ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ-ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΥΔΡΑΥΛΙΚΗ ΔΙΑΠΕΡΑΤΟΤΗΤΑ ΙΝΩΔΩΝ ΣΥΣΤΗΜΑΤΩΝ ΜΕ ΠΑΡΑΛΛΗΛΗ ΑΠΟΡΡΟΦΗΣΗ ΡΕΥΣΤΟΥ

ΦΩΤΙΑΔΗΣ ΦΩΤΙΟΣ

Επιβλέπων Καθηγητής

Δρ ΠΑΠΑΘΑΝΑΣΙΟΥ ΑΘΑΝΑΣΙΟΣ

ΒΟΛΟΣ 2017

Εξεταστική Επιτροπή

Πρώτος εξεταστής (Επιβλέπων)	Δρ. Παπαθανασίου Αθανάσιος, Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο Θεσσαλίας.
Δεύτερος εξεταστής	Δρ. Τσιακάρας Παναγιώτης, Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο Θεσσαλίας
Τρίτος εξεταστής	Δρ. Κερμανίδης Αλέξης, Επίκουρος Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο Θεσσαλίας

Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή της εργασίας κ. Αθανάσιο Παπαθανασίου, καθηγητή του τμήματος Μηχανολόγων Μηχανικών του Πανεπιστημίου Θεσσαλίας, για τη βοήθεια, τις γνώσεις, την παρότρυνση αλλά και για τις πολύτιμες συμβουλές του κατά την εκπόνηση της συγκεκριμένης διπλωματικής εργασίας.

Επιπλέον, για τη βοήθεια σε καίρια σημεία, την υποστήριξη και την παροχή πολύτιμων πληροφοριών θα ήθελα να ευχαριστήσω τον Ανδρέα Τσιαντή, ερευνητικό μέλος του τμήματος.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια μου για όλα τα εφόδια που μου έχει προσφέρει και για τη συνεχή και αμέριστη υποστήριξη τους.

Περίληψη

Η διπλωματική εργασία που ακολουθεί, αναφέρεται στην διερεύνηση του προβλήματος ροής ρευστού διαμέσου ενός σύνθετου υλικού. Το υλικό αποτελείται από πορώδεις δέσμες κυκλικής διατομής, οι οποίες είναι τοποθετημένες σε τετραγωνική διάταξη μέσα στην μάζα. Κάθε δέσμη απαρτίζεται από ίνες απείρου μήκους παρατεταγμένες σε εξαγωνική διάταξη. Ανάμεσα στις ίνες υπάρχει κενό, το οποίο κατά την διαδικασία του πειράματος καλύπτεται από ρευστό. Προκείμενου να αποφευχθεί υπολογιστικό φορτίο, εκμεταλλευόμενοι την συμμετρία του προβλήματος, απλοποιούμε τη γεωμετρία και περιοριζόμαστε στην μελέτη ενός μοναδιαίου κελιού δύο διαστάσεων. Στο κέντρο του υπάρχει μία δέσμη από ίνες τοποθετημένες σε εξαγωνική διάταξη. Για να φέρουμε εις πέρας τους απαραίτητους υπολογισμούς και τις προσομοιώσεις, θα χρησιμοποιηθεί το υπολογιστικό εργαλείο OpenFOAM. Το χαρακτηριστικό της ροής που ερευνάται σε όλη την έκταση της εργασίας είναι η υδραυλική διαπερατότητα. Πιο συγκεκριμένα, θα εξεταστεί υπολογιστικά η περίπτωση που το υγρό ρέει ανάμεσα από τις ίνες, χωρίς να απορροφάται. Στη συνέχεια, χρησιμοποιούνται εμπειρικοί τύποι ώστε να επαληθεύουν οι πειραματικοί υπολογισμοί. Αυτή η διαδικασία στοχεύει στη βέλτιστη δυνατή εξοικείωση με τον OpenFOAM και τα υπόλοιπα προγράμματα που χρησιμοποιούνται. Επιπλέον, εξετάζεται η περίπτωση ροής μέσα από το μοναδιαίο κελί με παράλληλη απορρόφηση των ινών. Σε αυτό το ενδεχόμενο παρατηρείται διόγκωση των ινών με αποτέλεσμα την αύξηση του στερεού κλάσματος όγκου μέσα στην δέσμη. Πρέπει να σημειωθεί ότι η απορρόφηση σταματά μόλις οι ίνες έρθουν σε επαφή μεταξύ τους, καθώς τότε είναι πρακτικά αδύνατη η ροή ρευστού ανάμεσα τους. Επομένως, με την συνεχή απορρόφηση το πορώδες της δέσμης μειώνεται, γεγονός που φαίνεται να επηρεάζει την διαπερατότητα. Τελικώς, μέσα από γραφήματα και διαγράμματα, προσπαθήσουμε να χαρακτηρίσουμε και να προσδιορίσουμε την συμπεριφορά της ροής και της διαπερατότητας σε σχέση με τον χρόνο απορρόφησης.

Abstract.

The following thesis is related to a computational analysis of viscous flow through a fibrous media. The latter are usually idealized as made up of square arrays of bundles in which individual fibers are packed I hexagonal arrangements. This dual porosity problem is characterized by different inter- and intra-tow porosities. In such arrangements a bi dimensional unit cell can be identified. Flow analysis through a unit cell provides the ability to compute the hydraulic permeability. To complete the necessary calculations and simulations, OpenFOAM will be used. In the first part the flow through rigid porous media, with no absorption, is examined. An empirical correlation will be used to verify the numerical experiments. This process aims at optimizing familiarity with OpenFOAM and rest of the programs that are used. In the second and last part the case flow through swelling porous media is studied. Swelling consists the result of fluid absorption by the fibers and seems to affect porosity and permeability. Through a number of simulations we try to determine behaviors of flow and permeability with respect to the time of absorption.

Table of Contents

Table of Figures

Chapter 1

1.1Porosity

Not all tangible objects are solid. Many of them contain void spaces. The percentage of the object that is empty of material defined as porosity. The visual symbol of porosity isφ and described mathematically by the following equation (Fetter, 2000).

$$\varphi = \frac{V_{void}}{V_{total}} \tag{1.1}$$

where

φ: porosity,[percentage]

V_{void} : Volume of empty space of material, $[m^3]$

V_{total} : Unit volume of material (solid and void), $[m^3]$

We can also calculate the percentage of the object that is solid material. It is defined as volume fraction, symbolized withΦ, and described mathematically by the equation

$$\Phi = \frac{V_s}{V_{total}} \tag{1.2}$$

Φ : Volume fraction, [percentage]

V_s: Volume of solid material $[m^3]$

The unit volume of material constitutes the sum of the void and solid space of the material:

$$V_{total} = V_{void} + V_s \tag{1.3}$$

Using the above equations we infer the following formula, which connects porosity and volumetric flow rate

$$\Phi = 1 - \varphi \tag{1.4}$$

1.2 Permeability

We have seen that some material contain a percentage of empty space and thus exhibit porosity. The quantity (amount) of the moving liquid or gas is dependent upon the porosity. For example, when the pores are too small the liquid flows through the medium with difficulty(Fetter, 2000). The ability of a porous media to allow fluids or gasses pass through is measured by coefficient called Permeability. The S.I. unit is m^2. The study of permeability and its characteristics can contribute to studies relating to disciplines such as biology, industrial production or geology. Permeability depends on the materials that are interacting such as a fluid that cannot pass through a material while a gas can easily. For example, carbonate in a soft drink can diffuse, even very slowly, out of the plastic bottles, which is not the case with glass or aluminum bottles. By the same token, various fluids with different characteristics may have different permeability to the same obstacle. In the case of a porous medium we study, permeability is affected by porosity (Papathanasiou, 2001). Porous materials that have the ability to absorb a percentage of the flowing fluid swell. The result is that the material's porosity changes over time. Such swelling porous media will be studied, in order to observe how permeability behaves.

1.3 Swelling

The majority of natural fiber materials undergo swelling when a wetting liquid comes in contact with the fiber. The crystallinity plays a major role in the swelling process. The amount of liquid retained by swollen fiber bundles varies from 5% to 100% of dry weight fibers. The retained liquid changes the porosity, the radius and the permeability of fibrous porous media.

The wetting of the fibers set the start of swelling and ends when the fibers reach an equilibrium state. In conclusion swelling is a time related process which means that the porosity, capillary pore radius and permeability are functions of time. So the governing equations we are going to use should be modified to include the swelling effects, since the parameters of the fibrous media change due to swelling (Masoodi & Pillai, 2010).

Swelling constitutes a special phenomenon that we encounter in our everyday life in wicking applications and products such as diapers, wipes and commercial wicks. Even inflammation is often characterized by swelling. When a finger start bleeding, for example, biochemical processes release proteins as "emergency signals" that bring in your body's immune cells, nutrients and hormones to fix the problem. Arteries swell, blood flow increases, and capillaries become more permeable so that, nutrients, white blood cells and hormones can move into the gap between cells. Hormones create blood clots to repair the damaged tissue and remove them when healing is completed. Swelling happens because fluid accompanies the hormones, nutrients and (white) blood cells. The fluid diffuses into the area and causes the swelling that can be the reason of increased pressure(Szalay, 2015).

1.4Study of Cases

The cases we study is devoted to a computational analysis of viscous flow of Newtonian liquid across unidirectional (one way) arrays of infinite fiber bundles. In the analysis of flow through fibrous media, the latter are usually idealized as made up of cylinders arranged in space in a defined periodic order. We consider systems made of infinitely long cylindrical bundles arranged in a square packing and with their axes perpendicular to the direction of flow. Each bundle consists of a number N of solid individual cylindrical filaments of radius R_f. Permeability is independent of the number of fibers when it is large (N>300) and depends exclusively on the porosity. These filaments are arranged in hexagonal packing within the bundle (Fig.1). Therefore such systems are characterized by an inter-tow (gaps between bundles) and intra-tows (void space within a bundle) porosity. Inter-tow and intra-tows porosities symbolized with φ_i and φ_t respectively. At low values of φ_t practically no fluid passes through the bundle. In such arrangements, a bi-dimensional unit cell(Martins, Laranjeira, Braga, & Mata, 2009) can be identified (Fig.2). In such a case the flow of liquid is characterized by the Darcy's law and the single phase continuity equation.

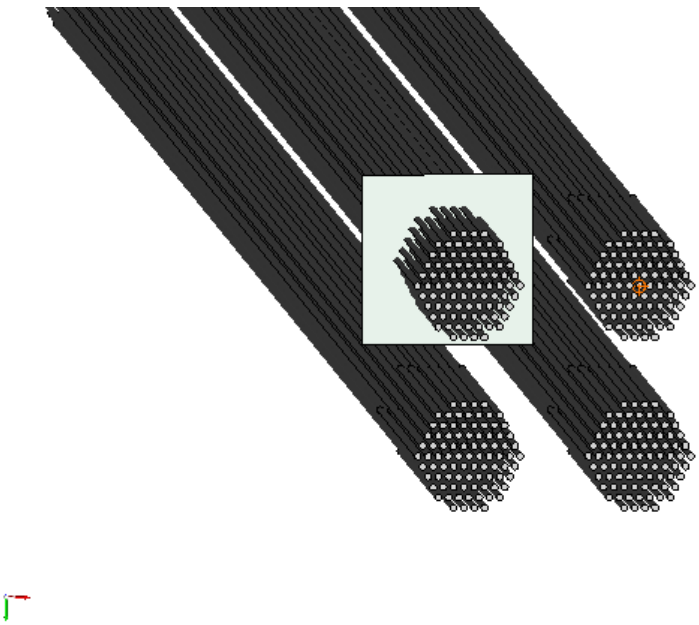


Figure 1: Infinite bundles in square packing, filled with hexagonal arranged fibers. The square of two dimensions (gray color) formed is a unit cell. (Visualized using Salome software)

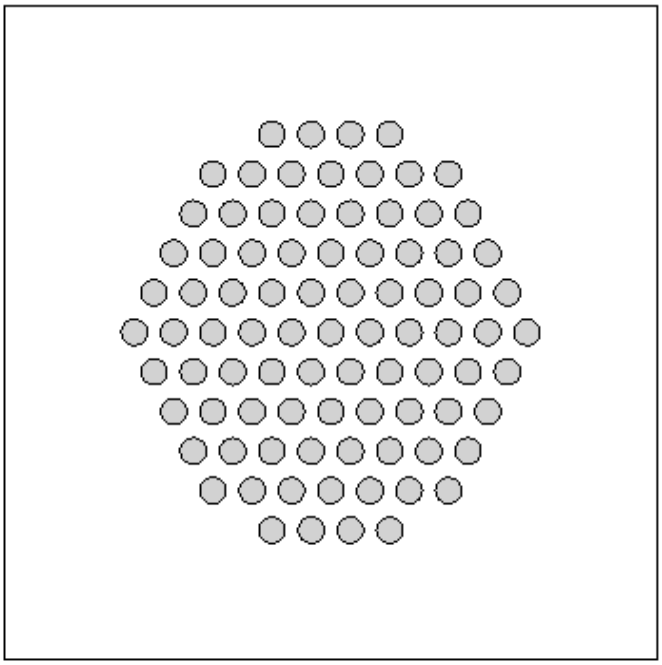


Figure 2: A two-dimensional unit cell (Visualized using Salome software)

1.4.1 Flow across a rigid porous media

In this case we study a flow through porous media that does not affect the fiber diameters. We ignore liquid absorption and swelling during the mold filling process. Furthermore the permeability remains the same (constant) in time. Therefore applies the following flow balance equation:

$$Q_{inlet} = Q_{outlet} \tag{1.5}$$

Where

Q_{inlet}: Volume flowrate of liquid at input$[m^3/s]$

Q_{outlet}: Volume flowrate of liquid at output$[m^3/s]$

1.4.2 Flow across a swelling porous media.

Applying the same approach with the exception that the fibrous media absorbs a fix percentage of the flowing liquid. The absorption leads to a swelling media during the mold filling process and as a result a continuous increase of the fiber diameters in time. So the following flow balance equation applies:

$$Q_{inlet} = Q_{ab} + Q_{outlet} \tag{1.6}$$

Where

Q_{ab}: Volume flowrate of liquid that the porous medium absorbs $[m^3/s]$

1.5 Darcy's law equation

A single-phase flow of a Newtonian liquid in porous media can be described by Darcy's law (Darcy,1856 as cited in (Vafai, 2015)) and the continuity equation. (Assume that we have a flow in the direction of the length of the porous medium and also this direction represents the x direction of a Cartesian coordinate system)

Darcy's law:

$$\mathrm{U} = \frac{\mathrm{K}}{\mu}\frac{dP}{\mathrm{dx}} \tag{1.7}$$

where

U =velocity of liquid through the porous medium $[m/s]$

A = cross-sectional are of sample $[m^2]$

K = permeability $[m^2]$

μ = viscosity $[Pa \cdot s]$

$\frac{dP}{\mathrm{dx}}$ = derivative of pressure with respect to x direction

Darcy's law was established under certain circumstances:

- Laminar flow
- Steady-state flow conditions
- Homogenous, isotherm and incompressible fluid
- Neglecting kinetic energy

In order to get this liquid to flow we need to apply pressure difference across this x direction (in order to get it to flow in x direction). Actually we are setting pressure boundary conditions.

$$P = P_1 \; at \; x = 0$$

And

$$P = P_2 \; at \; x = L$$

Darcy's equation can be written as:

$$\frac{U\mu}{AK}\mathrm{dx} = \mathrm{dP} \qquad (1.8)$$

By the integration of the above equation, we get:

$$\int_{x=0}^{x=L} \frac{U\mu}{AK}\mathrm{dx} = \int_{P_1}^{P_2} \mathrm{dP} \qquad (1.9)$$

In order to continue the solution of the above equation we must deepen the coefficient Q with the help of the equation of continuity, which is modeled on the previous assumptions.

$$\nabla(U\rho) = 0 \qquad (1.10)$$

Where

$\rho = liquid\ density[kg/m^3]$

$U = fluid\ velocity\ [m/s]$

The density is fixed because the fluid is incompressible so the final form of the continuity equation is:

$$\nabla(U) = 0 \qquad (1.11)$$

Zero divergence means that the amount going into a region equals the amount coming out. It is further concluded that the volumetric flow rate may be outside the equation's (1.9) totality as a constant value. Volumetric flow rate can alternative can be defined by:

$$Q = \iint_A \ UdA \qquad (1.12)$$

and for the flat cross-section we study the above equation can be written as :

$$Q = U \cdot A \qquad (1.13)$$

Consequently the Darcy's law equation for a rigid porous medium can be written in the following forms:

$$Q = -A\frac{K}{\mu}\frac{\Delta P}{L} \tag{1.14}$$

Or

$$U = -\frac{K}{\mu}\frac{\Delta P}{L} \tag{1.15}$$

In the case of a two-dimensional geometry or of a much greater width (infinite) than the length and height, the Darcy's law must be differentiated. Such a category we are studying. The flow rate per unit width must be entered in the formula:

$$q = \frac{Q}{W}\ [m^2/s] \tag{1.16}$$

where W=width of the medium [m]

Therefore, by combining the equations we take the law of Darcy that can be applied to a unit cell of two dimensions.

$$K = -q\frac{\mu}{H}\frac{L}{\Delta P} \tag{1.17}$$

Consequently Darcy's Law is valid only for laminar flow, which occurs for Reynold's number less than 1. Reynold's number is a dimensionless quantity in fluid mechanics that helps to predict flow patterns in different fluid flow situations and for flow through a unit cell is defined as(Zhang, Simon, Li, & Van de Ven, 2015):

$$Re = \frac{\rho UL}{\mu} = \frac{UL}{\nu} \tag{1.18}$$

where

ρ = fluid density [kg/m^{3}]

ν = kinematic viscosity [m^2/s]

U =fluid velocity [m/s]

In first step our main goal is to calculate the permeability and compare the numerical results with analytical models for flow through rigid porous medium. The flow rate through a cell can be obtained by integration of the velocity profile along the inlet or outlet boundaries. The velocity profile is obtained by solving the pertinent flow problem. We use OpenFOAM, a computer-aided design model which can model porous media flows. In chapter 3 is studied a single phase flow though porous medium based on Darcy's law without absorption. This means that the volume of fluid that passes through porous media per unit of time at the inlet is equal to the outlet. The Reynold's number is equal to zero because we study Newtonian liquids. Furthermore the cell geometry is standard (1x1) [in millimeters]. In chapter 4 we studied the flow of Newtonian liquid into a swelling porous medium. The geometry characteristics of the cell remain the same with the non-swelling study.

