full, 3D visualization of the simulation run using *ParaView* [16]. A Python® based visual filter has been programmed to enable the non-standard visualization of non-spherical particles.

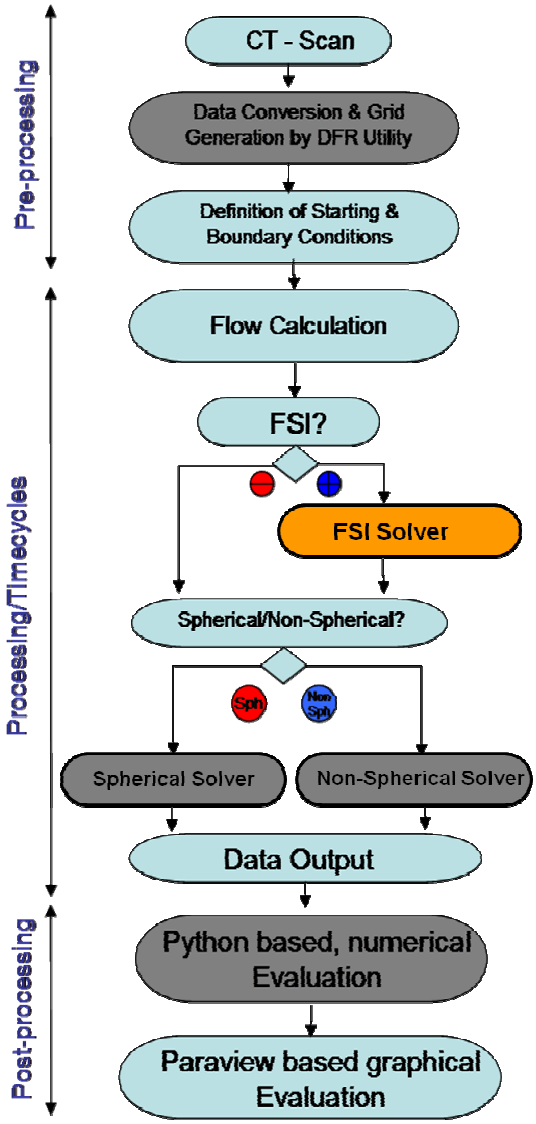


Figure 6.1: Activity diagram of total workflow, partitioned into pre-processing, processing and post-processing. Entities, coloured in grey have been created through intense development effort in the course of the development project but are not discussed in detail in this work. The entity coloured in orange was the centre of attention of this thesis.

The final version of the unified filtration solver will be equipped with an easy-to-use, graphical interface. This surface feature is supposed to direct the user through the entire process of fibre reconstruction, OpenFOAM® based FSI and/or dirt particle and deposition calculation as well as result evaluation. Furthermore it will enable the data transfer to an interlinked data-base where static and dynamic material properties can be stored. Figure 6.2 presents an overview of the main program modules e.g. the work packages and their relation within the workflow.

