Geometric Optimisation of Heat Transfer in Channels using Newtonian and Non-Newtonian Fluids

Marc Darren Stocks

2012

Geometric Optimisation of Heat Transfer in Channels using Newtonian and Non-Newtonian Fluids

by

Marc Darren Stocks

Submitted in partial fulfilment of the requirements for the degree

Masters in Engineering

in the

Faculty of Engineering, Built Environment and Information Technology

University of Pretoria

2012-12-05

Abstract

Title: Geometric optimisation of heat transfer in complex geometries using Newtonian and non-Newtonian fluids

Author: MD Stocks

Supervisors: Prof T Bello-Ochende and Prof JP Meyer

Department: Mechanical and Aeronautical Engineering

University: University of Pretoria

Degree: Masters in Engineering (Mechanical Engineering)

The continual advance in manufacturing processes has resulted in significantly more compact, high performance, devices. Consequently, heat extraction has become the limiting factor, and of primary concern. Therefore, a substantial amount of research has been done regarding high efficiency micro heat exchangers, employing novel working fluids.

This dissertation numerically investigated the thermal behaviour of microchannel elements cooled by Newtonian and non-Newtonian fluids, with the objective of maximising thermal conductance subject to constraints. This was done, firstly, for a two-dimensional simple microchannel, and secondly, for a three-dimensional complex microchannel. A numerical model was used to solve the governing equations relating to the flow and temperature fields for both cases. The geometric configuration of each cooling channel was optimised for Newtonian and non-Newtonian fluids, at a fixed inlet velocity and heat transfer rate. In addition, the effect of porosity on thermal conductance was investigated.

Geometric optimisation was employed to the simple and complex microchannels, whereby an optimal geometric ratio (height versus length) was found to maximise thermal conductance. Moreover, analysis indicated that the bifurcation point of the complex microchannel could be manipulated to achieve a higher thermal conductance.

In both cases, it was found that the non-Newtonian fluid characteristics resulted in a significant variation in thermal conductance as inlet velocity was increased. The

characteristics of a dilatant fluid greatly reduced thermal conductance on account of shear-thickening on the boundary surface. In contrast, a pseudoplastic fluid showed increased thermal conductance.

A comparison of the simple and complex microchannel showed an improved thermal conductance resulting from greater flow access to the conductive area, achieved by the complex microchannel.

Therefore, it could be concluded that a complex microchannel, in combination with a pseudoplastic working fluid, substantially increased the thermal conductance and efficiency, as opposed to a conventional methodology.

Acknowledgements

I would hereby like to thank Prof T Bello-Ochende for his constant guidance, supervision, and positive feedback. You allowed me not only to explore the problem in my own way.

I would like to acknowledge Prof JP Meyer for his technical assistance and guidance throughout my post-grad studies.

The funding obtained from the NRF, TESP, University of Stellenbosch/University of Pretoria, SANERI/SANEDI, CSIR EEDSM Hub and NAC is acknowledged and duly appreciated.

I would like to thank for Aerotherm for their training and assistance, which set me on my way in making the most of the CFD software.

Lastly I would like to thank my Family, for their constant support and understanding, as well as willingness to help – without you it would not have been possible.

Table of Contents

List of Figures

List of Tables

Nomenclature

Latin symbols

a	face area vector	m^2
A	total channel area	m^2
Be	Bejan number	—
C	global thermal conductance	—
C_p	specific heat	$J/kg \cdot K$
D	diameter of channel	m
E	energy	J
F	external forces	N
H	total channel height	m
HB	Herschel-Bulkley	—
I	identity matrix	—
k	thermal conductivity	$W/m \cdot K$
K	consistency	$Pa \cdot s^{1/2}$
L	total channel length	m
$\dot{m}$	mass flow rate	kg/s
n	power-law index	—
P	pressure	Pa

Pr	Prandtl number	—
q	total heat transfer	W
Q	activation energy	J
r	radius	m
R	global thermal resistance	—
Re	Reynolds number	—
Sv	svelteness number	—
t	time	s
T	temperature	K
U	average velocity	m/s
u, v, w	velocity vectors	m/s
v_g	grid velocity	m/s
V	volume	m^3
X	position vector	m
x, y, z	Cartesian coordinates	

Greek symbols

α	bifurcation point ratio	—
$\dot{\gamma}$	rate of strain	—
η	dynamic shear viscosity	$Pa \cdot s$
θ	bifurcation angle	rad
λ	coefficient of bulk viscosity	—
μ	viscosity	$kg/m \cdot s$
ρ	density	kg/m^3
σ	stress tensor	Pa

τ	deviatoric/ shear stress	Pa
τ_p/τ_0	yield stress	Pa
φ	porosity ratio	$-$
ϕ	particle concentration	kg/m^3
Φ	dissipation function	$-$
ψ	particle radius	m

Superscripts

$\widetilde{\cdot}$	Dimensionless variable

Subscripts

c	channel
f	fluid
in	inlet
i,j	numerical indices
max	maximum
$Newt$	Newtonian
s	solid
t	total

Chapter 1   Introduction

1.1   Background

Advances in the field of electronic manufacturing and design have resulted in a drive to increase the density of electronic systems, resulting in a higher heat generation density [1]. This has ensued in a requirement for novel methods of extracting more heat from smaller systems and increasing the efficiency of existing heat extraction systems.

For more than a decade, this drive has resulted in a significant amount of research being done in the field of microchannel heat exchangers and microchannel materials [2–5] in order to maximise the heat transfer density rate.

The microchannel heat exchangers are merely small heat exchangers, which transfer heat from the given system to a working fluid. Heat can be transferred by means of three mechanisms, namely:

- Thermal conduction: Energy is transferred as a result of microscopic vibration and collision of particles within a body, as a result of a temperature gradient. Thermal conduction can take place within virtually any matter.
- Convection: Energy is transferred as a result of fluid movements. Convection cannot take place within a solid, as particle movement is restricted.

- Thermal radiation: Energy is transferred by means of electromagnetic waves, which require no medium to propagate. All matter with a temperature above that of absolute zero will emit thermal radiation.

In a microchannel heat exchanger heat, is predominantly transferred by means of conduction and convection.

More recently, researchers have become interested in the conductive and convective impact of using various unconventional working fluids, including nanofluids to enhance microchannel cooling capacity.

Some research has been done regarding the flow of non-Newtonian fluids in microchannels in a heat transfer environment. These are fluids of which the flow properties diverge from standard Newtonian fluids. The most prevalent case is where the viscosity of the fluid is dependent on the shear rate.

1.2   Scope of Study

This dissertation focuses on the use of non-Newtonian fluids as a working fluid in a microchannel, with the goal to geometrically optimise the microchannel so as to maximise thermal conductance. The possible advantages of a complex microchannel, whereby a dendritic flow structure is developed, are investigated and compared to those of a simple (conventional) microchannel.

The scope of study can be broken down into the following deliverables:

- Investigate non-Newtonian flow and heat transfer characteristics in a simple microchannel;
- Investigate geometric optimisation of a two-dimensional simple microchannel;
- Investigate non-Newtonian flow and heat transfer characteristics in a complex microchannel;
- Investigate optimisation of a two-dimensional complex microchannel;
- Investigate optimisation of a three-dimensional complex microchannel.

Chapter 2   Literature

2.1   Introduction

The aim of this chapter is to give the reader a brief overview of the key concepts related to the dissertation. Section 2.2 commences with a background of non-Newtonian fluids, including the various categories and the non-Newtonian governing equations. A brief explanation is given of the power-law model, used to describe non-Newtonian viscosity, as well as a description of the non-Newtonian Reynolds number. This is followed by a background of complex flow structures in Section 2.3, including a brief overview of constructal theory and the resulting outcomes. Then, finally microchannels and geometrical optimisation are discussed in Sections 2.4 and 2.5 respectively.

2.2   Non-Newtonian Fluids

Other than water and air, most fluids found in nature do not adhere to the standard Newtonian characteristics. Typical examples in the human body are blood and mucus. Many different examples can be found in nature, such as suspensions of clay (Kaolin), mud slurries, lava and snow. Various manmade substances exhibit non-Newtonian behaviour such as toothpaste, ketchup, paint and molten rubber [6].

A non-Newtonian fluid is defined as a fluid that does not exhibit standard behaviour with the application of a shear stress. Fluid particles build up a micro-structure, which may become so dominant as to affect the macro-structure of the fluid. In this manner, the micro-structure is responsible for the unique physical characteristics and a non-linear stress-strain or stress-shear relationship. Therefore, the slope of a shear stress versus shear rate curve is not constant, i.e. the viscosity of non-Newtonian fluids is not constant.

2.2.1   Categories of Non-Newtonian Fluids

Non-Newtonian fluids include fluids such as blood, blood plasma, syrup and Bingham plastic, which can be categorised according to the following rheological characteristics [6–9]:

- Shear-thinning
- Shear-thickening
- Thixotropic
- Rheopectic
- Viscoplasticity
- Viscoelasticity

2.2.1.1   Shear-thinning

Fluids such as ketchup or toothpaste exhibit shear-thinning characteristics, in which case once a stress is applied, the fluid viscosity decreases. As a result, the fluid may form peaks and drip in large drops when poured, but flows freely under pressure. Bingham plastics seem to have shear-thinning and viscoelastic characteristics (see Section 2.2.1.6).

2.2.1.2   Shear-thickening

The opposite case is also possible where a fluid might flow freely under a low stress condition, however, once pressure is applied, the fluid builds resistance. Typical examples are mixtures of water and corn starch (Maizena), kaolin (clay) or slurry. Once a shear force has been applied, the particles in suspension are forced together, increasing the friction and thus a microstructure is formed. The water is forced out and one is left with a fluid that can

withstand a significantly higher shear stress. In certain extreme cases, it may even be possible to walk across such a fluid.

2.2.1.3   Thixotropic

The fluid, which is found to be viscous and thick under normal conditions, thins with the application of shear stress over a time period, thereby resembling shear-thinning characteristics. Upon standing, the fluid regains its yield strength and becomes more viscous again. The yield strength of the fluid significantly increases with the time of rest before a shear force is added [10]. Scot-Blair (1992) states that if this recovery is very rapid, the phenomenon is observed as structural viscosity (shear-thinning); if slow, it is observed as thixotropic [11].

2.2.1.4   Rheopectic

This is a fluid in which the stress needs to be increased in order to maintain the specific strain rate, hence the opposite of a thixotropic fluid. With an increase in shear rate, the viscosity of the fluid decreases gradually [11].

2.2.1.5   Viscoplasticity

These fluids behave plastically up to a specific yield stress, thereafter viscous flow is experienced. The fluid exhibits both viscous (time dependent) and plastic (load history dependent) characteristics [12].

2.2.1.6   Viscoelasticity

Elastic materials instantaneously deform with an addition of stress, where viscous fluids materials resist shear. Thus in viscoelastic fluids, there is an instantaneous elastic response, but the deformation also increases with time [13]. They show viscous behaviour in constant shear [14], and then "relax" to elastic behaviour.

Non-Newtonian fluid effects are varied and unusual and highly dependent on the specific microstructure. Therefore, there are many different models of suspensions and polymeric fluids (highly elastic material laced with polymers), each encapsulating a specific observed

effect [6]. Consequently, many of these models are designed with only a set of characteristics in mind and are not generally applicable to non-Newtonian fluids.

2.2.2 Governing Equations

The continuum approximation, which underlies the models, assumes that the dimensions of the flow field are far greater than those of the microstructure [6].

Conservation of mass

Assuming the fluid is incompressible, when examining the rate of change of momentum within a volume of fluid, the conservation of mass will yield the following:

$$\nabla \cdot \boldsymbol{u} = 0 \tag{2.1}$$

where $\boldsymbol{u}$ indicates the velocity field.

Momentum

Manipulating the momentum equation yields:

$$\rho \frac{D\boldsymbol{u}}{D_t} = -\nabla \cdot \boldsymbol{\sigma} + \boldsymbol{F} \tag{2.2}$$

where $\frac{D}{D_t} = \frac{\partial}{\partial_t} + \boldsymbol{u} \cdot \nabla$, ρ is the fluid density, $\boldsymbol{\sigma}$ the stress tensor and $\boldsymbol{F}$ external forces.

In the case of incompressible fluids, the stress tensor can be split into an isotropic part and pressure field.

$$\boldsymbol{\sigma} = -p\boldsymbol{I} + \boldsymbol{\tau} \tag{2.3}$$

Thus the momentum equation becomes:

$$\rho \frac{D\boldsymbol{u}}{D_t} = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \boldsymbol{F} \tag{2.4}$$

However, this is identical to that of a Newtonian fluid, therefore, in order to account for the non-Newtonian characteristics, the deviatoric stress tensor τ_{ij} is related to the fluid

properties. This is often done by relating the deviatoric stress to the rate of strain tensor $\dot{\gamma}_{ij}$ defined in [6] as:

$$\dot{\boldsymbol{\gamma}} = \nabla\mathbf{u} + (\nabla\mathbf{u})^{\mathrm{T}} \quad \text{or} \quad \dot{\gamma}_{ij} = \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \tag{2.5}$$

Some authors may add slight variations or include additional elements such as temperature, pressure, the strain tensor γ_{ij}, or particulate concentration.

Energy

In the case where the fluid is temperature dependent and where temperature may change, an energy equation is required. This is very important when modelling certain fluids such as snow (ice) or syrup, which may be Newtonian under fixed conditions but their viscosities are severely affected by temperature. Assuming thermal expansion is negligible, the energy equation is found to be:

$$\rho c \frac{DT}{Dt} = \frac{1}{2}\tau_{ij}\dot{\gamma}_{ij} + \nabla \cdot (\kappa\nabla T) \tag{2.6}$$

where c, κ and T is specific heat, conductivity and Temperature respectively. Any additional minor heat sources have been ignored, thus yielding an energy equation that describes the temperature field which forms as a result of advection, diffusion and frictional heating. This may affect the fluid microstructure, material properties, and consequently, fluid flow, according to the constitutive law.

2.2.3   Constitutive Models

In a classic Newtonian fluid, the deviatoric stress is linearly proportional to the rate of strain, where the coefficient of proportionality is viscosity:

$$\therefore \tau_{ij} = \mu\dot{\gamma}_{ij} \tag{2.7}$$

This, however, is not the case in a non-Newtonian fluid, where the shear stress is not directly proportional to the strain rate. In this case, the shear stress is determined by the microstructure of the fluid. Hence a more complex model is required, which is capable of

reflecting the macroscopic effects produced by the microstructure. Several different methods can be found in literature, the following are four different methods [6]:

- **Theoretical "kinetic" approach**

In this approach, a model of the molecular anatomy of the fluid is assembled and a kinetic theory built for the fluid microstructure. This can be done by investigating the flow around an idealised model polymer or emulsion droplet and the generation of constitutive equations for a dilute suspension through averaging procedures. Other possible methods include the representation of the fluid microstructure as a network of interacting elements. This kinetic method has, however, only recently become possible, yet it is still only applicable in simple fluids. A further drawback is that the mathematics is based upon physical approximations instead of asymptotic analysis [6].

- **Phenomenological approach**

This approach makes use of a model equation that describes how one would imagine the fluid microstructure will affect flow. This method was the first to be used in non-Newtonian fluids [6].

- **Improved phenomenological approach**

This approach was introduced in an attempt to improve upon the phenomenological approach. The method requires the writing down of the simplest kind of constitutive model that possesses the symmetries of the fluid[6].

- **Practical approach**

The fourth approach is experimental. One is to perform various experiments on the fluid and postulate a plausible stress-strain relationship. This is thus an empirical approach, which focuses merely on the macroscopic effects of the fluid, disregarding the microstructure. The advantage is that a reasonably accurate approximation can be obtained

from complex fluids, which would otherwise be exceedingly difficult or even impossible to model. The major drawback is, however, that the model is derived according to specific experimental conditions [6].

2.2.4   Temperature Dependence

Many non-Newtonian fluid microstructures are temperature dependent, thus the simplest manner to account for this is to define the viscosity as a function of temperature, such as the Arrhenius exponential dependence [6]:

$$\mu(T) = \mu_* \exp\left(\frac{Q}{RT}\right) \tag{2.8}$$

where μ_* is the viscosity at a reference temperature, R the universal gas law and Q the activation energy. Liquids are said to be strong if the viscosity is Arrhenius, and fragile if the viscosity is faster-than-Arrhenius [15].

2.2.5   Concentration Dependence

In suspensions, the concentration of the particles within the fluid may have a major impact on the microstructure, and therefore, the viscosity of the fluid. A fluid may in fact be Newtonian at lower concentrations, however, as the concentration is increased, it becomes non-Newtonian [16]. The Einstein relationship estimates the viscosity of the fluid in a dilute suspension of rigid spheres and solvent viscosity μ_o. Subsequently, the Einstein-Roscoe relation was developed in an attempt to extend the relation to mediums with higher concentrations [6]. The Einstein and Einstein-Roscoe relations are given below [6], [17]:

$$\mu = \mu_o \left(1 + \frac{5}{2}\emptyset\right) \tag{2.9}$$

$$\mu = \mu_o \left(1 - \frac{\emptyset}{\emptyset_{max}}\right)^{-a} \tag{2.10}$$

where $\emptyset$ and $\emptyset_m$ indicate particle concentration and maximum packing fraction, respectively, and a used for empirical matching.

Particle concentration has an additional effect on the yield strength of a viscoplastic fluid. It was noted that particles migrate away from areas with higher shear rates, thus the fluid at the centre of a channel will tend to have a lower viscosity than the walls where higher shear is experienced. It was found that very dilute polymer solutions can cause damping of the radial and cross-flow normal stresses and increase in the axial normal stress very close to the wall [6], [18]. In a cross-section, fluid with a lower viscosity will migrate towards the outer wall, while the high viscosity fluid remains in the centre. This produces a self-lubricating action [19].

In order to model concentration variations, the following conservation equation can be used:

$$\frac{D\emptyset}{Dt} + \nabla \cdot \left(J_c + J_\mu \right) = 0 \tag{2.11}$$

$$J_c = -K_c \psi^2 \emptyset \nabla(\emptyset \dot{\gamma}) \tag{2.12}$$

$$J_\mu = -K_\mu \dot{\gamma} \emptyset^2 \frac{a^2}{\mu} \frac{d\mu}{d\emptyset} \nabla \emptyset \tag{2.13}$$

Where J_c and J_μ are fluxes resulting from particle collision and spatially varying viscosity. K_c and K_μ are experimental constants and ψ particle radius.

2.2.6   Hysteresis

Hysteresis is a phenomenon often overlooked [6]. As the fluid microstructure changes, it does not necessarily reform in exactly the same manner again, and is said to disintegrate. This occurs in thixotropic fluid when shear-thinning occurs and the microstructure breaks down, it will not necessarily reform in the same manner, thus a hysteresis loop [11] occurs. Sudden changes to a fluid (flow rate or application of stress) may induce hysteresis in a fluid [19]. Hysteresis may also occur if the yield strength itself is time dependent [6].

2.2.7 Power-law Model

Non-Newtonian fluids are commonly referred to as power-law fluids, as a result of the Ostwald de Waele power-law model [6], [20], [21], which attempts to describe the viscosity as a power function as follows:

$$\mu(\dot{\gamma}) = K\dot{\gamma}^{n-1} \tag{2.14}$$

where K is the consistency and n the index. The higher the value of K, the more viscous the fluid. The index provides a measure for deviation from Newtonian behaviour, as the further the index is from unity, the more pronounced the non-Newtonian characteristics of the fluid will be.

- $n = 1$:   Reverts to Newtonian fluid
- $n > 1$:   Shear-thickening (viscosity increases with deformation)
- $n < 1$:   Shear-thinning

This empirical model is useful to fit to data, however it does not capture the results of yield stress, as would be found in the case of a viscoplastic fluid. One of the most common models that incorporate yield stress is the Herschel-Bulkley model [6], [22], given as:

$$\tau_{ij} = \left(K\dot{\gamma}^{n-1} + \frac{\tau_p}{\dot{\gamma}}\right)\dot{\gamma}_{ij} \qquad \text{For } \tau \geq \tau_p \tag{2.15}$$

$$\dot{\gamma} = 0 \qquad \text{For } \tau < \tau_p \tag{2.16}$$

where the parameter τ_p is the yield stress. The Bingham fluid model can be obtained by substituting $n = 1$ into equation (2.15). In this model, once the yield stress is overcome, the fluid flows as a Newtonian fluid, with strain rate proportional to the difference between the applied stress and yield stress [6].

2.2.8　　　Reynolds Number

The Reynolds number of a fluid is the ratio between inertial and viscous forces, and gives an indication of the particular flow regime of the fluid. The Reynolds number for a Newtonian fluid can readily be calculated using the following equation [23]:

$$Re_{Newt} = \frac{\rho D \bar{u}}{\eta_{Newt}}$$

(2.17)

where η is the dynamic shear viscosity, ρ the density, D the diameter and $\bar{u}$ the average velocity.

However, as the viscosity of non-Newtonian fluids is not constant, the Reynolds number for a non-Newtonian fluid is more complex to calculate. As a result, generalised equations were introduced to account for the viscosity characteristics.

Metzner and Reed [7], [15], [16] introduced a generalized Reynolds number for a pure power-law fluid:

$$Re_{genPowerLaw} = \frac{\rho D^n \bar{u}^{2-n}}{K \left(\frac{3n+1}{4n}\right)^n 8^{n-1}}$$

(2.18)

where K is the prefactor or consistency of the power-law. The equations for calculating a more complex viscosity relation, such as the Herschel-Bulkley relations, becomes exceedingly more complex. The generalised Reynolds number for laminar and fully developed duct flow, for non-Newtonian fluids with a viscosity relation following the Herschel-Bulkley equation, was determined to be [23]:

$$Re_{generalHB} = \frac{\rho D^n \bar{u}^{2-n}}{\left(\frac{\tau_0}{8}\left(\frac{D}{\bar{u}}\right)^n + K \left(\frac{3m+1}{4m}\right)^n 8^{n-1} + \frac{\eta_\infty (3m+1)}{4m}\left(\frac{D}{\bar{u}}\right)^{n-1}\right)}$$

(2.19)

with

12

$$m = \frac{nK\left(\frac{8\overline{u}}{D}\right)^n + \eta_\infty\left(\frac{8\overline{u}}{D}\right)}{\tau_0 + K\left(\frac{8\overline{u}}{D}\right)^n + \eta_\infty\left(\frac{8\overline{u}}{D}\right)} \qquad (2.20)$$

And η_∞ the constant viscosity in the very high shear rate range.

2.3　　　Complex Flow Structures

A complex flow structure attempts to break away from the standard flow design in an attempt to mimic natural design found in biological bodies and systems [24] [25]. The main objective is to promote higher flow access for maximising heat transfer while minimising flow resistance. The design movement is towards dendritic flow structures. These tree-type flow structures present the greatest relation between the volume that must be bathed and the port through which the stream enters or exits the volume [24]. Tree-type designs were proposed by constructal theory and have become useful in various applications, such as electronic cooling where space is limited.

2.3.1　　　Constructal Theory

Constructal theory is a principle-based method for constructing mechanisms in order to achieve their objectives optimally [26]. This is done by optimising volume shape, starting with the smallest element and progressing further. Constructal theory attempts to find this optimum by making use of various different methods. It is, therefore, not a method in itself, but a mind-set where the construction is not a given, but an outcome. In this manner, it has been proven that nature tends to the optimal solution. It is defined as: "For a flow system to persist in time (to survive) it must evolve in such a way that it provides easier and easier access to the currents that flow through it" [26].

The svelteness number is an indication of the performance of a flow system, defined as follows [27], [28]:

$$Sv = \frac{external\ flow\ length\ scale}{internal\ flow\ lenght\ scale}$$

(2.21)

Sv is a property of the global flow architecture. When Sv exceeds a number of 10, local losses become negligible in comparison with the distributed losses. These losses include junctions, bifurcations and bends.

As the constructal law and dendritic flow system attempts to find the optimal solution between various design parameters, priorities are placed upon the design variables. It is found to be better to optimise for a small flow resistance with a modest flow-uniformity and heat transfer. This, however, is application specific.

2.3.2 Bifurcation

By utilising Constructal theory, an optimal ratio can be found for the tube diameters and lengths as well as an optimal angle of bifurcation. The optimal ratio equates to $D1/D2 = 2^{1/3}$ and $L1/L2 = 2^{1/3}$ for laminar fully developed flow and an optimal angle of 75^0, see Figure 2.1 [24], [29], [30].
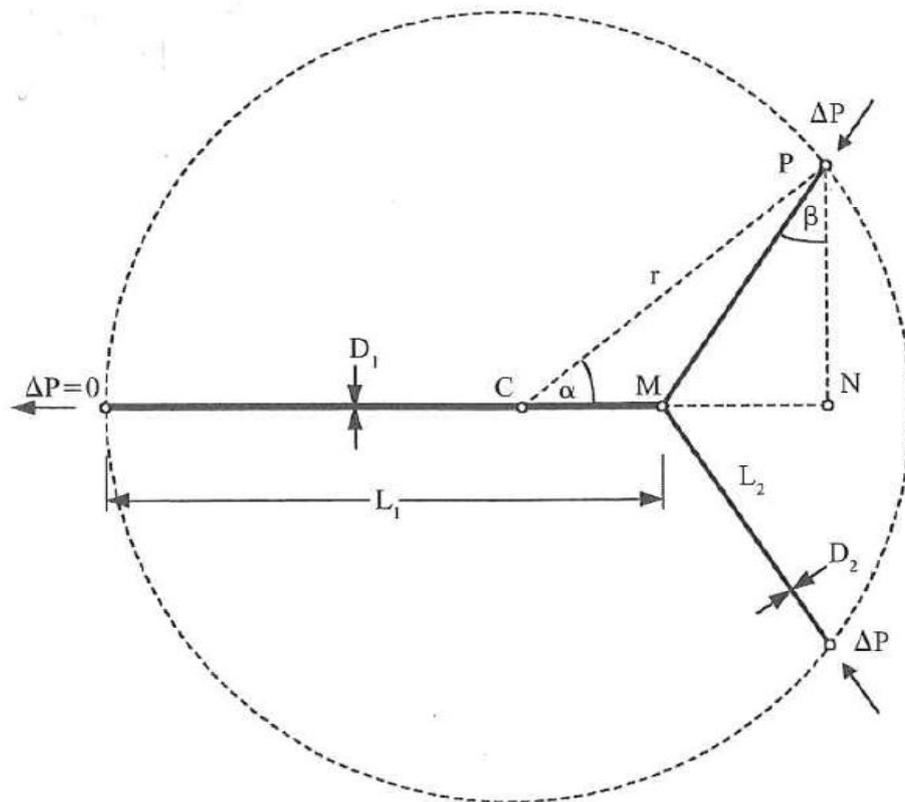
Figure 2.1 - Y-shaped construct of round tubes [24]

In a study done on the femoral arteries of hamsters [31], it was observed that the angles of multiple junctures along an artery will differ as depicted in Figure 2.2. The angle was obtuse

at the first juncture and progressively more acute for subsequent junctions. In the study, it was, however, not possible to find an optimal mathematical solution that matched the natural design. Changing viscosity, as well as the energy equation used, was deemed to be the main cause thereof.
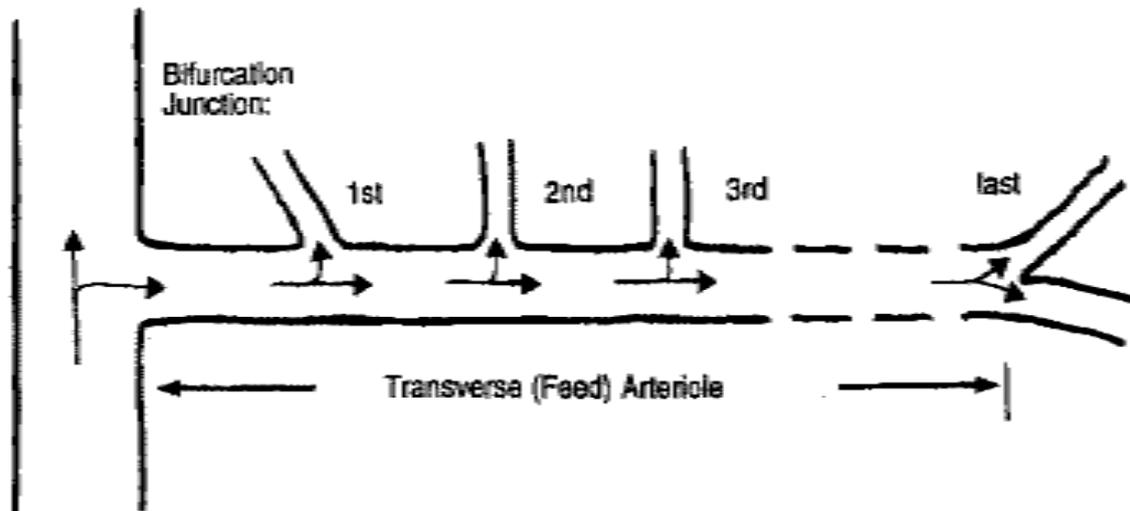
Figure 2.2 - Observed bifurcation junction angles in hamster artery [31]

It is especially important to minimise required energy use in biological systems, therefore nature itself has been optimised. Theoretical arguments suggest that in order to minimise the expenditure of energy the total volume blood flow should be minimised, and concurrently, the radii of arterial segments at bifurcations which follow a power-law [32]:

$$r_0^\gamma = r_1^\gamma + r_2^\gamma$$

(2.22)

where r_0, r_1 and r_2 are the radii of the parent, larger and smaller daughter segment respectively and γ is the bifurcation exponent. It is widely considered near optimum for $\gamma = 3$ for non-pulsate flow [32] and a necessary condition for equal shear stress for bifurcations [33] but not sufficient. With this exponent, it is possible to implement constant shear stress throughout the system, however, at the cost of flow homogeneity. Uneven splitting of flow is amplified by short frequently bifurcating segments of micro-vessels and, in turn, leads to a heterogeneous velocity field. The extent of phase separation is greatest in vessels of diameter < 30μm, and sensitive to the bifurcation angle [34].