Chapter 2

2.1 Introduction to OpenFOAM

OpenFOAM foremost constitutes a C++ library (a toolbox), used to create executables, known as applications. The applications are divided into two parts/categories. The first category, the solvers, is designed to solve a specific problem in continuum mechanics; on the other the second category, the utilities, is designed to perform tasks that involve data manipulation. The OpenFOAM distribution falls into numerous solvers and utilities covering a wide range of problems (Greenshields, 2016).

User can develop a direct relationship with the program, as they create additional solvers and utilities. OpenFOAM provides pre-processing and post-processing environments, while between the two processes the process of resolving takes place. The generally structure is illustrated in Figure 3.

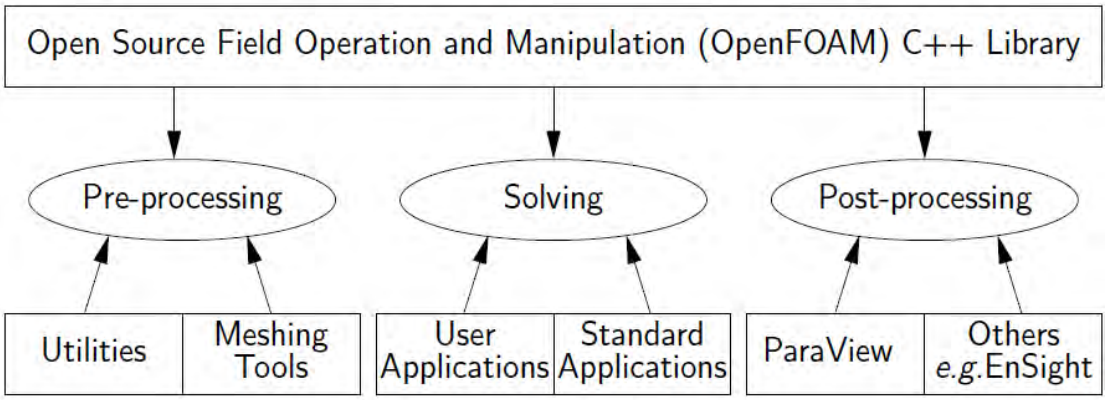


Figure 3: Overview of OpenFOAM operation (Greenshields,2015)

OpenFOAM provides users with tutorial and instructions in order to help familiarize users with the environment. In addition, in terms of engraving and projection of the outcomes, OpenFOAM provides a design tool called as ParaView.

Moreover OpenFOAM approaches many different types of problems which render impossible to have only a general form of solver. Each one of the solvers corresponds to a different problem. Below is briefly outlined the range of the cases that solvers are applicable:

- Incompressible flow
- Compressible flow
- Multiphase flow
- Direct Numerical Simulation
- Combustion
- Heat transfer
- Molecular dynamic methods
- Electromagnetism
- Stress analysis of solids
- Finance
- Monte Carlo methods simulation

2.2 Familiarity with the environment

This part describes how to pre-process, run and post-process a case involving incompressible flow in a bi-dimensional unit cell that in its center are located a number of solid cylindrical filaments arranged in periodic order (Fig.4). A pressure value is set at the left edge while at the right this value is zero. This pressure difference created between the two borders causes the flow of fluid to start. The flow is supposed laminar and will be worked out on a uniform mesh using the simpleFoam solver for laminar, incompressible liquid movement. SimpleFoam is a steady state solver that is using the SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) algorithm. Finally the fluid that flows across the unit cell is set to be Newtonian fluid this result in a low and fixed Reynolds number.

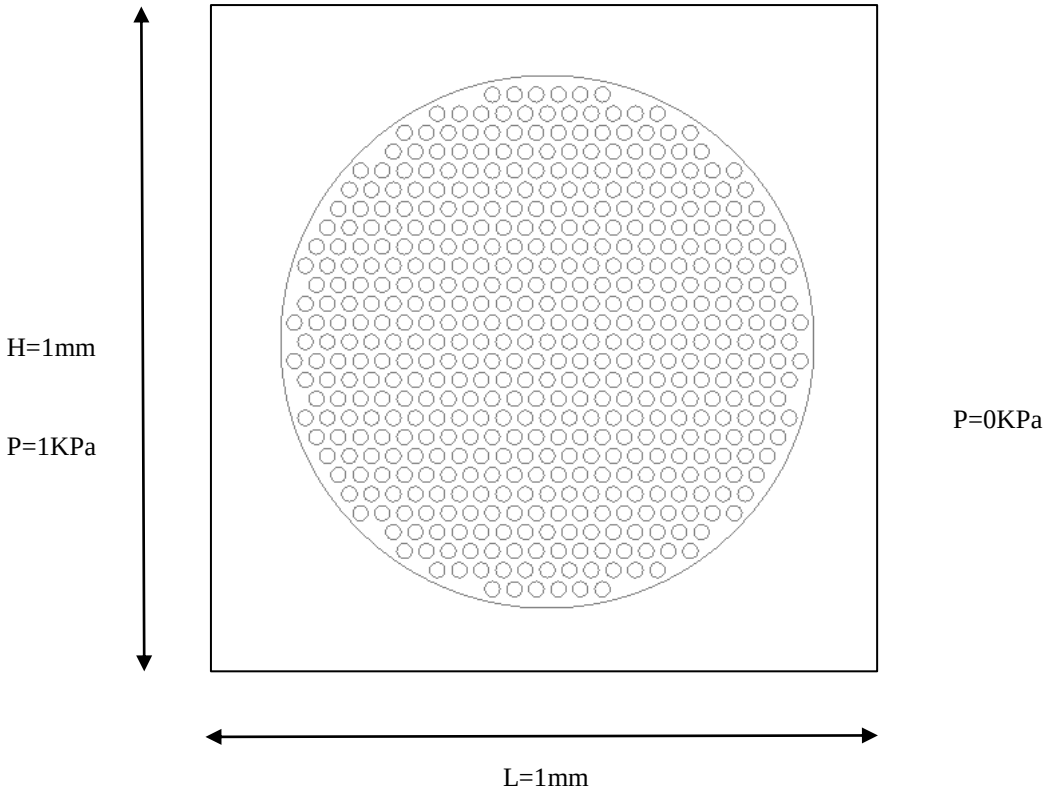


Figure 4: Filaments placed in the center of a cell in a periodic arrangement

2.2.1 SimpleFoam

Generally the flow of fluids is a phenomenon described by means of conservation of mass, momentum and energy. Because we have assumed incompressible flow the density is constant so the continuity equation limits to

$$\nabla \cdot \bar{U} = 0$$

$\bar{U} = \text{velocity [m/s]}$

On the other hand, we have the Navier-stokes equations in a three-dimensional system, ignoring the effects of gravity, assuming permanent conditions and uncompressed fluid; we end up with the following equation

$$\rho \bar{U} \nabla \bar{U} = -\nabla p + \nabla \cdot \bar{\bar{\tau}}$$

where

p=pressure

ρ=density

$\bar{\bar{\tau}} = \mu(\nabla \bar{U} + \nabla \bar{U}^T)$
$=$ Stoke′stress constitutive equation for incompressible viscous fluid

As we mentioned earlier SimpleFoam uses the SIMPLE algorithm which allows coupling the Navier-Stokes equations with an iterative proceeding

1. Setting boundary conditions.
2. Solve the discretized momentum equation to compute the intermediate velocity field.
3. Compute the mass fluxes at the cells faces.
4. Solve the pressure equation and apply under-relaxation.
5. Correct the mass fluxes at the cell faces.
6. Correct the velocities on the basis of the new pressure field.
7. Update the boundary conditions.
8. Repeat till convergence.

2.2.1.1 Pre-Processing

Data for mesh, physical properties, fields, control parameters etc are involved in a case simulation.

Mesh Generation:

The operations take place in a 3 dimensional Cartesian system. The geometries are also generated in 3D. OpenFOAM is a computational tool that is predetermined to solve in cases where its geometry is expanded into three dimensions. But it can be formulated to solve two-dimensional problems by defining an empty border in the third dimension boundary conditions, for which no solution is expected. OpenFOAM provides a mesh generator, blockMesh, which generates meshes form a description located in system folder, blockMeshDict file. Mesh generates by running on this file.

```
134. frontAndBack
135. {
136. type empty;
137. faces
138. (
139. (22 28 29 23)
140. (23 29 30 24)
141. (24 30 31 25)
142. (26 32 33 27)
143. (27 33 34 28)
144. (28 34 35 29)
145. (29 35 36 30)
146. (30 36 37 31)
147. (32 38 39 33)
148. (33 39 40 34)
149. (34 40 41 35)
150. (35 41 42 36)
151. (36 42 43 37)
152. (0 1 7 6)
153. (1 2 8 7)
154. (2 3 9 8)
155. (4 5 11 10)
156. (5 6 12 11)
157. (6 7 13 12)
158. (7 8 14 13)
159. (8 9 15 14)
160. (10 11 17 16)
161. (11 12 18 17)
162. (12 13 19 18)
163. (13 14 20 19)
164. (14 15 21 20)
165. );
166. }
```

Boundary conditions:

Each particular application will require appropriate conditions in order to various constants and functions of the solutions. The next step immediately afterwards the creation of the grid, the initial fields those are set up for our case, have to be checked. The case is plant to begin at time t=0s. That is the reason the initial field data is stored in directory called 0. Sub-directory 0 contains the pressure and velocity fields and the files named p and U respectively. In these files we can make interventions and changes in the initial and border conditions. Initial conditions amount to prescribe values of the unknowns at the start. For example we examine file p:

```
1.  dimensions      [0 2 -2 0 0 0 0];
2.
3.  internalField   uniform 0;
4.
5.  boundaryField
6.  {
7.  left
8.  {
9.  type            fixedValue;
10. value           uniform 1;
11. }
12.
13. right
14. {
15. type            fixedValue;
16. value           uniform 0;
17. }
18.
19. top
20. {
21. type            cyclicAMI;
22. }
23.
24. bottom
25. {
26. type            cyclicAMI;
27. }
28.
29. internal
30. {
31. type            zeroGradient;
32. }
33.
34. front
35. {
36. type                  empty;
37. }
38.
39. back
40. {
41. type                  empty;
42. }
43. }
44.
45. // ************************************************************************* //
```

cyclicAMI: enables two patched whose faces are non matching to be treated as they are physically connected. This boundary condition is usually used for sliding interface in rotating geometry cases.

zeroGradient: This boundary condition extrapolates the pressure (generally every quantity) to the boundary patch from the nearest cell value. The meaning is that the pressure is developed and its gradient is equal to zero in direction perpendicular to the boundary$\left(\frac{\partial P}{\partial x} = 0\right)$.

fixedValue: value of p is specified by a value (In our case at left boundary field p=1 and at right p=0 m^2/s^2)

empty: This condition characterizes the front and rear planes of the two-dimensional geometry, so they are posed as voids

Physical properties:

The physical properties are stored in directories whose name contains the word "Properties". For a simpleFoam case, the property that has to be specifies is the kinematic viscosity, stored in transportProperties. In the file the kinematic viscosity,ν, is symbolized as nu. The case we study is characterized by low Reynolds number equal to 10^{-3} and defined as:

$$Re = \frac{L|U|}{\nu}$$

where

L is the characteristic length

$|U|$ is the velocity

ν is the kinematic viscosity

Here $L = 1\,mm$,$|U| = 1\,m/s$, so that for $Re = 10^{-3}$, $\nu = 1\,m^2s^{-1}$. Below is specified the file entry for kinematic viscosity:

```
20.
21. nu              [0 2 -1 0 0 0 0] 1e-00;
22.
23. // ************************************************************* //
```

Control:

Controlling time, reading and writing of the solution data are important factors and read in, as input data, from the controlDict. This is actually a case control file located in system directory. The run stats at time t=0 s, consequently we set the startForm as startTime and afterwards starTime specified to be zero. Also we set as endTime 2000 steps if the case we are studying does not converge. Below are shown, in this case,the entries in the controlDict:

```
17.
18. application     simpleFoam;
19.
20. startFrom       startTime;
21.
22. startTime       0;
23.
24. stopAt          endTime;
25.
26. endTime         2000;
27.
28. deltaT          1;
29.
30. writeControl    timeStep;
31.
32. writeInterval   1;
33.
```

```
34. purgeWrite      2;
35.
36. writeFormat     binary;
37.
38. writePrecision  6;
39.
40. writeCompression off;
41.
42. timeFormat      general;
43.
44. timePrecision   6;
45.
46. runTimeModifiable true;
47.
48. //functions
49. //{
50. //   #includeFunc streamlines
51. //}
52.
53. // ************************************************************************* //
```

2.3.2.2 Running application

We choose to run OpenFOAM applications in the foreground, as a Linux executable. On this occasion we run simpleFoam solver, which is executed by entering the case directory and typing to the terminal window:

```
simpleFoam
```

2.3.2.3 Post-processing

By achieving convergence the results are saved to time directories and via a post-processing open-source tool, named ParaView (Fig.5), they can be viewed. The application that openFOAM uses for reading the cases is identified as paraFoam and it is running by typing paraFoam on the command line. It is essentially an extension of ParaView only for openFOAM cases.

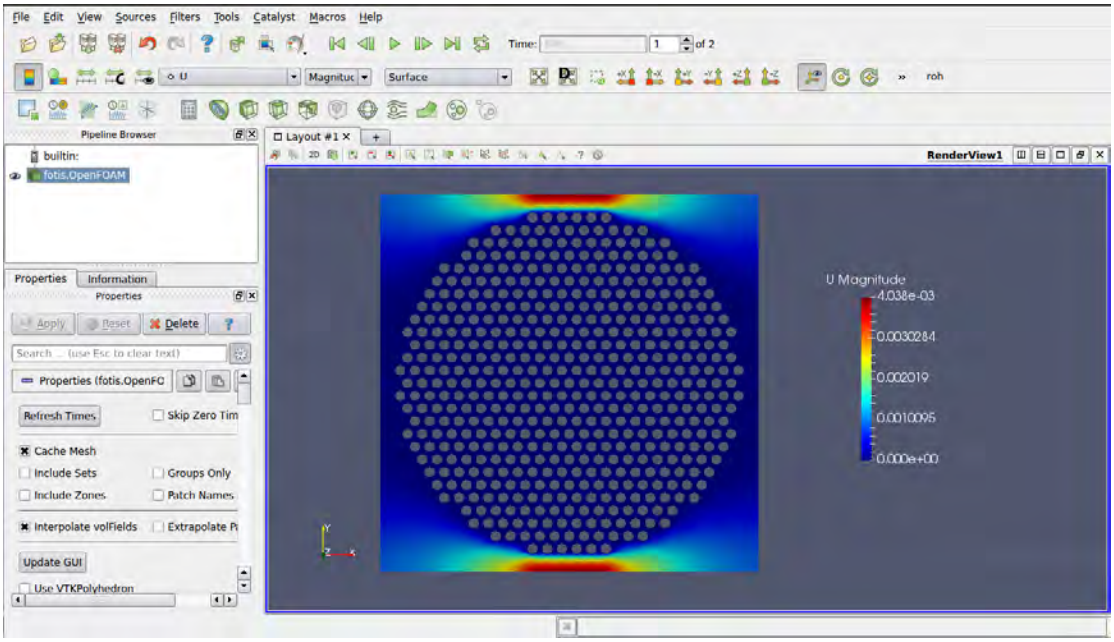


Figure 5: ParaView environment that the user does the appropriate processes he wants (Here is for example the velocity profile of a unit cell)

With the right use and experience paraView we can produce vector and streamline plots, through which we draw information on the problem. Furthermore by integrating over the inlet or the outlet surface we take the volumetric flow rate, Q. There is also an alternative solution to get the volumetric flow rate, by typing on the command line the next command.

```
postProcess -func 'flowRatePatch(name=right)' –latestTime
```

This function calculates the flow rate at the outlet for the latest simulation stored in time directory.

Chapter 3

In this section a validation of the solver simpleFoam solving simple cases is done. The results of the numerical simulation are compared with analytical expressions. Subsequently the permeability is evaluated analyzing the solution accuracy for different configurations of the SIMPLE algorithm.

3.1 Case description

The case consist long cylindrical bundles, arranged in a square packing, and closed system. Within each bundle are located a number of $N_f = 500$ rigid individual cylindrical filaments, arranged in hexagonal packing, of radius r. Our study is focuses on one of these square packaged bundles, which forms a unit cell. We assume that the dimensions of the cell are L=H=1mm, where L and H represent the length and height respectively. Priority constitutes to give substance to the problem and to do so, it must first be determined the volume fraction of the bundle (Φ_i) and of each individual filament (Φ_t). The volume fraction of the bundle in a two dimensional unit cell is defined as:

$$\Phi_i = \frac{\pi {R_t}^2}{H^2} \tag{3.1}$$

where R_t is the radius of the tow. The volume fraction of the filaments is defined as:

$$\Phi_t = N_f \left[\frac{R_f}{R_t}\right]^2 \tag{3.2}$$

where N_f is the number and R_f the radius of the cylindrical filaments in each tow.