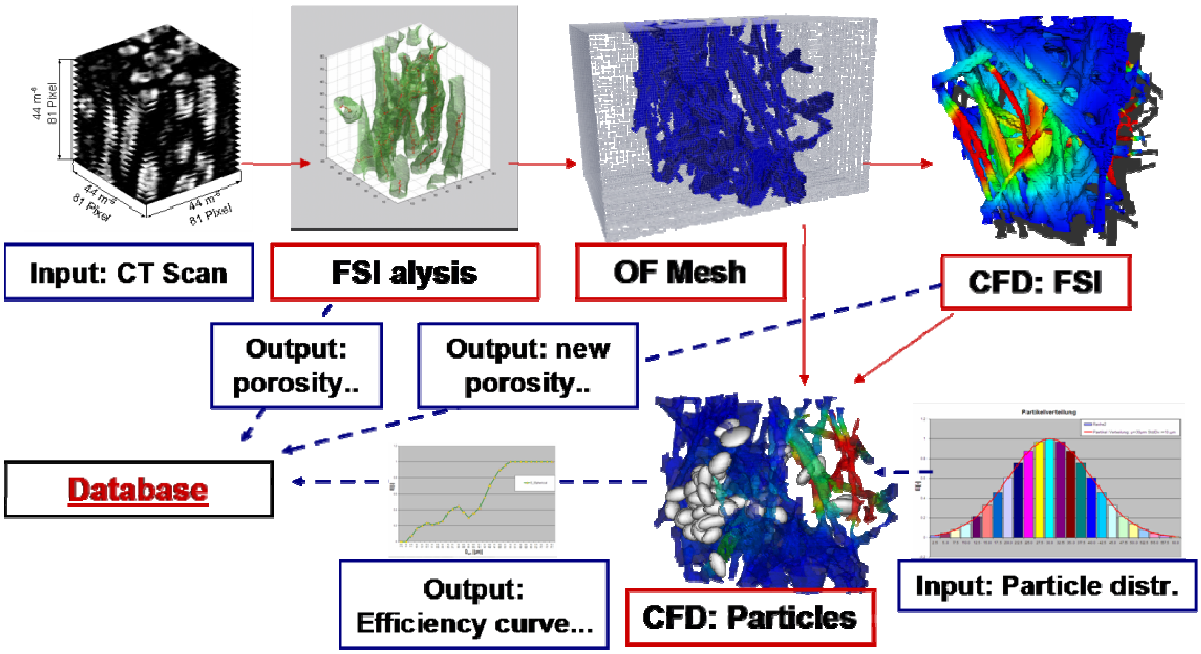


Figure 6.2: Sketch of the basic structure of the graphical user interface as it interlinks program modules according to the user workflow. Blue tags symbolize data in- or output. Dashed blue arrows symbolize data flow. Red tags symbolize program modules e.g. work packages and red arrows symbolize the user workflow. [2]

7 Conclusions and Outlook

The main task of this thesis was the creation of a fluid-structure interaction solver to investigate the change in the characteristics of an oil filter, with the deformation of the fibres due to the oil flow. The fluid conditions, for which the applied physics are valid, are highly viscous, relatively slow, incompressible and isothermal fluid flow. This flow leads to small deformation effects of the filter fibres, for which the algorithm was developed. In both regions, fluid and solid, different governing equations apply and hence have to be coupled. This coupling is handled by the pressure and shear force interaction terms of the Navier-Stokes equations for the fluid phase and Hooke's law for the solid phase. The underlying computational mesh is adjusted to the deformation of the solid region during each single time increment. The main advantage of this algorithm is that all of the steps listed above are realised by one single solver. Since the C++ based Open Source CFD platform OpenFOAM® features a strictly modular programming structure, the stand-alone development of a fluid-structure interaction solver was rendered possible and further on accomplished. At first, artificial, simple filter models were investigated. It turned out that new boundary conditions and a concept to model collision of fibres had to be developed. These features were programmed and also integrated into the new solver. In order to simulate realistic, from CT scans reconstructed fibre material, it was necessary to improve the computational side. Therefore the programming code was parallelised and a restart option was enabled. With that, the simulation of real fibres could be conducted within a reasonable time. Further on, the solver was thoroughly validated by plausibility checks and available experimental data. All in all this means that with this new FSI tool, considering the limitations cited above, simulation of filtration processes using any kind of hydraulic oil and filter fibre material are feasible.

Within the overall project, a Lagrangian, large particle model was developed [2-5] and combined with the FSI Solver. It enabled the possibility of simultaneous simulation of all relevant physical phenomena in one single finite volume solver.

A very important task was the verification of the significance of deformation of filter media to the overall filter characteristics. In experiments, a nonlinear behaviour of pressure drop in dependency of different inlet flow rates was observed. This effect was found to increase with decreasing Young's modulus. It was successfully verified in simulations with the newly developed FSI tool. Finally the particle deposition behaviour for filter materials with changing porosity was investigated. It was carried out with the help of the combined filtration tool, which comprises the FSI module and the Lagrangian particle model. With that it was successfully proven that the compression of the material highly influences the deposition efficiency. The overall solver will serve as a tool to simulate the filtration processes in every possible filter application. It can be utilized in the biotech industry, where metal meshes are used to grow bacteria. Another exemplary filtration application is the process of water purification, where dirt particles of different sizes are filtered out. In this case the solver will have to be extended with an additional force, which is the particle impact force acting onto the fibres. The new tool, developed in this thesis, serves well to investigate both phenomena.

Furthermore, as every other FSI solver suitable for small deformations and incompressible media, it can be used to simulate various other fluid-structure interaction phenomena. An important application arises in medical science, where the simulation of blood transport through an aorta would give deep insights into the behaviour of its vessel walls. Substantial development work would be required to extend the FSI solver capabilities towards simulating large material deformations or the handling of compressible fluids such as air. However, the benefit would be that for example the motion of parachutes could be studied in detail.