Four cases are depicted below, the first with constant flow and the second two with uniform shear stress.
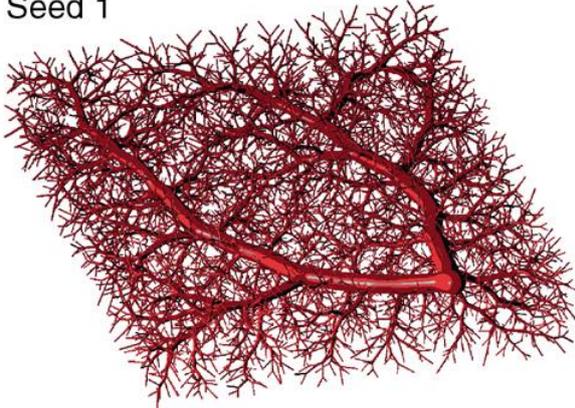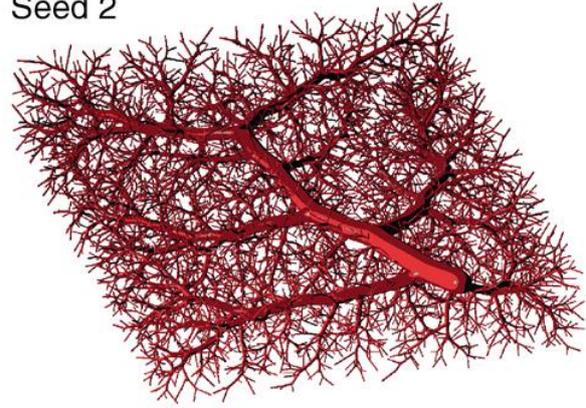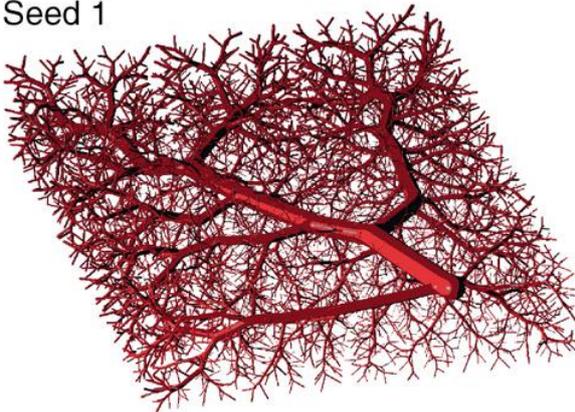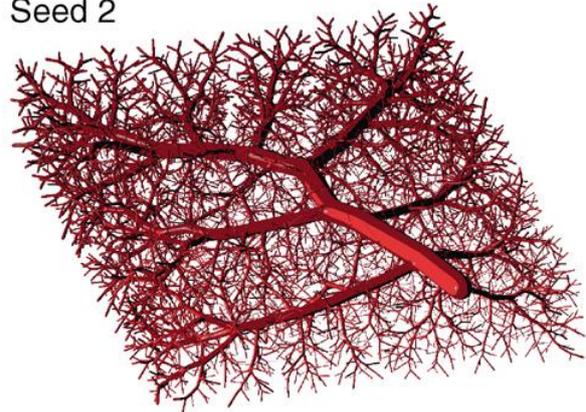
Figure 2.3 - Model generated by constrained constructive optimisation. Upper seeds (random sequences) for boundary conditions with *pre-set terminal pressure and flow* and lower seeds for *pre-set terminal pressure and uniform shear stress* [32]

2.3.2.1 Constrained Constructive Optimisation

Constrained constructive optimisation (CCO) is a model used to construct detailed optimal tree networks (Figure 2.3), according to a bifurcation law. Figure 2.4 depicts a tree structure created using CCO. CCO assumes a constant viscosity, which in the case of a non-Newtonian fluid such as blood is incorrect. However, it was found that, in this application, the effects on

shear stress induced by shear-rate-dependent viscosity are in the order of a few percent and may, therefore, for most purposes, be assumed negligible [33].
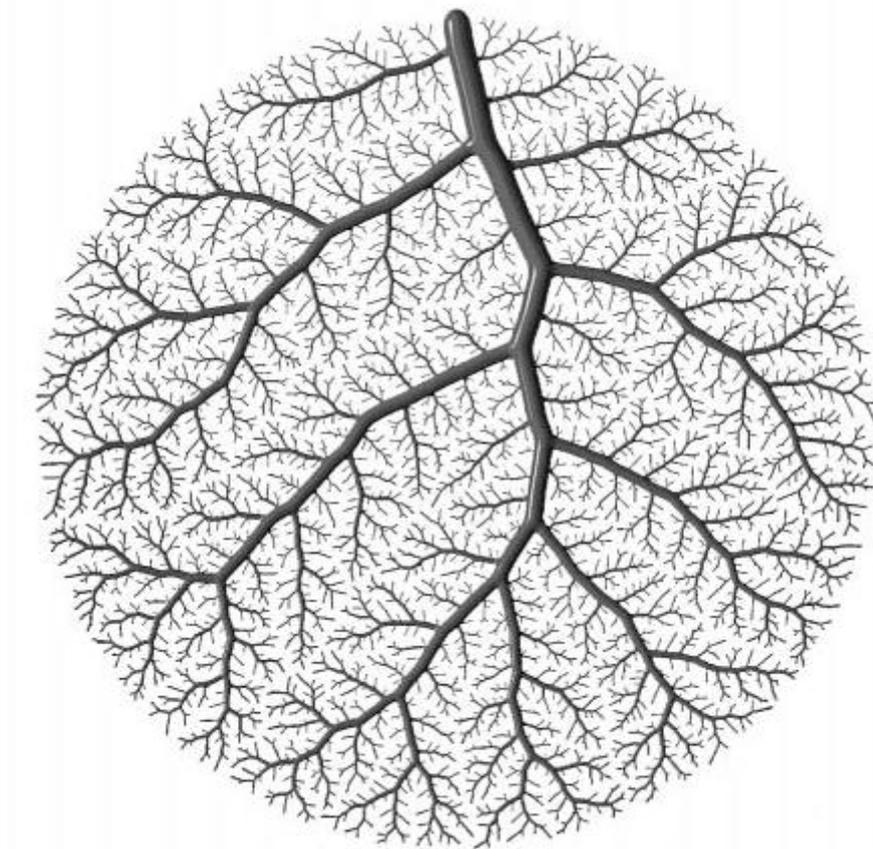
Figure 2.4 - Arterial tree model created with CCO [33]

The results obtained with CCO, under the constant viscosity assumption, closely match experimental data from a pig's left common carotid artery [33]. Interestingly, however, a study proposing an expansion of the initial work by Murray in Ref [30] indicates that if viscosity were assumed constant, the analysis would imply that only symmetric bifurcations are possible, leaving little room for heterogeneity [35].

Initial work by Murray proposes that the vascular system should be such that energy required (work) to maintain a specific required blood volume at a flow rate, should be minimised [30]. Murray's law states that the vascular tree is designed in order to balance the metabolic energy of a given blood volume and the energy required for blood flow, and is based on a minimisation principle for dissipated power [30]. It has been tested extensively

and appears valid in large arteries, even though it does not provide an accurate description of microcirculation [30].

2.3.3 Cross-section

With time, the duct cross-section of most natural biological systems tends towards a circular shape. Employing constructal theory and the freedom to morph, it can be found that a system will tend to minimum flow resistance, which would yield a polygon with the number of sides tending to infinity. Thus it yields a duct with a circular cross-section [28]. The design is, however, robust in nature and a polygonal cross-section with 10 sides performs nearly as well.

The cross-sectional size plays a major role in the total volume of fluid flow, as well as the fluid resistance. As the cross-section is increased, the flow resistance is lowered to the fourth power, which is represented by c_1/D^4 and volume increased quadratically ($c_2 D^2$) [24]. The terms c_1 and c_2 encompass all remaining factors. Thus the optimal ration between these opposing constraints would depend on the specific objective. The weight of the vessel/tube may be critical in the application, resulting in a higher flow resistance required. Similarly, the cost of fluid could impose a restriction on the total volume, hence the diameter. This is the case in the human body, where a trade-off is made between pumping power required (human heart) versus total blood volume. If vessels' cross-sections were to be smaller, a significantly larger heart would be required, however, less blood is necessary and vice versa [30].

2.3.4 Dendritic Heat Exchangers

According to the constructal theory, in order to maximise performance of a flow system, each volume element should perform at the same highest possible level of performance. The result is a heat exchanger with tree-shaped flow.

2.3.4.1 Trees Over Rectangular Area

A balanced parallel or counter-flow heat exchanger has two identical trees mated together. It is assumed the trees are in excellent thermal contact with each other. A tree is made up of many channels of decreasing sizes, where a channel has a length L_i, diameter D_i and $i = 0,1,2, \ldots, n$.

In constructal design, the smallest scale is primordial: the construction of the entire flow architecture starts with the smallest element scale (L_0, D_0) [24], [36]. Larger constructs are made by pairing these smallest elements. Dichotomy is thus an outcome and not as a result of assumptions. The channel lengths are given by the relation mentioned above: $L_i \cong 2^{0.5} L_i$ and the optimal ratio between diameters as $D_{i+1} = 2^{1/3} D_i$. This yields a heat exchanger as depicted in Figure 2.5.
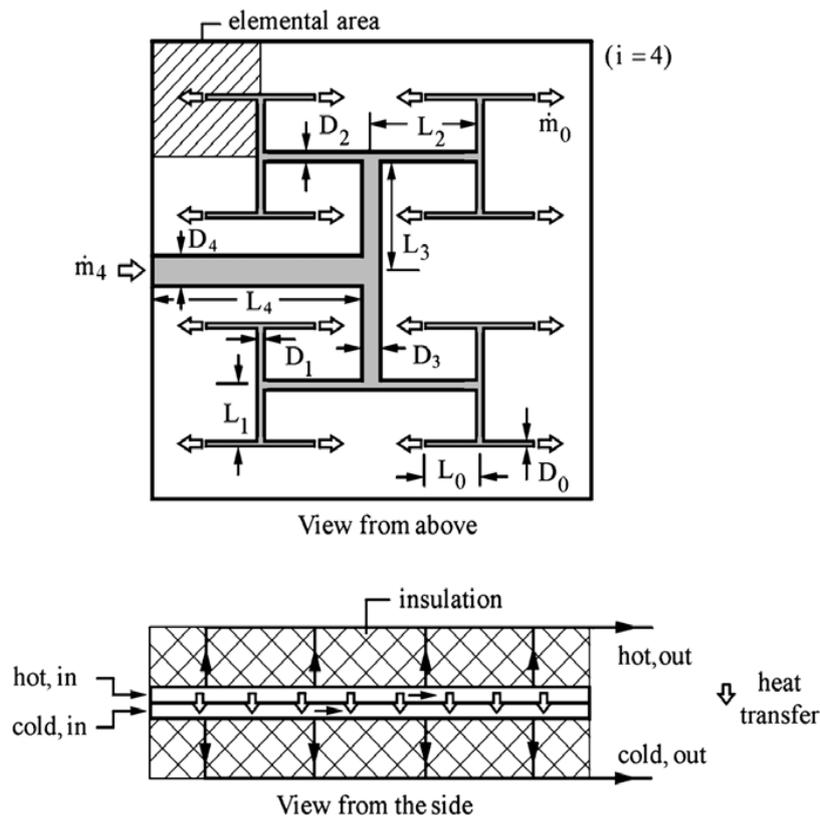
Figure 2.5 - Parallel flow of tree-shaped streams distributed over a square area [36]

The performance of such a tree design heat exchanger surpasses that of a conventional parallel tube heat exchanger [36]. It was found in [36] that for small mass flow rates, the

tree complexity (higher n) has little impact on the effectiveness. However, at higher mass flow rates the effectiveness increases with complexity, in a diminishing factor. Thus if the pumping power is available, it may significantly improve the heat exchanger performance by increasing the complexity. Complexity, however, is optimised, not maximised [27], [36].

2.3.4.2 Disk-Shaped Tree Structures

In search of a simpler and more compact configuration, Heitor and Reis [26] investigated flow in a tram shaped as a disc between the centre and the rim. These discs are placed in thermal contact with each other, forming a cross-flow as one flows from the centre outwards and the other inwards to the centre. It was found that a larger cross-sectional area with smaller radial distance yielded improved performance, which leads the fluid through a tree network of radial and quasi-radial ducts as shown in Figure 2.6.

The counter-flow heat exchanger consists of two trees, sandwiched together. Optimal tree architectures are based on the assumption that the ducts are slender enough with a Reynolds number sufficiently small that the flow in the ducts is fully developed and laminar, with negligible junction losses [36]. The layouts of the ducts have three features, namely the number of ports on the rim (n_0), the number of pairing levels (n_p) and the number of ducts connected to the centre (n_n), of which only two are degrees of freedom [24], [36]. Optimal tree structures can be found in Ref [37].
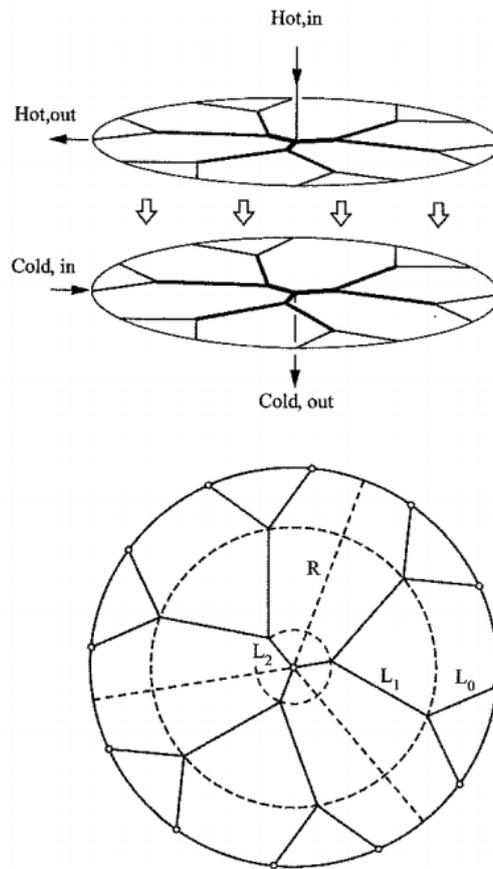
Figure 2.6 - Tree network of channels spread on a disk [24].

2.3.4.3   Square-Shaped Tree Structures

A simplification of the round disk shape is to make use of a tree between a point and a line, described in [37] obtained via path minimisation. The architecture follows the same principle as in Figure 2.6, as the fluid flows from a centre point to a side and vice versa. The structure is made up of four identical quadrants (Figure 2.7(a)), which are added together. This results in an overlap of the ducts at the diagonals. If the structure is optimised numerically, the two diagonal ducts bow slightly to produce the result seen in Figure 2.7 (b) [24]. This design, however, has only one degree of freedom, being the number of bifurcations [37].
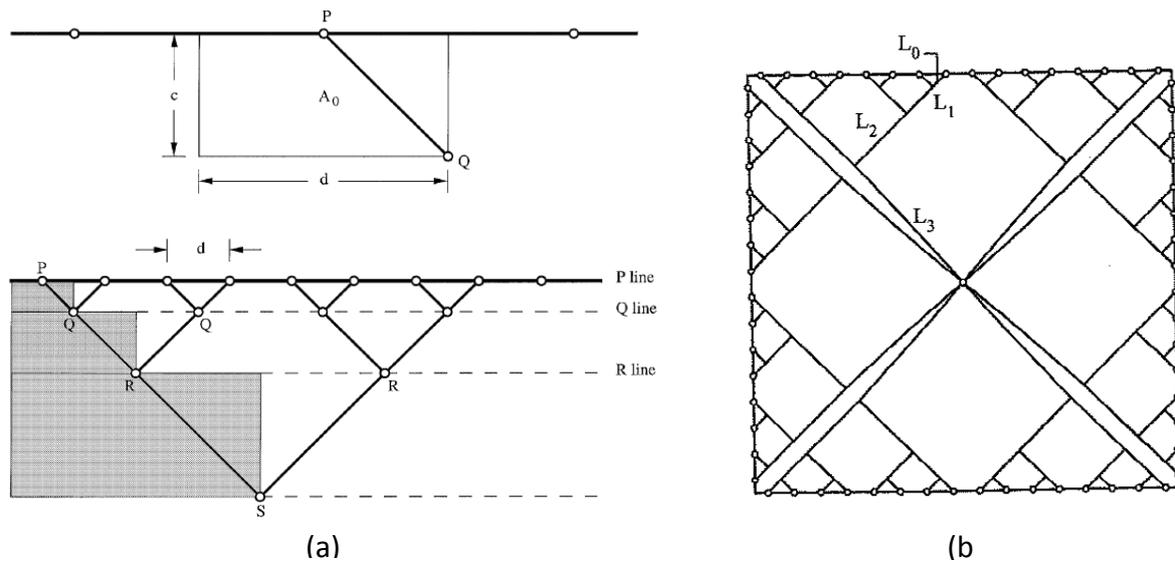
Figure 2.7 - (a) Construction of minimal-length tree between a line and a point [37]; (b) Qualitative presentation of the minimal-resistance tree of channels connecting a square rim with its centre [24]

2.3.5   Cost of Computing

It was found in [37] that it is possible to rapidly generate near-optimal tree structures by simply minimising the length of each duct for each area element. The performance of these minimal-length trees is 10-20% inferior to that of fully optimised equilibrium structures, but the cost of computing is orders of magnitude smaller, especially when the number of pairing levels increase [27].

23

2.4 Microchannels

A rapidly increasing requirement to achieve higher heat transfer density rate, especially of electronic devices, has promoted research into the field of microchannels and microchannel materials. This has resulted in research being done in order to more accurately predict the heat transfer coefficients in microchannels as well as the correlation between experimental en theoretical results [2], [5], [38].

Li and Peterson [39] numerically determined an optimum shape microchannel, which was found to significantly improve heat transfer performance. Unlike hypothesised, the authors found the optimal microchannel to hold the advantage to be a feasibly manufacturing design. Further research by Bello-Ochende et al. in Ref [40] employed constructal theory principles in order to obtain an optimal microchannel, minimising peak wall temperatures.

The research has extended towards the fluid-flow within the microchannel in a heat transfer environment. Toh et al. found numerically in Ref [3] that lowering the Reynolds number results in lower frictional losses as a result of a reduced viscosity of the water. This would subsequently lower the pumping power required.

Microchannels hold the potential to enhance the heat extraction from electronic devices, which will promote the development of smaller, higher-performance electronic devices.

2.5 Geometrical Optimisation

A practical form of constructal theory is found in geometrical optimisation, whereby the geometric ratio of a geometry is to be optimised. The optimisation is constrained by a certain parameter, usually total volume or cross-sectional area, and optimised according to an objective function, typically heat transfer, by changing the geometric ratio.

The heat transfer characteristics in a microchannel show that the geometric aspect ratio has a significant effect on the efficiency of microchannel heat sinks. In [41], Bello-Ochende et al. utilised geometric optimisation in order to determine the optimal configuration for a three-dimensional microchannel. The total volume was fixed by the length and cross-sectional area constraints. This allowed the only design parameter to be in the form of the cross-sectional shape and the ratio of the internal thickness of the vertical and horizontal substrate. The authors found that, using geometric optimisation through scale analysis and intersecting the asymptotes method, an optimal ratio exists in order to maximise heat transfer over a fixed pressure drop. Later, in Bello-Ochende et al. [40], an optimal geometry for a microchannel was numerically determined.

2.6 Conclusion

In this chapter a background was given on literature related to the dissertation. Non-Newtonian fluids, the characteristics thereof sand relevant equations were given. This included the Ostwald de Waele power-law model predicting the viscosity and Reynolds number. Complex flow structures, with the focus on constructal theory, were discussed. A brief overview of research done on microchannels was put forward. Finally, geometrical optimisation was discussed, which will be employed in this dissertation.

Chapter 3   Numerical Modelling

3.1   Introduction

This chapter attempts to give an overview of the numerical modelling techniques employed in order to solve a specific engineering problem. The governing equations used to characterise the flow and energy transfer are put forward in Section 3.3. A brief overview is given of the finite volume discretisation method in Section 3.4. Finally, the SIMPLE algorithm, used for solving the governing equations, is discussed in Section 3.5.

Numerical analysis was done with the use of computational fluid dynamics (CFD) software, namely StarCCM+, which was selected for its robustness, high quality mesh generation and ability to model non-Newtonian fluids.

3.2 Procedural Overview

The numerical modelling can be broken down into three sections, namely:

Pre-processing: In order to simulate a system, a CFD model which accurately captures the physical model must be created. Boundary conditions and material properties are defined according to the physical model.

Once the geometry has been created, it is subdivided into small volumes, referred to as cells. These cells are used to calculate the discretised governing equations at each point. The mesh, which is the collection of cells, is refined at certain key areas according to heat transfer or fluid flow.

Processing: Once the model has been created simulation takes place, where the discretised governing equations are solved numerically.

Post-Processing: This stage involves the collection and evaluation of data obtained. In addition to the raw data, the user has access to various visual representations, including flow and temperature fields.

3.3 Governing Equations

The governing equations are a set of partial differential equations, mathematically representing the conservation laws of physics. The laws are captured in the conservation of mass (continuity), momentum and energy.

3.3.1 Conservation of Mass (Continuity)

The equation for the conservation of mass is given by Ref [42]

$$\frac{\partial \rho}{\partial t} + \nabla(\rho \vec{U}) = 0 \tag{3.1}$$

where ρ is the density, t the time and $\vec{U}$ the velocity vector.

Thus for a steady-state solution, equation (3.1) is reduced to:

$$\nabla(\rho \vec{U}) = 0 \tag{3.2}$$

3.3.2   Conservation of Momentum

The conservation of momentum equations, the Navier-Stokes equations, relate Newton's second law, in which the rate of change of momentum of a fluid is proportional to the forces applied. For incompressible flow, neglecting the gravitational effect, the equation is given by:

$$\rho(\vec{U} \cdot \nabla \vec{U}) = -\nabla P + \mu \nabla^2 \vec{U} \tag{3.3}$$

where P is the total pressure and μ the dynamic viscosity.

3.3.3   Conservation of Energy

The conservation of energy equation is derived from the first law of thermodynamics, which states that the rate of change of energy of a system is equal to the sum of the work and heat added to the system: in standard form, the equation becomes [42]:

$$\rho \frac{DE}{Dt} = div(k\nabla T) + \Phi \tag{3.4}$$

where k and T is the thermal conductivity and temperature of the fluid respectively and Φ the dissipation function given by:

$$\Phi = \mu \left\{ 2 \left[\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 + \left(\frac{\partial w}{\partial z} \right)^2 \right] \right.$$

$$\left. + \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)^2 + \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right)^2 \right\} \qquad (3.5)$$

$$+ \lambda \left(\nabla \vec{U} \right)^2$$

where λ represents the coefficient of bulk viscosity. For incompressible flow at low velocity, the dissipation function can be neglected, thus equation (3.5) becomes:

$$\rho C_p \left(\vec{U} \cdot \nabla T \right) = k_f \nabla^2 T \qquad (3.6)$$

where $\nabla^2 = \partial^2 / \partial x^2 + \partial^2 / \partial y^2 + \partial^2 / \partial z^2$ and C_p is the specific heat capacity.

For a volume occupied by a solid, the energy equation reduces to:

$$k_s \nabla^2 T = 0 \qquad (3.7)$$

Flow is assumed steady, laminar, incompressible and two-dimensional with negligible heat transfer due to radiation and natural convection. Buoyancy forces are considered negligible.

3.4 Finite Volume Method

StarCCM+ makes use of the finite volume discretisation method in order to discretise the integral form of the governing equations. The result is a set of equations which can be solved for each control volume or cell. The segregated flow and segregated fluid temperature models of StarCCM+ are used to simulate the fluid flow and energy in conjunction with a second-order upwind scheme to model the convection-diffusion effect in the transport equations. The segregated models allow the equations to be solved simultaneously in an explicit manner [43].

The finite volume method is similar to the finite difference method, with the advantage that the control volumes, thus the mesh, need not be structured. In addition, the numerical flux is conserved from one discretised cell to its neighbour, which makes the scheme attractive

for CFD analysis [44]. The governing equations are solved at these special locations, typically, the nodes with a control volume surrounding them. The discretised governing equations used by StarCCM+ take on the following form [43]:

Conservation of mass (continuity) equation

$$\sum_f \dot{m}_f = \sum_f (\dot{m}_f^* + \dot{m}_f') = 0$$

(3.8)

where $\dot{m}_f^*$ is the uncorrected face mass flow rate with flow correction $\dot{m}_f'$.

Momentum equation

From the Navier-Stokes equations, it is possible to obtain a finite volume discretised equation for momentum:

$$\frac{d}{dt}(\rho X v V)_0 + \sum_f [v\rho(v - v_g) \cdot a]_f = -\sum_f (p\boldsymbol{I} \cdot a)_f + \sum_f \boldsymbol{T} \cdot a$$

(3.9)

where X is the position vector, v the velocity and v_g grid velocity, V represents volume, T is the viscous stress tensor, a is the face area vector and I the identity matrix.

Energy equation

$$\frac{d}{dt}(\rho E V_0) + \sum_f \{[\rho H(\mathrm{v} - \mathrm{v}_g) + \boldsymbol{q}'' \cdot \mathrm{a} - (\mathbf{T} \cdot \mathrm{v})] \cdot \mathrm{a}\}_f = (\mathrm{f} \cdot \mathrm{v} + s)V_0$$

(3.10)

3.5   SIMPLE Algorithm

The semi-implicit method for pressure-linked equations (SIMPLE) algorithm was originally put forward by Patankar and Spalding (1972) [42], [45], [46]. The algorithm makes use of correction factors to calculate the pressure on a staggered grid. The sequence of operations for the SIMPLE algorithm is given by Figure 3.1 [42], [45]:
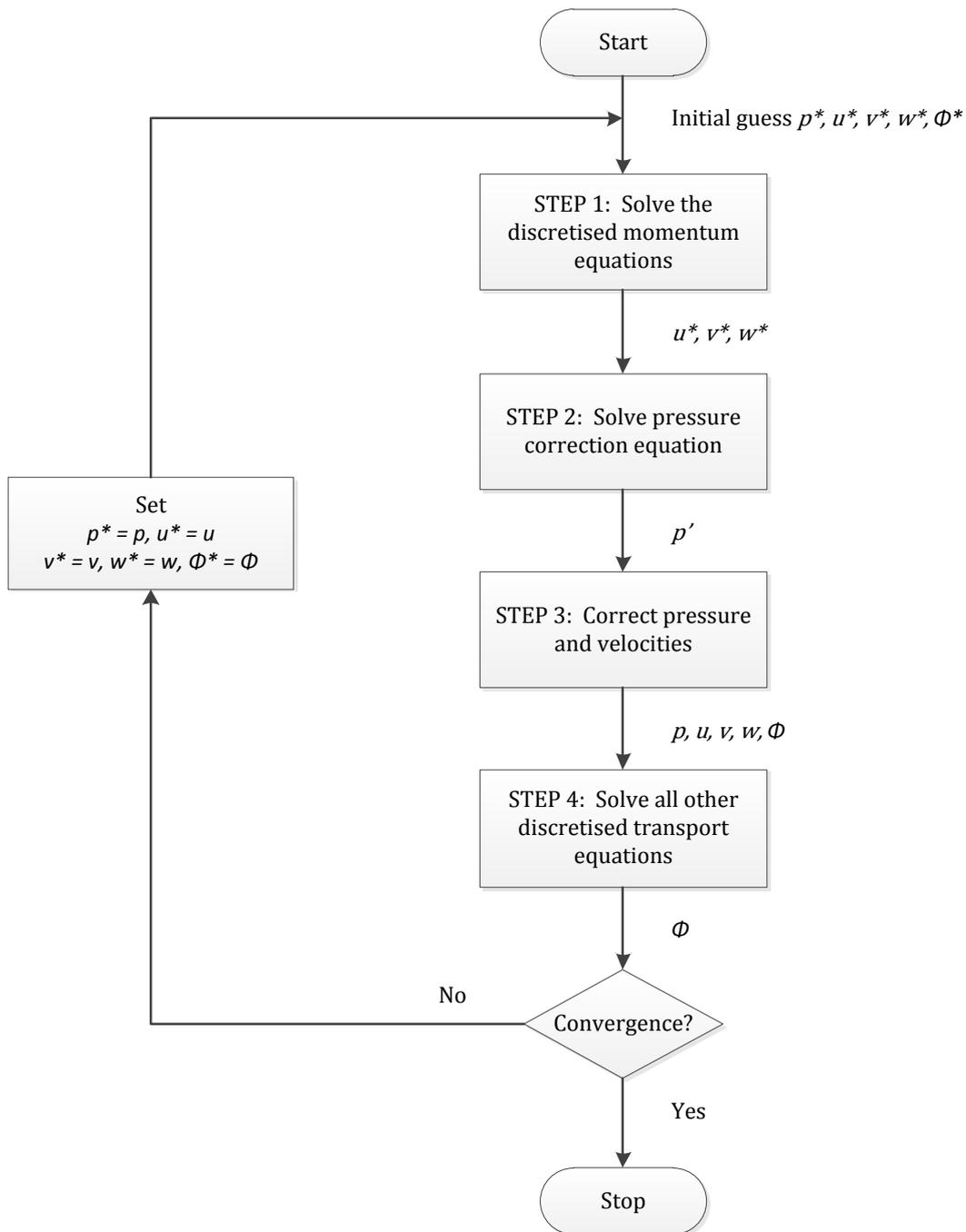
Figure 3.1 - SIMPLE algorithm

3.6   Non-Newtonian Fluid

In order to model non-Newtonian fluid characteristics, the Ostwald de Waele power-law model [6], [20], [21] was used, according to which viscosity is a function of consistency (K) and the power-law index (n)

$$\mu(\dot{\gamma}) = K\dot{\gamma}^{n-1}$$

(3.11)

Three index (n) values were used, namely:

- $n = 0.5$ Pseudoplastic (shear-thinning)
- $n = 1$ Newtonian
- $n = 1.5$ Dilatant (shear-thickening)

This allowed for testing of both dilatant and pseudoplastic non-Newtonian fluids compared to the benchmark Newtonian fluid. The goal was to simulate with both a shear-thickening and shear-thinning non-Newtonian fluid as to determine the effectiveness thereof as a working fluid.