In order to solve the problem we will create a program that will accept the number of fibers, and volume fractions of the bundle and filaments as data. It will be able to create the cell containing the bundle and the fibers. Furthermore the program will be

capable of placing the filaments inside the bundle in hexagonal array. Follows an analysis of how the program places the fibers.

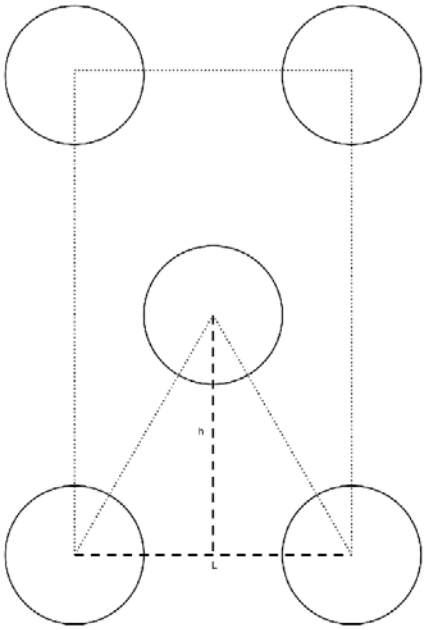


Figure 6: 1D structure for hexagonal arrays of filaments.

In Figure 6 appears a lattice of the filaments that are packed in hexagonal array. The horizontal distance between their centers is called L while the vertical h. To develop a relationship between these two dimensions, we need to deepen the geometry of the image. It is observed that between the two circles of the lower layer and the circle in the middle layer are created two isosceles right triangles, which allows us to apply the Pythagoras theorem to one of them and as a result we get:

$$h = L\frac{\sqrt{3}}{2} \tag{3.3}$$

In addition, calculating the area of the square created by joining the outer centers results:

$$4\frac{\pi {R_f}^2}{4} + \pi {R_f}^2 = \Phi_t \cdot L \cdot 2h \rightarrow L = \left(\frac{2\pi {R_f}^2}{\Phi_t\sqrt{3}}\right)^{\frac{1}{2}} \tag{3.4}$$

The program works with coordinates, so starting from the point opposite the center of the big circle begins to place the fibers along the horizontal axis at a distance L between them. Just before the fibers get out of the bundle, the process of horizontal placement stops. Then we move at a height h in the next up and down levels and start the horizontal placement process on the new levels. With the only difference that the first point at which the positioning begins, shifts horizontally by L/2 relative to the first point from the previous layer. This happens only at the first point of each layer, so the distance between every fiber in each plane remains L. The output of the program is .geo format file, so mesh can be generated and be processed by the OpenFOAM. Even though we work in two dimensions, the program produces a three-dimensional cell (L=H=W=1mm) in order to be able to run on the openFOAM (see Chapter 2.2).A part of the code is place in Appendix A.

Ultimately, the user only needs to give values in the volume fractions and the number of the filaments and will get a visual result of the case we are studying as it appears in the Figure 7.

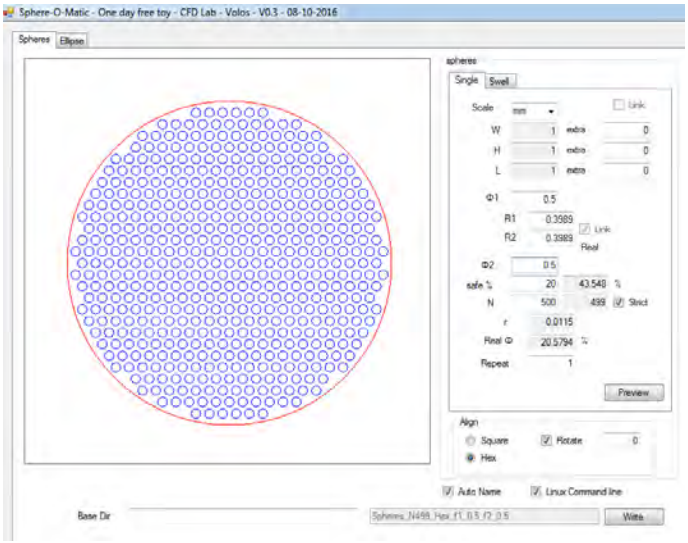


Figure 7: Visual result of the program that creates the unit cell

The user will notice that the number of filaments in the bundle, at fixed R_f and R_f , may vary a bit along with φ_i and φ_t because of the way the bundle is filled.

3.2 Parameters

We are studying linear flow of a Newtonian fluid. Thus the following assumptions are made:

Density: ρ= $10^3 [kg/m^3]$

Reynolds number: Re = 10^{-3}

Dynamic viscosity: μ=$10^3 [Pa \cdot s = kg/(s \cdot m)]$, calculated from eq. (1.18)

3.3 Boundary conditions

Pressure

In order to make the fluid flow we have to apply pressure difference on the inlet (left border) and outlet (right border) .With openFOAM it is possible to determine the kinematic pressure values in the boundary layers border. The kinematic pressure is given by the following relationship:

$$p = \rho \cdot P \qquad (3.5)$$

Where

p= Kinematic Pressure $[m^2/s^2]$

We choose to set:

$$p = 1 \; at \; left \; boundary$$

And

$$p = 0 \; at \; left \; boundary$$

That means the pressure difference $\Delta p = -1 \; m^2/s^2$ or $\Delta P = -1000 \; Pascal$

Velocity

The case we study has single phase flow across x direction as identifying trait. That means that setting the velocity in such direction perpendicular to the boundary, is a priority. That requirement is satisfied by setting the gradient of velocity equal to zero in x direction.

$$\frac{\partial U}{\partial x} = 0 \qquad (3.6)$$

And because we study a plane cross-sectional that also means:

$$\frac{\partial Q}{\partial x} = 0 \qquad (3.7)$$

In detail, the boundary conditions directly applicable to Openfoam are set out in the Appendix B.

3.4Mesh and Mesh Dependency

Mesh generates a discrete illustration of the geometry we are dealing with. By dividing gap into elements, makes possible for the equations to be estimated. A boundary zone is able to depict inlet or outlet boundaries within the geometry. Mesh generates with utilizing the mesh generator provided supplied with OpenFOAM. This command runs the Gmsh, an implement that converts the .geo file format geometries, which have been created with the visual basic program, into .msh file format. Numerical results may vary depending on the elements number we are using. The more elements exist in our mesh the less is the grid spacing. The finest grid solution is

considered to be the one with zero grid spacing. The computational load is heavy because simulations are part of a design study and more elements means additional calculations. Thus suitable solution will be a reliable and coarser grid, which as a result decreases considerably the time required for the simulations. In order to find the appropriate grid, spatial convergence must be studied. For the study of spatial convergence a method based on Richardson extrapolation is chosen(Slater, 2006).

3.4.1Spatial grid convergence

The Richardson extrapolation method allows us to obtain the highest order estimate for the value of the unknown (f) at zero grid spacing, from a range of lower order values, which are obtained using progressively finer meshes.

The value of a function f yielded by a simulation can be expressed by the series:

$$f = f_{h=0} + g_1 h + g_2 h^2 + g_3 h^3 + \ ... \qquad (3.8)$$

where

h is the grid spacing

g_1, g_2, and g_3 are functions independent of h.

$f_{h=0}$ is the continuum value at zero grid spacing.

With the assumption of n-order solution and by computing the quantity f on two grids of spacing h_1 and h_2 where $h_1 < h_2$, a generalized form of Richardson extrapolation, that estimates the continuum value at zero grid spacing, is:

$$f_{h=0} \cong f_1 + \frac{f_1 - f_2}{r^n - 1} \qquad (3.9)$$

Where r is the grid refinement ratio: $r = h_2/h_1$

For a large number of CFD computations are to be performed just like in the case we study, we may wish to use the coarser grid h. The continuum value $f_{h=0}$ considered to be n+1 order accurate. Richardson extrapolation can be applied for the solution at each grid point.

The estimated fraction error for f_2 coefficient defined as:

$$E_2 = \frac{\frac{f_2 - f_1}{f_1} \cdot r^n}{r^n} \tag{3.10}$$

where $\frac{f_2 - f_1}{f_1} = \varepsilon$ is the relative error.

The order of convergence n for three levels of grid is defined as:

$$n = ln\left(\frac{f_3 - f_2}{f_2 - f_1}\right) / ln(r) \tag{3.11}$$

In the following we will assume that f is the volumetric flow rate calculated at the outlet border. The $f_{h=0}$ thenis an estimate of quantity f in the limit as the grid spacing goes to zero. By using $f_{h=0}$ we report the value as the finest estimate of another volumetric flow rate f with grid spacing h≠0 from the CFD study; an estimation of the discretization error between $f_{h=0}$ and f can also be obtained.

By using a consistent numerical analysis that supplies an approach of the actual result as the grid resolution approaches zero (Roache, 1998). This analysis is based on a grid convergence index (GCV) that grants with a more consistent manner in presenting the results of grid convergence studies. The GCI can be computed by using two or three grid levels depending on the accuracy of the estimation of the order of the convergence we want to succeed.

The GCI measures the percentage the computed volumetric flow rate Q is away the asymptotic numerical value $Q_{h=0}$. This percentage indicates the distance between the solution and the asymptotic value. If GCI has a minor value then the computation is within the asymptotic range. GCI on the fine grid is defined as:

$$GCI_{fine} = \frac{F_S|\varepsilon|}{(r^n-1)} \tag{3.12}$$

where F_S is a factor of safety equal to 3.0 or 1.25 for two or three grid comparison respectively. The GCI on the coarser grid:

$$GCI_{coarser} = \frac{r^n F_S|\varepsilon|}{(r^n-1)} = r^n GCI_{12} \tag{3.13}$$

With the observation of two GCI values, computed over three grids, we can verify if each level of grid yield solutions which belong to the asymptotic range of convergence.

$$GCI_{23} = r^n GCI_{12} \rightarrow \ GCI_{23}/(r^n GCI_{12}) = 1 \tag{3.14}$$

If the result of the equation $GCI_{23}/(r^n GCI_{12}) \approx 1$ is true then the three grids selected can be considered reliable.

For the case we study the finer grid considered to be twofold the previous coarser grid ($h_3 = 2h_2 = 4h_1 \rightarrow r = 2$). Mesh dependency study takes place in a unit cell with $\varphi_i = 0.5$ and $\varphi_t = 0.7$ inter- and intra-tow porosities respectively.

Grid	Normalized Grid Spacing	Flow rate per unit width, q $[2/s]$
1	1	2.65373e-03
2	2	2.66144e-03
3	4	2.68732e-03

Table 1: Contains flow rates for three normalized meshes.

The first step is to determine the order of convergence adapting the eq. (3.11) to the numerical results of volumetric flow rate.

$$n = \frac{\ln\left(\frac{2.86732-2.66144}{2.66144-2.65373}\right)}{\ln(2)} = 1.74705$$

Now we are able to calculate the volumetric flow rate at zero grid spacing using the eq. (3.9):

$$Q_{h=0} = 2.650458 \cdot 10^{-3}$$

We have selected to estimation Q by using three levels of grid so the factor of safety has to be $F_S = 1.5$ .Below are calculated the GCI for the grids 1 and 2 as well for grids 2 and 3, using eq. (3.13):

$$GCI_{12} = 0.15409\% \ \ and \ \ GCI_{23} = 0.515761\%$$

Now we are able to test if the calculated values were in the in the asymptotic range of convergence using the eq. (3.14)

$$0.515761/(2^{1.74705} \cdot 0.154097) = 1.002905$$

The result is approximately one. This means that solutions can be considered reliable since they are well within the asymptotic range of convergence.

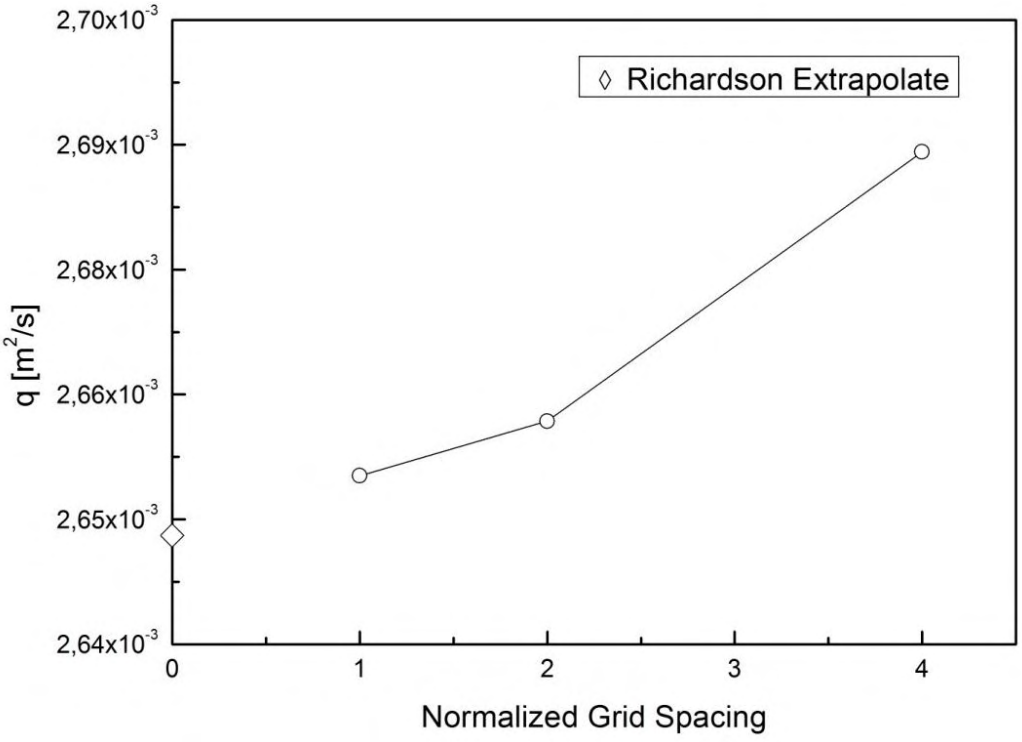


Figure 8: The convergence of computational results with respect to the normalized grid spacing

Based on a FORTRAN program, we can save the computational load (see Appendix C). The program accepts the three solutions as data and calculates what we analyzed above. The output for the results of Table (1) is:

```
--- VERIFY: Performs verification calculations ---
1.000 2.65373e-03
2.000 2.66144e-03
4.000 2.68732e-03

Number of data sets read =           3

     Grid Size     Quantity

     1.000000      0.002654
     2.000000      0.002661
     4.000000      0.002687

Order of convergence using first three finest grid
and assuming constant grid refinement (Eqn. 5.10.6.1)
Order of Convergence, p =      1.74705

Richardson Extrapolation: Use above order of convergence
and first and second finest grids (Eqn. 5.4.1)
Estimate to zero grid value, f_exact =     2.650459E-03

Grid Convergence Index on fine grids. Uses p from above.
Factor of Safety =      1.25000

   Grid     Refinement
   Step      Ratio, r       GCI(%)
   1  2      2.000000      0.154099
   2  3      2.000000      0.515766

Checking for asymptotic range using Eqn. 5.10.5.2.
A ratio of 1.0 indicates asymptotic range.

 Grid Range     Ratio
 12 23      1.002905

--- End of VERIFY ---


Press RETURN to close window...
```

Figure 9: FORTRAN output

An alternative method that in our case we will use it as verification is to solve the flow problem by continuously increasing the number of grid elements. Prices in the range of ±1% of the convergence value are considered acceptable. Here is a diagram showing how the volumetric flow rate varies with the number of mesh elements. (Studying the same case with Richardson extrapolation)

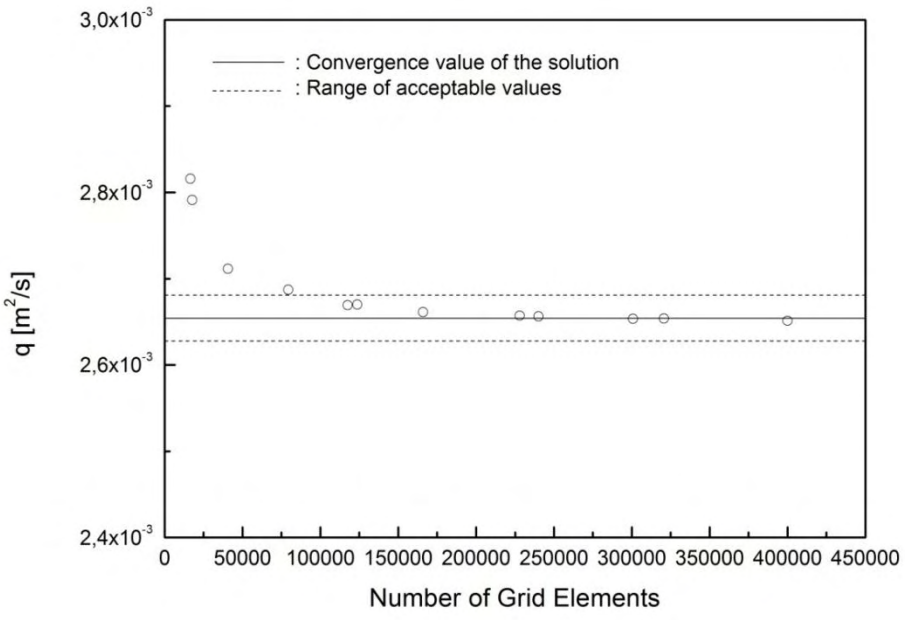


Figure 10: Convergence of values with the increase in the number of grid elements.