The final aim of the project was to supply a virtual design circle for development and optimization of new high performance filter materials without a need for performing time consuming expensive experimental work.

8 Nomenclature

a	arbitrary quantity
A	area [m²]
A_f	cross section [m²]
d	particle diameter [m]
D	displacement of solid [m]
D_f	diffusion coefficient
E	Young's modulus [N/m²]
E(p)	efficiency for multiple processor run
E_{iP}	particle deposition efficiency []
F_p	pressure force [N/m²]
F_s	spring force [N]
F_τ	traction force [N/m²]
g	gravity [m/s²]
G	shear modulus [N/m²]
k	spring constant [N/m]
k_f	permeability [m²]
L	length [m]
n	number of particles []
$\underline{n}$	normal vector []
p	pressure [N/m²]
Δp	pressure drop [N/m²]
S(p)	speed-up for multiple processor run
$S\phi$	source term
t	time [s]
T(1)	time for single processor run
T(p)	time for multiple processor run
u	velocity [m/s]
V	volume [m³]
$\dot{V}_f$	flow rate [m³/s]
Δx	sliding length distance [m]

Δx_f filter sample thickness [m]

x_i coordinate directions [m]

X_b body force [N/m³]

ε strain [m/m]

Γ diffusion coefficient

η dynamic viscosity [Pas]

λ Lamè coefficient [N/m²]

μ Lamè coefficient [N/m²]

ν Poisson number []

Φ transport property

ρ density [kg/m³]

ρ_f fluid density [kg/m³]

ρ_P particle density [kg/m³]

σ normal stress [N/m²]

τ shear stress [N/m²]

9 List of Figures

10 References

[1] M.Mataln; G.Boiger; W.Brandstätter; and B.Gschaider: Simulation of Particle Filtration Processes in Deformable Media, Part 1: Fluid-Structure Interaction, ICE Strömungsforschung GmbH, Montanuniversitaet Leoben, Int.Journal of Multiphysics, Vol. 2 (No.2), pp. 179 - 189, 2008

[2] G.Boiger: Development and verification of a novel lagrangian, (non-)spherical dirt particle and deposition model to simulation fluid filtration processes using OpenFOAM, Ph.D. Thesis, ICE Strömungsforschung, 2009

[3] G.Boiger; M.Mataln; W.Brandstätter; and (2009): Simulation of Particle Filtration Processes in Deformable Media, Part 3.1: Basic concepts and particle-fluid force implementation of a non-spherical dirt particle solver, ICE Strömungsforschung GmbH, Montanuniversitaet Leoben, Austria, Int.Journal of Multiphysics, 2010

[4] G.Boiger; M.Mataln; W.Brandstätter; and (2009): Simulation of Particle Filtration Processes in Deformable Media, Part 3.2: Interaction modelling and solver verification of a non-spherical dirt particle solver, ICE Strömungsforschung GmbH, Montanuniversitaet Leoben, Austria, Int.Journal of Multiphysics, 2010

[5] G.Boiger; M.Mataln; W.Brandstätter; and B.Gschaider: Simulation of Particle Filtration Processes in Deformable Media, Part 2: Large Particle Modelling, ICE Strömungsforschung GmbH, Montanuniversitaet Leoben, Int.Journal of Multiphysics, Vol. 2 (No.2), pp. 191 - 206, 2008

[6] www.mathworks.com, Matlab and Simulink for Technical Computing: 2010

[7] I.M.Spiliotis and B.G.Mertzios: A Fast Parallel Skeleton Algorithm on Block Represented Binary Images, Democritus University of Thrace, Greece, Elektrik, Vol.1, pp. 161 - 173, 1997

[8] http://www.opencfd.co.uk/openfoam, OpenFOAM, 2010

[9] openfoam-extend.svn.sourceforge.net, OpenFoam Extensions, 2010

[10] OpenCFD: OpenFOAM User guide, Version 1.6, OpenCFD Limited, 9 Albert Road, Caversham, Reading, Berkshire RG4 7AN, United Kingdom, 2009

[11] Bjarne Stroustrup: The C++ Programming Language, Special Edition, 2000

[12] William H.Press; Saul A.Teukolsky; William T.Vetterling; and Brian P.Flannery: Numerical Recipies in C, Special Edition, 1992

[13] H.Schade; E.Kunz; and F.Kameier: Strömunglehre, ISBN: 978-3-11-018972-8, 20-7-2007