It was found with preliminary simulation that by using a higher consistency factor K, the overall viscosity of the fluid increased. Therefore, it allowed for better testing with regard to the shear-thinning and -thickening action, whereby the viscosity is increased or decreased in the presence of shear force. Subsequently, a consistency of 20 $Pa \cdot s^{1/2}$ [47] was used, which closely resembles the consistency of human blood [11].

3.7   Conclusion

In this section, the numerical modelling process was discussed. This included an overview of the numerical procedure to be followed. The governing equations responsible for the flow and heat transfer were described and subsequently given in their discretised form, accompanied with an explanation of the finite volume discretisation method used by the CFD software. The process undertaken by the SIMPLE algorithm in order to numerically

solve the governing equations was put forward. Finally, the numerical method for implementing the non-Newtonian fluid viscosity with accompanied parameters was briefly discussed. The range for the power-law exponents was fixed for $n \in \{0.5, 1, 1.5\}$ with accompanied consistency of $20\ Pa.s^{1/2}$.

Chapter 4   Optimisation and Automation

4.1   Introduction

In this chapter, firstly, the optimisation process (Section 4.2) is discussed and the objective function, namely thermal conductance, is put forward. This is followed by the automation process (Section 4.3) required to automate the CFD software, the reason it was required and the implementation thereof.

4.2   Optimisation

The phenomenon of optimisation has become more prevalent in research fields, where the efficiency of a system is to be maximised according to a given set of constraints. In this dissertation, geometric optimisation was conducted, in which the geometry of the microchannel was to be optimised to enhance heat transfer. This included the modification of the geometric ratio (height versus length) or in the case of a complex microchannel, the bifurcation point.

An objective function was required to evaluate the optimisation goal, being to maximise heat transfer through the microchannel, at various inlet velocities and porosity values, utilising Newtonian and non-Newtonian fluids. Thus, thermal conductance was programmed as the objective function. Thermal conductance in dimensionless form, for a two-dimensional model, is defined as follows [41]:

$$C = \frac{q''L}{k_f(T_{max} - T_{in})} = \frac{q}{Lk_f(T_{max} - T_{in})} \qquad (4.1)$$

$$\therefore C = \frac{q}{Lk_f\Delta T} \qquad (4.2)$$

here q is defined as the heat flux, k_f the thermal conductivity of the fluid, L the length of the computational domain, and ΔT the excess temperature. Manipulating equation (7.3) yields the following for a constant heat flux across the characteristic length $A^{1/2}$:

$$C = \frac{q}{A^{1/2}k_f\Delta T} \qquad (4.3)$$

where the thermal conductance is now a function of ΔT alone.

Similarly for a three-dimensional case the thermal conductance can be given as:

$$C = \frac{q}{V^{1/3}k\Delta T} \qquad (4.4)$$

where $V^{1/3}$ is given as the characteristic length.

Thermal conductance is a dimensionless form of expressing the ratio of the heat transfer rate to largest excess temperature, expected to occur in the exit plane of the microchannel. The dimensionless thermal resistance, R, is given by the reciprocal of thermal conductance:

$$R = \frac{1}{C} \qquad (4.5)$$

As each the design variable could be isolated and investigated independently the optimisation problem was rendered unimodal. Thus optimisation algorithms were

redundant and a simple line search methodology could be employed to obtain the optimal configuration, for each flow configuration.

4.3 Automation

CFD software is generally written to simulate a specific model, of which the parameters are clearly defined and only a few iterations are required. This is done manually by the user and typically requires the manipulation of the geometry of the model in a GUI (graphical user interface) fashion, followed by redefining of the boundary conditions and mesh parameters. However, in the case of an optimisation problem, numerous iterations of design parameter augmentation and geometry updates are required. Consequently, this method of user input becomes extremely cumbersome and time consuming. It is therefore a requirement to automate the CFD software.

As StarCCM+ is written in the Java language [9, 18], it is possible to control StarCCM+ with the use of a Java macro script. A macro script was written such that it allowed the user to completely automate the simulations and iterate through various alterations of the design parameters.

Figure 4.1 illustrates the entire simulation process, which would be automated by means of the macro script. As the optimisation only required iteration through various, pre-determined, geometries and corresponding inlet velocities, two separate macro scripts were created. The first, as indicated in Figure 4.1, created all the geometries, which were exported. Secondly, another macro script was written to import the geometries, initialise the mesh and design parameters, run the simulation and export the data. This was iterated for each geometric configuration and inlet velocity.

The use of two separate macro scripts held numerous advantages, including:

- Increased robustness and stability: it was found after several iterations that the software became increasingly more unstable, if the geometry creation was included in the simulation iteration.

- Significant resource efficiency: as both geometric and inlet velocity iterations were conducted, the resources required by the software to modify the geometry within each iteration were significantly greater.

- Computational time-saving: as each successive inlet velocity was to iterate through a set of identical geometries, a considerable amount of CPU time was saved by recycling the exported geometries.

- Ease of debugging: separate macro scripts allowed for easier debugging and fault-finding.
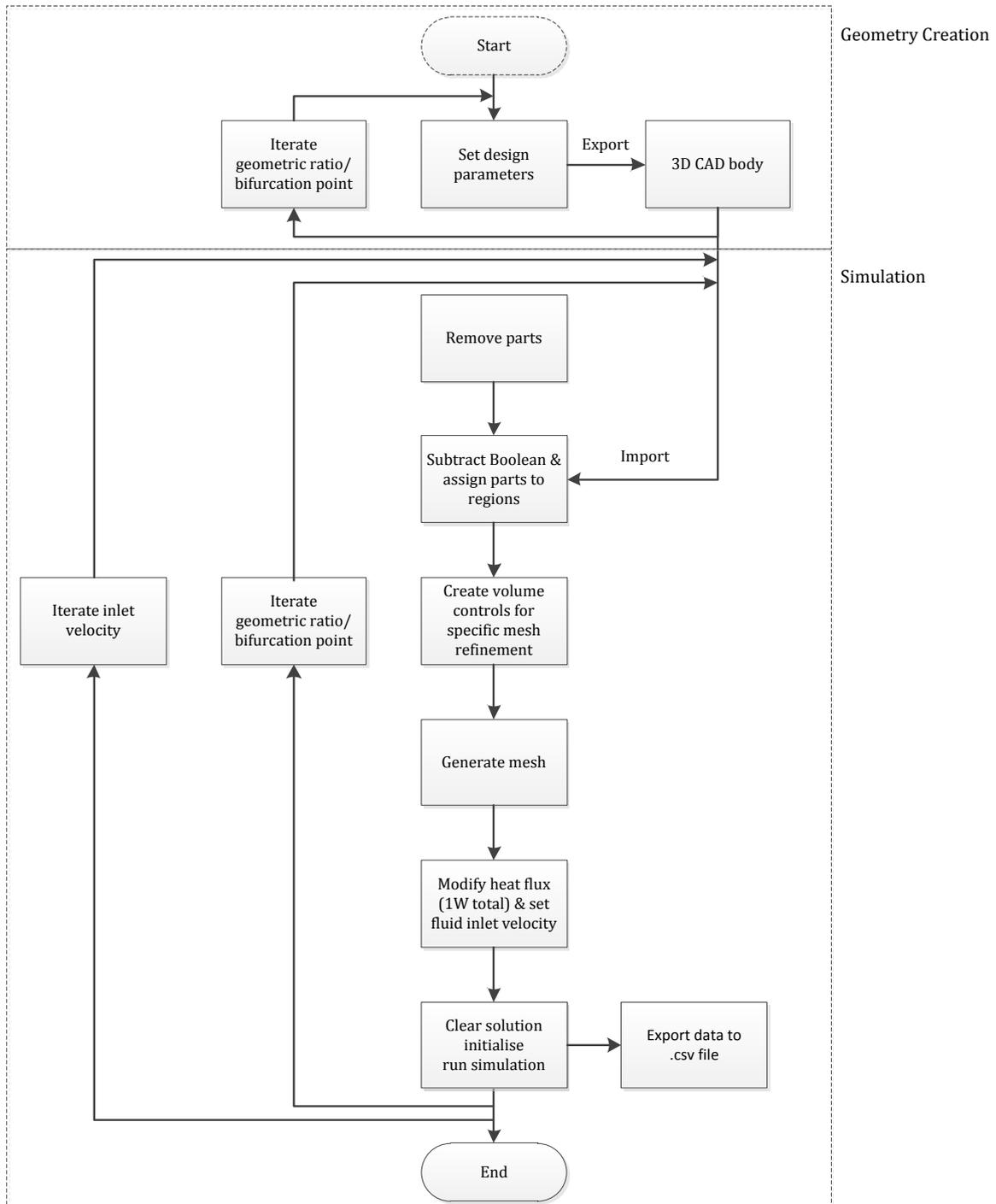
Figure 4.1 - Java macro script

4.4　　Conclusion

In this chapter, the process of numerical optimisation with its requirements was given. It was stated that, as the problem was essentially unimodal, the use of optimisation algorithms was not required. The objective function, being thermal conductance, was defined for both a two-dimensional and three-dimensional problem. The objective function was to be maximised. However, as the optimisation routine required a significant amount of CFD configurations to be simulated, an automation process was required. In order to achieve this, two Java macro scripts were written. It was found best to make use of two separate macro scripts, which improved the stability of the routine, improved the resource efficiency and reduced computational time requirements. The process followed by the macro script was illustrated in this chapter.

Chapter 5   Simple Microchannel

5.1   Introduction

Chapter 5 is dedicated to the analysis of the simple microchannel. The two-dimensional numerical model is related to the physical model in Section 5.2, by means of the boundary conditions. CFD solutions are heavily dependent on the mesh quality; therefore, it is essential to conduct a mesh refinement study to ensure mesh independence. The mesh independence study is described in Section 5.2.1, followed by the numerical simulation. Finally, the optimisation results are given and discussed in Section 5.4.

5.2   Computational Model

Consider a section of microchannel with a total area (A) fixed at 1×10^{-4} m as shown in Figure 5.1. The height, length and duct diameter are given by H, L and D respectively. The ratio between the area of the solid and fluid is constrained by the porosity ratio (φ). A uniform heat transfer rate (q) is applied to the solid from the top and bottom and heat is extracted by the working fluid.
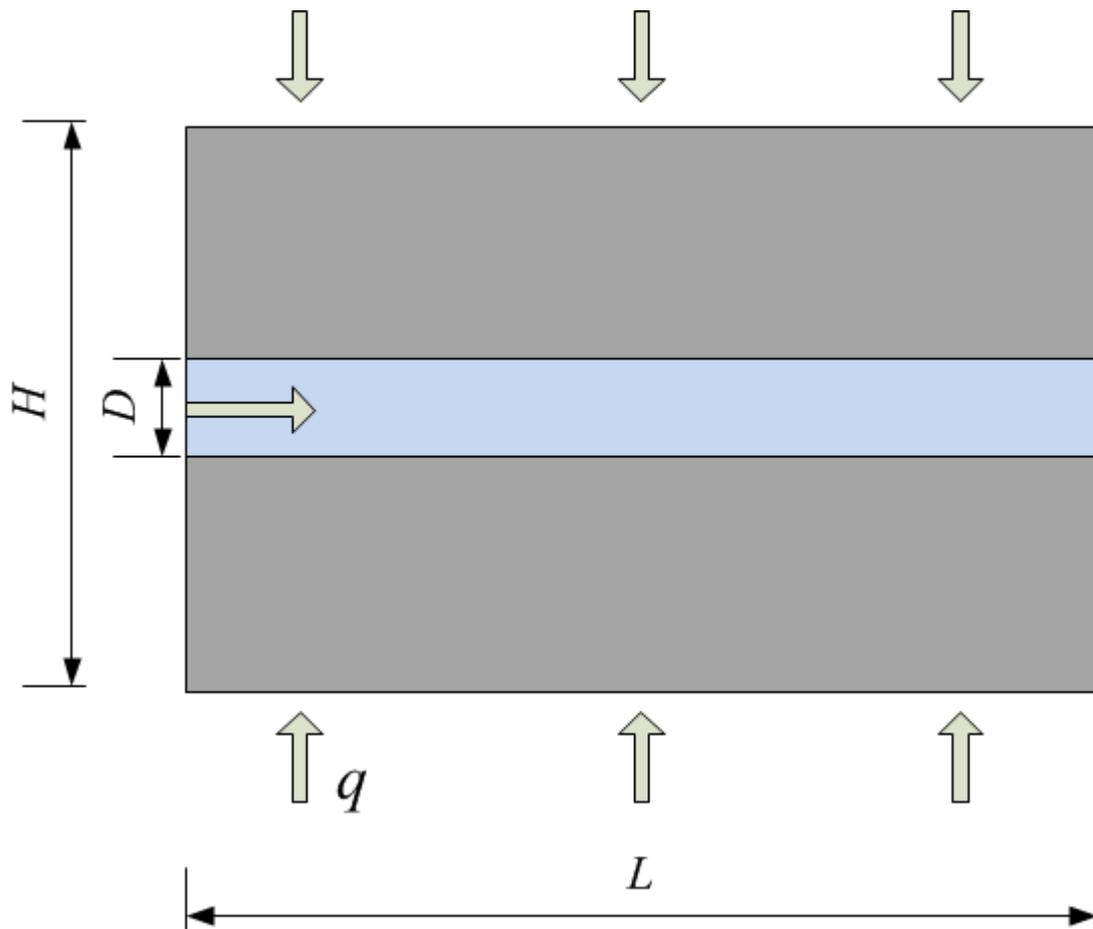
Figure 5.1 - Microchannel configuration

A numerical model was created in StarCCM+ to accurately resemble the microchannel configuration given in Figure 5.1. A three-dimensional model, with unit thickness in the z-direction, was created. The fluid flow was simulated in the xy-plane only. This was to simulate a two-dimensional microchannel. The result was a microchannel as follows:
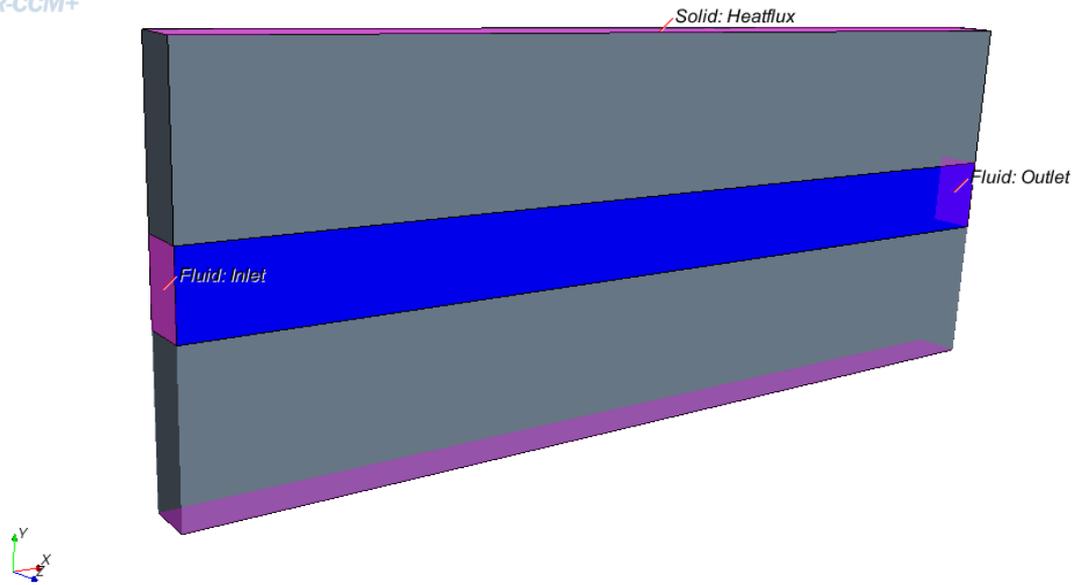
Figure 5.2 - Simple microchannel numerical representation

The inlet boundary condition was set to a velocity inlet:

$$\vec{U} = u_{in} \tag{5.1}$$

$$T = T_{in} \tag{5.2}$$

with no slip occurring on the walls of the channel and a pressure outlet was defined for the fluid outlet. The thermal boundary condition consisted of uniform heat flux, calculated to ensure a constant total heat transfer rate ($1W$), which was applied to the top and bottom of the solid:

$$k_s \frac{\partial T}{\partial y} = -q'' \tag{5.3}$$

The remaining boundaries were modelled as symmetry planes with free slip, so as to represent an infinite channel, of which a two-dimensional section in the xy-plane was considered. Thus it can be stated that:

$$\frac{\partial \vec{U}}{\partial (x, y)} = 0 \tag{5.4}$$

$$\frac{\partial T}{\partial(x,y)} = 0 \tag{5.5}$$

Convergence of the solution was met once the residuals, which included of energy, continuity and velocity, were below 10^{-4}.

5.2.1   Mesh Refinement

In order to accurately model the physics of the flow and heat transfer through the microchannel, a structured mesh, using predominantly hexahedral cells, was used. The StarCCM+ trimmer cell mesher provided a robust and efficient mesh with surface refinement [43]. Region-specific meshing was employed where a higher level of mesh refinement was done in the fluid region.

As the model was created to simulate the flow of non-Newtonian fluid, the mesh required adequate refinement in areas such as the walls where the shear force was higher, thus enabling the model to accurately capture the non-Newtonian characteristics in these regions. Consequently, prism layers were added (Figure 5.3) along the fluid-solid interface to accurately model the fluid boundary layer and non-Newtonian characteristics, as well as the heat transfer over the interface.

With the purpose of obtaining mesh-independent numerical solutions, a mesh refinement study was conducted. It was found that 15 prism layers were able to capture the fluid boundary layer, as depicted in Figure 5.4. As mentioned, it was required to reduce the cell size of the fluid region, while the cell size of the solid region could be increased, thereby reducing redundant computational time.

Once scaling with regard to regions and the prism layers were added, the cell size was systematically reduced in order to ascertain the point at which mesh independence would be obtained. A summary of the study is captured in Table 5.1. The StarCCM+ meshing tool makes use of a base size, from which all mesh calculations are done, thus assisting in systematically reducing the cell size of the mesh. It can be seen that 30912 cells produced a 1.76% change in the result, after which a doubling in the amount of cells merely produced a

0.6% and subsequently 0.3% change. Thus a mesh of 30 912 cells was deemed satisfactory to ensure mesh independence. The result was a mesh as depicted in Figure 5.5.

Table 5.1   Mesh refinement study: simple microchannel

Cells	Base size (m)	Thermal Resistance	% change
21400	2.2×10^{-4}	50.7984	-
25586	2.0×10^{-4}	50.6758	0.241
30912	$\mathbf{1.8 \times 10^{-4}}$	**49.7835**	**1.761**
66304	1.6×10^{-4}	49.4732	0.623
117400	1.4×10^{-4}	49.3278	0.294
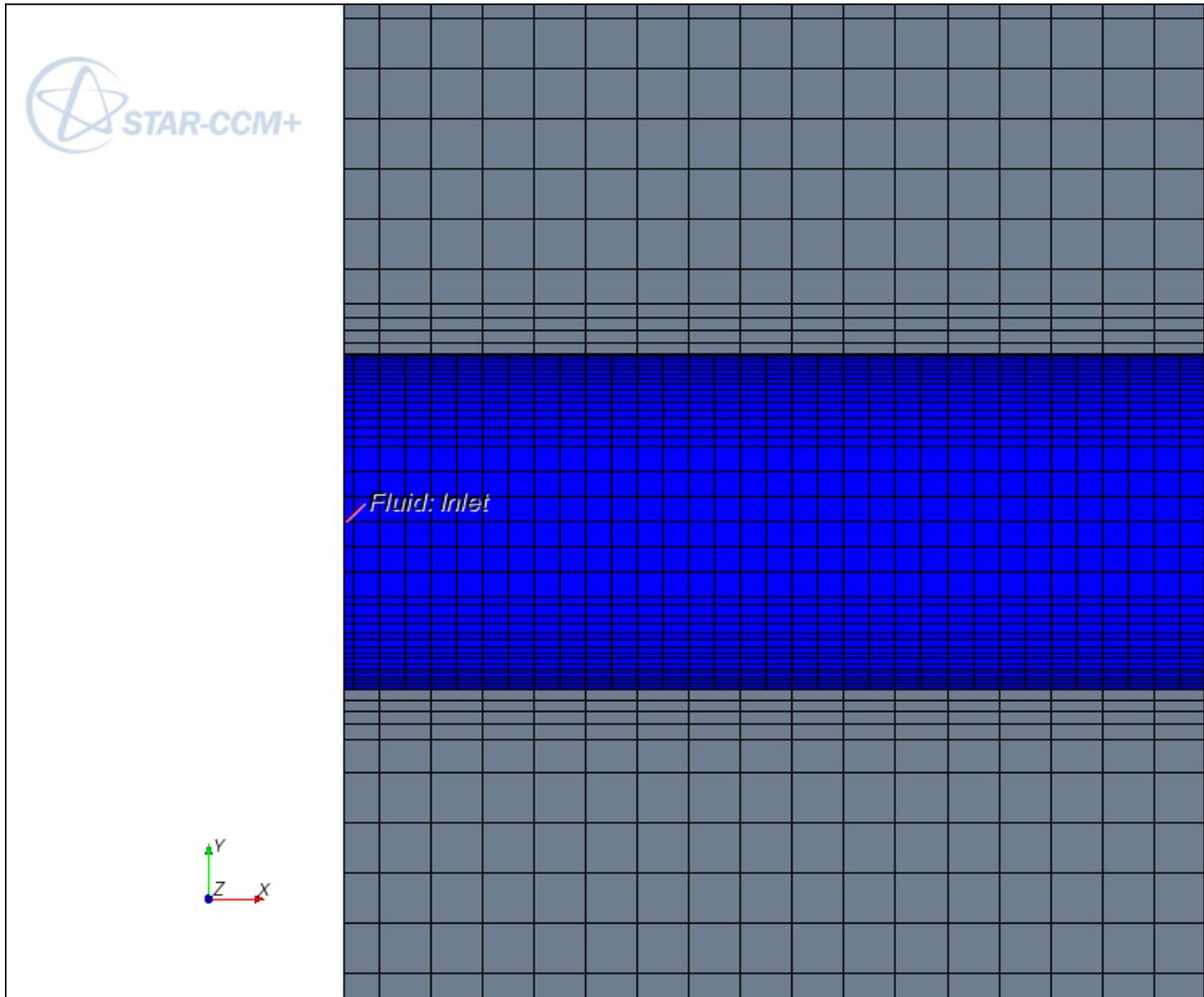
Figure 5.3 - Simple microchannel prism layer mesh
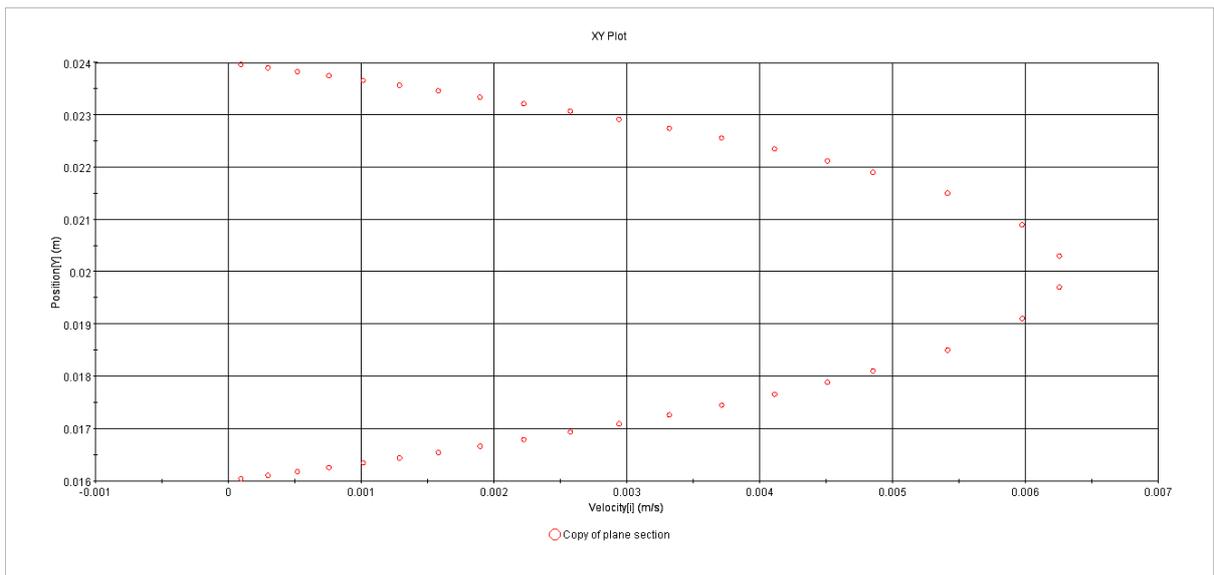
Figure 5.4 - Simple microchannel velocity profile generated by StarCCM+

45
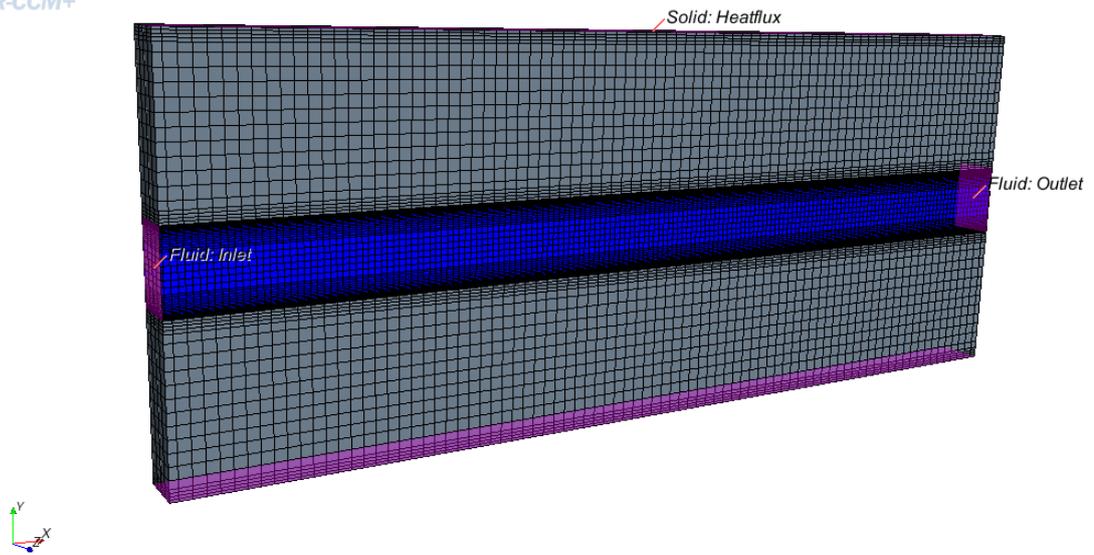
Figure 5.5 - Simple microchannel mesh