As we can notice the convergence value is the same. We choose to work our case study with grid number 3 from Richardson extrapolation study. The elements of this mesh are just over one hundred thousand and as a result we get reliable solutions and saving computational load. In the end, it is worth noting that the number of elements is varied depending on a coefficient, called Mesh.CharacteristicLengthFactor, which can be changed through the .geo format file.

3.5 Numerical Experiments

Once the appropriate number of grid elements has been set, we can start running simulations for various loopholes. The cases we simulate will only differ in the proportion occupied by the porous medium in the unit cell ($\Phi_1 \; or \; \varphi_i$) but also in the percentage occupied by the fibers within the porous medium ($\Phi_2 \; or \; \varphi_t$). Having now defined all the parameters we can see the grid to understand how the void space of the cell divides into elements, before the simulation begins.

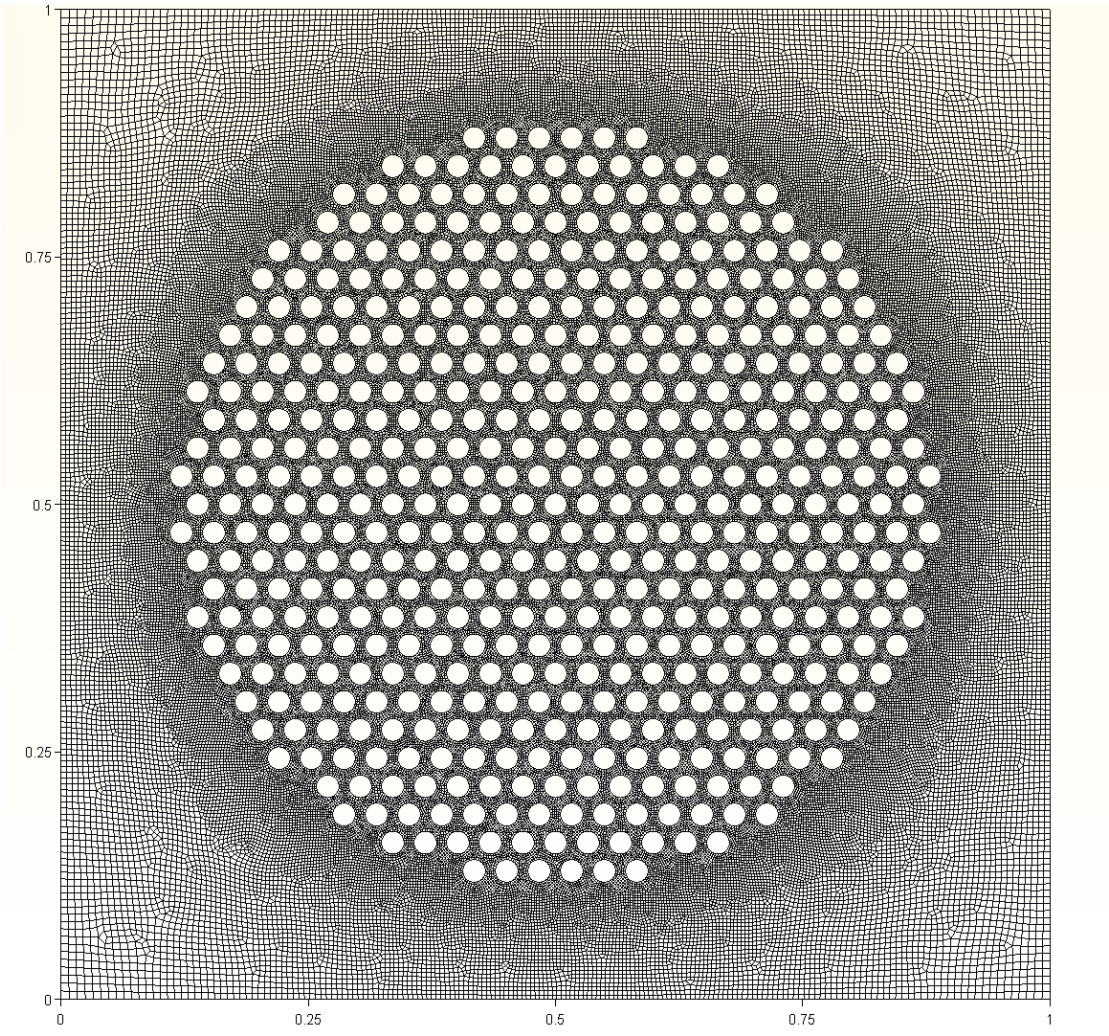


Figure 11: Mesh of unit cell for φ_i=0.5 and φ_t=0.5 porosities

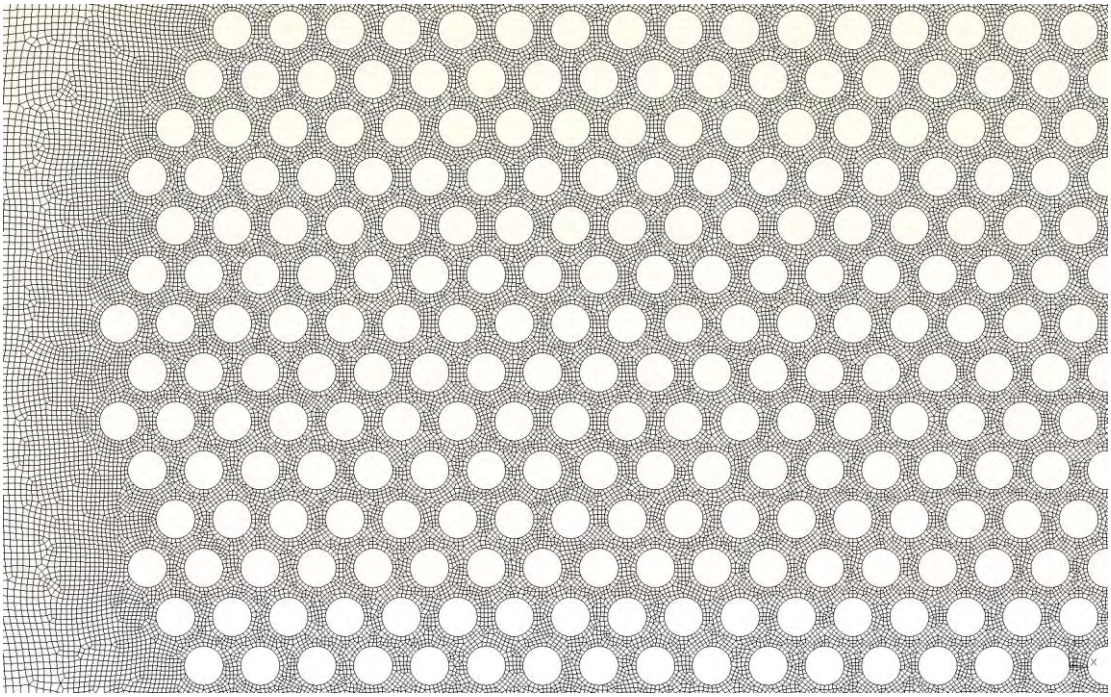


Figure 12: Zoom in mesh of unit cell for φ_i=0.5 and φ_t=0.5

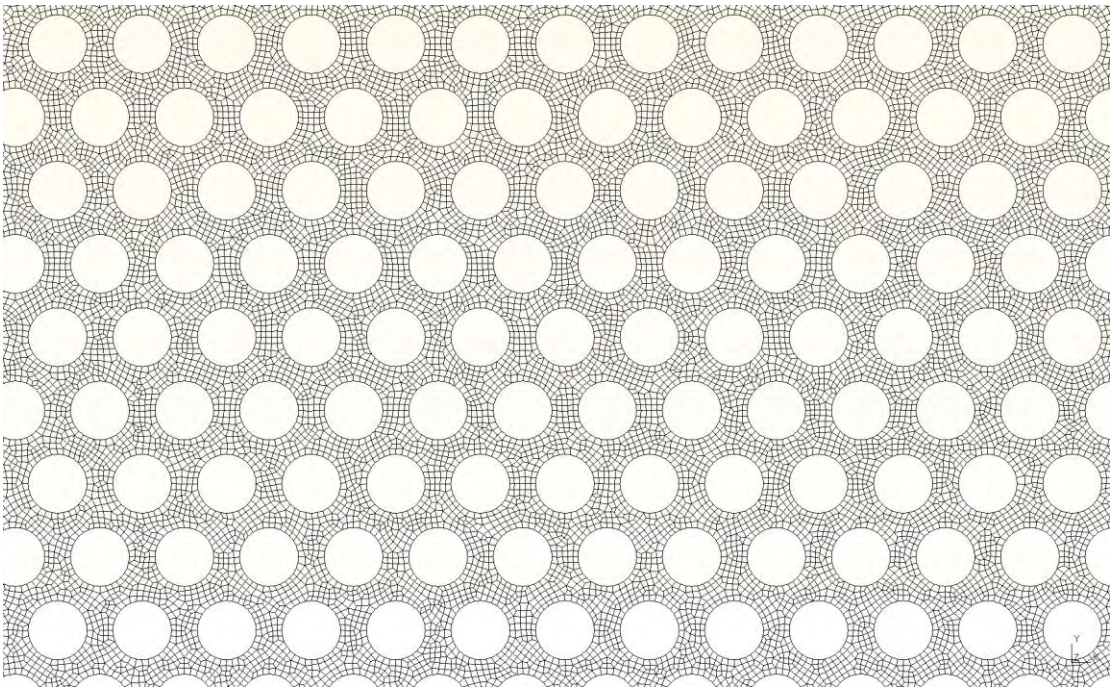


Figure 13: Zoom in internal mesh of unit cell for φ_i=0.5 and φ_t=0.5

After the simulation is completed, we can graphically present our numerical results using the design environment paraView. The following figures show how the velocity profile develops as well as how the relative drop in pressure.

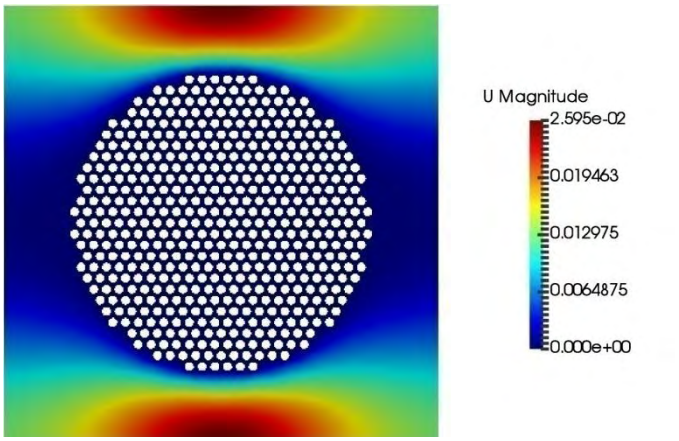


Figure 14: Velocity Profile of unit cell with φ_i=0.5 and φ_t=0.5 porosities

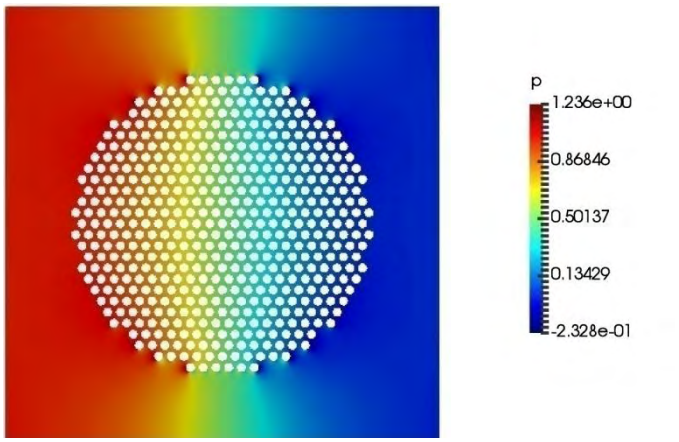


Figure 15: Pressure Drop of unit cell with φ_i=0.5 and φ_t=0.5 porosities

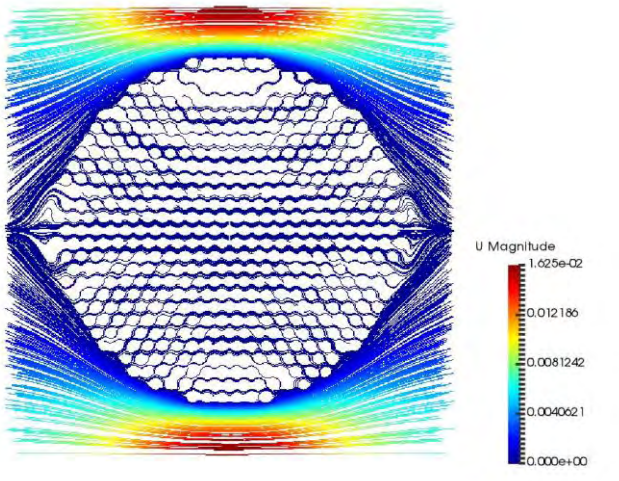


Figure 16: Stream Lines of unit cell with φ_i=0.5 and φ_t=0.5 porosities

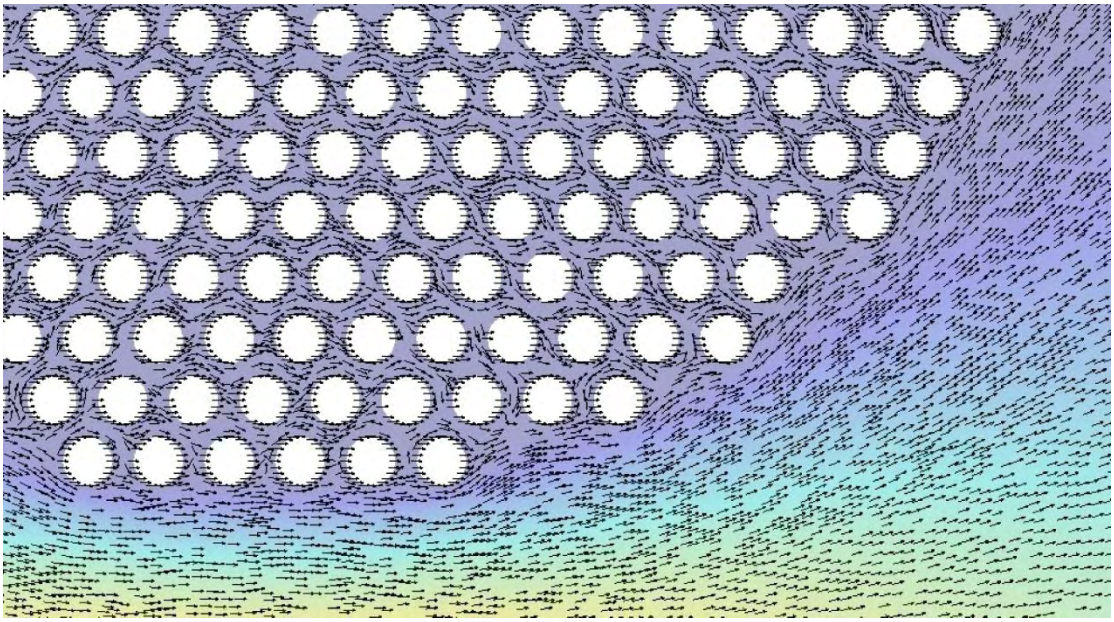


Figure 17: Velocity Vectors (zoom in unit cell with φ_i=0.5 and φ_t=0.5 porosities)

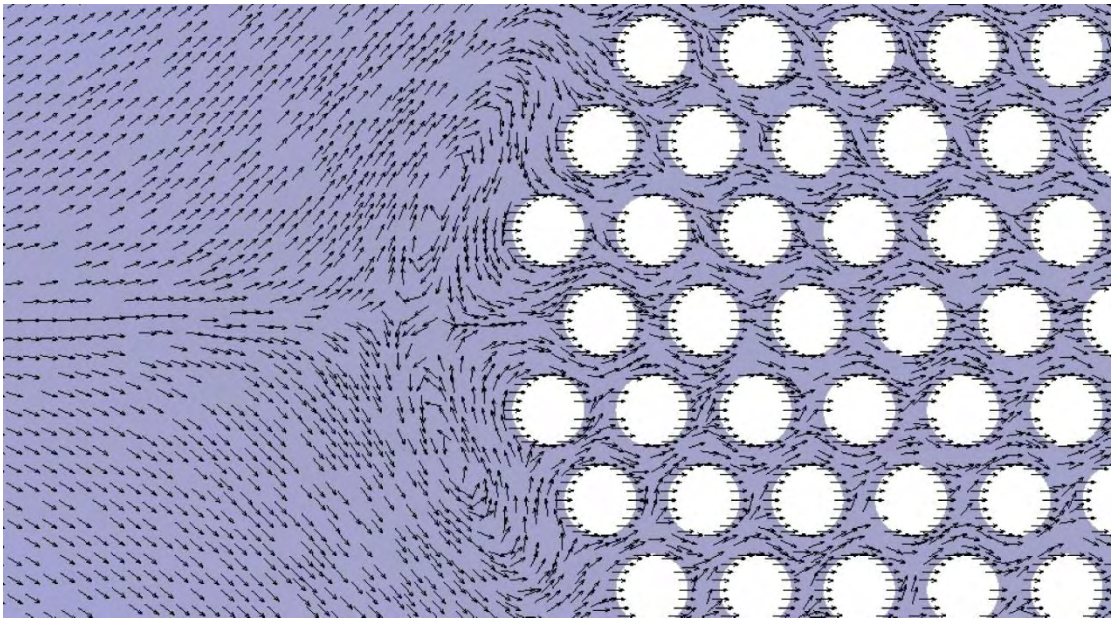


Figure 18: Velocity Vectors (zoom in unit cell with φ_i=0.5 and φ_t=0.5 porosities). There is a recirculation of fluid

At this stage, 60 cases were obtained, studied and their permeability calculated on the right border. To calculate the permeability we use Darcy's law based eq. (1.17):

$$q = -H\frac{K}{\mu}\frac{\Delta P}{L}$$

From the data needed to calculate permeability most have already been reported and characterized. Only the volumetric flow rate is unknown, but as noted in a previous chapter, is calculated using Openfoam. Actually the flow rate is obtained by integration of the velocity profile along the outlet boundaries. By knowing that L =H =10^{-3}m, A=$10^{-6}\ m^2$, μ = 10^3 Pa · s, ΔP= -10^{-3} Pa, we are also able to determine the permeability K for all the 60 cases (see Appendix D).