[14] http://www.gnu.org/licenses/gpl.html, GNU General Public License: 2010

[15] Fluent Inc.: Fluent 6.2 User Guide, Centerra Resource Park, 10 Cavendish Court, Lebanon, NH 03766, USA, 2005

[16] J.Ahrens; B.Geveci; and Ch.Law: Paraview: An End-User Tool for Large Data Visualization, Handbook, Edited by C.D.Hansen and C.R.Johnson, Elsevier, 2005

[17] C.Lanczos: The variational principles of mechanics, 4th edition, ISBN: 0-486-65067-7, 1970

[18] K.-J.Bathe: Finite Element Procedures, Englewood Cliffs, New Jersey, ISBN-10: 0133014584, 1995

[19] K.H.Huebner; D.L.Dewhirst; D.E.Smith; and T.G.Byrom: The Finite Element Method for Engineers, 4th edition, ISBN: 0-471-37078-9, 2001

[20] H.G.Weller; G.Tabor; H.Jasak; and C.Fureby: A tensorial approach to computational continuum mechanics using object-oriented techniques, Computers in Physics, Vol.12, No. 6, pp. 620 - 631, 1998

[21] W.Bohl and W.Elmendorf: Technische Stroemungslehre, Vogel Fachbuch, Kamprath Reihe, Auflage 13, 2005

[22] O.Univ.-Prof.Dr.F.D.Fischer: Fluid Mechanics, lectures notes, Montanuniversitaet Leoben, Austria, 2002

[23] W.Brandstätter: Flow and Combustion Modelling, lecture notes, Montanuniversitaet Leoben, Austria, 2005

[24] G.Polya and J.Kilpatrick: The Stanford Mathematics Problem Book, ISBN-10:0486469247, 2009

[25] W.Brandstätter: Advanced Simulation Techniques, lecture notes, Montanuniversitaet Leoben, Austria, 2004

[26] H.Jasak and H.G.Weller: Application of the finite volume method and unstructured meshes to linear elasticity, Int.Journal for numerical methods in engineering, 48, pp. 267 - 287, 1999

[27] H.Jasak: Error analysis and estimation for finite volume method with application to fluid flows, Ph.D. thesis, Imperial College, University of London, 1996

[28] Dipl.-Math.Clemens Helf: Eine Finite-Volumen-Methode in allgemeinen Zellen fuer die Euler-Gleichungen mit integrierter, selbst-adaptiver Gittergenerierung . 2002

[29] OpenFOAM programmer's guide, Version 1.4.1, OpenCFD Limited, 9 Albert Road, Caversham, Reading, Berkshire RG4 7AN, United Kingdom, 2007

[30] OpenCFD: OpenFOAM programmer's guide, Version 1.6, OpenCFD Limited, 9 Albert Road, Caversham, Reading, Berkshire RG4 7AN, United Kingdom, 2009

[31] www.mathworks.com, Matlab and Simulink for Technical Computing, 2010

[32] www.ice-sf.at, ICE Strömungsforschung GmbH, Hauptplatz 13, 8700 Leoben, Austria, 2010

[33] www.open-mpi.org, Open MPI: Open Source High Performance Computing: 2010

[34] S.Balay; W.D.Gropp; L.C.McInnes; and B.F.Smith: Efficient management of parallelism in object oriented numerical software libraries, Modern Software Tools in Scientific Computing, Birkhäuser Press, pp. 163 - 202, 1997

[35] A Ecer; J Hauser; P Leca; and J Periaux (eds): Parallel Computational Fluid Dynamics. New Trends and Advances, N-H Elsevier: Amsterdam, 1995

[36] P.Amestoy; A.Guermouche; J.L'Excellent; and S.Pralet: Hybrid scheduling for the parallel solution of linear systems, Parallel Computing 32 (2), pp. 136 - 156, 2006

[37] ISO 4548-12:2000-02: Methods of test for full-flow lubricating oil filters for internal combustion engines - Part12: Filtration efficiency using particle counting, and contaminant retention capacity, 2000

[38] G.Reiss: Improvement and presentation of two novel, innovative filter testing methods, Master thesis, ICE Strömungsforschung GbmH, 2010

[39] M.Lutz: Python kurz&gut, Köln: O'Reilly Verlag 2005. ISBN: 3-89721-511-X, 2005

[40] W.Schroeder; K.Martin; and B.Lorensen: The Visualization Toolkit An Object-Oriented Approach To 3D Graphics. 4th Edition., Kitware, Inc. Publishers. ISBN 1-930934-19-X, 2006