5.3   Numerical Results and Discussion

The steady-state numerical results, simulating both Newtonian and non-Newtonian fluids, are given in this section. As mentioned, convergence is only achieved once the residuals decrease below 10^{-4}. Subsequently, the temperature field error is reduced to a value of 10^{-4} as well. An example of the residuals is given in Figure 5.6. It can be seen that the residual for energy has decreased below 10^{-4} and remained constant, while the momentum and continuity residuals have reduced below 10^{-5}. This was deemed sufficiently accurate.
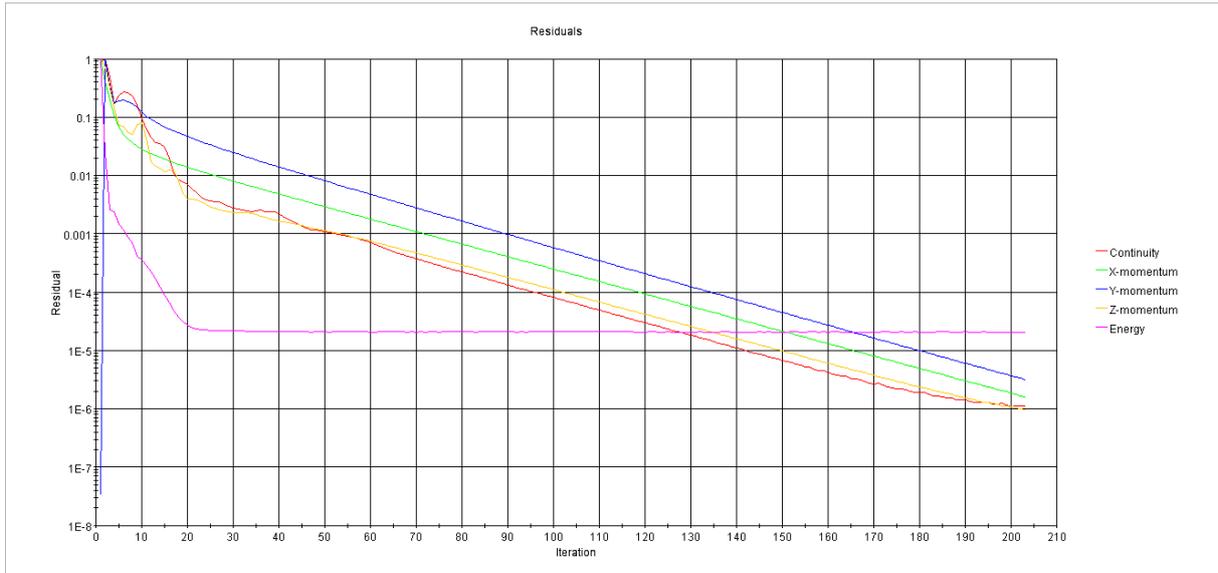
Figure 5.6 - Residuals plot generated by StarCCM+ indicating convergence

Figure 5.7 to Figure 5.9 below depict the temperature field for the three fluids with varying power-law exponents (n = {0.5; 1; 1.5}). It can be seen that the maximum temperature is nearly 9 degrees Kelvin lower when the pseudoplastic fluid is used, compared to the dilatant fluid. This equates to a 13% rise in excess temperature (the difference between fluid inlet and maximum microchannel temperature). Similarly, the maximum temperature of the microchannel with a Newtonian working fluid is nearly 3 degrees Kelvin higher than that of the pseudoplastic fluid, an approximate 5% rise in excess temperature. This indicates the role non-Newtonian characteristics may have upon the heat transfer capabilities of a microchannel.
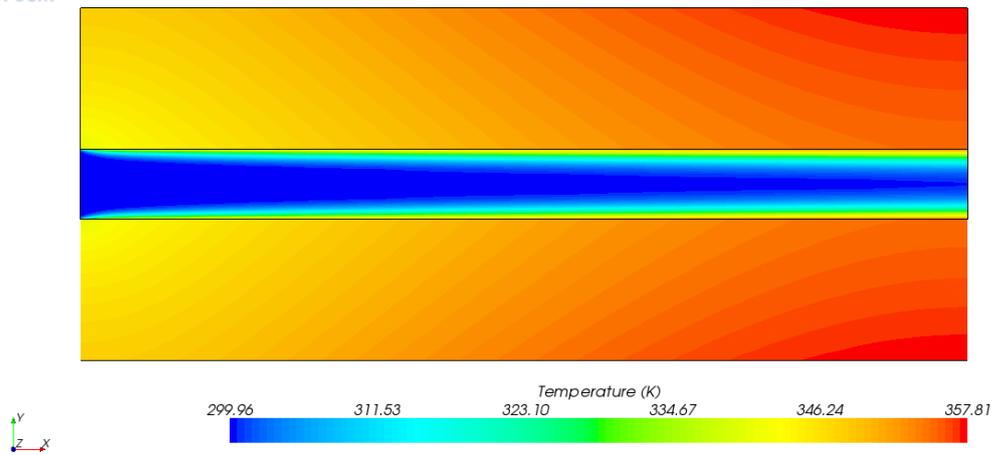
Figure 5.7 - Simple microchannel temperature field (Pseudoplastic fluid, n = 0.5)
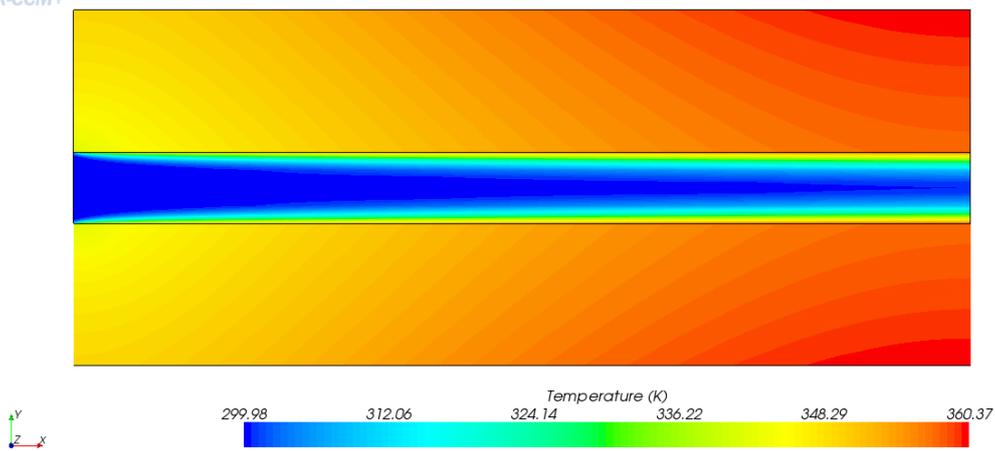
Figure 5.8 - Simple microchannel temperature field (Newtonian fluid, n = 1)
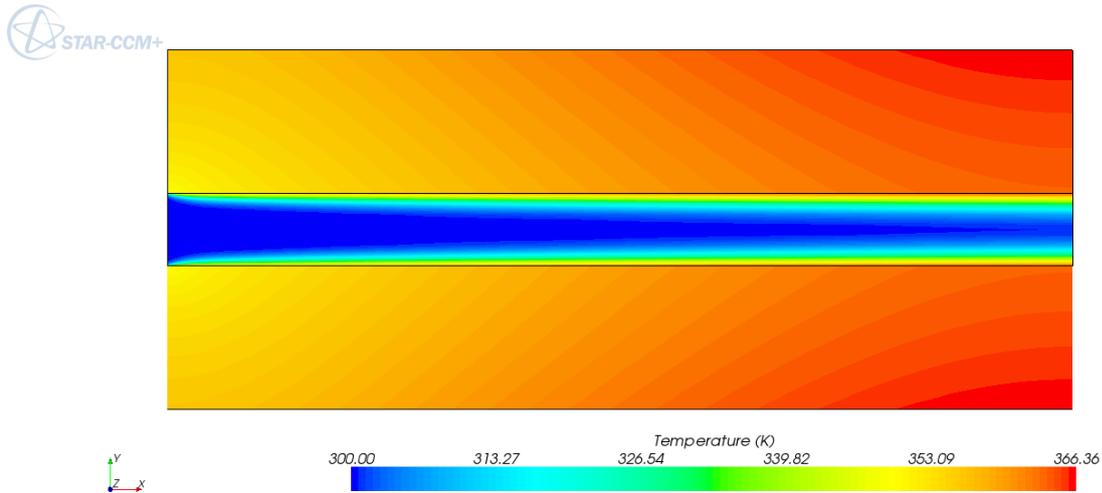
Figure 5.9 - Simple microchannel temperature field (Dilatant fluid, n = 1.5)

Figure 5.10, generated by StarCCM+, plots the fluid velocity at the various nodes in the working fluid channel. It indicates the flow field at a specific point, where the flow is fully developed. It can be observed that the shear-thinning action of the pseudoplastic fluid (a) results in increased fluid velocity in the boundary layer, as opposed to the Newtonian fluid. In contrast, the dilatant fluid experiences a shear-thickening in this boundary layer region, where shear force is increased at the wall. This reduces the fluid velocity at the fluid-solid interface. However, as continuity is to be maintained, the fluid furthest from the boundary is forced to increase in velocity. The result is a less homogeneous velocity field, with higher peak fluid velocity in the centre of the channel.

It can be seen that the peak fluid velocity of the dilatant fluid tends towards 0.1m/s, where that of the pseudoplastic fluid is merely 0.0825 m/s. This results in an increase of fluid velocity of 17%, in order to maintain continuity, and subsequently, the flow rate.

© University of Pretoria
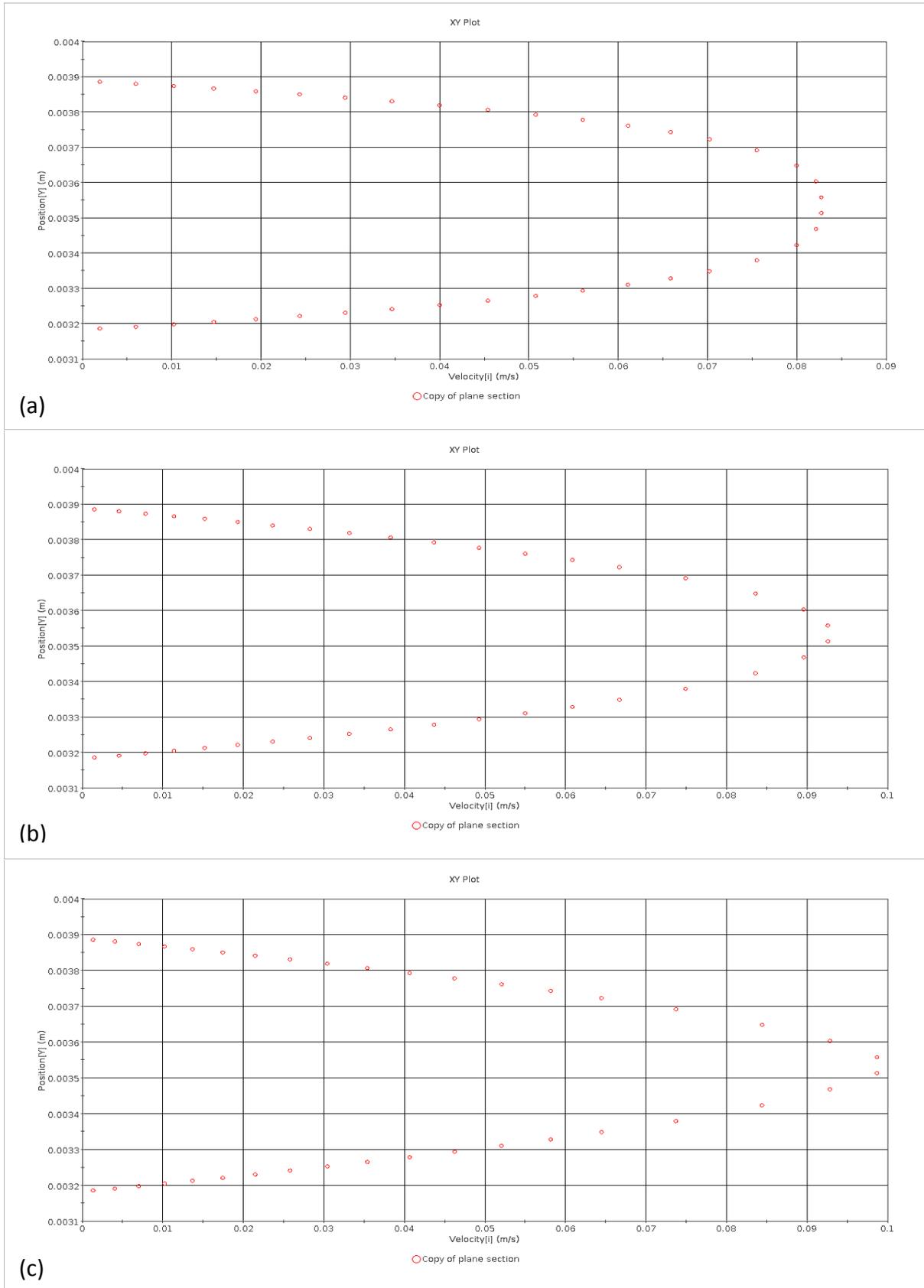
Figure 5.10 - The fluid flow velocity distribution extracted from StarCCm+ for pseudoplastic (a); Newtonian (b) and dilatant (c) fluid

5.4   Simple Microchannel Optimisation

The objective was to obtain an optimal geometric configuration, which maximises the thermal conductance derived in Section 4.2. The ratio between the channel height and length is manipulated such that heat transfer is maximised, while the area of the two-dimensional microchannel is constrained.

5.4.1   Mathematical Formulation

In order to perform geometric optimisation, an expression for the duct diameter D is required as a function of the geometric ratio, given by H/L.

If stated that the porosity ratio φ is a measure of the ratio between the solid and fluid, then for two-dimensional analysis, it can be stated that:

$$\varphi = \frac{A_s}{A} \tag{5.6}$$

The respective areas can be given as:

$$Total\ Area = A = LH \tag{5.7}$$

$$Area\ of\ Solid = A_s = L(H - D) \tag{5.8}$$

$$Area\ of\ Channel = A_c = DL \tag{5.9}$$

Now, if φ is fixed, then equation (5.6) can be written as:

$$\varphi = \frac{L(H - D)}{LH} \tag{5.10}$$

$$\therefore \varphi = \frac{H - D}{H} \tag{5.11}$$

or

$$\varphi = \left(1 - \frac{D}{H}\right) \tag{5.12}$$

Manipulating to solve for H yields the following:

$$H = \frac{D}{1 - \varphi} \tag{5.13}$$

And from equation (5.7):

$$H = \frac{A}{L} \tag{5.14}$$

Thus combining equation (5.13) and (5.14), it is possible to obtain an equation for D in relation to A, L and φ:

$$\frac{D}{1 - \varphi} = \frac{A}{L} \tag{5.15}$$

$$\therefore D = \frac{A}{L}(1 - \varphi) \tag{5.16}$$

where the geometric ratio of H/L is left as a design variable.

For this model, three different materials with known thermal coefficients were tested, namely silicon, aluminium and copper.

5.5   Numerical Results and Discussion

The flow was simulated at various inlet velocity ranges for porosity ratios of 0.95, 0.9 and 0.8. Three materials with known thermal conductivity were used as the solid material, namely silicon ($k = 124\,W/mK$), aluminium ($k = 237\,W/mK$) and copper ($k = 398\,W/mK$). A constant heat transfer rate of $1W$ was applied to the top and bottom boundary of the solid. This was done for three power-law exponents in order to obtain a comparison between Newtonian and non-Newtonian fluid. For each fluid-solid configuration, geometric optimisation was done so as to obtain a maximum global thermal conductance, from which an optimal duct was found, as illustrated in Figure 5.11. Compared to Figure 5.7, it can be seen that the exit temperature of the fluid is substantially higher in the optimal configuration.

As the Reynolds number is a function of viscosity and is affected by non-Newtonian fluids (see Section 2.2.8), the thermal conductance of the various configurations and fluids is compared according to a fixed inlet velocity.

Figure 5.12 indicates the process of geometric optimisation, whereby an optimal geometric ratio H/L is obtained for silicon for a porosity of 0.95 and 0.05 m/s inlet velocity. The optimal ratio H/L in Figure 5.12 is shown to vary only slightly at 0.6, 0.6 and 0.7 for the pseudoplastic, Newtonian and dilatant fluid respectively. However, the Newtonian characteristics can be seen to have a larger effect on the excess temperature, at these optimum values.

Figure 5.11 - Temperature distribution and boundary layer formation of characteristic geometry
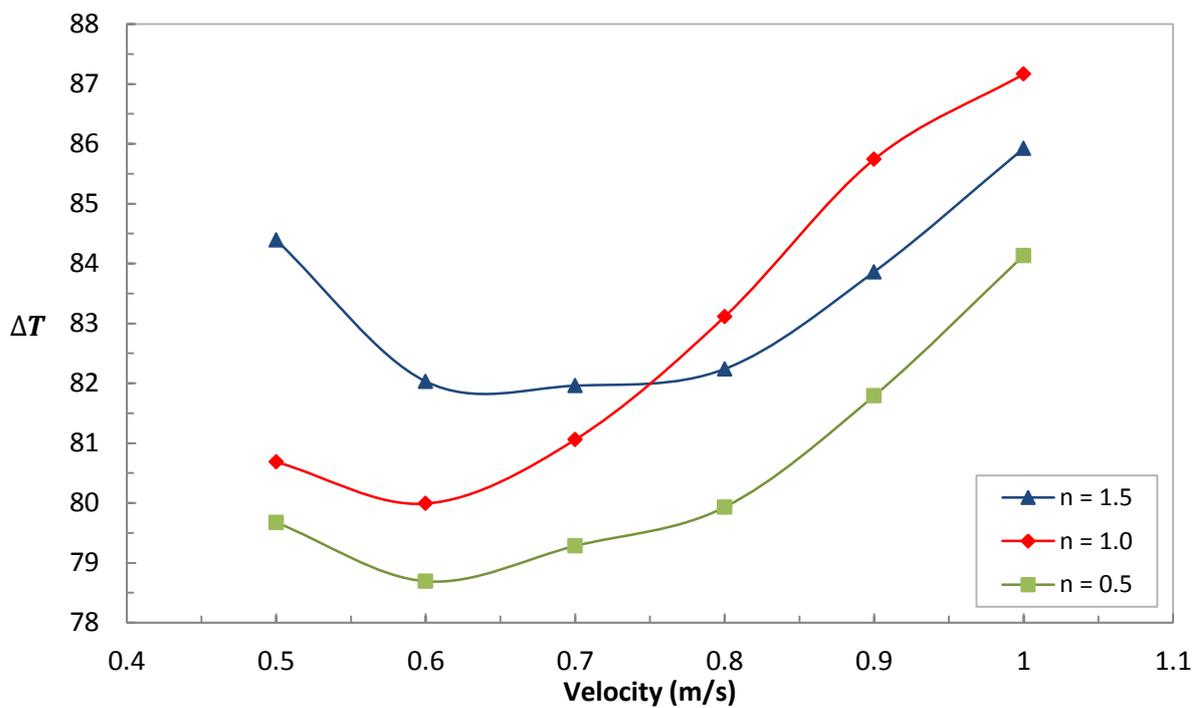
Figure 5.12 - Optimal simple microchannel geometric configuration sensitivity for silicon and three fluids (v_{in}= 0.05 m/s ; φ = 0.95)

The optimal geometric ratio H/L is found to be dependent on the inlet velocity. Figure 5.13 indicates this relation, for an inlet velocity range of 0.014 m/s ($Re_{Newt} \sim 150$) to 0.062 m/s ($Re_{Newt} \sim 700$). As the inlet velocity, and therefore Re, is increased, the thermal boundary layer thickness is reduced and the two boundary layers only converge further downstream. Thus the optimal configuration is lengthened. Consequently, it can be stated that as $v_{in} \sim \infty$ thus $H/L \sim 0$. A further increase of L beyond the optimum will result in an overworked fluid, and ΔT_{min} will systematically rise again as seen in Figure 5.12. In addition, Figure 5.13 indicates that the H/L ratio is less sensitive to inlet velocity increase as the porosity is decreased (i.e. the percentage of fluid in relation to solid is increased). Therefore, at a porosity of 0.8, the H/L ratio remains fairly constant throughout the inlet velocity range.
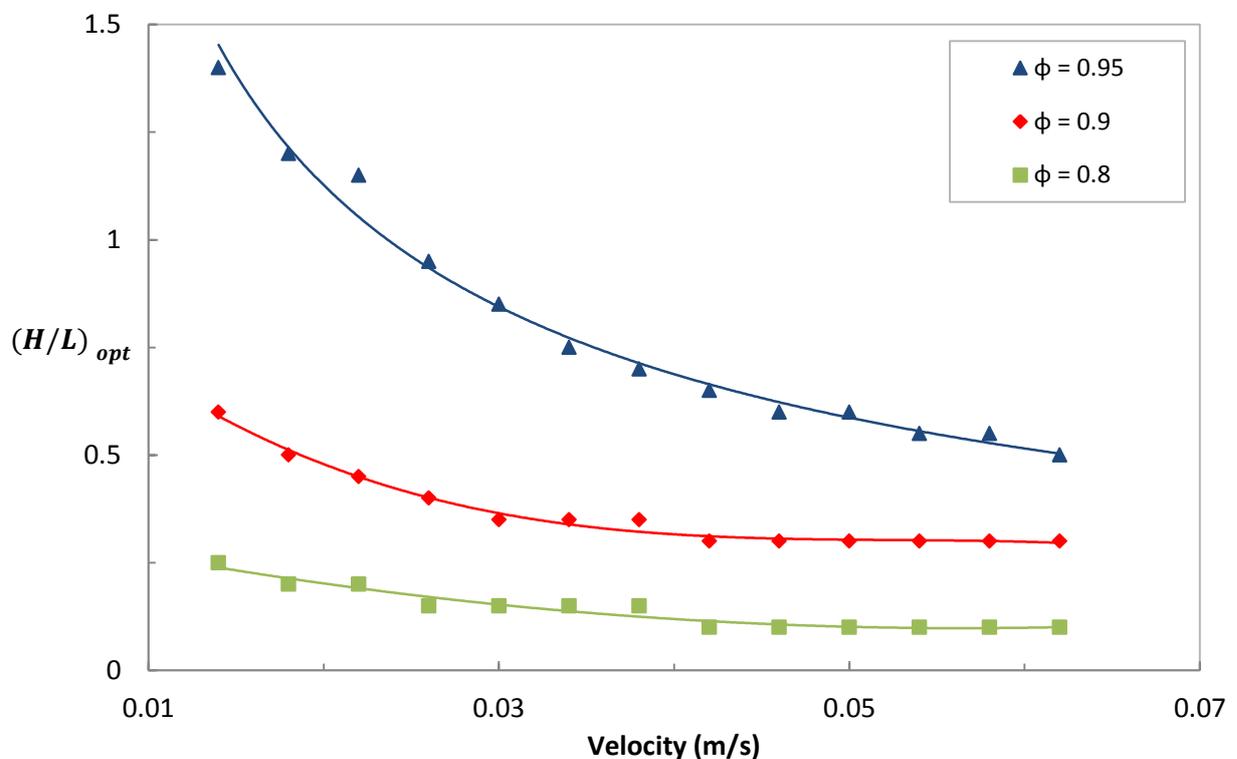
Figure 5.13 - Optimal configuration change as a result of inlet velocity increase for a silicon substrate using a Newtonian working fluid

As the inlet velocity is increased, the optimal ΔT_{min} obtained is seen to decrease. Figure 5.14 indicates the asymptotic nature of the temperature to inlet velocity. It can be concluded, from this figure, that a further increase in inlet velocity will have a decreasing

effect on the excess temperature. As a higher inlet velocity increases the pressure drop and required pumping power, it will merely reduce overall system efficiency.
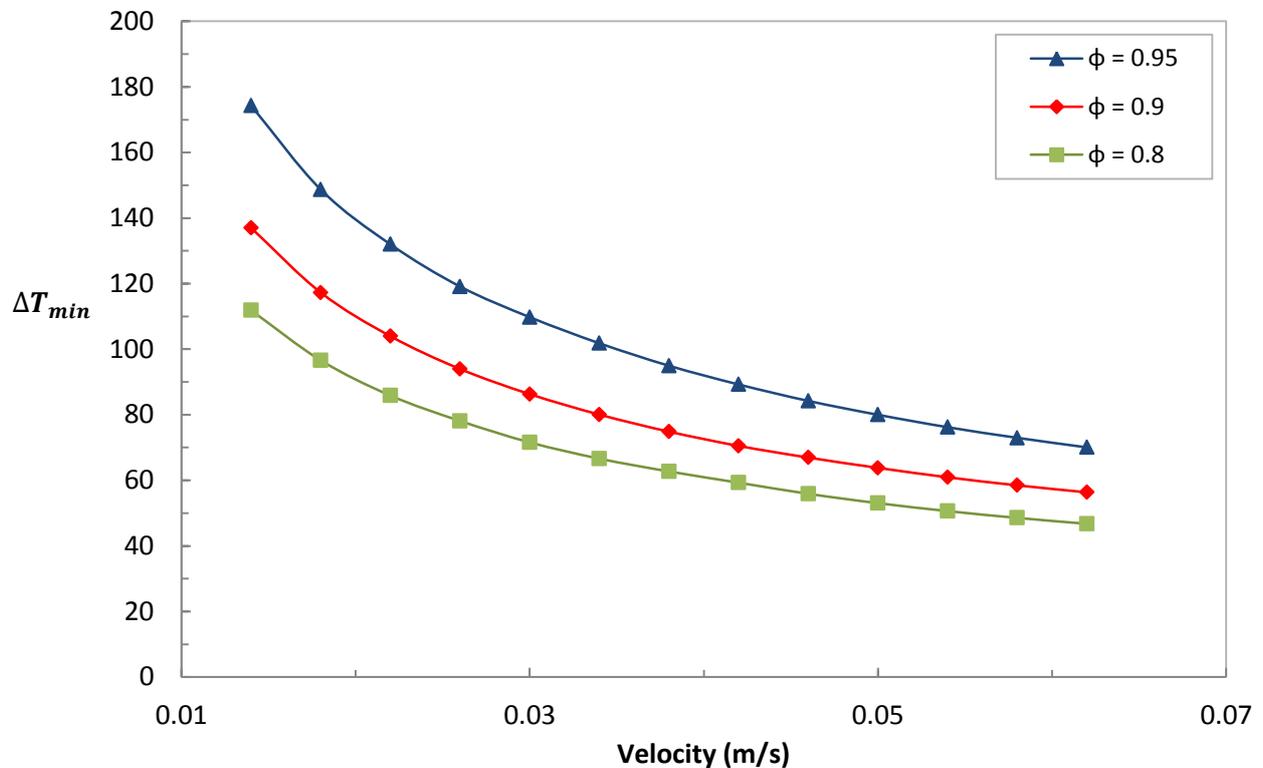
Figure 5.14 - Excess temperature change as a result of inlet velocity and porosity increase for a Newtonian fluid in silicon

The thermal conductance, derived above, for silicon in the range of $0.014\ m/s \leq v_{in} \leq 0.062\ m/s$ and $0.5 \leq n \leq 1.5$ is summarised in Figure 5.15. As the velocity increases, the thermal conductance increases approximately linearly for a Newtonian fluid. However, at higher inlet velocities, the effect of the non-Newtonian characteristics becomes more pronounced. It was found that at higher inlet velocities, the dilatant (shear-thickening; $n = 1.5$) characteristics result in a reduced thermal conductance. This may be attributed to the action of shear-thickening, where force is increased at the wall, which in, turn leads to a thickening of the dilatant fluid. As a result, the heat transfer coefficient at the wall is reduced and heat transfer is significantly impaired, resulting in a decrease in thermal conductance of 13% from the benchmark Newtonian fluid. The opposite effect is found

with a pseudoplastic fluid, whereby a shear-thinning action is experienced at the wall, further increasing the heat transfer coefficient and resulting in an increase in the thermal conductance. Figure 5.15 captures this divergence from the Newtonian fluid. This phenomenon will be accentuated as the viscosity (higher consistency K) of the fluid is increased.
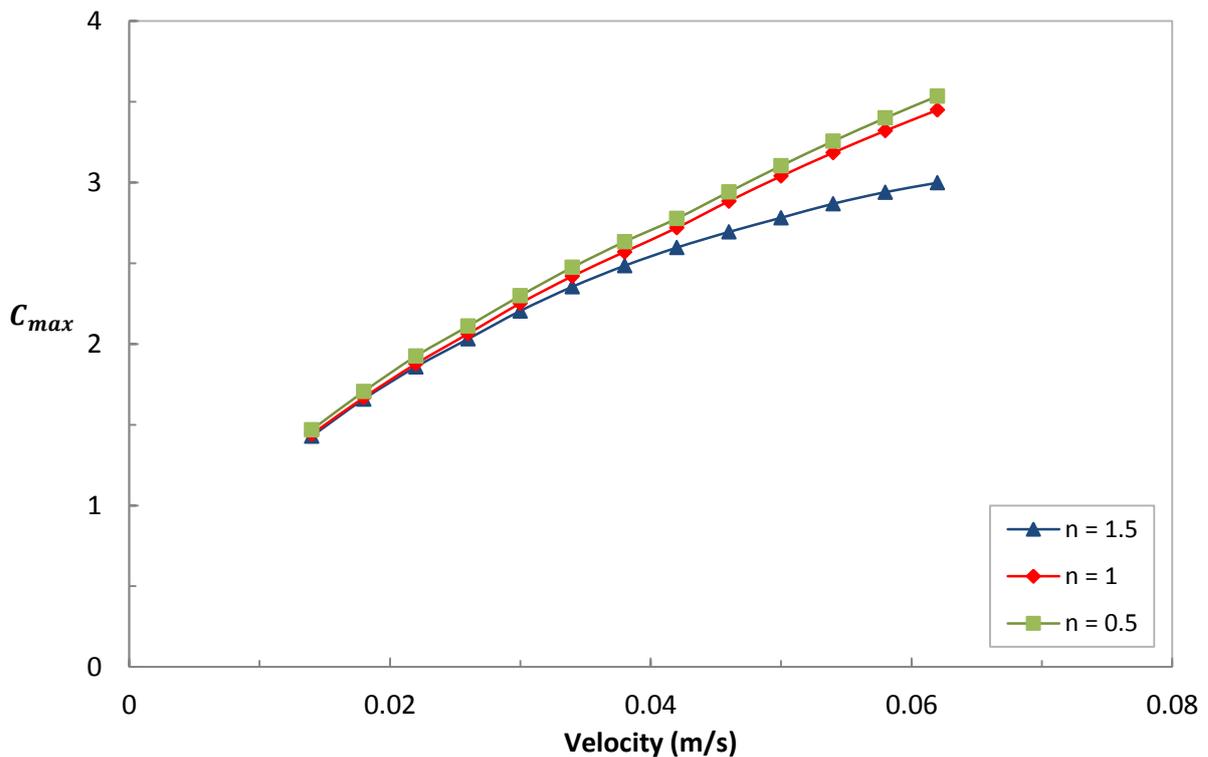
Figure 5.15 - The optimal thermal conductance, in silicon, as a function of the power-law exponents and velocity (φ = 0.8)

A summary of data concerning the sensitivity of thermal conductance to the thermal conductivity of the solid material used in the microchannel, as well as the porosity ratio, is given in Figure 5.16. The figure illustrates that the effect of a variation in the solid material thermal conductivity, from $124\,W/mK$ of silicon to $398\,W/mK$ of copper, accounts for only an approximate 3% increase in global thermal conductance. This leads to the conclusion that the thermal conductance is insensitive to the solid material used. However, as seen in Figure 5.16, a decrease in porosity leads to a higher mass flow rate and, in turn, results in a higher thermal conductance.