In the next section the experimental results will be compared to the analytical ones through Papathanasiou mathematical model.

3.6 Comparison with Earlier Results

Numerical results are compared to earlier empirical models which have the form of a dimensionless correlation. The overall permeability (K_p) of a dual porosity system is affected by the flow through the inter- and intra- tow spaces. The latter is related to the permeability of the tow itself (K_{tow}) while the former will be related to permeability of a system with the same inter-tow porosity but made up of impermeable tows (Ks).Consequently the overall permeability can be expressed in a general form:

$$K_p = f\,(K_{tow}, \mathrm{K_s}) \tag{3.15}$$

The work of Papathanasiou (Papathanasiou, 2001)suggests an empirical model with two adjustable parameters (α, n) to calculate the permeability,$\mathrm{K_p}$,of the unit cell .

$$K_p = K_s\left(1 + \alpha\left(\frac{K_s}{K_{tow}}\right)^{n-1.5}\right) \tag{3.16}$$

Or in terms of the dimensionless quantity $Y = K_p {K_s}^{1/2}\left({K_{tow}}^{3/2}\right)$

$$Y \equiv \frac{K_p {K_s}^{0.5}}{{K_{tow}}^{1.5}} = \alpha\left(\frac{K_p}{K_{tow}}\right)^n\left(1 + \frac{1}{\alpha}\left(\frac{K_s}{K_{tow}}\right)^{1.5-n}\right) \tag{3.17}$$

$\mathrm{K_s}$ and K_{tow}can be calculated by Gebart's model (Gebart,1992 as cited in (Ngo & Tamma, 2004)):

$$K = C\left(\sqrt{\frac{V_{fmax}}{1-\varphi}} - 1\right)^{5/2} R^2 \tag{3.18}$$

Fiber arrangement	C	V_{fmax}
Quadratic	$\frac{16}{9\pi\sqrt{2}}$	$\frac{\pi}{4}$
Hexagonal	$\frac{16}{9\pi\sqrt{6}}$	$\frac{\pi}{2\sqrt{3}}$

Table 2: Parameters of Permeability's equation (3.18)

To calculate K_s we set $\varphi = \varphi_i$, $R = R_{tow}$ and $V_{fmax} = \frac{\pi}{4}$ depending on the square packing. To calculate K_{tow} we use $\varphi = \varphi_t$, $R = R_f$ and $V_{fmax} = \frac{\pi}{2\sqrt{3}}$ because of having hexagonal arrangement. The ratio K_s/K_{tow} is a function of φ_i and φ_t . For the SH arrays we study the ratio can be expressed in a general form:

$$\frac{K_s}{K_{tow}} = \sqrt{3}\left(\frac{R_{tow}}{R_f}\right)^2 \left(\frac{\sqrt{\frac{\pi}{4(1-\varphi_i)}}-1}{\sqrt{\frac{\pi}{2\sqrt{3}(1-\varphi_t)}}-1}\right)^{2.5} \qquad (3.19)$$

In Appendix E the results of the equations (3.17), (3.18), (3.19) for a variety of porosity values, with parameters α= 3.0 and n = 0.56 are shown .It should be noted that the values for the parameters n and αobtained by Papathanasiou research.

Figure 19 shows Papathanasiou model and experimental results plotted as suggested by eq.(3.17).

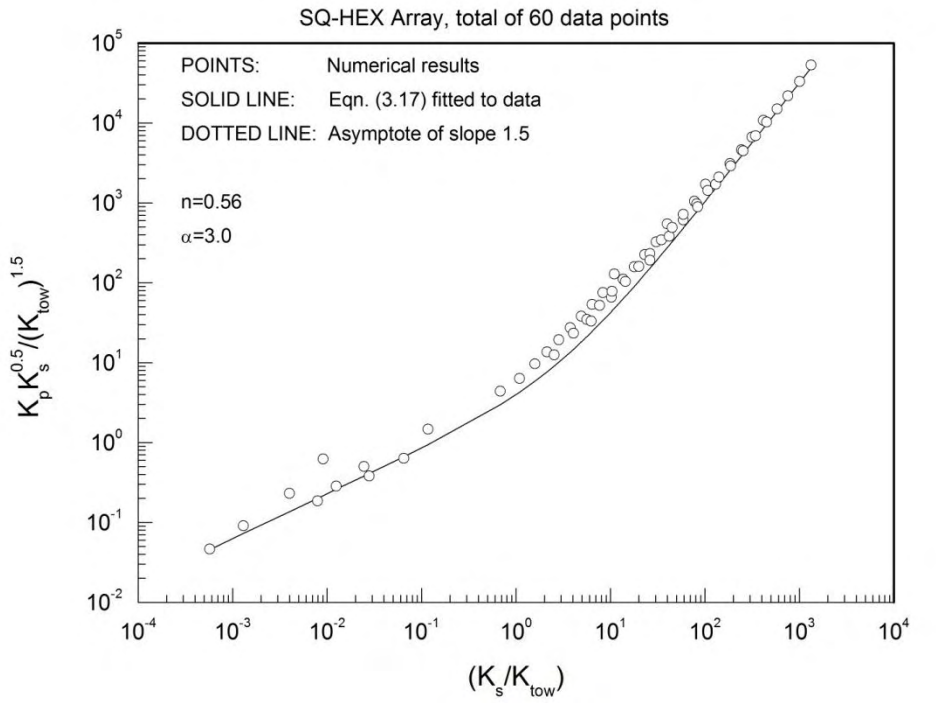


Figure 19: Numerical results for the permeability (K) of SQ arrays plotted on a log-log scale as suggested by Eq.(3.17)

It can be seen that the numerical data collapse on a curve described by eq. (3.17) with α=3.0 and n=0.56. As anticipated, at high values of (K_s/K_{tow}) the numerical data approach an asymptote with a slope of 1.5 on the log-log graph:

$$Y \to \left(\frac{K_s}{K_{tow}}\right)^{1.5} \tag{3.20}$$

At low values of (K_s/K_{tow}) the higher scatter is observed and specially due to the lower values of inter-tow porosity φ_i combined also with lower values of intra-tow porosity φ_t . At low levels φ_i of, small changes in the location of the filaments in the parameter of the tow tend to have large influence on the flow resistance of the unit cell. Nevertheless this applies in individual cases and so most of the low values of (K_s/K_{tow}) can be expressed as:

$$Y \to \alpha\left(\frac{K_s}{K_{tow}}\right)^{n-1.5} \tag{3.21}$$

As we can see at low values of (K_s/K_{tow}), there is dependence on the parameters α and n, while at higher values this does not apply.

Chapter 4

4.1 Case Description

As in the previous chapter, it is also contemplated that the flow of a neuronal fluid through a unit cell. The difference is found in the boundary conditions applied to the bundle and hence to the fibers. Essentially, the fibers absorb a proportion of the fluid flowing between them. The consequence of this phenomenon is the increase in the diameter of the fibers, while the diameter of the bundle remains constant. The bundle stops absorbing fluid only when the radius of each filament begins to osculate each other. At that time, the bundle is filled and no fluid passes through the porous medium. For the purpose of this study, it is assumed that the flow at the input remains constant until the phenomenon is complete. As a consequence there is a limitation on the generalization of the results and is a flaw in the study. To make it more conspicuous, the fluid supply at the inlet for a stack of staggered dimensions is not entirely stable but varies within a small range that changes along with the fiber geometry. A representative value is selected within this range and is considered to be the flow rate used to determine the size absorbed by the porous medium. The rate of growth is linearly proportional to the percentage change of the volume fraction.

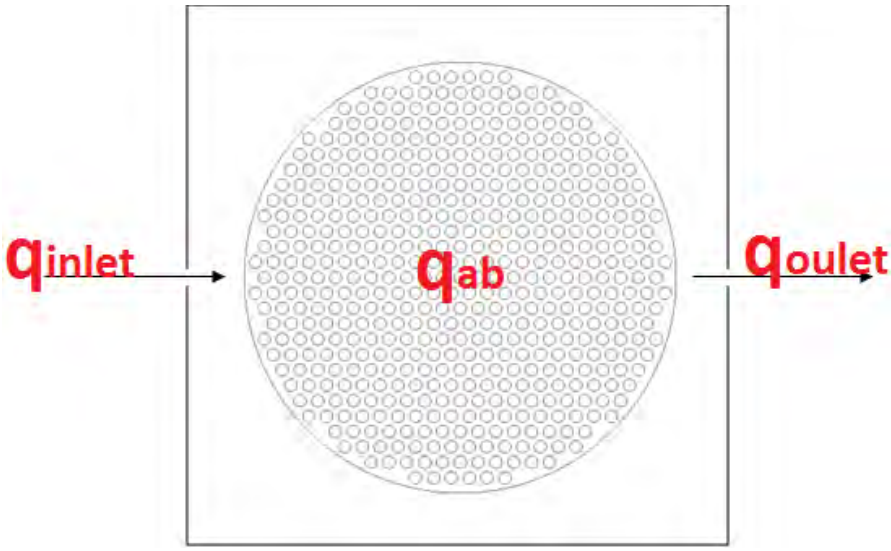


Figure 20: Appears where flow rate is present

Therefore, for a steady flow, the absorbed flow rate per unit width is defined as:

$$q_{ab} = ab_{rate} \cdot q_{inlet} \tag{4.1}$$

where ab_{rate} =percent of inflow absorbed by the fibers. As mentioned in chapter 1.4, the amount of liquid retained by the expanded bundles of fibers ranges between 5.5 and 100%. The flow rate is divided and absorbed equally by the fibers. With the assumption that the absorption of each fiber remains steady until complete blockage of the flow path by the expanding fibers, the rate of the increase of each fiber is given by the following equation:

$$\Delta r = q_{ab}/(2\pi r N_f) \tag{4.1}$$

where Δr is the rate of the increase of the radius measured in m/s. Here it is also assumed that the density of the absorbed fluid is the same as the density of the flowing fluid, in other words, that the absorption coefficient of Pillai et al (2010) is 1. At each time step the radius value is defined by:

$$r_t = dt \cdot \Delta r + r_0 \tag{4.3}$$

Where

r_0 is the initial radius [m]

dt is the time step [s]

r_t is the radius of the fiber after dt time steps .

It should be noted that unlike the previous chapter, we will study the dimensionless permeability $K/{R_f}^2$. Using the visual basic program, at each level of inter-tow

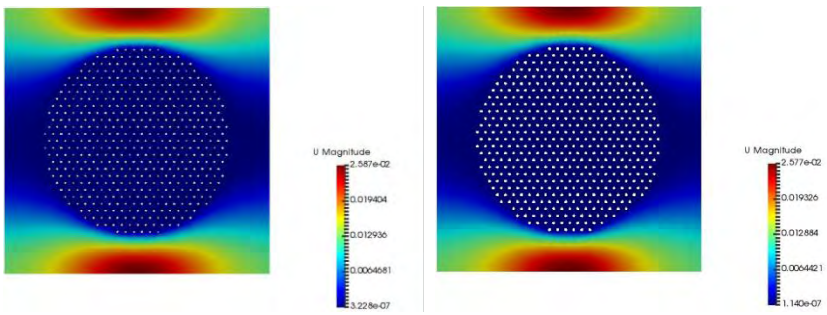
porosity the number of the fibers in the tow is varied along with φ_t , whereas in fact the number of fibers and their arrangement remain constant. With the study of dimensionless permeability, we emphasize each fiber separately, and we do not deal with number within the bundle. The behavior of the non-dimensional permeability for various inter-tow porosities and absorption rate values will be studied. Initial fibers' porosity is large and the absorption continues until their radius are united (at that time no fluid passes through the bundle).

.4.2 Boundary Conditions

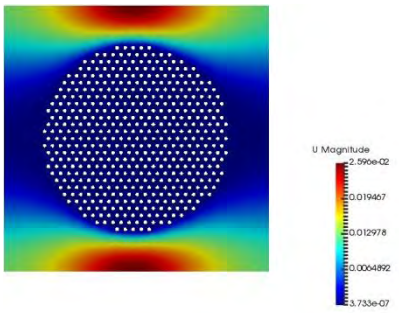
The only difference in boundary conditions with respect to Chapter 3 is at velocity. In particular, in the border conditions of velocity there is a change in the internal boundary. Once there is absorption of liquid from the fibers, which was not done in the previous chapter, a new boundary has to be set which will set the percentage absorbed by the porous medium. (For OpenFOAM's U boundary conditions see Appendix F)

4.3 Computational Results

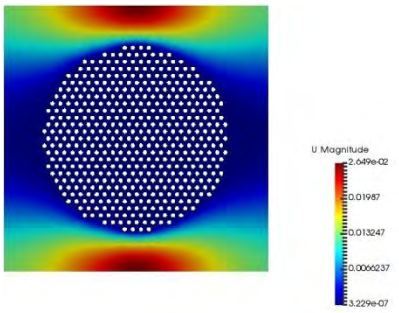
After the simulations have been completed, variations in the velocity profile and pressure distribution due to the absorption over time are observed.



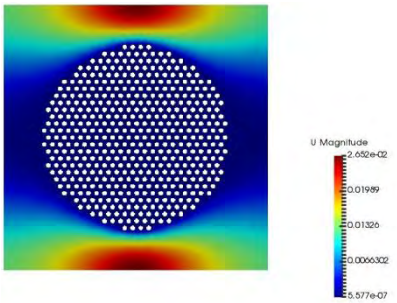
t=0 s t=98 s



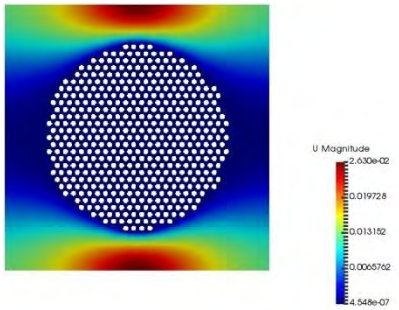
t=196 s



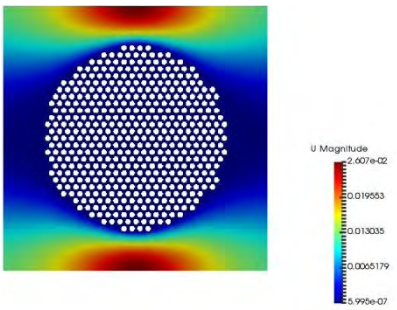
t=294 s



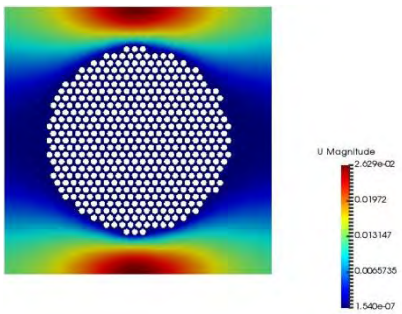
t=392 s



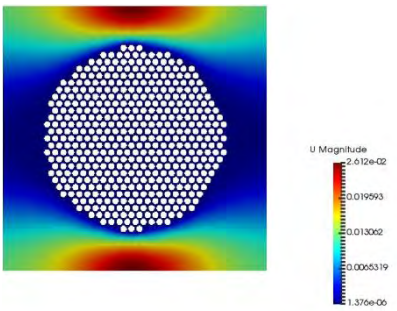
t=490 s



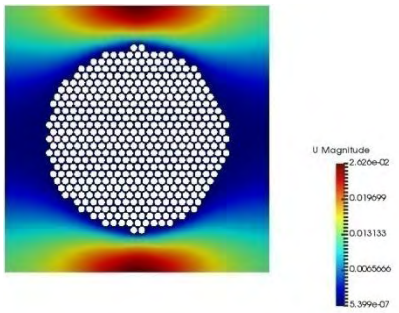
t=588 s



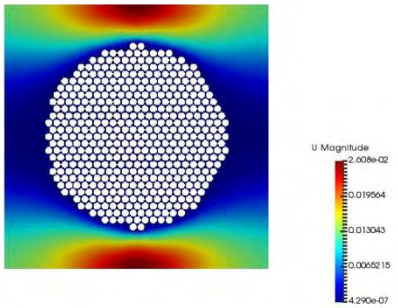
t=686 s



t=784 s



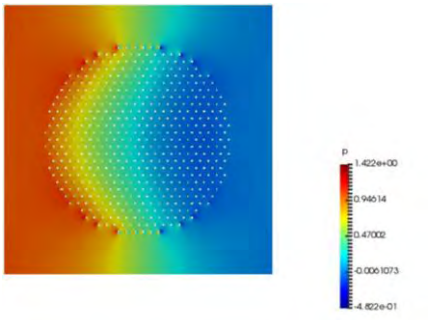
t=882 s



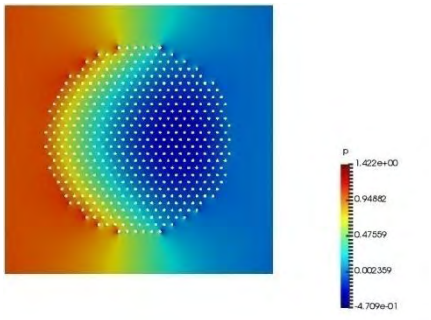
t=980 s

Figure 21: Shows how velocity profile of a unit cell with $\varphi_i = 0.6$ changes thought time

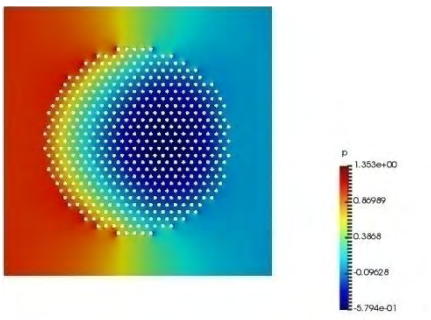
It is noted that the velocity profile remains virtually unchanged, at the right boundary. The factor that determines the velocity, primarily, is the percentage occupied by the bundle within the unit cell (Φ_i *or* φ_i), which remains stable during the course of the study.



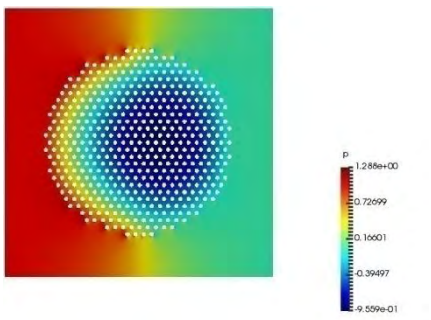
t=0 s



t=98 s



t=196 s



t=294 s

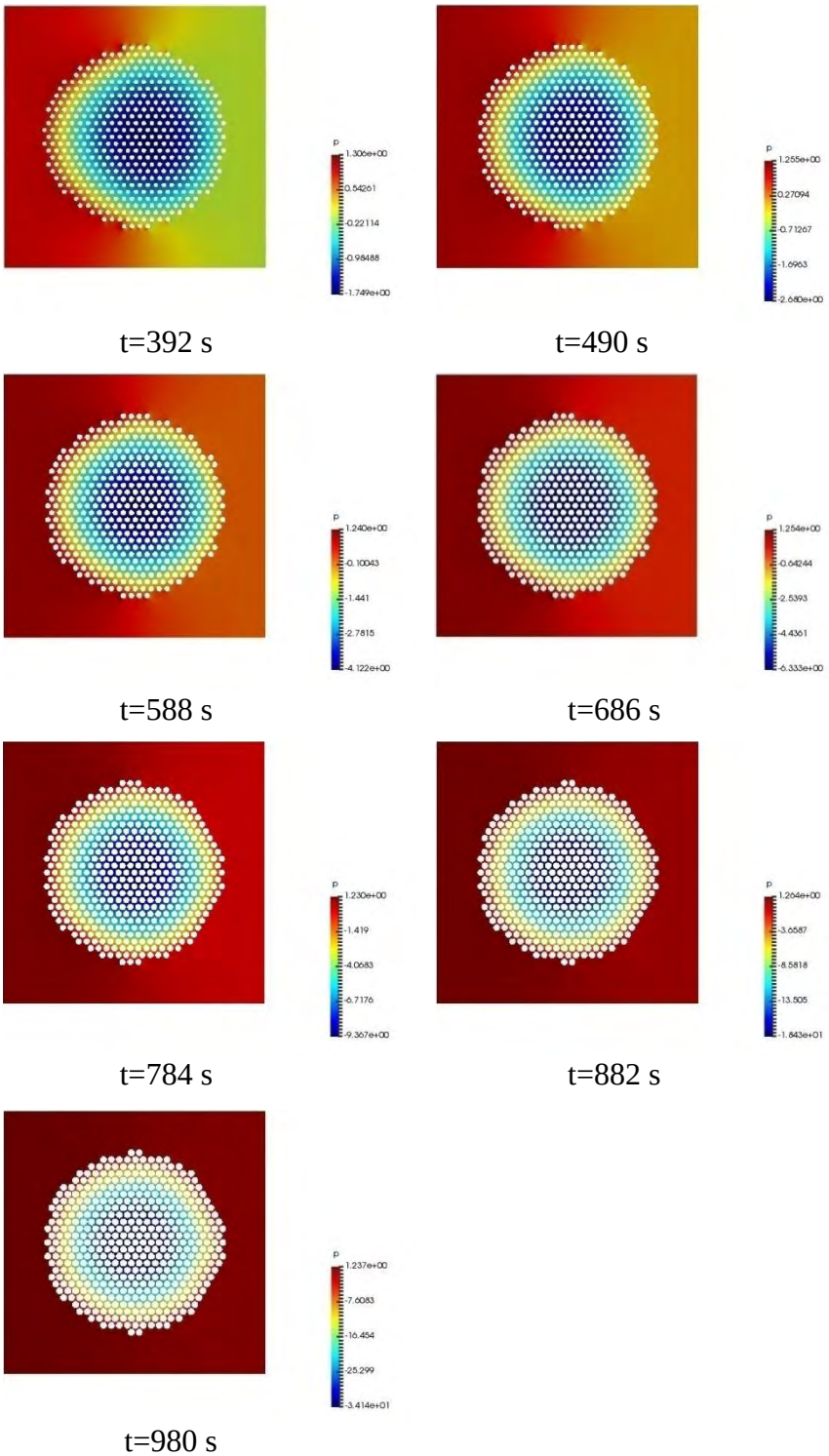


Figure 22: Shows how pressure drop of a unit cell with $\varphi_i = 0.6$ changes thought time

Unlike the velocity profile, the pressure distribution shows obvious differences as time passes. By increasing the radius of the fibers render it more difficult for fluid to

flow between them. However, the amount to be absorbed remains constant. For this reason, the pressure drop inside and outside the porous medium grows as the fiber radius increases. Therefore in conclusion the pressure distribution within the unit cell is distributed according to the percentage of fiber occupied within the bundle ($\Phi_t \; or \; \varphi_t$).

A taste of how the liquid moves into the cell is given by observing the vectors of the velocity and the stream lines.

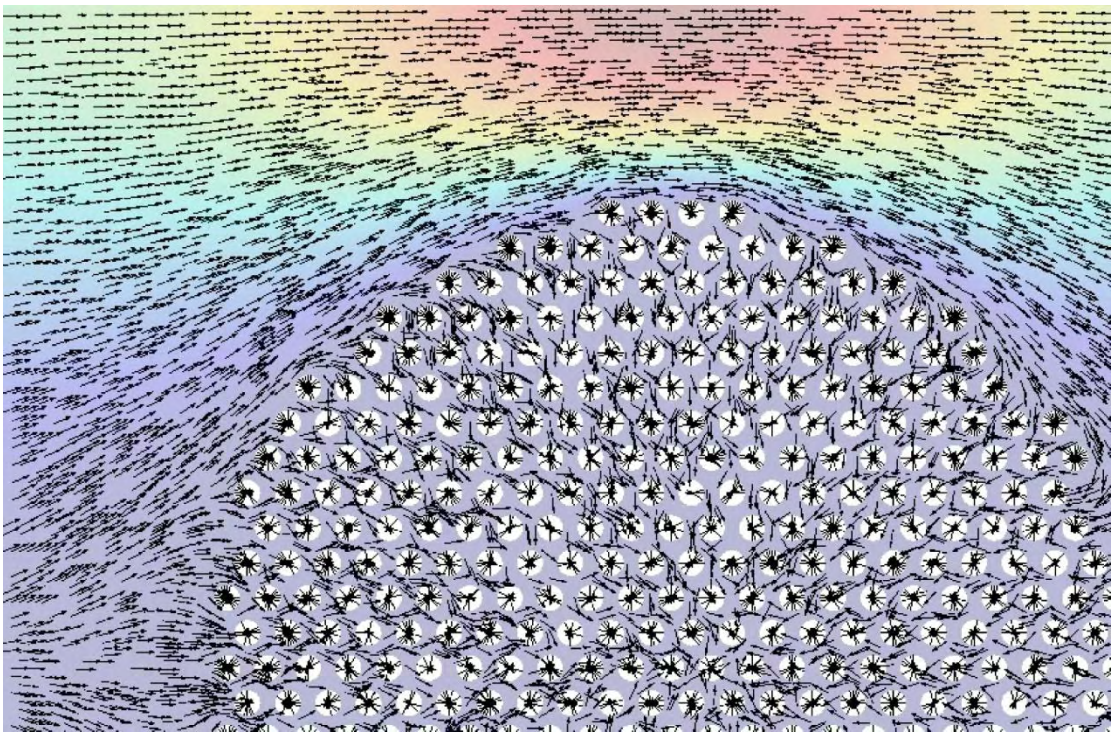


Figure 23: Velocity vectors of swelling porous media at random time

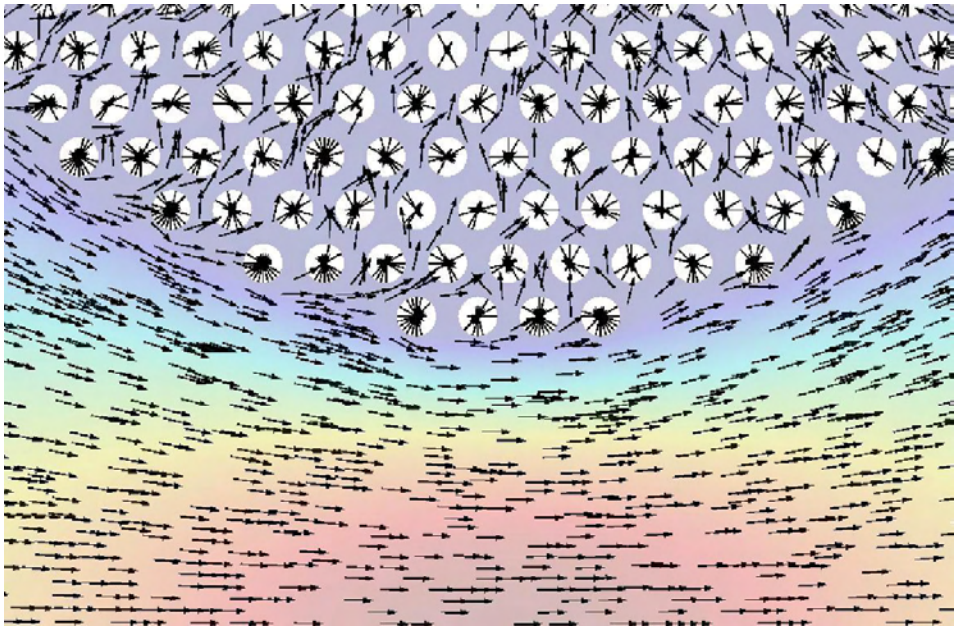
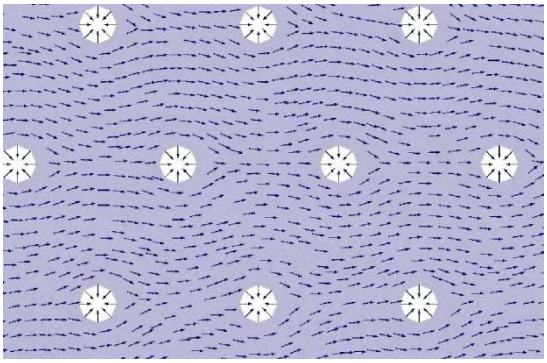
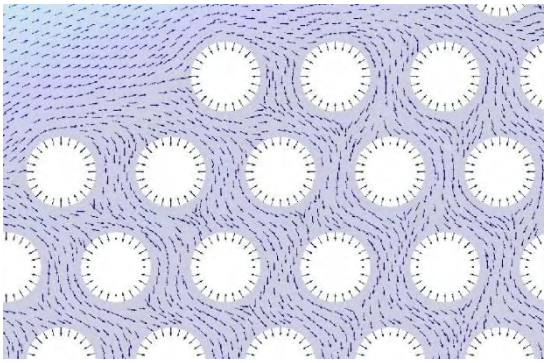


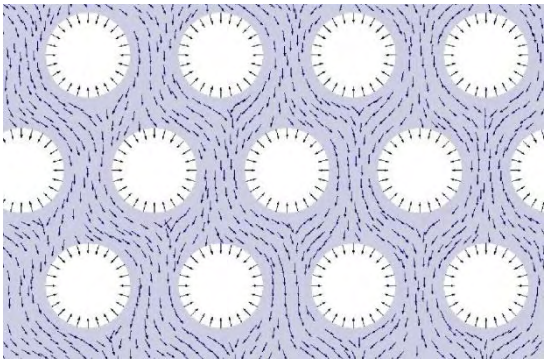
Figure 24: Velocity Vectors



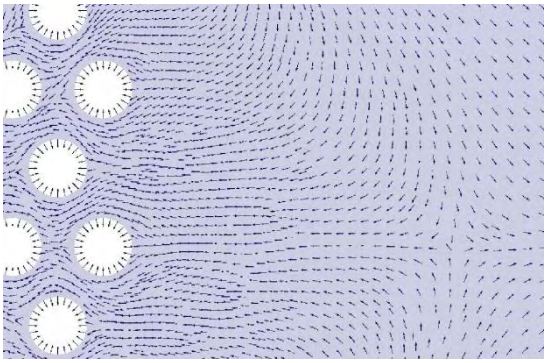
(a) Horizontal flow of fluid into the porous medium



(b) Fluid flow inclined into the porous medium



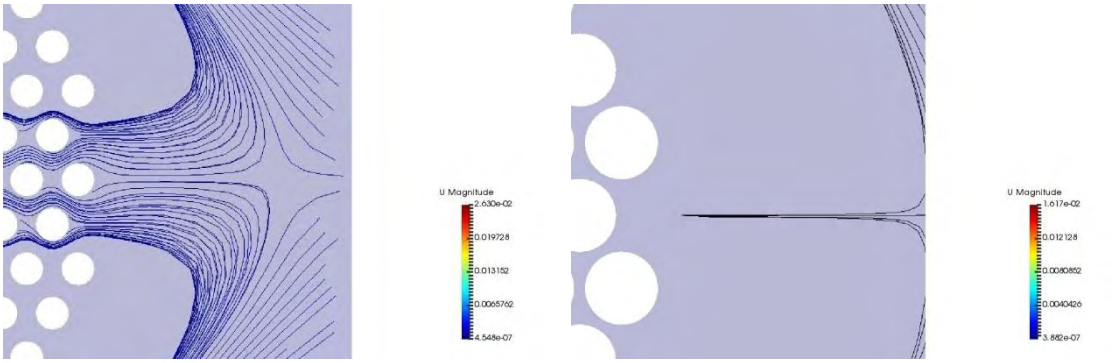
(c) Vertical flow of liquid into the porous medium



(d) Liquid flow when leaving the porous medium

Figure 25: Zoom in the fluid flow between the fibers

Simulations show that not all inter-tow porosities give good results. This is for large Φ_i (or small φ_i).Because of the large percentage occupied by the bundle within the cell; the flow does not advance fully developed, with the result that the medium in turn does not absorb the percentage set. The solver to cover the absorbed amount provides flow to the cell from the right border (Fig. 26). This phenomenon becomes more intense with the increase of Φ_t, with the result that more and more flow is provided from the right. The purpose is to avoid such cases because they give incorrect permeability measurements and for this reason cases with small inter-tow volume fractions will be studied ($\Phi_i < 0.5$).



i. Flow in a unit cell with $\Phi_i = 0.4$ ii. Flow in a unit cell with $\Phi_i = 0.5$

Figure 26: Shows that there is a flow rate from the right border for a fraction of volume greater than 50%

For this reason the volume fraction of the bundle studied is equal to 40% ($\Phi_i = 0.4$).The numerical experiments are plotted on a graph of axes $K/{R_f}^2\ vs\ t$, so can be observed how the dimensionless permeability changes over time. In the case below, the fibers absorb 5% of the flow ($ab_{rate} = 5\%$).

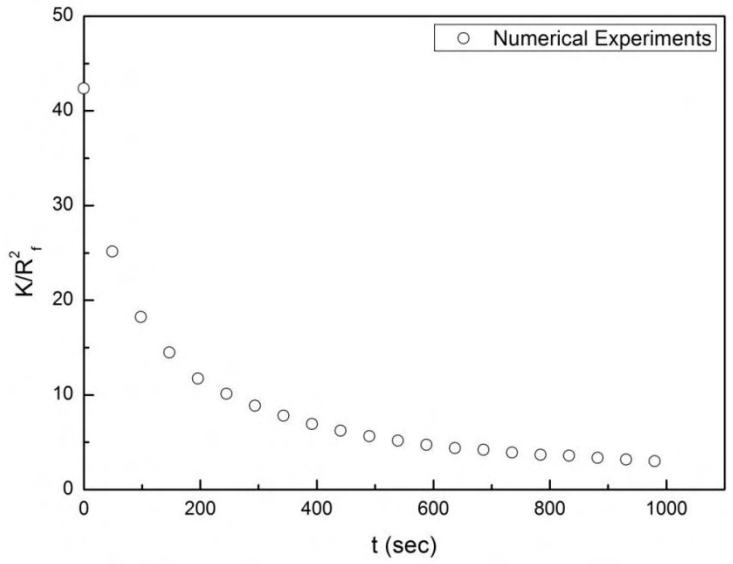


Figure 27: Numerical experiments for the dimensionless permeability ($K/{R_f}^2$) of SQ arrays plotted versus time. The percent of inflow absorbed by the fibers is 5% and bundle's porosity is 0.6.

By observing figure 28 is obvious that the dimensionless permeability reduces to time. Such form charts can be described by power law equations (Appendix G for additional geometries). This becomes clearer if we observe the above diagram on logarithmic axes.