Figure 5.16 substantiates the conclusion that the heat transfer mechanism at play within the microchannel is predominantly convectively driven. Although the conductivity of the solid material will contribute, the limiting factor is the convective capability of the working fluid and its ability to remove energy from the system.
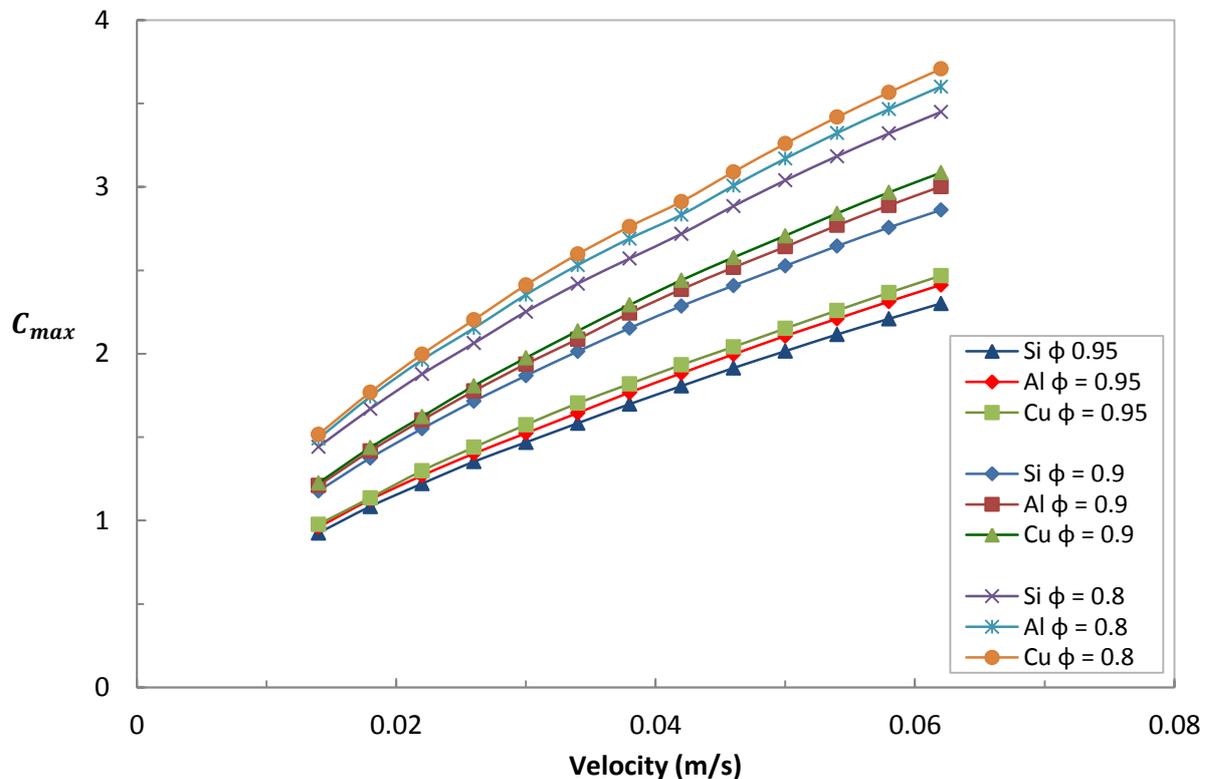
Figure 5.16 - Thermal conductance sensitivity to solid material and porosity variations for a Newtonian fluid

Granted that the increased thermal conductance for a pseudoplastic fluid over a Newtonian fluid is relatively small at low inlet velocities (or Re_{Newt}), the fluid holds the advantage of becoming less viscous as shear force is increased, typically at locations such as a wall or where high flow resistance is encountered. Thus the pseudoplastic nature of the fluid will result in a reduced viscosity, which, in turn, will result in lower flow resistance. As a consequence, the total pressure drop across a system can be significantly reduced, thereby increasing system efficiency. In Figure 5.17 the pressure drop with each fluid was normalized to a reference value, namely the pressure drop for the dilatant fluid at an inlet velocity of $0.062 \ m/s$. Thus Figure 5.17 indicates the substantial increase in normalised pressure drop across the microchannel for a dilatant fluid ($n = 1.5$) compared to a

Newtonian fluid and conversely the reduced pressure drop incurred by a pseudoplastic (n = 0.5) fluid. The pseudoplastic non-Newtonian characteristic results in a significantly lower pressure drop across the microchannel. The increased pressure drop is accompanied by an increased maximum fluid velocity encountered by the dilatant fluid. This phenomenon is discussed and depicted above in Section 5.3 and Figure 5.10 respectively.
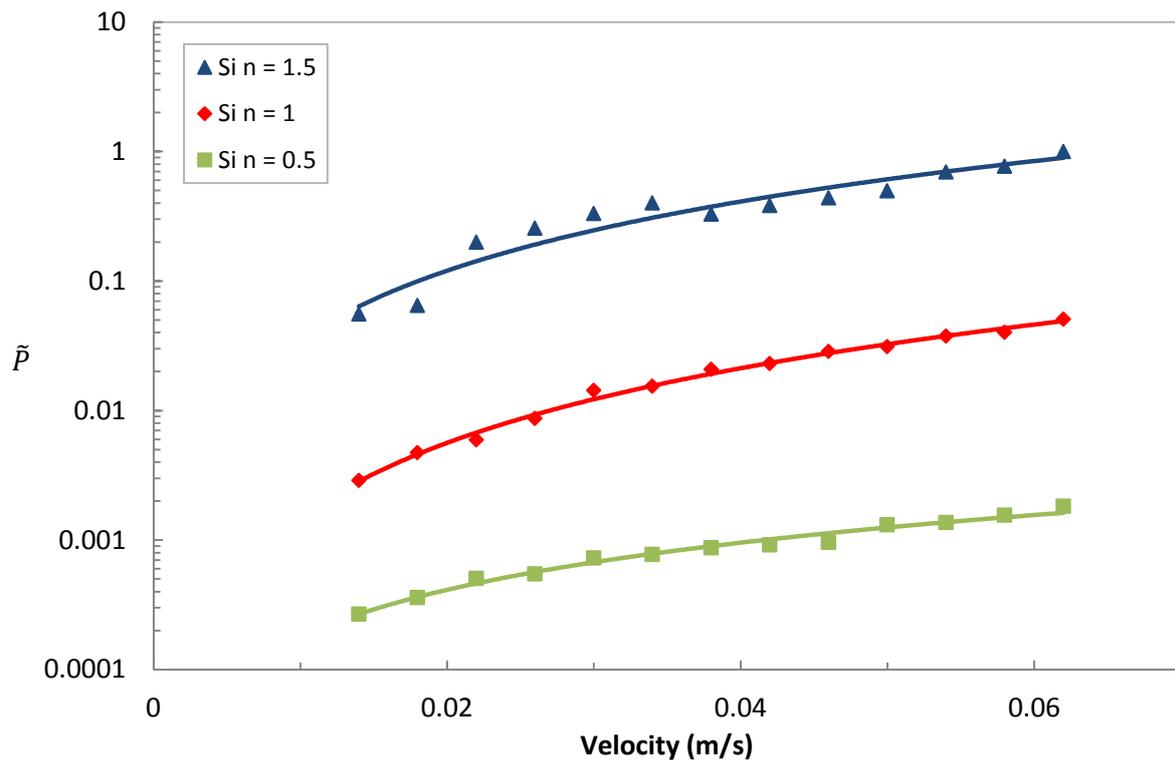
Figure 5.17 - Normalised pressure drop for pseudoplastic versus Newtonian fluid

Chapter 6   2D Complex Microchannel

6.1   Introduction

In Chapter 6, the two-dimensional complex microchannel is discussed. The model, as well as complications encountered, is briefly discussed in Section 6.2 concluding with the reasoning behind abandoning the model.

6.2   Model

Following on the results obtained for the two-dimensional simple microchannel, the non-Newtonian flow in a complex microchannel was to be simulated. This would allow for comparison between the effectiveness of a simple versus complex microchannel. As a result, the configuration depicted in Figure 6.1 was modelled using StarCCM+.
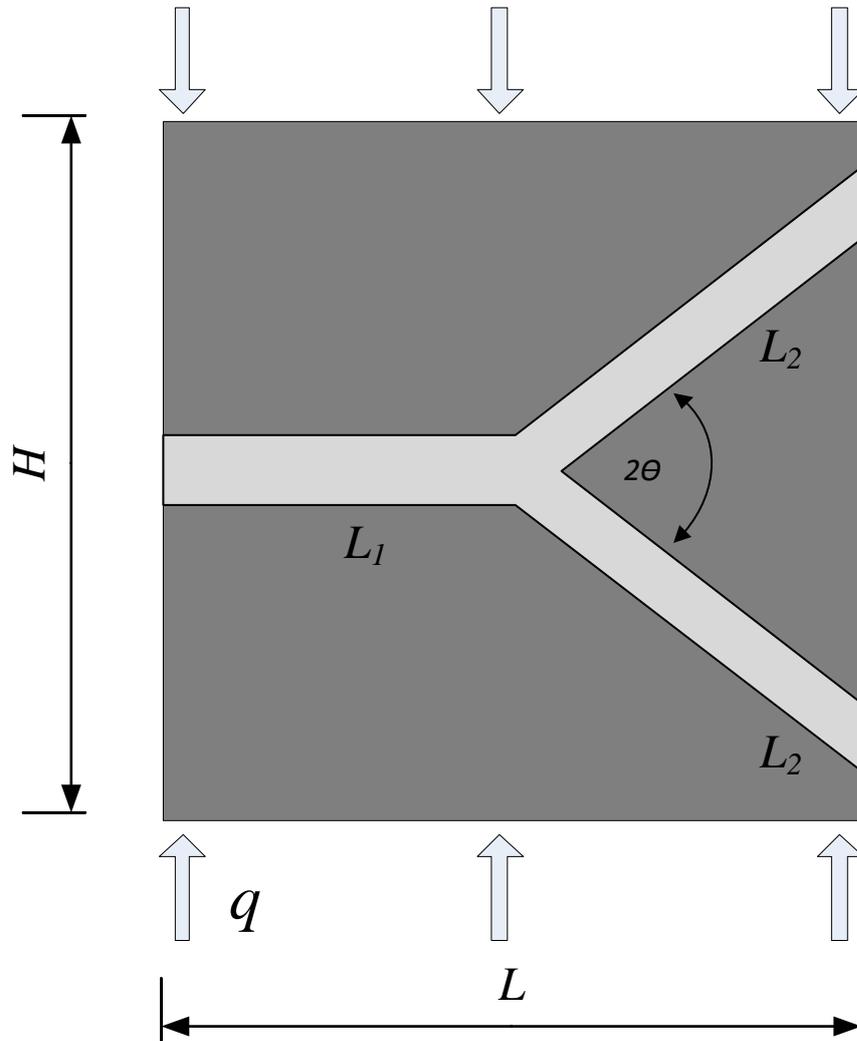
Figure 6.1 - 2D Complex microchannel

Preliminary simulations, however, highlighted a critical error. As the heat flux is supplied from the top and bottom of the microchannel, the fluid bifurcating isolated a triangular shape of the solid material between the fluid branches. Thus a numerical result was obtained as illustrated in Figure 6.2. This further resulted in a dramatic increase in excess temperature as the effective conductive heat transfer area was reduced, rendering only the outer wall of the working fluid effective.

As a result of the preliminary findings, it was decided to abandon any further investigation into two-dimensional complex microchannels and rather advance to a three-dimensional model, where conduction would eliminate the possibility of insulating the triangular region.

61
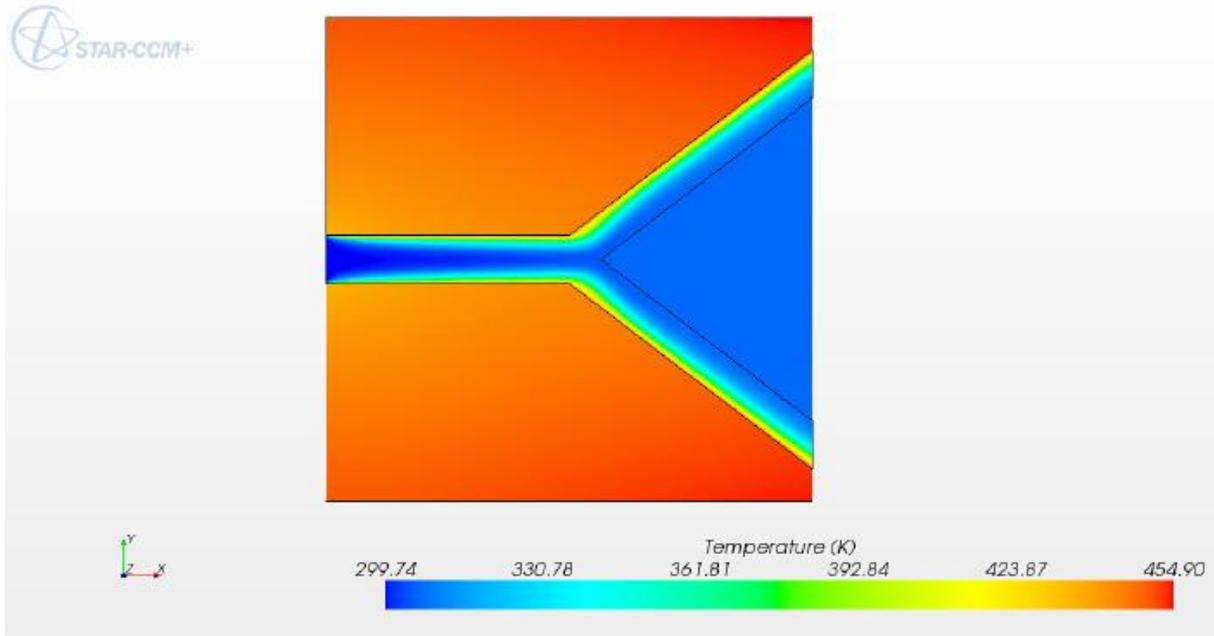
Figure 6.2 - Two-dimensional complex microchannel temperature field

Chapter 7   3D Complex Microchannel

7.1   Introduction

In Chapter 7, the three-dimensional complex microchannel is discussed and simulation results given. The computational model with mesh refinement is discussed in Section 7.2. The complex microchannel is optimised in terms of the geometric ratio (Section 7.4) as well as the bifurcation point in Section 7.5.

7.2   Computational Model

Now consider a three-dimensional microchannel as depicted in Figure 7.1. This physical model can be deconstructed into several individual microchannel elements, as seen in Figure 7.2. The element, depicted in Figure 7.2, has a total volume (V) fixed at $2 \times 10^{-7} \, m^3$. The depth is fixed at $0.002 \, m$ yielding a total microchannel cross-sectional area of $1 \times 10^{-4} \, m^2$. A uniform heat transfer rate (q) is applied to the solid from the top and bottom and heat extracted by the fluid.

The equations for conservation of mass, momentum and energy equations are respectively:

$$\nabla\left(\rho\vec{U}\right) = 0 \tag{7.1}$$

$$\rho\left(\vec{U} \cdot \nabla\vec{U}\right) = -\nabla P + \mu\nabla^2\vec{U} \tag{7.2}$$

$$\rho C_p\left(\vec{U} \cdot \nabla T\right) = k_f\nabla^2 T \tag{7.3}$$

where $\nabla^2 = \partial^2/\partial x^2 + \partial^2/\partial y^2 + \partial^2/\partial z^2$, the Cartesian coordinates (x, y) and velocity component (u, v) are defined in Figure 7.2. The variables are defined in the nomenclature. For a volume occupied by a solid, the momentum (7.2) and energy (7.3) reduce to:

$$\vec{U} = 0 \tag{7.4}$$

$$k_s\nabla^2 T = 0 \tag{7.5}$$
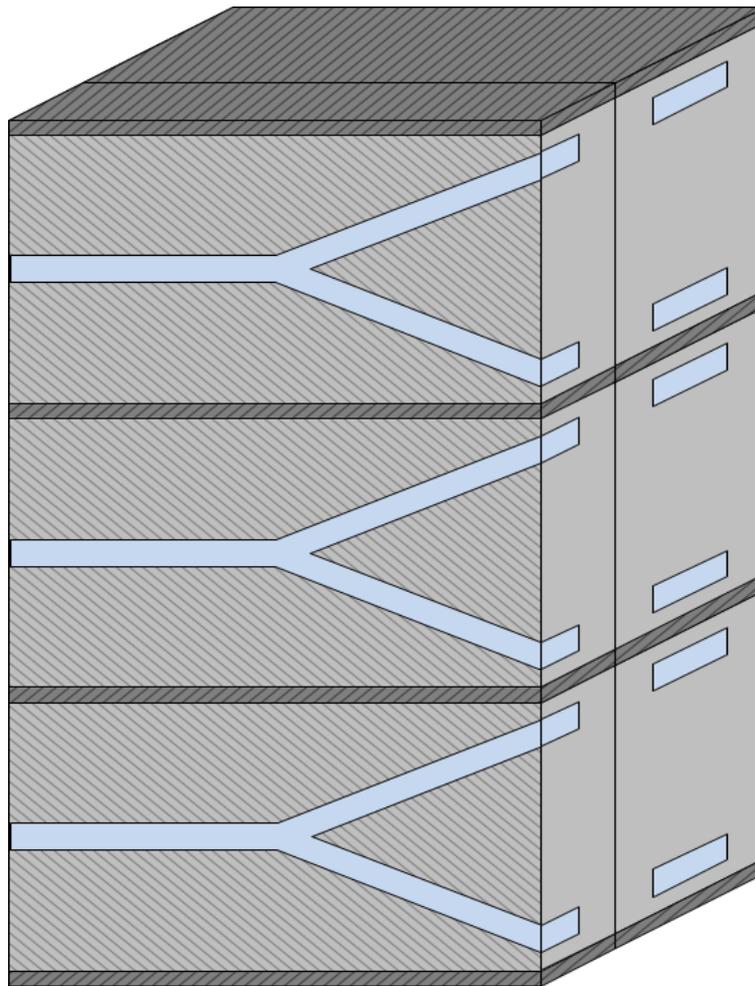
Figure 7.1 - Complex microchannel, the physical model
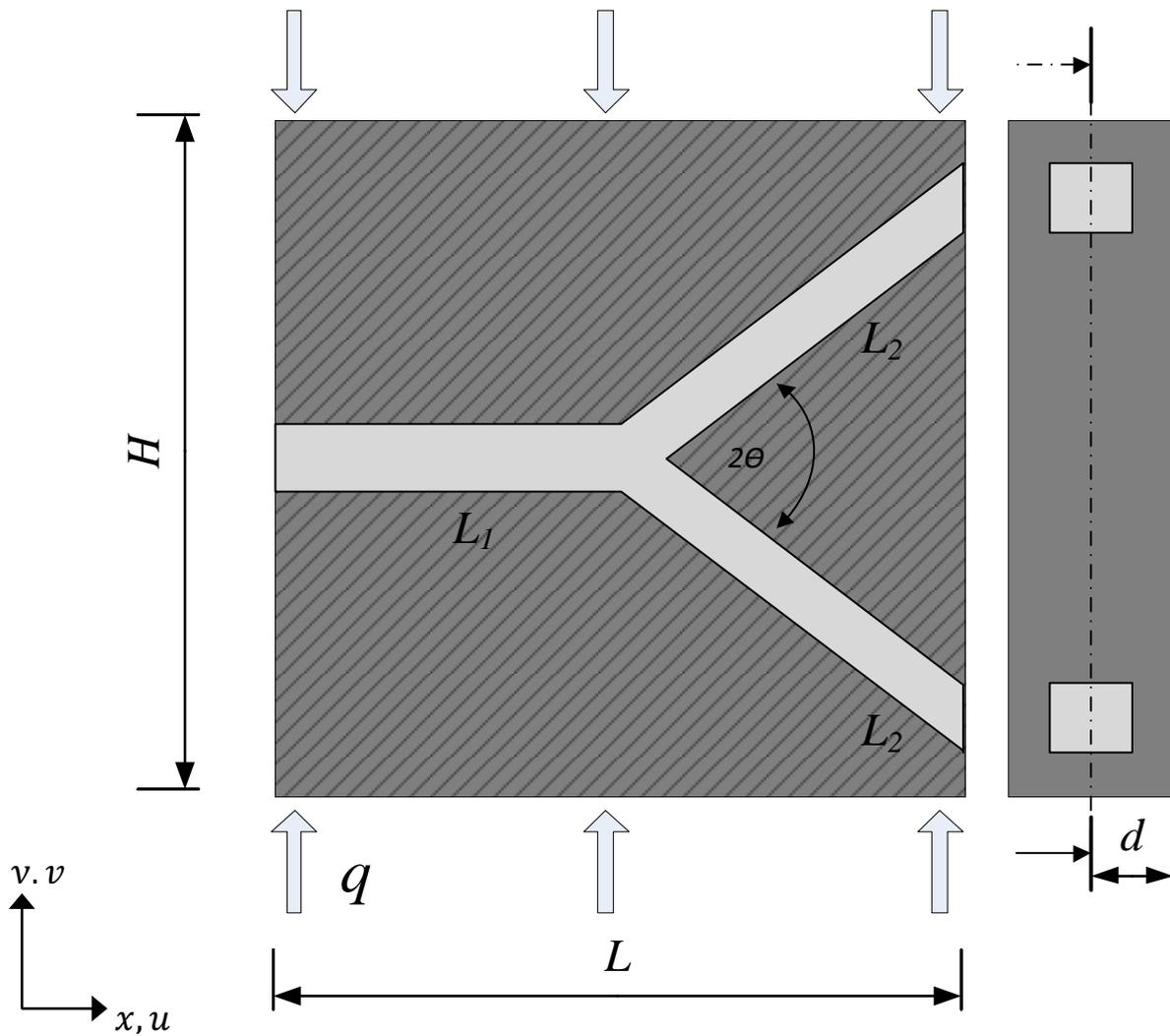
Figure 7.2 - Section view of complex microchannel, the computational model

A three-dimensional model was created in StarCCM+, accurately resembling Figure 7.2, in order to simulate non-Newtonian flow through the complex microchannel. As the microchannel element was symmetrical in the z-axis, only half of the channel need be simulated, as illustrated in Figure 7.3. This dramatically reduced computational time, with no detrimental effect on the accuracy.

The boundary conditions for the model were set as follows:

The inlet boundary condition was set to inlet velocity:

$$\vec{U} = u_{in} \qquad (7.6)$$

65

$$T = T_{in} \tag{7.7}$$

No-slip boundary conditions were set on the fluid-wall interface. A pressure outlet boundary was defined for both the exits. The thermal boundary condition consisting of uniform heat flux, calculated to ensure a constant total heat transfer rate ($1W$), was applied to the top and bottom of the solid:

$$k_s L \frac{\partial T}{\partial y} = -q' \tag{7.8}$$

The remaining four boundaries (sides) were modelled as symmetry planes with free slip, as to represent an infinite channel, thus

$$\frac{\partial \vec{U}}{\partial (x, y)} = 0 \tag{7.9}$$

$$\frac{\partial T}{\partial (x, y)} = 0 \tag{7.10}$$
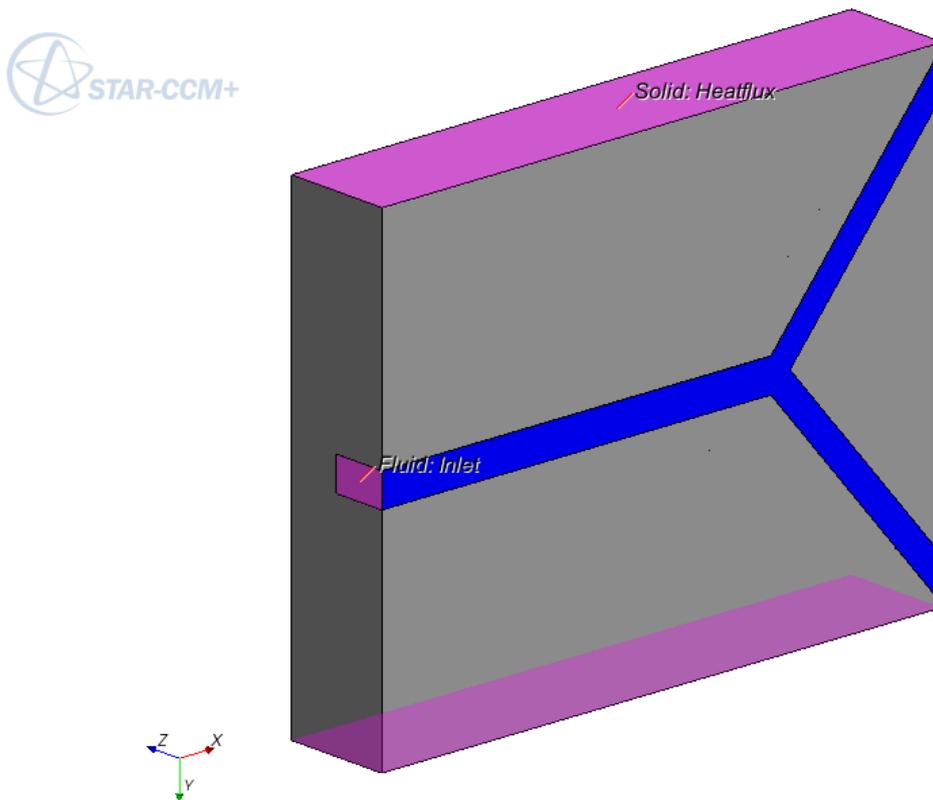
Figure 7.3 - Complex microchannel numerical model

Convergence of the solution was met once the residuals, which included energy, continuity and velocity, were below 10^{-4}.

7.2.1 Mesh Refinement

A mesh refinement study, which included extensive mesh refinement and optimisation, was conducted. As the three-dimensional model occupied a larger volume, along with the bifurcation of the fluid, the model required significantly more cells than that of a two-dimensional case. For this reason, aggressive mesh optimisation techniques were incorporated to reduce the required cells and refine where needed.

It was found that a structured mesh required less nodes than an unstructured mesh, with the additional advantage of lower cell skewness, and was therefore used. The StarCCM+ trimmer mesher was used with fluid-region refinement, prism layers and high cell growth. In order to ensure a high quality mesh at the sharp angles of the bifurcation, prism-layer reduction was enabled, which decreased the amount of prism layers at an acute angle. This ensured that a higher quality mesh was preserved.

Once it was certain that the prism-layers captured the entire boundary layer, the cell size was systematically reduced in order to obtain mesh independence. Table 7.1 gives a summary of the mesh refinement study. Using the StarCCM+ mesher, it was possible to set a base size, from which all the cell sizes were calculated. It was found that the simulation results converged for a mesh of more than 146 245 cells, accounting for a mere 0.08% change if the number of cells was doubled, as opposed to a 0.45% change in result if the number of cells was to be halved. Thus a mesh of 146 245 cells was deemed sufficient to obtain mesh independence.

The result was a mesh as depicted in Figure 7.4. As can be seen, extensive mesh refinement was conducted around the fluid region, with the cell size increasing further into the solid region. Figure 7.5 indicates the mesh refinement and prism layers in the fluid region.