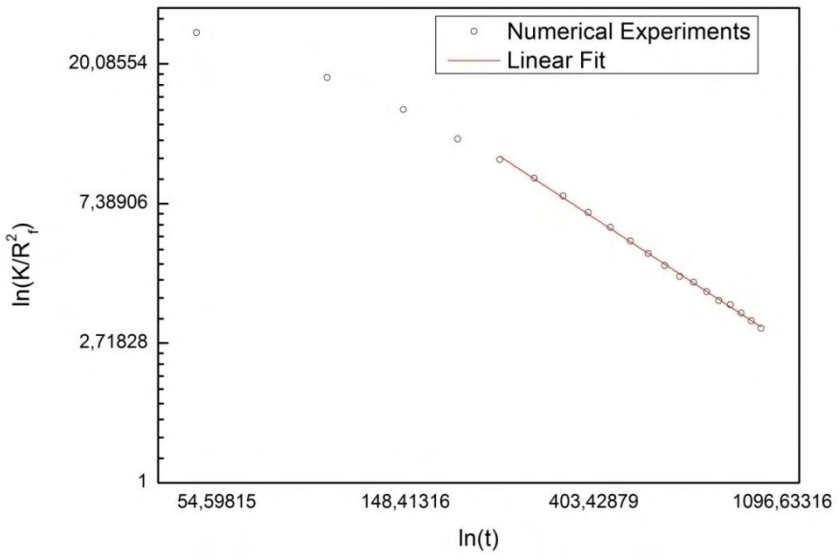


Figure 28: Dimensionless permeability ($K/{R_f}^2$) versus time plotted on a ln-ln graph

Observing the logarithmic diagram it is concluded that for long times the dimensionless permeability decreases with a steady slope. Thus the points satisfy the same linear equation. The relationship that satisfies non-dimensional permeability in relation to time is as follows:

$$\frac{K}{R_f^2} = \frac{A}{B+t^n} \qquad (4.4)$$

where A, B, n are constants and in particular for the factor n claim it is smaller than the unit (n<1).

We used Curve Expert, a program that helps to approach, by way of example, the equation that satisfies our results (Fig. 29). Giving different values to its constant equations, it is observed that the results of the experiment are approached by the equation, especially with the passage of time.

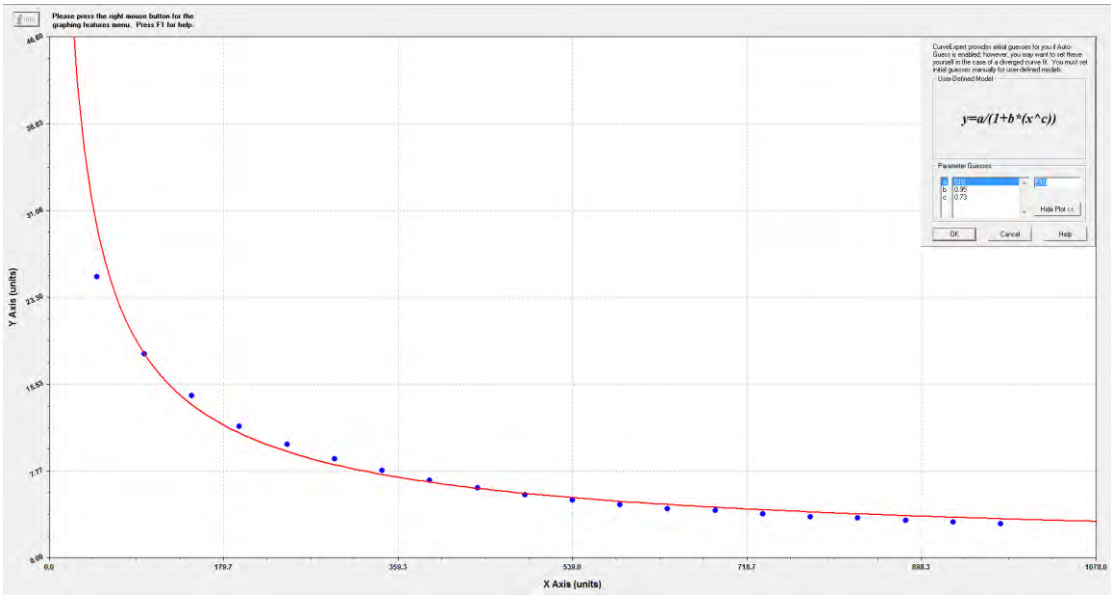


Figure 29: Numerical results for the dimensionless permeability ($K/{R_f}^2$) of S-Q arrays plotted versus time as suggested by eq. (4.4). The final values of the variables A, B and n were given after testing with the program Curve Expert.

From a physical point of view, in the case we study some ambiguities are observed, such as the continuous absorption of a constant amount of liquid from the fibers throughout the experiment. Our experiment is computational and so we are allowed to do this without obstacles. What really deserves to be pointed out is that in real-time experiments, similar behavior of dimensionless permeability over time is being noticed.

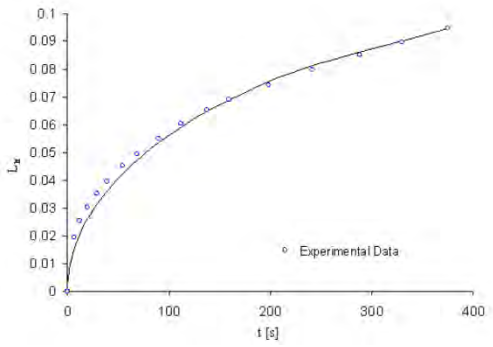


Figure 30: The above diagram is encountered for wicking paper-like swelling porous media (Masoodi & Pillai, 2010) and the y axis of the diagram is a function of permeability.

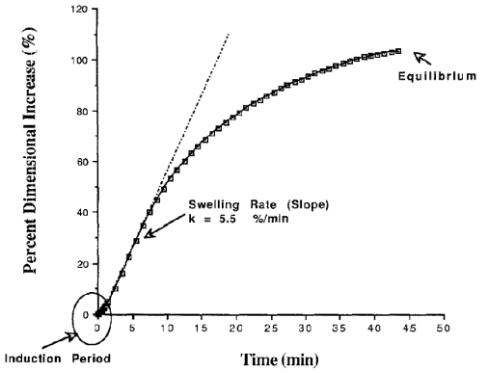


Figure 31: The above diagram is encountered in a study of swelling of cellulose fiber (Mantanis, Young, & Rowell, 1995) and the y axis of the diagram is a function of dimensionless permeability.

Figure (30) y-axis is characterized by the following relation $Y \sim \sqrt{\frac{K(t)}{{R_f}^2}t^2}$ (In this particular case the rate of growth is linearly proportional to the percentage change of the volume fraction, just like in the case we study). Figure (31) y-axis is characterized by a dimensionless relation $Y \sim \int \frac{K}{{R_f}^2}dt$. In the first case to make a comparison, we set a y-axis equal to $\sqrt{\frac{K(t)}{{R_f}^2}t^2}$ with [m] units and in the second y-axis equal to $\int \frac{K}{{R_f}^2}dt$ .The diagrams we get is similar to these of the figures (30) and (31) ,which gives us the conviction that further study will lead to a generalized model.

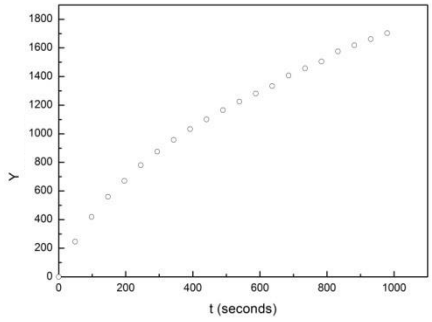


Figure 32: A similar behavior is observed with that of Figure 30, while Y= $\sqrt{\frac{K(t)}{R_f^2}t^2}$.

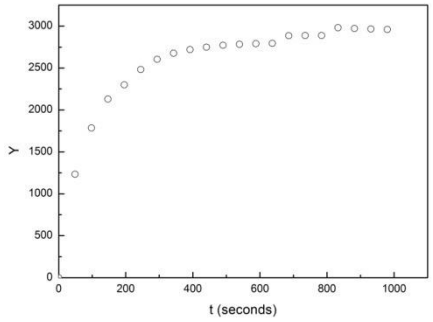


Figure 33: A similar behavior is observed with that of Figure 31, while Y= $\int \frac{K}{R_f^2} dt$.

4.4 Proposals for further study

As mentioned, the number of fibers is not a static and depends on the porosity. In later studies, it would be ideal that the way in which the fibers are placed in a hexagonal configuration is made in such a way that their number is stable. The result will be that the permeability can be directly studied, not its non-dimensional form.

Another issue for discussion is the drop in pressure that drives the fluid. We can study scenarios with a greater pressure drop and see if the flow develops well enough to have proper results for even more porosities (for $\Phi_i > 0.5$).

APPENDIX

Appendix A

Part of the Visual Basic program that results in the geometry of the cell. (The part that fills the bundle)

```
PrivateSub Fill_Spheres(ByVal write AsBoolean, ByVal nm AsString)
Dim i AsInteger, j AsInteger
Dim x AsSingle, y AsSingle
Dim pen AsNewPen(Color.Blue, 1)
Dim e AsGraphics
Dim vol AsSingle = H.Text() * H.Text()
Dim Rc AsSingle, Veff AsSingle, Rc2 AsSingle
Dim Ac AsSingle
Dim Lh AsSingle, Hh AsSingle
Dim iRf AsInteger
Dim check AsSingle
Dim cList AsNewList(OfPointF)
Dim r_x1 AsSingle, r_x2 AsSingle, r_y1 AsSingle, r_y2 AsSingle
Dim r_step AsSingle, caount AsSingle
Dim cp AsPointF, cp2 AsPointF
Dim k AsInteger

        vol = Math.PI * R1.Text() * R1.Text()
        Veff = vol * F2.Text()
        Ac = Veff / CSng(N.Text())

        Rc2 = Math.Sqrt(Ac / Math.PI)

        Veff = vol
        Ac = Veff / N.Text()
        Rc = Math.Sqrt(Ac / Math.PI)

        check = 500 * Math.PI * Rc2 * Rc2
If Rc2 = 0 ThenExit Sub

        safe_r.Text() = (100 * (Rc - Rc2) / Rc2).ToString("0.000")
        safe_r.BackColor = safe_r.BackColor
        safe_r.ForeColor = Color.Black
IfCSng(safe_r.Text()) <CSng(safe_limit.Text()) Then safe_r.ForeColor =
Color.Red

        e = pb.CreateGraphics
        i = 1
        iRf = 0

If align_hex.Checked() = TrueThen
            r_x1 = -(0.5 / Rc) : r_x2 = -r_x1
            r_y1 = -(0.7 / Rc) : r_y2 = -r_y1
            r_step = 0.5

If strict.Checked() = FalseThen
```

```
                r_x1 = 0 : r_x2 = 1 / Rc
                r_y1 = 0 : r_y2 = 1.4 / Rc
                r_step = 1.5
EndIf

            Lh = 0
Hh = 0
            cp.Y = 0.5
            cp.X = 0.5
            caount = 0
For i = r_y1 To r_y2

                cp.Y = cp.Y + Hh
                cp.X = -0.5 + (Lh / 2) * caount
For j = r_x1 To r_x2



If Rot_check.Checked() = TrueThen
                        cp = RotatePoint(cp)
EndIf

IfMath.Sqrt((cp.X - 0.5) ^ 2 + (cp.Y - 0.5) ^ 2) < (CSng(R1.Text()) - Rc2) Then
                        iRf += 1
                        e.DrawEllipse(pen, 500 * (cp.X - Rc2) + 1, 500 * (cp.Y
- Rc2) + 1, Rc2 * 1000, Rc2 * 1000)
For k = 0 To Repeat.Text() - 1
                            cp2 = cp
                            cp2.X += k
                            cList.Add(cp2)
Next

EndIf
                    Lh = Math.Sqrt((2 * Math.PI * (Rc2 ^ 2)) / (F2.Text() *
Math.Sqrt(3)))
                    cp.X = cp.X + Lh
Next
                caount = caount - 1
                Hh = (Lh * Math.Sqrt(3)) / 2
Next
            Lh = Math.Sqrt((2 * Math.PI * (Rc2 ^ 2)) / (F2.Text() *
Math.Sqrt(3)))
            Hh = (Lh * Math.Sqrt(3)) / 2
            cp.Y = 0.5
            cp.X = -0.5
            caount = -1
For i = r_y1 To r_y2

                cp.Y = cp.Y - Hh
                cp.X = -0.5 + (Lh / 2) * caount
For j = r_x1 To r_x2



If Rot_check.Checked() = TrueThen
                        cp = RotatePoint(cp)
EndIf

IfMath.Sqrt((cp.X - 0.5) ^ 2 + (cp.Y - 0.5) ^ 2) < (CSng(R1.Text()) - Rc2) Then
                        iRf += 1
```

```
                        e.DrawEllipse(pen, 500 * (cp.X - Rc2) + 1, 500 * (cp.Y
- Rc2) + 1, Rc2 * 1000, Rc2 * 1000)
For k = 0 To Repeat.Text() - 1
                            cp2 = cp
                            cp2.X += k
                            cList.Add(cp2)
Next

EndIf
                    Lh = Math.Sqrt((2 * Math.PI * (Rc2 ^ 2)) / (F2.Text() *
Math.Sqrt(3)))
                    cp.X = cp.X + Lh
Next
                caount = caount - 1
                Hh = (Lh * Math.Sqrt(3)) / 2
Next
EndIf

If align_square.Checked() = TrueThen
            r_x1 = -(0.5 / Rc) : r_x2 = -r_x1
            r_y1 = -(0.7 / Rc) : r_y2 = -r_y1
            r_step = 0.5

If strict.Checked() = FalseThen
                r_x1 = 0 : r_x2 = 1 / Rc
                r_y1 = 0 : r_y2 = 1.4 / Rc
                r_step = 1.5
EndIf

For i = r_x1 To r_x2

For j = r_y1 To r_y2

                    cp.X = r_step - Rc * (2 * i)
                    cp.Y = r_step - Rc * (2 * j)
If Rot_check.Checked() = TrueThen
                        cp = RotatePoint(cp)
EndIf

IfMath.Sqrt((cp.X - 0.5) ^ 2 + (cp.Y - 0.5) ^ 2) < (CSng(R1.Text()) - Rc2) Then
                        iRf += 1
                        e.DrawEllipse(pen, pb.Width() * (cp.X - Rc2) + 1,
pb.Height() * (cp.Y - Rc2) + 1, Rc2 * pb.Width() * 2, Rc2 * pb.Height() * 2)
For k = 0 To Repeat.Text() - 1
                            cp2 = cp
                            cp2.X += k
                            cList.Add(cp2)
Next
EndIf
Next
Next
EndIf

        Real_f.Text() = (100 * (iRf * Rc2 * Rc2 * Math.PI) / (H.Text() *
H.Text())).ToString("0.0000")
        Real_N.Text() = iRf.ToString()
        r.Text() = Rc2.ToString("0.0000")

If write = TrueThen WriteMesh(cList, Rc2, nm)
EndSub
```

Appendix B

OpenFOAM's boundary conditions for pressure and velocity for flow through porous media are listed:

Velocity:

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                      |                                            |
| \\      /  F ield              | OpenFOAM: The Open Source CFD Toolbox      |
|  \\    /   O peration          | Version:  4.0                              |
|   \\  /    A nd                | Web:      www.OpenFOAM.org                 |
|    \\/     M anipulation       |                                            |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volVectorField;
    object      U;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

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

internalField   uniform (0 0 0);

boundaryField
{
    left
    {
     type               zeroGradient;
    }

    right
    {
        type            zeroGradient;
    }

    top
    {
        type            cyclicAMI;
    }

    bottom
    {
        type            cyclicAMI;
    }

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

        front
        {
                type                    empty;
        }

        back
        {
                type                    empty;
        }
}

// ************************************************************************* //
```

Pressure:

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  4.0                                   |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      p;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

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

internalField   uniform 0;

boundaryField
{
    left
    {
        type            fixedValue;
        value           uniform 1;
    }

    right
    {
        type            fixedValue;
        value           uniform 0;
    }

    top
    {
        type            cyclicAMI;
    }

    bottom
    {
        type            cyclicAMI;
    }

    internal
    {
        type            zeroGradient;
    }

        front
        {
                type                    empty;
        }

        back
        {
                type                    empty;
        }
}

// ************************************************************************* //
```