Table 7.1   Mesh refinement study: complex microchannel

Cells	Base size (mm)	ΔT_{min}	% change
26043	1.4	12.677368	-
32210	1.2	12.560608	0.921
46424	1.0	12.710358	-1.192
77903	0.8	12.684174	0.206
146245	**0.6**	**12.626709**	**0.453**
338342	0.4	12.636658	-0.079
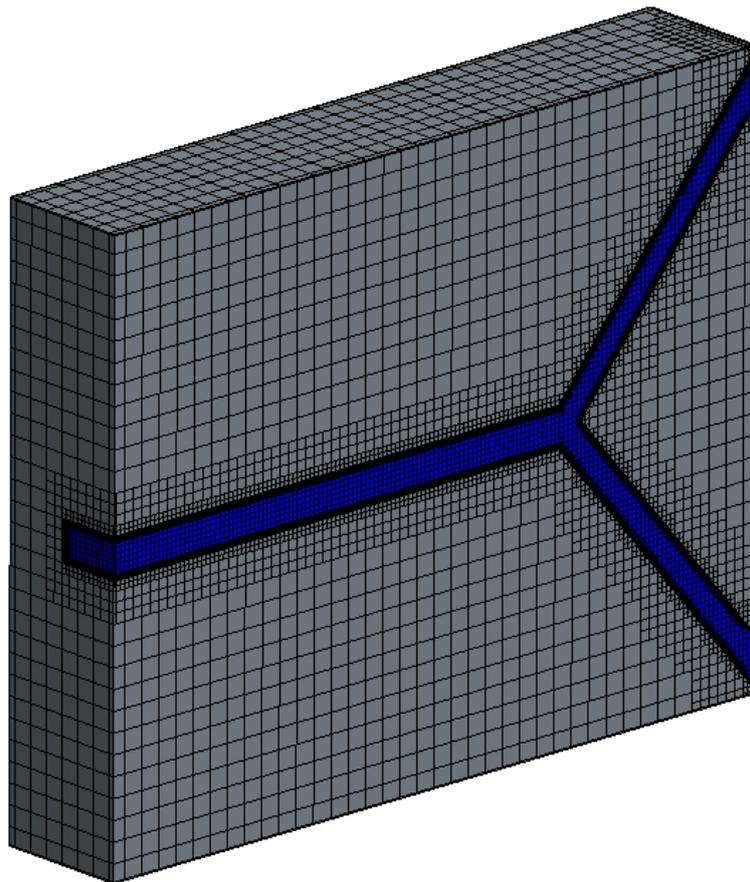
Figure 7.4 - Complex microchannel mesh
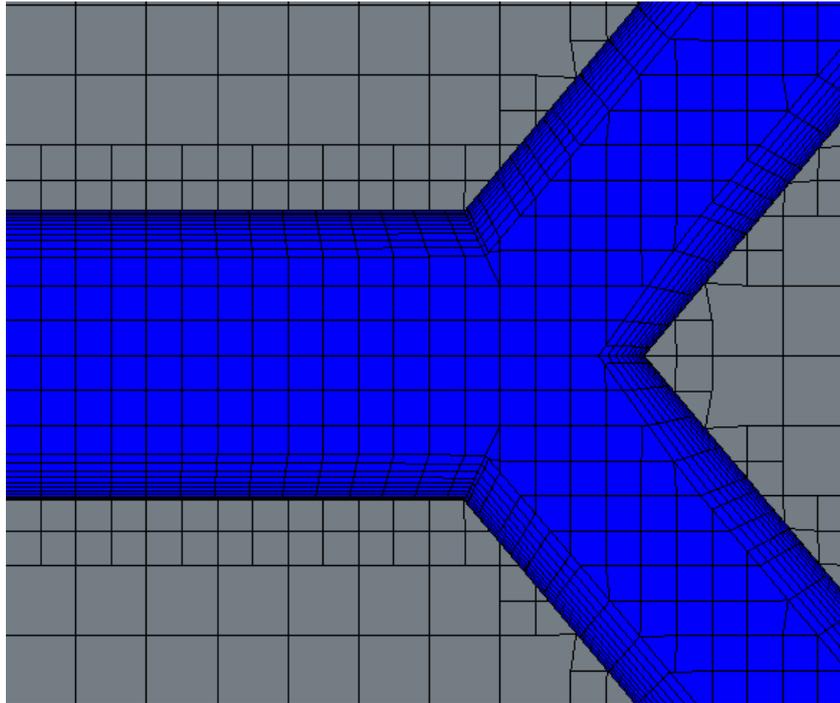
Figure 7.5 - Mesh refinement in the fluid region

7.3 Complex Microchannel Optimisation

In this section, an attempt was made to optimise the complex microchannel, as to maximise the thermal conductance, and subsequently, investigate the effectiveness of a non-Newtonian working fluid compared to that of a Newtonian. The complex microchannel can be optimised in two fashions, namely by optimising the bifurcation point, and secondly, by optimising the geometric ratio. Both these cases will be investigated independently. However, as the mathematical model is generic, it is discussed in Section 7.3.1.

7.3.1 Mathematical Formulation

The fluid channel branches are fixed at the optimum position to minimise flow resistance, which has been found to be $L_1 = 2^{1/3}L_2$ with the duct diameters being $D_1 = 2^{1/3}D_2$ [24], [29], [30] for laminar flow. The height (H) and length (L) are design parameters and are free

to morph, subject to the global constraint of volume. The ratio between the fluid and solid material is constrained by the porosity ratio (φ).

The length of the branches is given by:

$$L_1 = \alpha L \tag{7.11}$$

$$L_{2_x} = L - L_1 \tag{7.12}$$

$$L_{2_y} = \frac{H}{2} - D_2$$

$$\approx 0.9\frac{H}{2} \tag{7.13}$$

The angle of bifurcation is represented by θ

$$\theta = \tan^-\left(\frac{L_{2_y}}{L_{2_x}}\right) \tag{7.14}$$

Substituting equations (7.12) and (7.13) into (7.14) yields the following angle:

$$\theta = \tan^-\left(\frac{0.45H}{(1-\alpha)L}\right) \tag{7.15}$$

As indicated above, the porosity ratio (φ) is a measure of the ratio between the solid and fluid:

$$\varphi = \frac{V_s}{V_t} \tag{7.16}$$

Where the solid volume can be given by

$$V_s = V_t - V_c \tag{7.17}$$

Now, if the depth of the section of microchannel is d, and the fluid channel $d/2$ (see Figure 7.2), then V_s can be approximated as:

$$V_s \approx HLd - \frac{d}{2}(D_1L_1 + 2D_2L_2) \tag{7.18}$$

Taking into account the optimal diameter and length ratio given above:

$$D_1 = 2^{1/3}D_2 \tag{7.19}$$

$$L_1 = 2^{1/3}L_2 \tag{7.20}$$

From equation (7.12), the length L_1 can be written as:

$$\left(1 + \frac{\cos\theta}{2^{1/3}}\right)L_1 = L \tag{7.21}$$

$$\alpha L_1 = L \tag{7.22}$$

$$\therefore \alpha = \left(1 + \frac{\cos\theta}{2^{1/3}}\right) \tag{7.23}$$

Substituting into equation (7.18) yields:

$$V_s = HLd - \frac{d}{2}D_2\left(2^{1/3}\alpha L + \frac{2(1-\alpha)L}{\cos\theta}\right) \tag{7.24}$$

Thus, equation (7.16) can be manipulated to:

$$\varphi = 1 - \frac{D_2}{2H}\left(2^{1/3}\alpha + \frac{2(1-\alpha)}{\cos\theta}\right) \tag{7.25}$$

As the porosity value is fixed, it can be used to determine the diameter of the duct branches. Therefore, equation (7.25) can be rewritten as follows:

$$D_2 = \frac{2H(1-\varphi)}{\left(2^{1/3}\alpha + \frac{2(1-\alpha)}{\cos\theta}\right)} \tag{7.26}$$

71

7.4   Geometric Ratio Optimisation

As the objective is to obtain an optimal geometric ratio in order to maximise thermal conductance, subject to the global constraints of volume and porosity, the ratio H/L is set as a design parameter and free to morph. The bifurcation point, defined by the parameter α, is constrained to a constant value subject to equation (7.20) in order to minimise flow resistance.

The parameters θ and α are design parameters calculated in the macro, using equation (7.15) and (7.23) respectively. However, as they are interdependent, the solutions to the parameters are solved iteratively in the macro, to a specified accuracy.

An extract from the macro is given in Figure 7.6 below:

```
51        while (err > 0.00001) {
52
53            alpha1 = 1/(1 + (Math.cos(Math.toRadians(Theta)))/(Math.pow(2,des)));
54
55            double temp = 0.35*Hb/((1-alpha1)*Lb);
56
57            Theta = Math.toDegrees(Math.atan(temp));
58
59            err = Math.abs((alpha-alpha1)/alpha1);
60
61            alpha = alpha1;
62
63        }
```

Figure 7.6 - Macro Extract

Figure 7.7 and Figure 7.8 illustrate the variation in geometric ratio, whereby the geometry is elongated. The bifurcation point is, however, maintained constant to satisfy the equation (7.20).

72
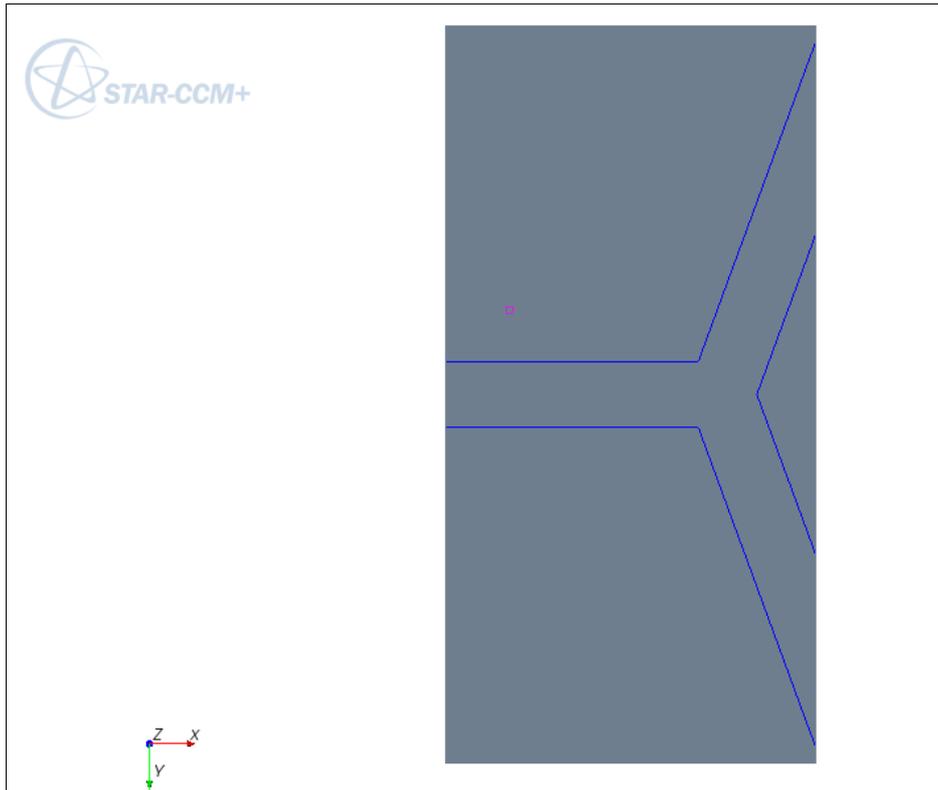
Figure 7.7 - Complex microchannel H/L = 2
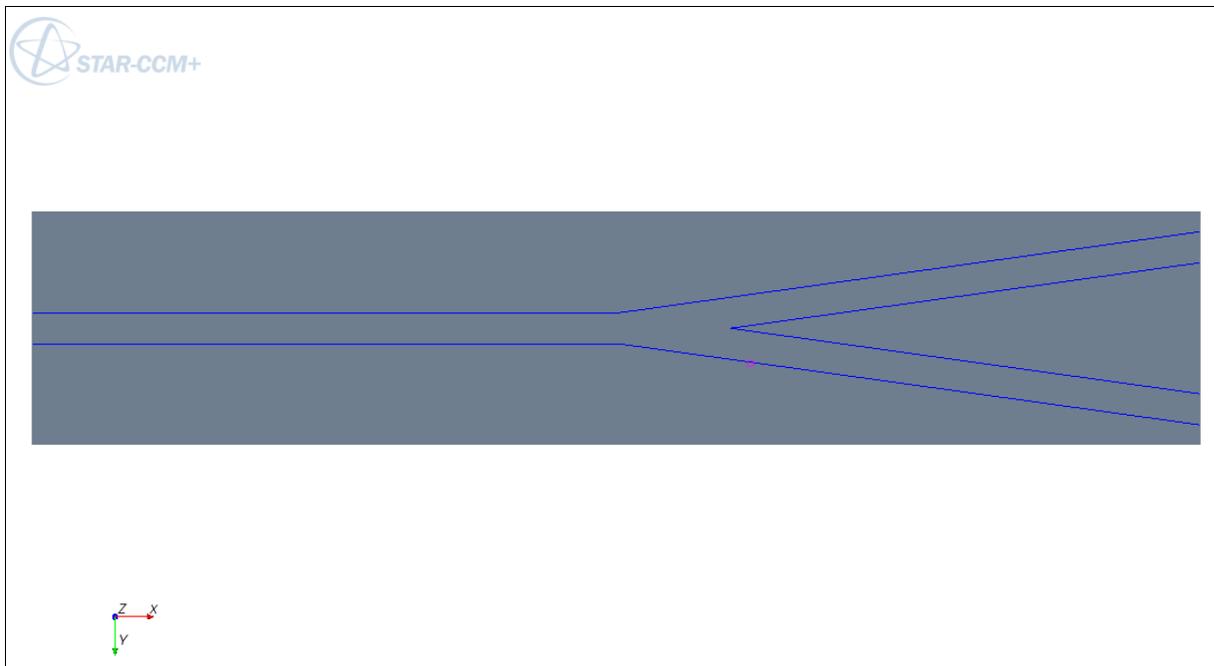
Figure 7.8 - Complex microchannel H/L = 0.2

7.4.1 Numerical Results and Discussion

The three-dimensional flow through a complex microchannel was simulated at various inlet velocity ranges for porosity ratios of 0.95, 0.9 and 0.85. The material used for the microchannel was silicon with known thermal conductivity ($k = 124 \, W/mK$). A constant heat transfer rate ($1W$) was applied to the top and bottom boundary of the solid. This was done for the same three fluids, as in the simple case, with power-law exponents ($n = 0.5; \; n = 1; \; n = 1.5$) in order to obtain a comparison between Newtonian ($n = 1$) and non-Newtonian fluids. Geometric optimisation was done for each fluid-solid configuration to obtain a maximum global thermal conductance. The numerical result obtained is an optimised complex microchannel, such as depicted in Figure 7.9.
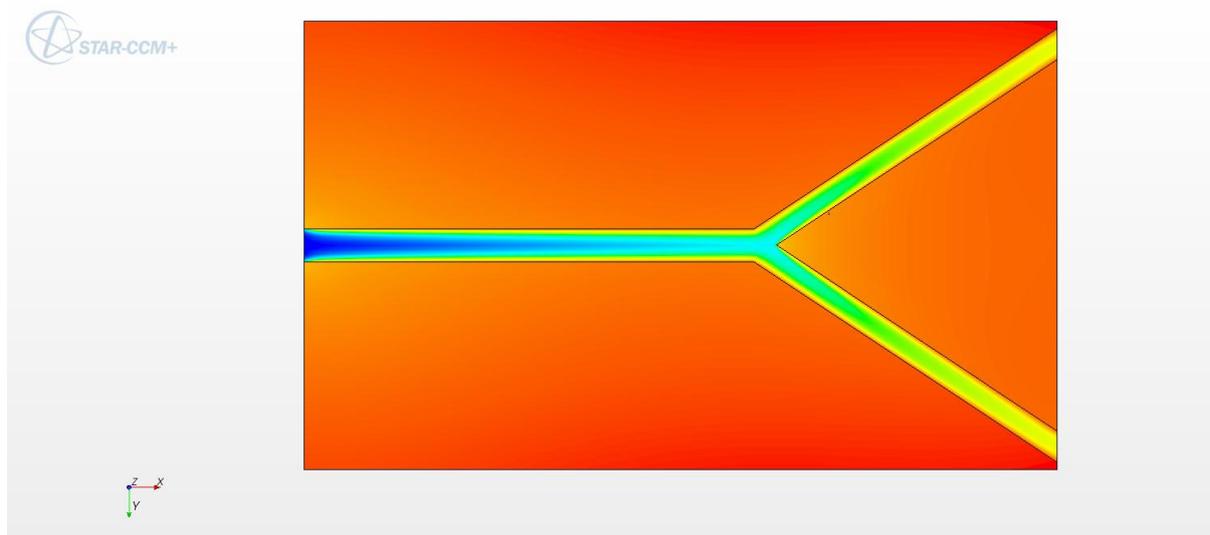
Figure 7.9 - Optimal geometry

Figure 7.10 indicates the process of geometric optimisation whereby an optimal geometric ratio H/L for the complex microchannel is obtained, for a porosity of 0.9 and an inlet velocity of 0.07 m/s. From the figure, it can be seen that the optimal geometric ratio H/L is 0.6 for both the pseudoplastic ($n = 0.5$) and Newtonian ($n = 1$) fluid, where it is 0.7 for a dilatant fluid ($n = 1.5$). This is only a slight variation. However, the non-Newtonian characteristic of the dilatant fluid can be seen to have a significant effect on the excess temperature.

As in the case of the two-dimensional simple microchannel, the optimal complex geometric ratio is seen to be highly dependent on the inlet velocity. Figure 7.11 indicates this relation for an inlet velocity range from $0.3 \ m/s$ ($Re_{new} \sim 340$) to $0.1 \ m/s$ ($Re_{new} \sim 1100$). Again it can be seen that the optimal geometry is lengthened with the increase in inlet velocity ($v_{in} \sim \infty$ thus $H/L \sim 0$). As the heat transfer is convectively driven, a decrease of L below the optimum will allow an underworked fluid to escape the heat transfer domain, resulting in an increased excess temperature (ΔT_{min}) in the conductive space. Similarly, a further increase of L beyond the optimum will result in an overworked fluid, and ΔT_{min} will systematically rise again as seen in Figure 7.10.
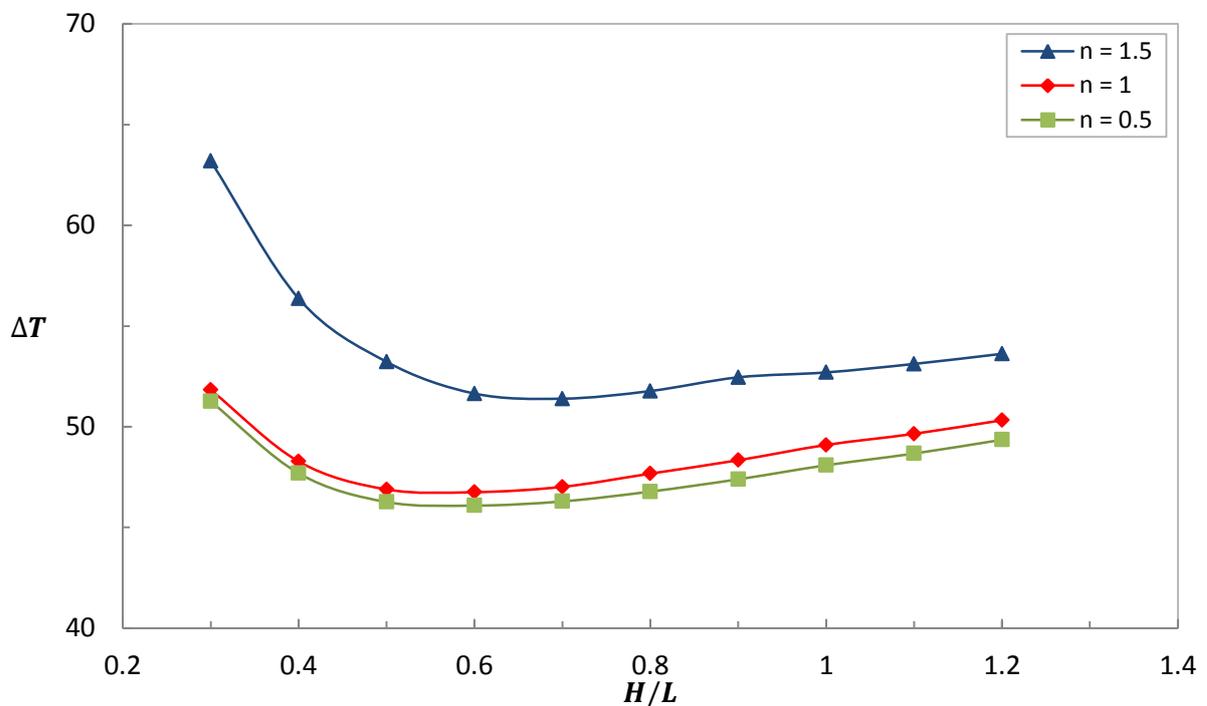
Figure 7.10 - Excess temperature versus geometric ratio at inlet velocity of 0.07 $\boldsymbol{m/s}$ and $\boldsymbol{\varphi}$ = 0.9
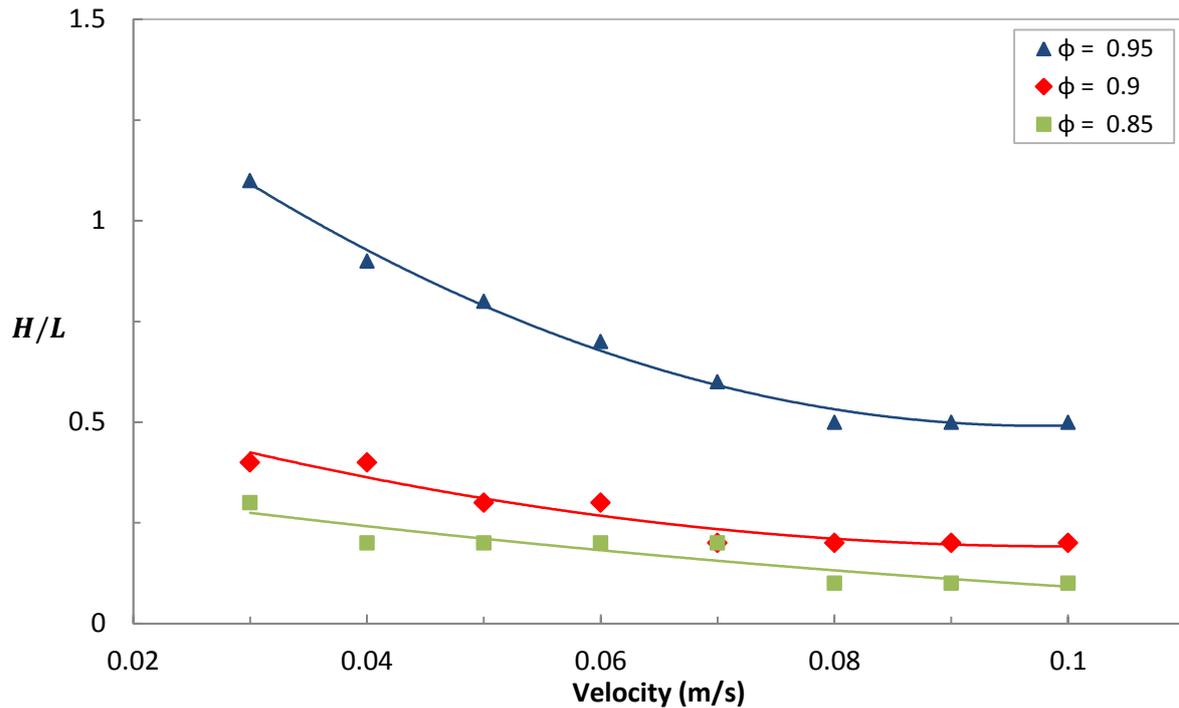
Figure 7.11 - Optimal geometric ratio as a function of inlet velocity for a Newtonian fluid

It is clear from Figure 7.11 that the H/L ratio is less sensitive to variation in inlet velocity as the porosity is decreased (i.e. the percentage of fluid in relation to solid is increased). Thus at a porosity of 0.85, the H/L ratio remains fairly constant throughout the inlet velocity range.

It was found that as the inlet velocity increased, the optimal ΔT_{min} obtained decreased, however, in Figure 7.12, this relation can be seen to have an asymptotic nature. Thus it can be concluded that an additional increase in inlet velocity will have a reduced effect on ΔT_{min}. As a higher inlet velocity increases the pressure drop and required pumping power, it will merely reduce overall system efficiency.
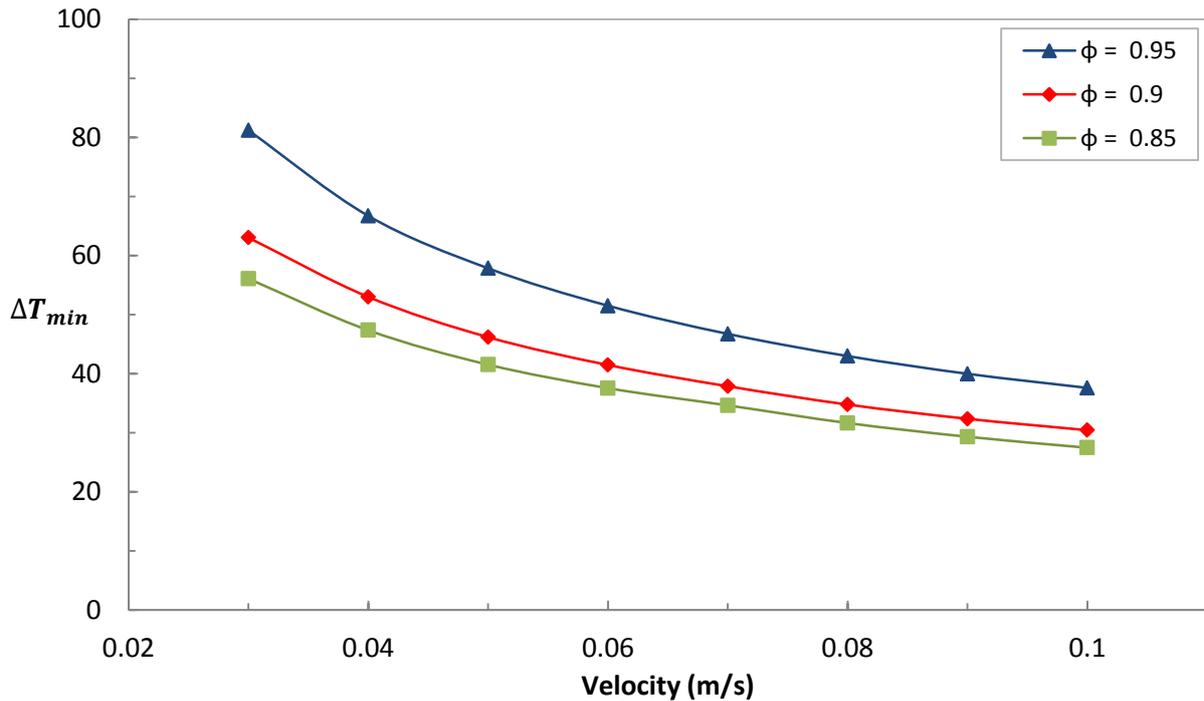
Figure 7.12 - Optimal ΔT_{min} versus inlet velocity for a Newtonian fluid at various porosity ratios

The thermal conductance, derived above for the range of $0.03\ m/s \leq v_{in} \leq 0.1\ m/s$ and $0.5 \leq n \leq 1.5$, is summarised in Figure 7.13 and Figure 7.14. At lower inlet velocities, the thermal conductance increases approximately linearly. However, at higher inlet velocities, the effect of the non-Newtonian characteristics becomes more pronounced. It can be seen from Figure 7.13 that the dilatant fluid ($n = 0.5$) characteristics result in a reduced thermal conductivity. This may be attributed to the action of shear-thickening, where at the wall, the shear force is increased, which leads to a thickening of the dilatant fluid. The bifurcation of the complex structure leads to a further increase in shear force, decreasing the thermal conductance of the fluid. As a result, the heat transfer coefficient at the wall is reduced and heat transfer is significantly impaired, causing a decrease in thermal conductance of 23% from the benchmark Newtonian fluid.

In contrast, the shear-thinning action experienced at the wall for a pseudoplastic fluid, leads to an increased fluid velocity at the thermal boundary, therefore increasing the convective heat transfer coefficient at the fluid-solid interface. As seen in Figure 7.13, this results in an increased thermal conductance. This divergence pattern between the dilatant and

pseudoplastic fluid will be accentuated as the viscosity (higher consistency K) of the fluid is increased.
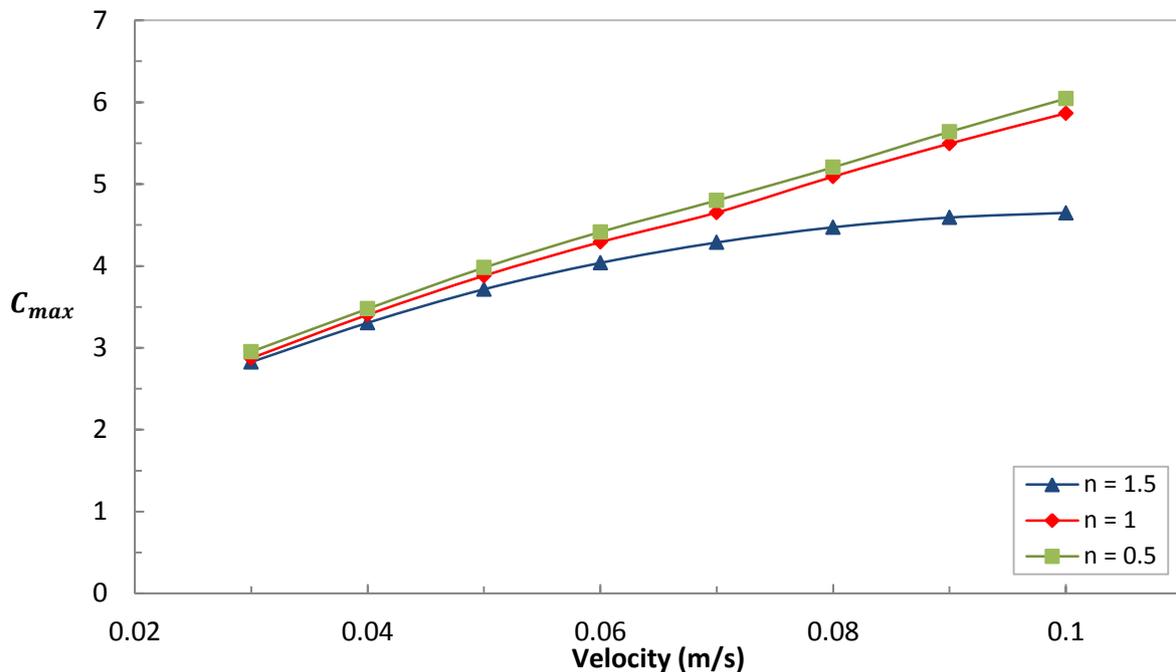
Figure 7.13 - Divergence in thermal conductance for the power-law exponents as a function of inlet velocity ($\varphi = 0.85$)

Figure 7.14 indicates the effect of porosity on the thermal conductance. Reducing the porosity leads to a higher mass flow rate, which, in turn, leads to a higher thermal conductance. It can be noted that the divergence pattern between the fluids is similar for all porosities, however, slightly more pronounced at the lower value of 0.85.

Figure 7.15 illustrates the significant increase in normalised pressure drop across the complex microchannel for a dilatant fluid ($n = 1.5$) compared to a Newtonian fluid and conversely the dramatic reduction in normalised pressure drop ($\sim 70\%$) for a pseudoplastic fluid ($n = 0.5$). Therefore, it can be stated for a complex microchannel as well, that the total pressure drop across a system can be significantly reduced with the use of a pseudoplastic fluid. Coupled with the increase in thermal conductance achieved by a pseudoplastic fluid, albeit small, such reduced pressure drop may bring about significantly increased system efficiency.
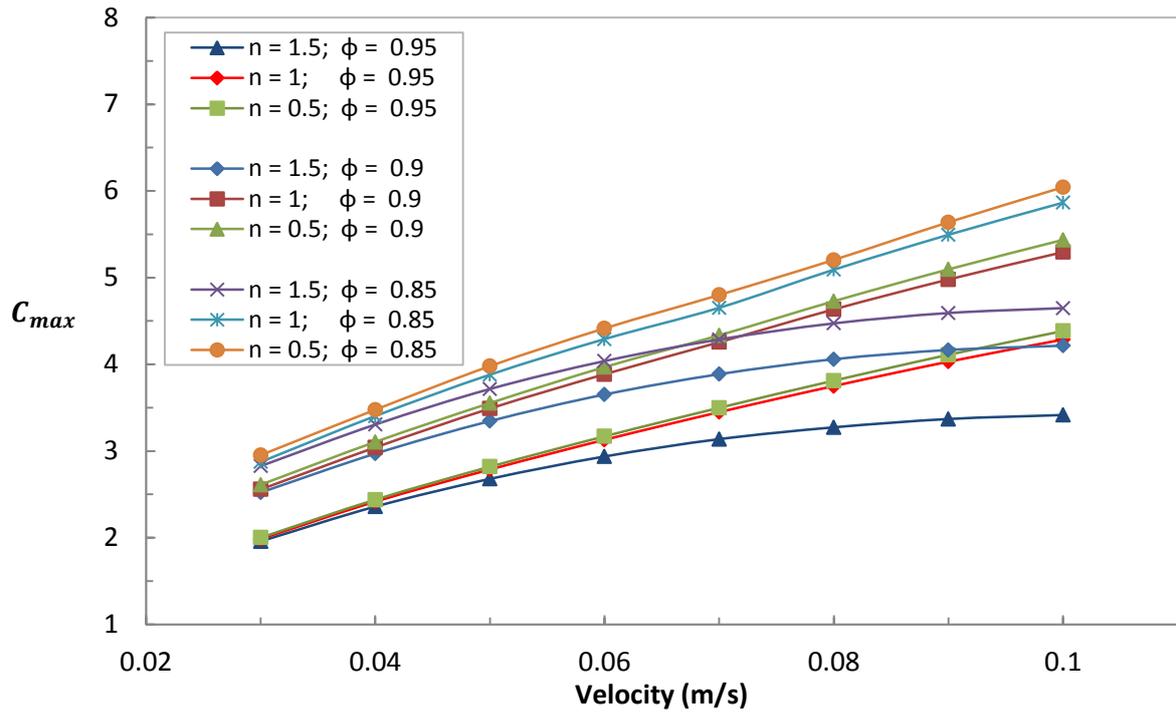
Figure 7.14 - A comparison of thermal conductance between various configurations as a function of velocity
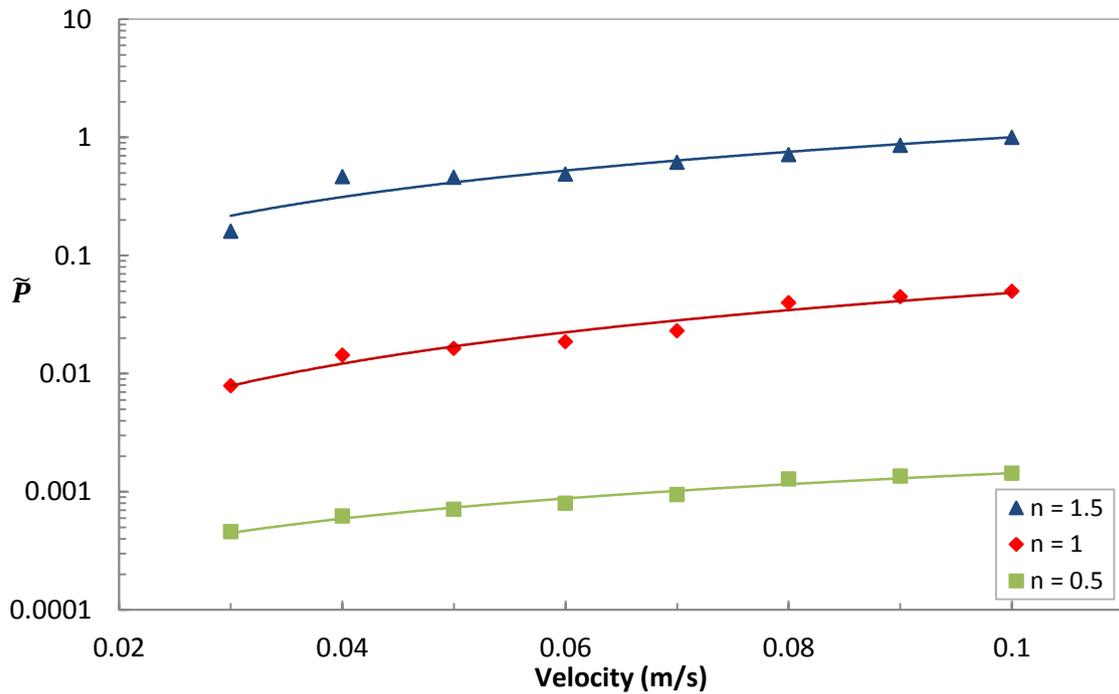
Figure 7.15 - Normalised pressure drop versus inlet velocity, $\varphi = 0.95$

7.5   Bifurcation Point Optimisation

For the bifurcation point optimisation, consider a microchannel element as depicted in Figure 7.2, where the geometric ratio H/L is fixed to a constant value, leaving the bifurcation point (L_1/L_2) free to morph. The objective is to find the optimal point at which bifurcation occurs, in order to maximise thermal conductance subject to the constraint of volume and aspect ratio, and so doing, investigate the effect of utilising a non-Newtonian working fluid.

The model used for the complex ratio optimisation was modified to allow the factor α, which defined the bifurcation point, to be manipulated. The geometric ratio H/L was set to a constant value of 0.8.

Thus from Section 7.3.1, the diameter of the branches are given as:

$$D_2 = \frac{(1-\varphi)H}{\left(2^{1/3}\alpha + \frac{2(1-\alpha)}{cos\theta}\right)} = 2^{-1/3}D_1 \tag{7.27}$$

where α is a design parameter and free to morph.

Figure 7.16 and Figure 7.17 below indicate the bounds in which optimisation was conducted. The optimal value for α in each case was found to be between 0.01 and 0.9, in which it was allowed to vary.
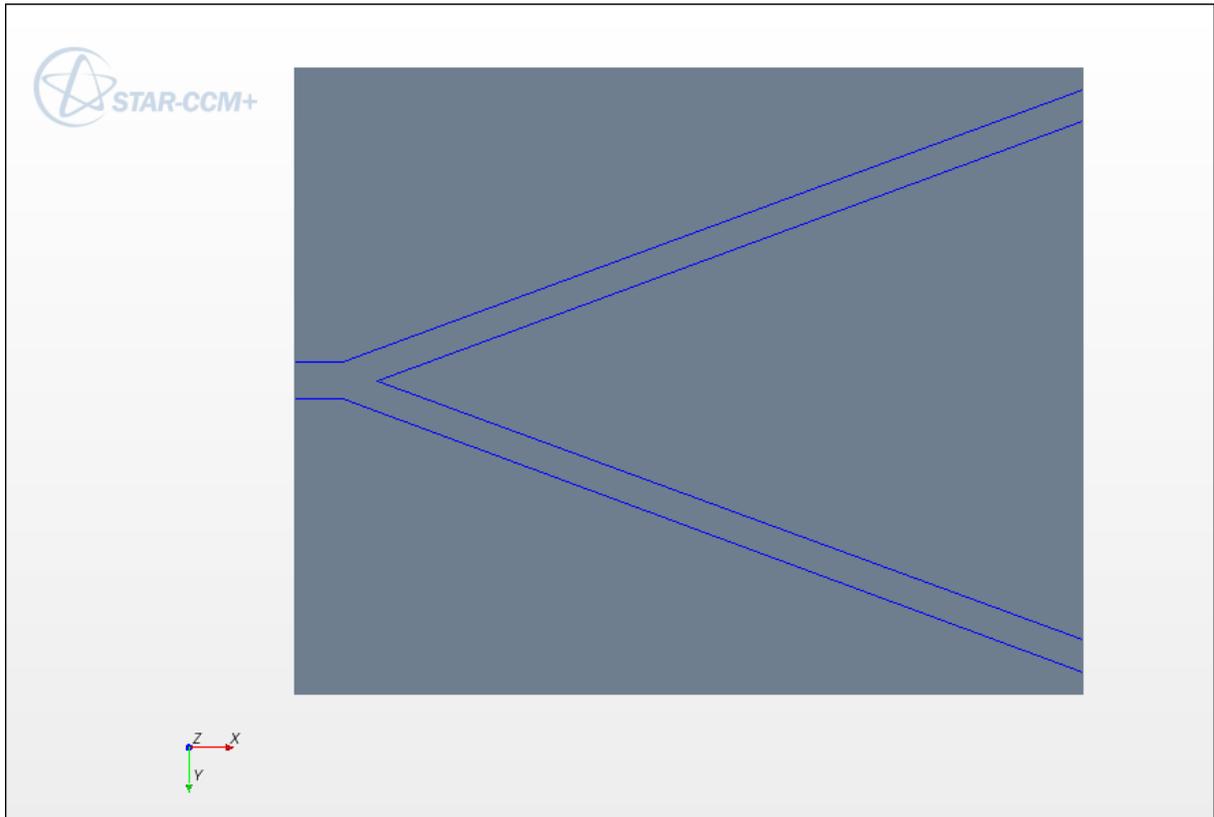
Figure 7.16 - Complex microchannel α = 0.05
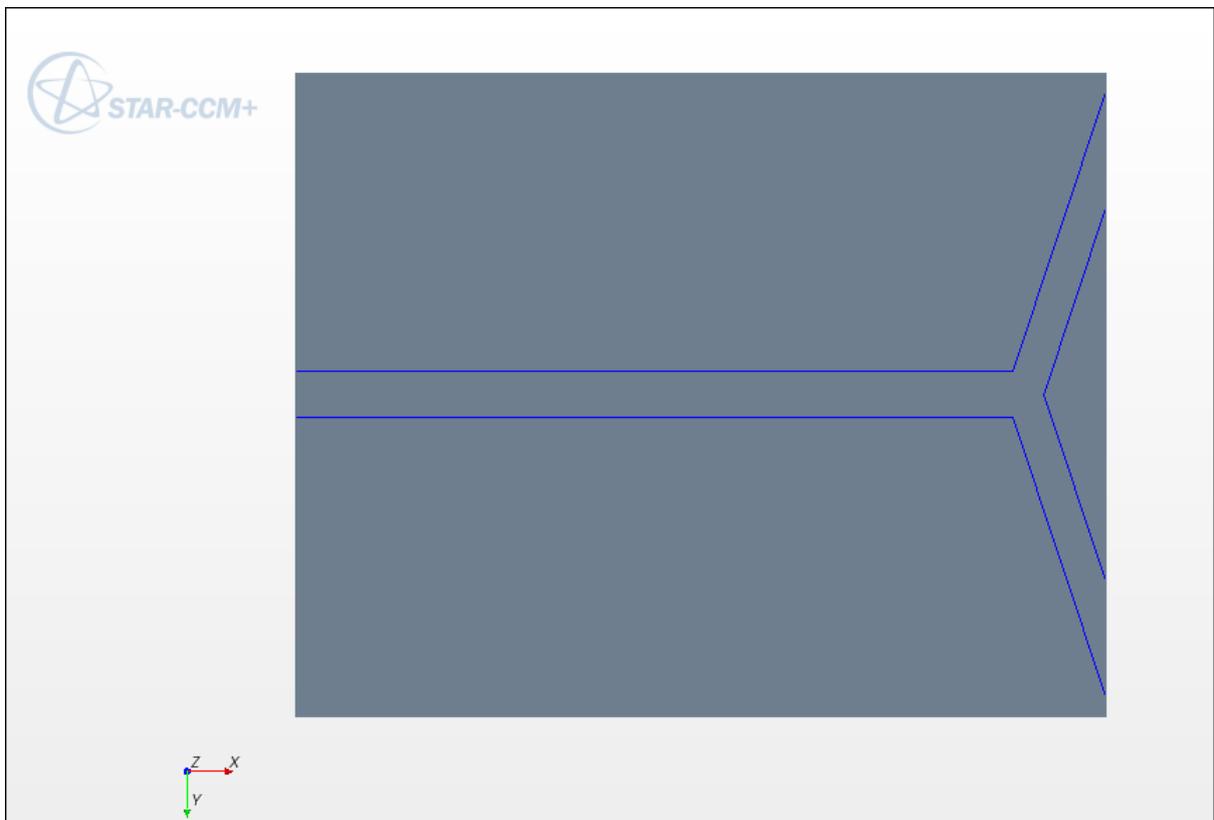
Figure 7.17 - Complex microchannel α = 0.9

7.5.1   Numerical Results and Discussion

Flow through the complex microchannel was simulated for two porosity values, namely 0.95 and 0.9. Again the three power-law fluid exponents were used ($n \in \{0.5; 1; 1.5\}$), simulating Newtonian and non-Newtonian flow. The objective was to determine the optimal point at which the flow channel should bifurcate, so as to maximise the thermal conductance and obtain a relation between inlet velocity and bifurcation point. Again silicon ($k = 124\ W/mK$) was used as solid for the microchannel.

By allowing the bifurcation point (α) to morph to its optimum value, the following data in Figure 7.18 was obtained. It can be noted that as the inlet velocity is increased, so α is reduced. Thus, the bifurcation point is moved closer to the fluid inlet. In this manner, the fluid has access to a greater area of the conductive space, increasing the total heat transfer capacity of the working fluid. However, as the inlet velocity is decreased, α is increased, decreasing the dichotomous area of the fluid. This prevents the fluid from being overworked, in which case the excess temperature of the solid in the exit plane will increase. This phenomenon can be observed in Figure 7.19, where if the bifurcation point α is further increased from the optimum point, the excess temperature increases. Similarly, if α is reduced below this point, the fluid is underworked and heat transfer capacity is lost as the fluid escapes the convective space, again resulting in a rise in excess temperature.

It can be seen in Figure 7.18 that the porosity ratio has an effect on the bifurcation point. A decrease in porosity (higher percentage fluid) results in a lower optimum α. In addition, it can be observed that the bifurcation point becomes increasingly less sensitive to velocity increases, as the porosity is reduced.
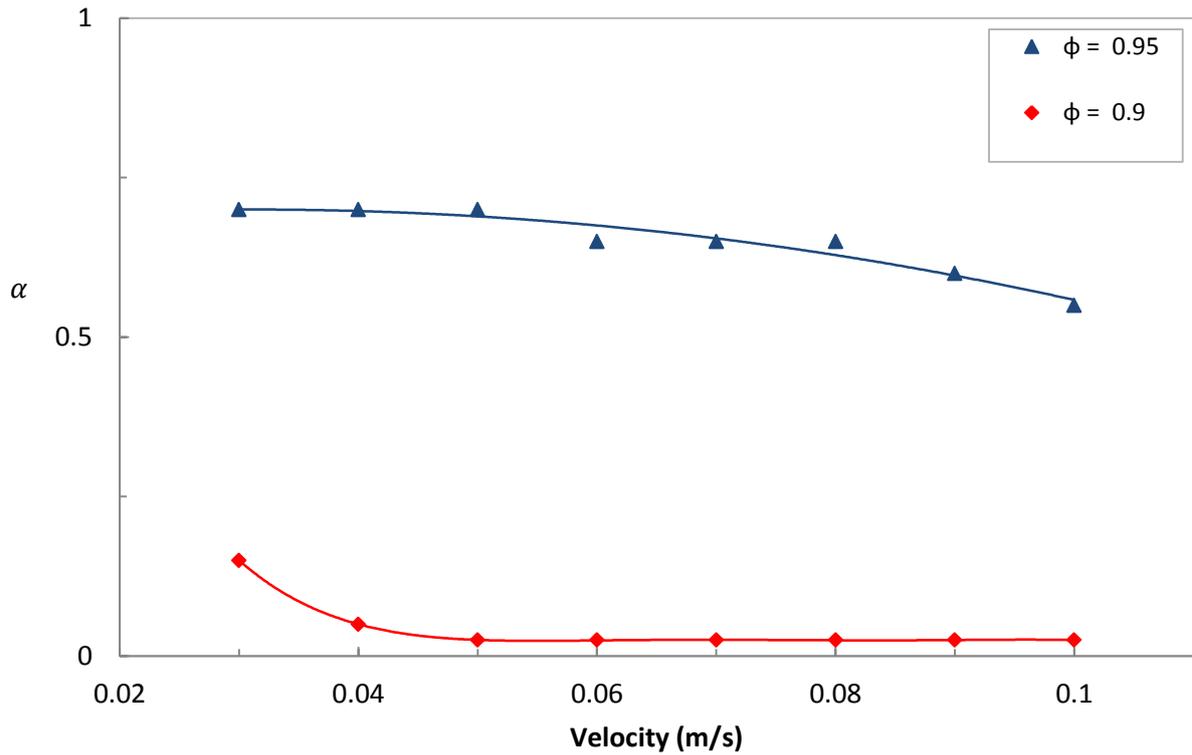
Figure 7.18 - Bifurcation point versus inlet velocity for a Newtonian fluid
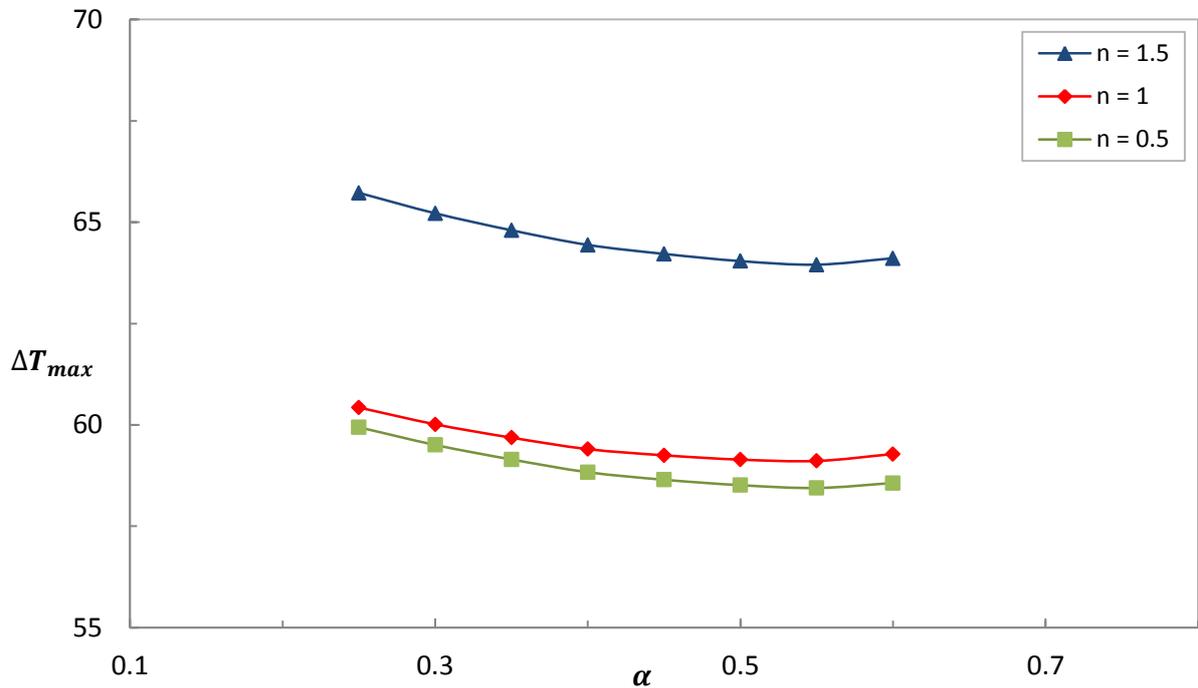
Figure 7.19 - Excess temperature variation as a function of bifurcation point for φ = 0.95 for the various power-law exponents

Figure 7.19 indicates the effect of the power-law exponent distinguishing from dilatant, Newtonian and pseudoplastic fluid. It can be noted that the bifurcation point remains fairly constant with the change in fluid properties, however, the excess temperature is reduced.

Similarly, in Figure 7.20, the increase in inlet velocity as well as a lower porosity value will reduce the overall excess temperature, which can be directly attributed to the increase in the mass flow rate of the fluid.

As in the previous sections, it can be observed in Figure 7.21 that the thermal conductance is increased as the inlet velocity increases. This occurs as the excess temperature is lowered through an increased heat transfer rate. Furthermore, the shear-thickening characteristic of the dilatant fluid again becomes more evident at higher inlet velocities where shear force at the walls and bifurcation point is increased, reducing the heat transfer ability of the fluid.

By comparing the thermal conductance of the optimal bifurcation point to that of the optimal geometry ratio in Figure 7.14, it can be observed that by manipulating the optimal bifurcation point alone has a lesser contribution upon increasing the thermal conductance than the geometric ratio. However, when used in conjunction, an optimal geometry can be obtained.
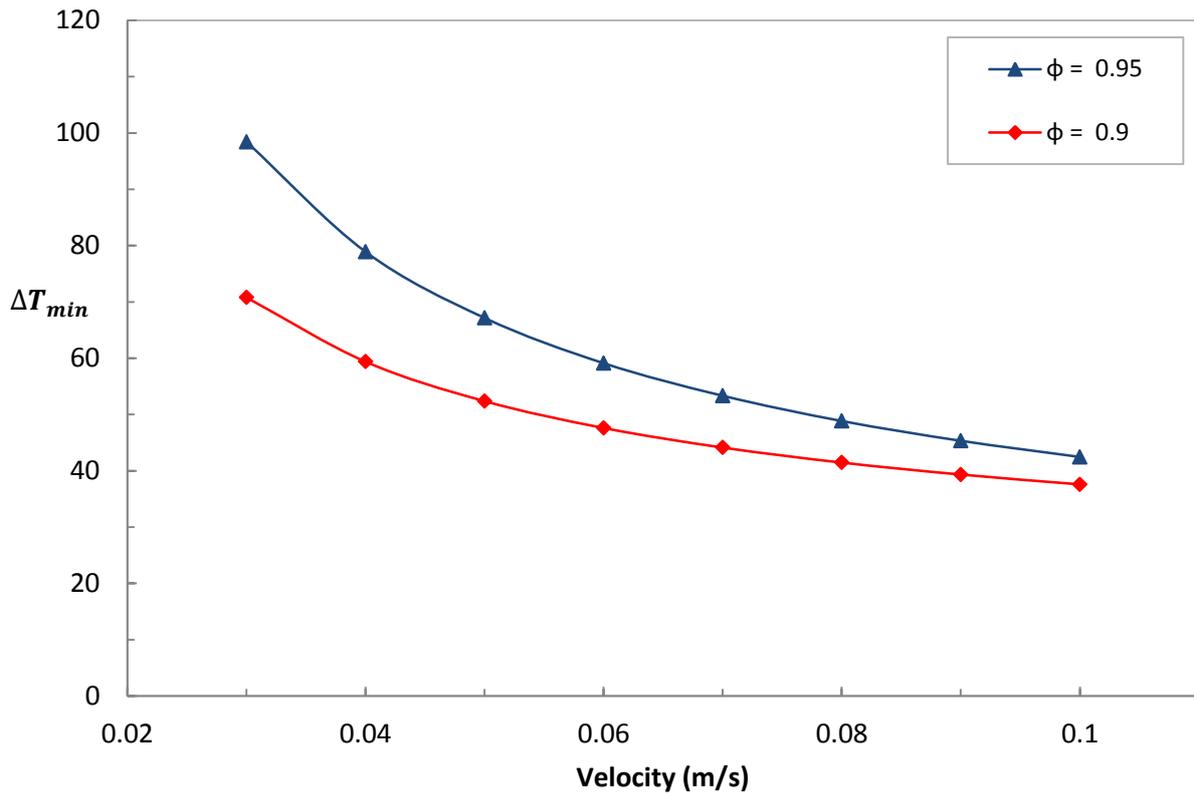
Figure 7.20 - Excess temperature versus inlet velocity for a Newtonian fluid
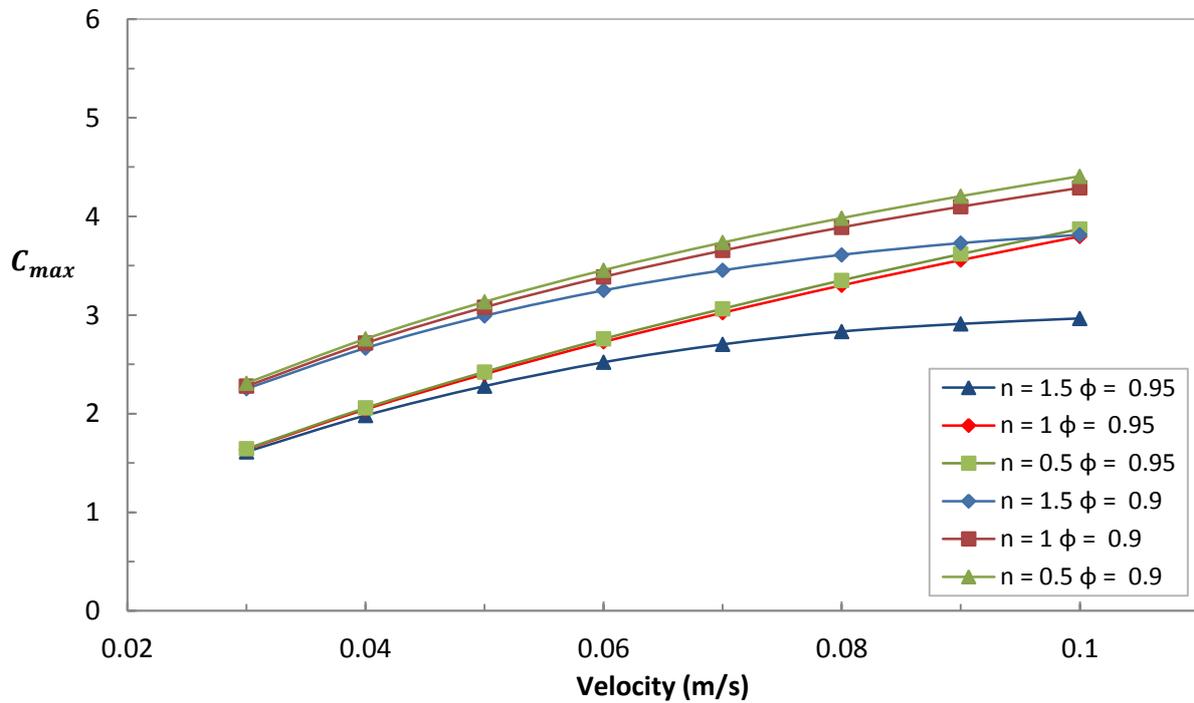
Figure 7.21 - Thermal conductance dependence on porosity and the power-law exponents as a function of inlet velocity