Appendix C

Examing spatial grid convergence with the help of the following FORTRAN program:

```
!-- File  verify.f90  ----------------------------------------------
-------
!
!  Performs several verification calculations given a file of grid
spacings
!  and some observed quantity corresponding to each grid spacing.
!
!  Computes:
!  - order of convergence
!  - Richardson extrapolation to zero grid spacing
!  - grid convergence indices (GCI)
!
!---------------------------------------------------------------------
------

   PROGRAM verify

IMPLICIT NONE

   INTEGER :: n
   INTEGER :: nd

   INTEGER, PARAMETER :: ndmax = 10

   REAL :: ratio
   REAL :: fexact
   REAL :: fsafe
   REAL :: p

   REAL, DIMENSION(ndmax) :: f
   REAL, DIMENSION(ndmax) :: gcif
   REAL, DIMENSION(ndmax) :: r
   REAL, DIMENSION(ndmax) :: x

!---------------------------------------------------------------------
------

!..Write Header.

   WRITE(*,*) ' '
   WRITE(*,*) '--- VERIFY: Performs verification calculations ---'

!..Read in the file, determine the number of data points, and output.

   DO n = 1, ndmax
     READ(*,*,end=10) x(n), f(n)
   ENDDO

10 CONTINUE
```

```
   nd = n - 1

   WRITE(*,*) ' '
   WRITE(*,*) 'Number of data sets read = ', nd

   WRITE(*,*) ' '
   WRITE(*,*) '     Grid Size     Quantity'
   WRITE(*,*) ' '

   DO n = 1, nd
     WRITE(*,'(2(F14.6))') x(n), f(n)
   ENDDO

!..Compute the grid refinement ratio, r, between each pair.

   DO n = 1, nd-1
     r(n) = x(n+1) / x(n)
   ENDDO

!..Estimate the order of convergence using the first three data pairs
!..and assuming that the grid refinement ratio is constant, r(1) =
r(2).
!..This is done using Eqn. 5.10.6.1 of (Roache, 1998).

   p = log( ( f(3) - f(2) ) / ( f(2) - f(1) ) )  /  log( r(1) )

   WRITE(*,*) ' '
   WRITE(*,*) 'Order of convergence using first three finest grid '
   WRITE(*,*) 'and assuming constant grid refinement (Eqn. 5.10.6.1)
'
   WRITE(*,*) 'Order of Convergence, p = ', p

!..Perform Richardson extrapolation to estimate a zero grid value of
f.

   fexact = f(1) + ( f(1) - f(2) ) / ( r(1)**p - 1.0 )

   WRITE(*,*) ' '
   WRITE(*,*) 'Richardson Extrapolation: Use above order of
convergence'
   WRITE(*,*) 'and first and second finest grids (Eqn. 5.4.1) '
   WRITE(*,*) 'Estimate to zero grid value, f_exact = ', fexact

!..Compute Grid Convergence Index (GCI) for each fine grid using Eqn.
5.6.1
!..from Roache's book. Use factor of safety as recommended on page
123.

   IF ( nd .ge. 3 ) then
     fsafe = 1.25
   ELSE
     fsafe = 3.0
   ENDIF

   DO n = 1, nd-1
     gcif(n) = fsafe * ( abs( f(n+1) - f(n) ) / f(n) ) / ( r(n)**p -
1.0 )
   ENDDO
```

```
   WRITE(*,*) ' '
   WRITE(*,*) 'Grid Convergence Index on fine grids. Uses p from
above.'
   WRITE(*,*) 'Factor of Safety = ', fsafe
   WRITE(*,*) ' '
   WRITE(*,*) '  Grid     Refinement            '
   WRITE(*,*) '  Step      Ratio, r       GCI(%)'
   DO n = 1, nd-1
     WRITE(*,'(2x,i2,i3,2(f14.6))') n, n+1, r(n), gcif(n)*100.0
   ENDDO

!..Examine if asymptotic range has been achieved by examining ratio
!..of Eqn. 5.10.5.2 of Roache's book is one.

   IF ( nd .ge. 3 ) then

     WRITE(*,*) ' '
     WRITE(*,*) 'Checking for asymptotic range using Eqn. 5.10.5.2.'
     WRITE(*,*) 'A ratio of 1.0 indicates asymptotic range.'
     WRITE(*,*) ' '
     WRITE(*,*) ' Grid Range    Ratio'
     DO n = 1, nd-2
       ratio = r(n)**p * gcif(n) / gcif(n+1)
       WRITE(*,'(2x,i1,i1,i2,i1,f14.6)') n, n+1, n+1, n+2, ratio
     ENDDO

   ENDIF


!..Write Trailer.

   WRITE(*,*) ' '
   WRITE(*,*) '--- End of VERIFY ---'
   WRITE(*,*) ' '

!-----------------------------------------------------------------------
------

    END PROGRAM verify
```

Appendix D

Volume Fraction		Porosity		q[m^2/s]	K[m^2]
Φ1	Φ2	φ_i	φ_t		
0.7	0.4125	0.3	0.5875	0.000268	0.000268
0.7	0.37125	0.3	0.62875	0.000238	0.000238
0.7	0.33	0.3	0.67	0.000252	0.000252
0.7	0.28875	0.3	0.71125	0.000267	0.000267
0.7	0.2475	0.3	0.7525	0.000288	0.000288
0.7	0.20625	0.3	0.79375	0.00031	0.00031
0.7	0.165	0.3	0.835	0.000336	0.000336
0.7	0.12375	0.3	0.87625	0.00037	0.00037
0.7	0.0825	0.3	0.9175	0.000426	0.000426
0.7	0.04125	0.3	0.95875	0.00059	0.00059
0.65	0.4125	0.35	0.5875	0.000556	0.000556
0.65	0.37125	0.35	0.62875	0.0005	0.0005
0.65	0.33	0.35	0.67	0.000519	0.000519
0.65	0.28875	0.35	0.71125	0.000539	0.000539
0.65	0.2475	0.35	0.7525	0.000567	0.000567
0.65	0.20625	0.35	0.79375	0.000596	0.000596
0.65	0.165	0.35	0.835	0.000629	0.000629
0.65	0.12375	0.35	0.87625	0.000671	0.000671
0.65	0.0825	0.35	0.9175	0.000737	0.000737
0.65	0.04125	0.35	0.95875	0.000801	0.000801
0.6	0.4125	0.4	0.5875	0.001012	0.001012
0.6	0.37125	0.4	0.62875	0.000924	0.000924
0.6	0.33	0.4	0.67	0.000948	0.000948
0.6	0.28875	0.4	0.71125	0.00098	0.00098
0.6	0.2475	0.4	0.7525	0.001011	0.001011
0.6	0.20625	0.4	0.79375	0.001047	0.001047
0.6	0.165	0.4	0.835	0.001088	0.001088
0.6	0.12375	0.4	0.87625	0.00115	0.00115
0.6	0.0825	0.4	0.9175	0.001219	0.001219
0.6	0.04125	0.4	0.95875	0.001277	0.001277
0.5	0.4125	0.5	0.5875	0.00262	0.00262
0.5	0.37125	0.5	0.62875	0.002449	0.002449
0.5	0.33	0.5	0.67	0.002495	0.002495
0.5	0.28875	0.5	0.71125	0.002538	0.002538
0.5	0.2475	0.5	0.7525	0.002585	0.002585
0.5	0.20625	0.5	0.79375	0.002637	0.002637
0.5	0.165	0.5	0.835	0.002698	0.002698

0.5	0.12375	0.5	0.87625	0.002793	0.002793
0.5	0.0825	0.5	0.9175	0.002883	0.002883
0.5	0.04125	0.5	0.95875	0.002917	0.002917
0.4	0.4125	0.6	0.5875	0.005643	0.005643
0.4	0.37125	0.6	0.62875	0.005378	0.005378
0.4	0.33	0.6	0.67	0.005428	0.005428
0.4	0.28875	0.6	0.71125	0.005588	0.005588
0.4	0.2475	0.6	0.7525	0.005549	0.005549
0.4	0.20625	0.6	0.79375	0.005622	0.005622
0.4	0.165	0.6	0.835	0.005736	0.005736
0.4	0.12375	0.6	0.87625	0.005831	0.005831
0.4	0.0825	0.6	0.9175	0.005949	0.005949
0.4	0.04125	0.6	0.95875	0.006035	0.006035
0.78	0.04125	0.22	0.95875	0.000237	0.000237
0.77	0.0825	0.23	0.9175	0.000195	0.000195
0.77	0.04125	0.23	0.95875	0.000256	0.000256
0.75	0.04125	0.25	0.95875	0.000304	0.000304
0.78	0.12375	0.22	0.87625	0.000134	0.000134
0.78	0.4125	0.22	0.5875	4.99E-05	4.99E-05
0.76	0.20625	0.24	0.79375	0.000124	0.000124
0.77	0.165	0.23	0.835	0.000125	0.000125
0.78	0.28875	0.22	0.71125	6.39E-05	6.39E-05
0.76	0.04125	0.24	0.95875	0.000278	0.000278

Appendix E

φ_i	φ_t	K_s/K_{tow}	K_p	$K_p{K_s}^{1/2}({K_{tow}}^{3/2})$
0.22	0.95875	0.00057	0.0002344	0.045762
0.22	0.87625	0.001302	0.0001078	0.07271
0.22	0.71125	0.003997	3.762E-05	0.136429
0.23	0.95875	0.007921	0.0002754	0.200431
0.22	0.5875	0.00911	1.738E-05	0.216862
0.23	0.9175	0.012544	0.0001791	0.25978
0.23	0.835	0.024694	9.519E-05	0.381432
0.24	0.95875	0.027985	0.0002994	0.409626
0.25	0.95875	0.06493	0.0003192	0.665314
0.24	0.79375	0.117526	8.023E-05	0.944765
0.3	0.95875	0.682155	0.0004035	2.984972
0.3	0.9175	1.085843	0.0002876	4.273087
0.3	0.87625	1.580992	0.0002247	5.865134
0.3	0.835	2.126706	0.0001886	7.679019
0.35	0.95875	2.530417	0.0005788	9.070759
0.3	0.79375	2.838362	0.0001618	10.16262
0.3	0.7525	3.741495	0.0001422	13.51812
0.35	0.9175	4.050517	0.0004637	14.71843
0.3	0.71125	4.926474	0.0001272	18.2619
0.35	0.87625	5.632072	0.0004086	21.26363
0.4	0.95875	6.236452	0.0009259	23.93576
0.3	0.67	6.416612	0.000116	24.7499
0.35	0.835	7.699953	0.0003699	30.77568
0.3	0.62875	8.330832	0.0001073	33.8789
0.4	0.9175	10.19268	0.0008063	43.55049
0.35	0.79375	10.42256	0.0003419	44.79595
0.3	0.5875	11.00703	0.0001001	48.01146
0.35	0.7525	13.63707	0.0003229	63.31831
0.4	0.87625	14.38576	0.0007499	67.91567
0.35	0.71125	17.85583	0.0003081	90.52206
0.4	0.835	19.93829	0.000711	105.0596
0.35	0.67	23.14434	0.000297	128.7706
0.4	0.79375	26.12571	0.0006866	152.1872
0.5	0.95875	26.2785	0.0023419	153.4214
0.35	0.62875	30.40636	0.0002879	187.9706
0.4	0.7525	34.68311	0.0006669	226.1139
0.35	0.5875	40.00786	0.0002809	276.7335
0.5	0.9175	41.94421	0.0022403	295.9608

0.4	0.71125	45.13904	0.0006528	328.6014
0.5	0.87625	58.89765	0.0021901	481.4108
0.4	0.67	59.24862	0.0006414	485.5551
0.4	0.62875	77.44826	0.0006328	715.8553
0.5	0.835	81.92664	0.0021544	776.9156
0.6	0.95875	84.02536	0.0054371	806.0963
0.4	0.5875	101.4483	0.000626	1061.67
0.5	0.79375	107.653	0.0021322	1158.179
0.6	0.9175	130.2571	0.0053554	1532.485
0.5	0.7525	140.6143	0.0021153	1715.282
0.5	0.71125	184.3047	0.0021021	2557.796
0.6	0.87625	188.3249	0.0053084	2640.784
0.5	0.67	243.8159	0.0020915	3872.233
0.6	0.835	253.9616	0.0052807	4113.826
0.5	0.62875	315.0653	0.002084	5667.636
0.6	0.79375	344.2198	0.0052594	6465.388
0.5	0.5875	415.6598	0.0020776	8562.188
0.6	0.7525	450.8791	0.005245	9665.845
0.6	0.71125	580.2121	0.0052345	14081.77
0.6	0.67	757.1665	0.0052257	20957.57
0.6	0.62875	1005.113	0.0052186	32009.62
0.6	0.5875	1334.867	0.0052131	48939.3

Appendix F

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  4.0                                   |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volVectorField;
    object      U;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

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

internalField   uniform (0 0 0);

boundaryField
{
    left
    {
        type            zeroGradient;
    }
    right
    {
        type            zeroGradient;
    }

    top
    {
        type            cyclicAMI;
    }

    bottom
    {
        type            cyclicAMI;
    }

    internal
    {
        type flowRateInletVelocity;
        volumetricFlowRate  -0.000055;
    value       uniform (0 0 0); // placeholder
    }

        front
        {
                type                    empty;
        }

        back
        {
                type                    empty;
        }
}

// ************************************************************************* //
```

Appendix G

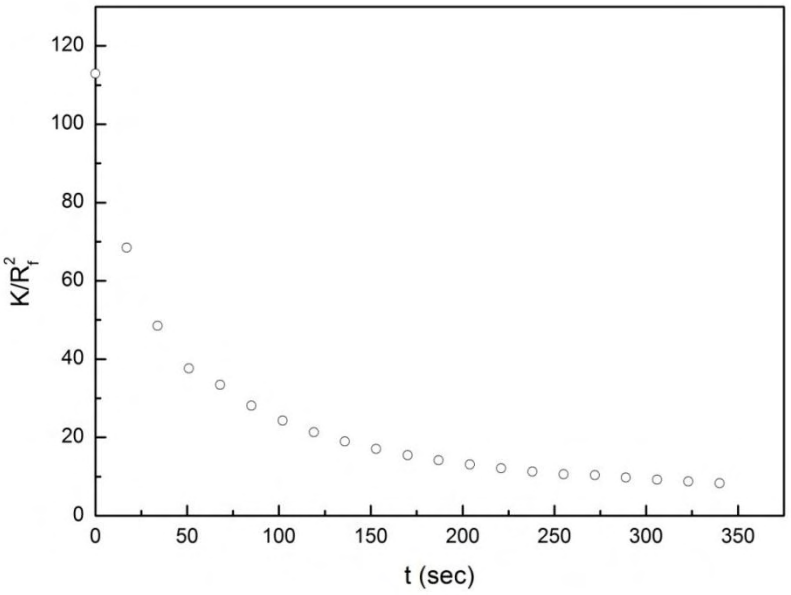


Figure 34: Numerical experiments for the dimensionless permeability ($K/{R_f}^2$) of SQ arrays plotted versus time. The percent of inflow absorbed by the fibers is 5% and bundle's porosity is 0.7.

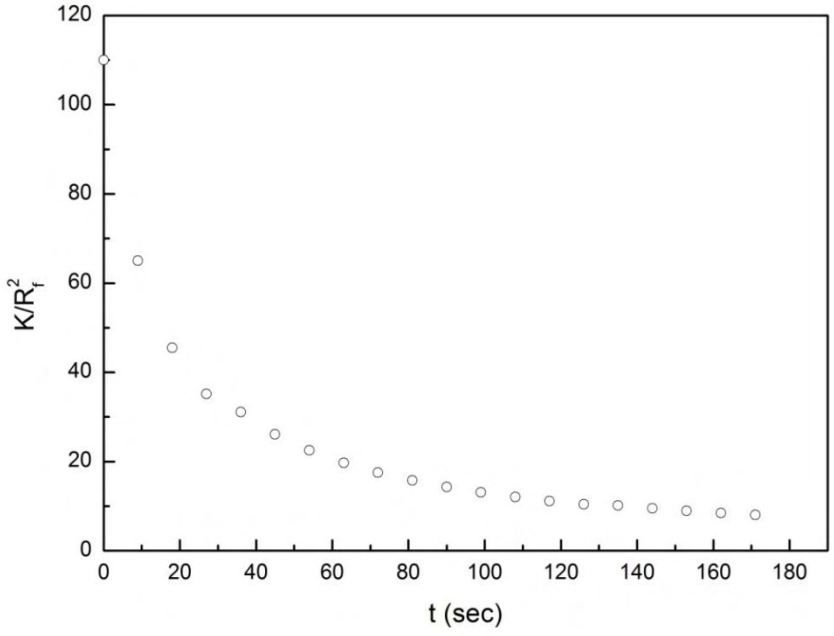


Figure 35: Numerical experiments for the dimensionless permeability ($K/{R_f}^2$) of SQ arrays plotted versus time. The percent of inflow absorbed by the fibers is 5% and bundle's porosity is 0.7.

References

Fetter, C. W. (2000). *Applied hydrogeology*: Prentice hall.

Greenshields, C. J. (2016). Openfoam user guide. *OpenFOAM Foundation Ltd, version* Retrieved 19 January, 2017, from http://foam.sourceforge.net/docs/Guides-a4/OpenFOAMUserGuide-A4.pdf

Mantanis, G., Young, R., & Rowell, R. (1995). Swelling of compressed cellulose fiber webs in organic liquids. *Cellulose, 2*(1), 1-22.

Martins, A. A., Laranjeira, P. E., Braga, C. H., & Mata, T. M. (2009). Modeling of transport phenomena in porous media using network models. *Progress in Porous Media Research, ISBN*, 978-971.

Masoodi, R., & Pillai, K. M. (2010). Darcy's law-based model for wicking in paper-like swelling porous media. *AIChE journal, 56*(9), 2257-2267.

Ngo, N., & Tamma, K. (2004). Complex three-dimensional microstructural permeability prediction of porous fibrous media with and without compaction. *International journal for numerical methods in engineering, 60*(10), 1741-1757.

Papathanasiou, T. (2001). Flow across structured fiber bundles: a dimensionless correlation. *International journal of multiphase flow, 27*(8), 1451-1461.

Roache, P. J. (1998). *Verification and validation in computational science and engineering* (Vol. 895): Hermosa Albuquerque, NM.

Slater, J. W. (2006). Examining spatial (grid) convergence. *Public tutorial on CFD verification and validation, NASA Glenn Research Centre, MS* Retrieved 20 March, 2017, from https://www.grc.nasa.gov/www/wind/valid/tutorial/spatconv.html

Szalay, J. (2015). Inflammation: Causes, Symptoms & Anti-Inflammatory Diet Retrieved 26 February, 2017, from https://www.livescience.com/52344-inflammation.html

Vafai, K. (2015). *Handbook of porous media*: Crc Press.

Zhang, C., Simon, T. W., Li, P. Y., & Van de Ven, J. D. (2015). Numerical modeling of three-dimensional heat transfer and fluid flow through interrupted plates using unit cell scale. *Special Topics & Reviews in Porous Media: An International Journal*, *6*(2).