Chapter 8 Comparison of Simple and Complex Microchannel

An identical process of geometric optimisation was followed for both microchannel models. It is observed from the results obtained that a similar trend persisted in both cases, whereby an optimal geometric ratio for each inlet velocity existed. At this ratio, the excess temperature was minimised and thermal conductance was maximised. However, it was found that the complex microchannel produced a higher thermal conductivity at similar condition. Figure 8.1 summarises this enhanced thermal conductance obtained from the complex microchannel, as opposed to a simple microchannel. In addition, when comparing Figure 5.11 to Figure 7.9, it can be noted that the highest temperature for the simple microchannel is found at the exit plane, i.e. at the furthest point from the fluid. In Figure 7.9 – the complex microchannel – the fluid is directed to this specific point, which assists in reducing the specific hot spot temperature and increasing the thermal conductance of the microchannel. Thus, the complex structure allows the fluid greater access to the conductive space, resulting in a lower excess temperature and maximising thermal conductance.
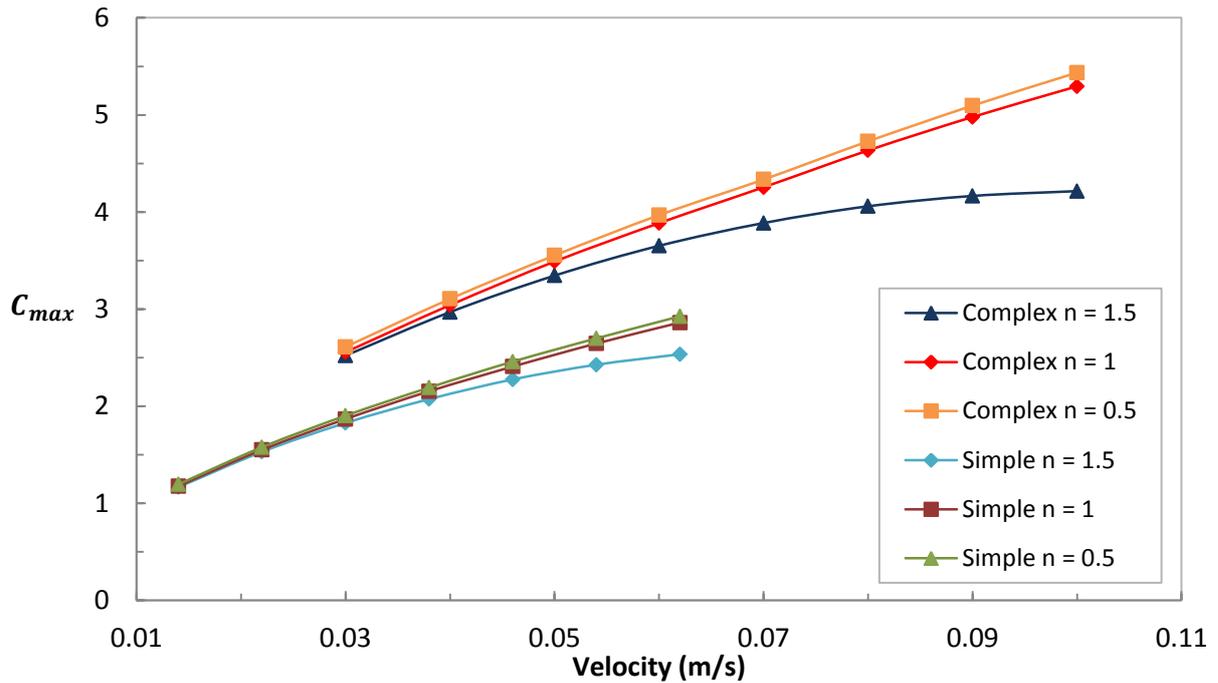
Figure 8.1 - Thermal conductance of a simple microchannel (silicon) compared to a complex microchannel for porosity of 0.9

Figure 8.2 indicates that the normalised pressure drop for the complex microchannel is only marginally larger than that of the simple microchannel. This is achieved by utilising the optimal duct length and diameter ratio for the complex microchannel. Moreover, at a higher inlet velocity, the optimal geometric ratio for a simple microchannel is small (low H/L ratio). This results in a very elongated geometry whereby the channel diameter is reduced and flow resistance is increased. A comparison between Figure 5.13 and Figure 7.11 shows that the geometric ratio for a simple microchannel is significantly lower than that of the complex microchannel. Consequently, it can be extrapolated that at higher inlet velocities, the pressure drop across a simple microchannel may surpass that of a complex microchannel.
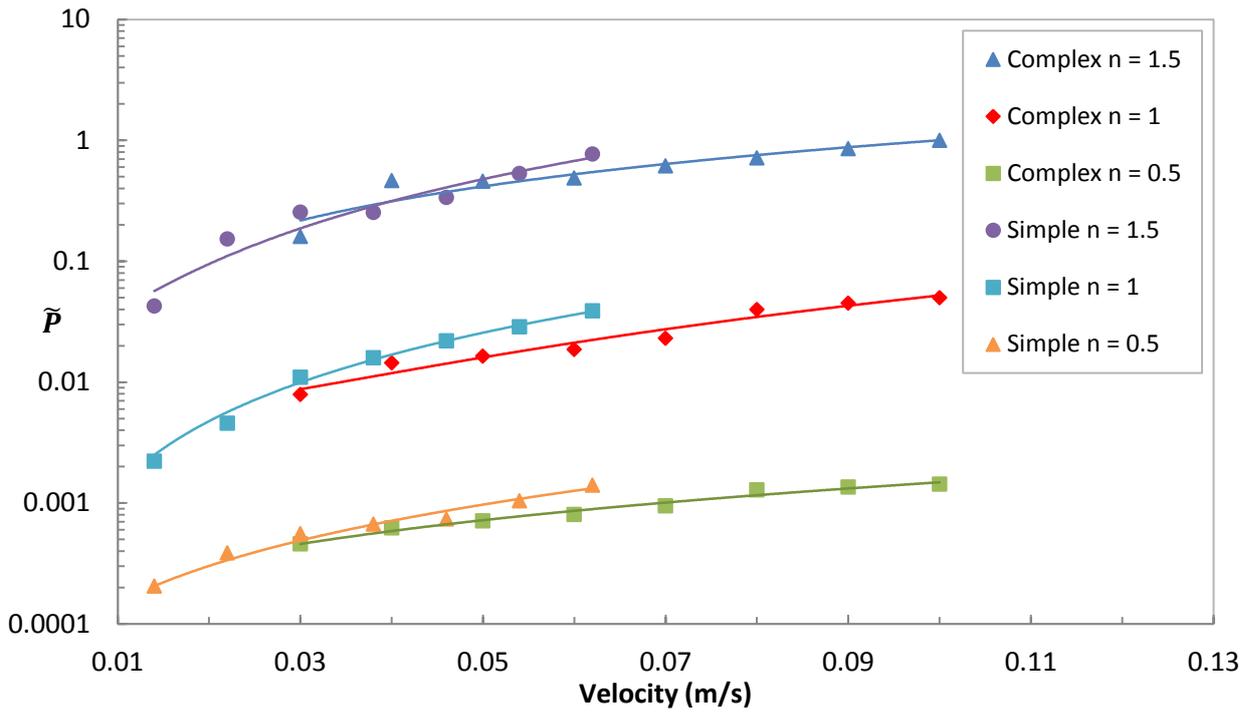
Figure 8.2 - Normalised pressure drop for a simple microchannel compared to a complex microchannel (silicon, $\varphi = 0.95$)

Chapter 9   Scale Analysis

The numerical results obtained were compared to those of Bello-Ochende et al. [41], where results were based on a three-dimensional microchannel heat sink with heat flux from below. By manipulating the Reynolds number, it is possible to write it as a function of the Bejan number:

The Bejan number [40][24] is given by:

$$Be = \frac{\Delta P V^{2/3}}{\alpha \mu} \tag{9.1}$$

The Reynolds number for Newtonian fluid flow through a three-dimensional microchannel, with characteristic length $V^{1/3}$, can be written as:

$$Re = \frac{\rho U V^{1/3}}{\mu} \tag{9.2}$$

and the pressure drop over a flow section is given by:

$$\Delta P = \frac{1}{2} \rho U^2 \approx \rho U^2 \tag{9.3}$$

$$\therefore U = \left(\frac{\Delta P}{\rho}\right)^{1/2} \tag{9.4}$$

Now, substituting equation (9.4) into (9.2) yields:

$$Re = \frac{\rho \left(\frac{\Delta P}{\rho}\right)^{1/2} V^{1/3}}{\mu} \qquad (9.5)$$

which can be manipulated to:

$$Re = \frac{(\alpha\mu\rho)^{1/2}}{\mu} \left(\frac{\Delta P V^{2/3}}{\alpha\mu}\right)^{1/2} \qquad (9.6)$$

Thus, from equation (9.1), the Reynolds number can be defined as follows:

$$Re = \left(\frac{\alpha\rho}{\mu}\right)^{1/2} (Be)^{1/2} \qquad (9.7)$$

which can be rewritten as:

$$Re = \left(\frac{1}{Pr}\right)^{1/2} (Be)^{1/2} = \left(\frac{Be}{Pr}\right)^{1/2} \qquad (9.8)$$

In Ref [41] Bello-Ochende et al. predict, using theoretical analysis, that the maximum thermal conductance is given by:

$$C_{max,\ theory} = 0.6 Be^{0.49} \qquad (9.9)$$

From Figure 9.1, the computational model leads to a thermal conductance relation as a function of Reynolds number as follows:

$$C_{max} = 0.13 Re^{0.59} \qquad (9.10)$$

$$\therefore C_{max} = 0.13 \left(\left(\frac{Be}{Pr}\right)^{1/2}\right)^{0.59} = 0.13 \left(\frac{Be}{Pr}\right)^{0.3} \qquad (9.11)$$

$$\therefore C_{max} \approx Be^{0.3} \qquad (9.12)$$

Thus the data obtained shows similar trends as found in Bello-Ochende et al. [41] and [40].

Similarly, the results for the simple microchannel show comparable tendencies to those of Bello-Ochende et al. The maximum thermal conductance for the simple microchannel can be stated as:

$$C_{max} \approx Be^{0.293} \tag{9.13}$$
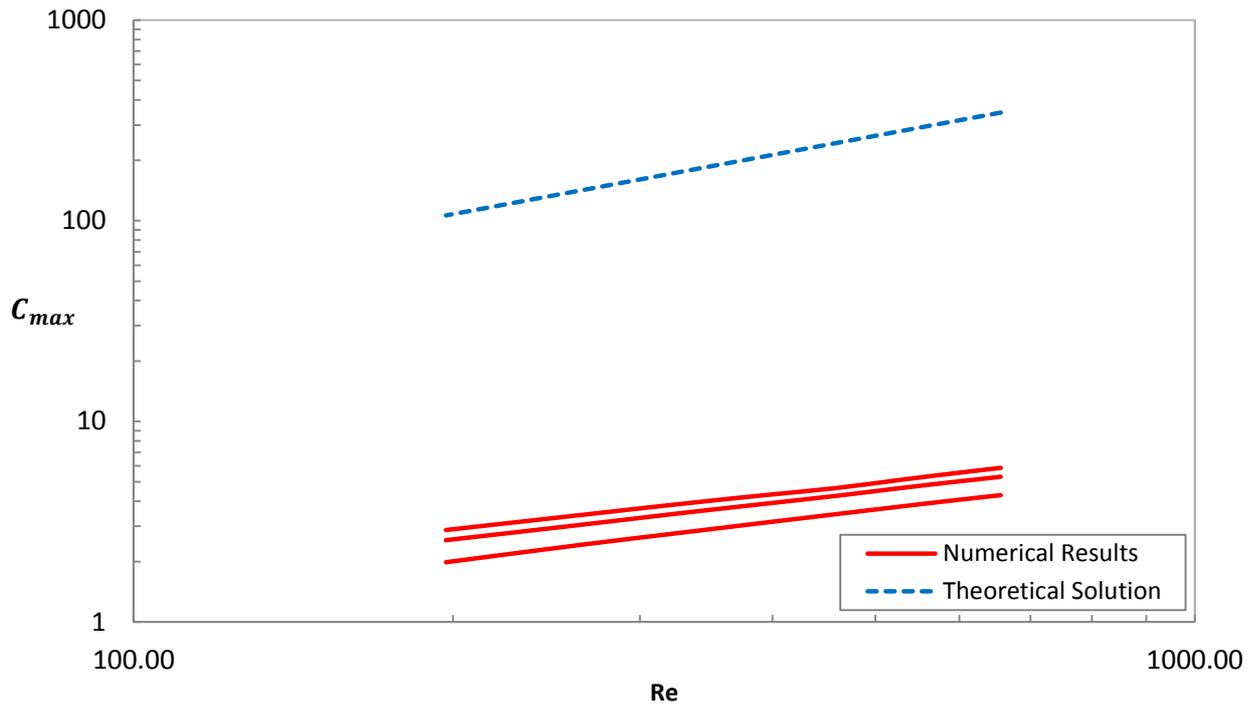
Figure 9.1 - C_{max} versus Reynolds number (Complex results)

Chapter 10   Summary, Conclusion and Recommendations

10.1   Summary

In this dissertation, the effect of utilising a non-Newtonian working fluid, as opposed to a Newtonian, in a microchannel was investigated. The objective was to employ geometric optimisation techniques in order to maximise the thermal conductance of both a simple and complex microchannel, over a range of flow configurations. The result was an optimised microchannel, for both the simple and complex case, which could be used to critically evaluate the effect of a non-Newtonian power-law working fluid as well as the effect of a complex as opposed to a simple microchannel.

10.2   Conclusion

It was found that an optimum geometry existed for each velocity inlet and porosity value, whereby the thermal conductance was maximised. Furthermore, the thermal conductance can be enhanced by increasing the velocity or decreasing the porosity value (i.e. higher ratio of fluid to solid). A dilatant (shear-thickening) non-Newtonian fluid ($n = 1.5$) proved to greatly reduce the thermal conductivity.

The pseudoplastic non-Newtonian (shear-thinning; $n = 0.5$) effect on the thermal conductance was, in both models, only marginally increased over that of the Newtonian fluid ($n = 1$). However, as this effect was achieved in areas of higher shear stress, such as at the wall, the fluid became less viscous and hence the pressure drop over the channel would reduce. This should reduce the pumping requirements of such a fluid and, in addition to the marginal increase in thermal conductance, may significantly increase the efficiency of the system.

In conclusion, the analysis shows that a complex microchannel affords the working fluid greater access to the conductive space and therefore increases the thermal conductance of the microchannel to a substantial extent.

10.3   Recommendations

The work done in this dissertation focussed on the maximising of heat transfer with the use of non-Newtonian fluids in a complex microchannel. It was coincidently found that the power-law exponent had a pronounced effect on the pressure drop across the microchannel. It would therefore be recommended that the research be extended into the effect of firstly, using a complex microchannel as opposed to conventional designs, as well as the effect the power-law exponent could have upon the flow resistance, the objective being to minimise the flow resistance while maximising the thermal conductance of a microchannel.

In addition, the dissertation focussed on a microchannel experiencing a constant heat transfer rate, cooled by the working fluid entering over a range of inlet velocities. Therefore, it can be recommended that the study be extended to investigate the effect of a non-Newtonian fluid, in a complex microchannel, with a constant pressure drop across the channel. This will require the application of a constant heat flux on the microchannel.

References

[1]   I. Mudawar, "Assessment of high-heat-flux thermal management schemes," *IEEE Transaction on Component and Packaging Technologies*, vol. 24, no. 2, pp. 122–141, 2001.

[2]   M. M. Rahman, "Measurement of heat transfer in microchannel heat sinks," *International Communications in Heat and Mass Transfer*, vol. 27, no. 4, pp. 495–506, 2000.

[3]   K. C. Toh, X. Y. Chen, and J. C. Chai, "Numerical computation of fluid flow and heat transfer in microchannels," *International Journal of Heat and Mass Transfer*, vol. 45, pp. 5133–5141, 2002.

[4]   J. Li and G. P. Peterson, "Three-dimensional numerical optimization of silicon-based high performance parallel microchannel heat sink with liquid flow," *International Journal of Heat and Mass Transfer*, vol. 50, pp. 2895–2904, 2007.

[5]   A. G. Fedorov and R. Viskanta, "Three-dimensional conjugate heat transfer in the microchannel heat sink for electronic packaging," *International Journal of Heat and Mass Transfer*, vol. 43, 2000.

[6]   N. J. Balmforth and R. V Craster, "Geophysical Aspects of Non-Newtonian Fluid Mechanics." pp. 34–51, 2001.

[7]   A. B. Metzner and J. C. Reed, "Flow of non-Newtonian fluids - correlation of the laminar, transition , and turbulent-flow regions," *AlChE Journal*, vol. 1, no. 4, pp. 434–440, 1955.

[8]   J. P. Hartnett and T. F. Irvine Jr., "Heat transfer in non-Newtonian fluids, by A. B. Metzner," in *Advances in Heat Transfer*, 1965.

[9]   M. R. Kamal and A. Mutel, "Rheological properties of suspensions in Newtonian and non-Newtonian fluids," *Journal of Polymer Engineering*, vol. 5, no. 4, 2011.

[10]   P. Coussot and F. Gaulard, "Gravity flow instability of viscoplastic materials: The ketchup drip," *Physical Review E*, vol. 72, no. 3, pp. 1–5, Sep. 2005.

[11]   J. F. Tropea; C. Yarin and A.L. Foss, *Springer Handbook of Experimental Fluid Mechanics*, Vol 1. 2007.

[12]   R. Von Mises, "Physical Foundations," in *Advances in Applied Mechanics, vol. 9*, 1966, pp. 244 – 368.

[13]   W. F. Hosford, *Mechanical Behaviour of Materials*. Cambridge University Press, 2005.

[14]   M. M. Denn, "Issues in viscoelastic fluid mechanics," *Annual Review of Fluid Mechanics*, vol. 22, pp. 13–32, 1990.

[15]   D. J. Lacks, "Arrhenius viscosity in fragile liquids due to non-Newtonian effects," *Journal of Chemistry and Physics*, vol. 118, no. 4, pp. 1593–1595, 2003.

[16]   I. M. Krieger, B. P. Madow, and S. H. Maron, "Rheology of synthetic latex. II Concentration dependence of flow i type VGR-S Latex," *Journal of Colloid Science*, vol. 6, no. 6, pp. 584–591, 1951.

[17]   F. L. Saunders, "Rheological properties of monodisperse latex systems," *Journal of Colloid Science*, vol. 16, pp. 13–22, 1961.

[18]   F. T. Pinho and J. H. Whitelaw, "Flow of non-Newtonian fluids in a pipe," *Journal of Non-Newtonian Fluid Mechanics*, vol. 34, pp. 129–144, 1990.

[19]   N. L. Kissi and J. M. Piau, "The different capillary flow regimes of entagled polydimethylsiloxane polymers: Macroscopic slip at the wall, hysteresis and cork flow," *Journal of Non-Newtonian Fluid Mechanics*, vol. 37, pp. 55–94, 1990.

[20]   M. Barkhordari and S. G. Etemad, "Numerical study of slip flow heat transfer of non-Newtonian fluids in circular microchannels," *International Journal of Heat and Mass Transfer*, vol. 28, pp. 1027–1033, 2007.

[21]   H. W. Kim, D. R. Jeng, and K. J. DeWitt, "Momentum and heat transfer in power-law fluid flow over two-dimensional or axisymmetrical bodies," *Journal of Heat Transfer*, vol. 26, no. 2, pp. 245–259, 1983.

[22]   H. S. Tang and D. M. Kalyon, "Estimation of the parameters of Herschel-Bulkey fluid under wall slip using a combination of capillary and squeeze flow viscometers," *Rheologica Acta*, vol. 43, pp. 80 – 88, 2004.

[23]   K. Madlener, B. Frey, and H. K. Ciezki, "Generalized Reynolds number for non-Newtonian fluids," *Progress in Propulsion Physics*, vol. 1, pp. 237–250, Sep. 2009.

[24]   A. Bejan and S. Lorente, *Design with Constructal Theory*. 2008.

[25]   J. Lee, S. Lorente, a. Bejan, and M. Kim, "Vascular structures with flow uniformity and small resistance," *International Journal of Heat and Mass Transfer*, vol. 52, no. 7–8, pp. 1761–1768, Mar. 2009.

[26]   A. H. Reis, "Constructal Theory: From Engineering to Physics, and How Flow Systems Develop Shape and Structure," *Applied Mechanics Reviews*, vol. 59, no. 5, p. 269, 2006.

[27]   S. Lorente and A. Bejan, "Svelteness, freedom to morph, and constructal multi-scale flow structures," *International Journal of Thermal Sciences*, vol. 44, no. 12, pp. 1123–1130, Dec. 2005.

[28]   A. Bejan, "Constructal theory of pattern formation," *Hydrology and Earth System Sciences Discussions*, vol. 3, no. 4, pp. 1773–1807, 2006.

[29]   W. R. Hess, "Das Prinzip des kleinsten Kraftverbrauches im Dienste hämodynamischer Forschung," *Archiv für Anatomie und Physiologie*, pp. 1–62, 1914.

[30]   C. D. Murray, "The physiological principle of minimum work. The vascular system and the cost of blood volume," *Proceedings of the National Academy of Sciences of the Unite States of America*, vol. I, no. 12, pp. 207–214, 1926.

[31]   M. D. S. Frame and I. H. Sarelius, "Energy Optimization and Bifurcation in the Microcirculation," *Microvascular Research*, vol. 50, pp. 301 – 310, 1995.

[32]   W. Schreiner, R. Karch, M. Neumann, F. Neumann, S. M. Roedler, and G. Heinze, "Heterogeneous Perfusion is a Consequence of Uniform Shear Stress in Optimized Arterial Tree Models," *Journal of Theoretical Biology*, vol. 220, no. 3, pp. 285–301, Feb. 2003.

[33]   W. Schreiner, F. Neumann, R. Karch, M. Neumann, S. M. Roedler, and A. End, "Shear stress distribution in arterial tree models, generated by constrained constructive optimization.," *Journal of Theoretical Biology*, vol. 198, no. 1, pp. 27–45, May 1999.

[34]   J. Lee and N. P. Smith, "Theoretical modeling in hemodynamics of microcirculation.," *Microcirculation (New York, 1994)*, vol. 15, no. 8, pp. 699–714, Nov. 2008.

[35]   T. Alarcón, H. M. Byrne, and P. K. Maini, "A design principle for vascular beds: the effects of complex blood rheology.," *Microvascular Research*, vol. 69, no. 3, pp. 156–72, May 2005.

[36]   V. D. Zimparov, a. K. da Silva, and a. Bejan, "Constructal tree-shaped parallel flow heat exchangers," *International Journal of Heat and Mass Transfer*, vol. 49, no. 23–24, pp. 4558–4566, Nov. 2006.

[37]   S. Lorente, W. Wechsatol, and A. Bejan, "Tree-shaped flow structures designed by minimizing path lengths," *Science*, vol. 45, pp. 3299–3312, 2002.

[38]   P.-S. Lee, S. V. Garimella, and D. Liu, "Investigation of heat transfer in rectangular microchannels," *International Journal of Heat and Mass Transfer*, vol. 48, no. 9, pp. 1688–1704, Apr. 2005.

[39]   J. Li, G. . Peterson, and P. Cheng, "Three-dimensional analysis of heat transfer in a micro-heat sink with single phase flow," *International Journal of Heat and Mass Transfer*, vol. 47, no. 19–20, pp. 4215–4231, Sep. 2004.

[40]   T. Bello-Ochende, J. P. Meyer, and F. U. Ighalo, "Combined Numerical Optimization and Constructal Theory for the Design of Microchannel Heat Sinks," *Numerical Heat Transfer , Part A: Applications: An International Journal of Computation and Methodology*, vol. 58, no. 11, pp. 882–889, 2010.

[41]   T. Bello-Ochende, L. Liebenberg, and J. P. Meyer, "Constructal cooling channels for micro-channel heat sinks," *International Journal of Heat and Mass Transfer*, vol. 50, no. i, pp. 4141–4150, 2007.

[42]   H. K. Versteeg and W. Malalasekera, *Computational Fluid Dynamics: The Finite Volume Method*, II. England: Prentice Hall, 2007.

[43]   CD-Adapco, *StarCCM+ User guide*, 6.02.007 ed. 2011.

[44]   R. Eymard, T. Gallouët, and R. Herbin, "Finite Volume Methods," *Handbook of Numerical Analysis*, 2006.

[45]   S. V. Patankar, *Numerucak Heat Transfer and Fluid Flow*. Hemisphere, 1980.

[46]   R. Yin, W. K. Chow, and H. Kong, "Comparison of Four Algorithms for Solving Pressure-Velocity Linked Equations in Simulating Atrium Fire," vol. 4, no. 1, pp. 24–35, 2003.

[47]   D. R. Lester, M. Rudman, and G. Metcalfe, "Optimisation of heat transfer in non-Newtonian fluids with chaotic advection," pp. 1–6, 2006.

[48]   CD-Adapco, *StarCCM+ User guide*, 7th ed. 2012.

Appendix A

Macro Scripts

Geometry Update

```
1    // STAR-CCM+ macro: Complex_updateGeom.java
2    package macro;
3    import java.util.*;
4    import java.lang.Math.*;
5    import star.cadmodeler.*;
6    import star.common.*;
7    import star.base.neo.*;
8    public class Complex_updateGeom extends StarMacro {
9      public void execute() {
10       execute0();
11     }
12     private void execute0() {
13       Simulation simulation_0 =
14         getActiveSimulation();
15       CadModel cadModel_0 =
16         ((CadModel) simulation_0.get(SolidModelManager.class).getObject("3D-CAD Model 1"));
17             double r = 1.5;   //r = 0.8 for k
18             double Hb = 1;
19             double Lb = 1;
```

```java
20          double D1 = 1;
21          double D2 = 1;
22          double D3 = 1;
23          double L1 = 1;
24          double L2 = 1;
25          double L3 = 1;
26          double Theta2 = 0;
27          double Theta = 75;
28          double phi = 0.9;
29          double k = 1;
30          double k1 = 0;
31          double A = 0.0001;
32          double des = 1/3;
33          double err = 100;
34          while (r <= 1.5) {                        // (r <= 1.5)    (k <= 0.95)
35            double val = (A/r);    //
36            Lb = Math.sqrt(val);
37            Hb = r*Lb;
38            while (err > 0.00001) {
39                    k1 = 1/(1 + (Math.cos(Math.toRadians(Theta)))/(Math.pow(2,des)));
40                                    //Set if using H/L as design paramater
41                    double temp = 0.42*Hb/((1-k1)*Lb);
42                    Theta = Math.toDegrees(Math.atan(temp));
43                    err = Math.abs((k-k1)/k1);
44                    k = k1;
45            }
46            Theta2 = 360-Theta*2;
47            L1 = k*Lb;
48            L2 = (Lb - L1)/(Math.cos(Math.toRadians(Theta)));
49            L3 = L2;
50            D2 = (1-phi)*Hb/(Math.pow(2,des)*k + 2*(1-k)/Math.cos(Math.toRadians(Theta)));
51            D1 = Math.pow(2,des)*D2;
52            D3 = D2;
53  // Set Parameter
54          //Sketch 1
55      Sketch sketch_0 =
56        ((Sketch) cadModel_0.getFeatureManager().getObject("Sketch 1"));
57      LengthDimension LengthDimension_1 =
58        ((LengthDimension)      sketch_0.getConstraintManager().getObject("LengthDimension
59  1"));
```

```
60      LengthDimension_1.getLength().setValue(Hb);
61          LengthDimension LengthDimension_2 =
62      ((LengthDimension)      sketch_0.getConstraintManager().getObject("LengthDimension
63  2"));
64      LengthDimension_2.getLength().setValue(Lb);
65          //Sketch 2
66      Sketch sketch_1 =
67      ((Sketch) cadModel_0.getFeatureManager().getObject("Sketch 2"));
68      LengthDimension LengthDimension_3 =
69      ((LengthDimension)      sketch_1.getConstraintManager().getObject("LengthDimension
70  1"));
71      LengthDimension_3.getLength().setValue(L1);
72      LengthDimension LengthDimension_4 =
73      ((LengthDimension)      sketch_1.getConstraintManager().getObject("LengthDimension
74  2"));
75      LengthDimension_4.getLength().setValue(L2);
76      LengthDimension LengthDimension_5 =
77      ((LengthDimension)      sketch_1.getConstraintManager().getObject("LengthDimension
78  3"));
79      LengthDimension_5.getLength().setValue(L3);
80          AngleDimension AngleDimension_1 =
81           ((AngleDimension)   sketch_1.getConstraintManager().getObject("AngleDimension
82  1"));
83          AngleDimension_1.getAngle().setValue(Theta2);
84          LengthDimension LengthDimension_6 =
85           ((LengthDimension)
86  sketch_1.getConstraintManager().getObject("LengthDimension 4"));
87          LengthDimension_6.getLength().setValue(D1);
88          LengthDimension LengthDimension_7 =
89           ((LengthDimension)
90  sketch_1.getConstraintManager().getObject("LengthDimension 8"));
91          LengthDimension_7.getLength().setValue(D2);
92          LengthDimension LengthDimension_8 =
93           ((LengthDimension)
94  sketch_1.getConstraintManager().getObject("LengthDimension 9"));
95          LengthDimension_8.getLength().setValue(D3);
96      cadModel_0.update();
97  cadModel_0.exportModel(resolvePath("…Insert Path…/Complex_Phi_" + phi + "_r-ratio_" +
98  r + ".x_b"));
99          r = r + 0.05;
```

```
100        double rnew = r*100;
101        rnew = Math.round(rnew);
102        r = rnew/100;
103        err = 100;
104      }
105    }
106
```

Simulate and Export

```java
1   // STAR-CCM+ macro: Complex_to_csv_k.java
2   package macro;
3   import java.util.*;
4   import java.lang.Math.*;
5   import java.io.*;
6   import java.nio.*;
7   import star.common.*;
8   import star.base.neo.*;
9   import star.base.report.*;
10  import star.vis.*;
11  import star.meshing.*;
12  import star.energy.*;
13  import star.flow.*;
14  public class Complex_to_csv_k extends StarMacro {
15    BufferedWriter bwout = null;
16    public void execute() {
17      execute0();
18    }
19    private void execute0() {
20    Simulation simulation_0 =
21     getActiveSimulation();
22        double r = 0.8;
23        double q = 0;
24        double L = 0;
25        double Phi = 0.95;
26        double v = 0.08;
27        double k = 0.05;
28        double A = 0.0001;
29        while (v >= 0.08) {
30        // Set velocity
31              Region region_5 =
32                simulation_0.getRegionManager().getRegion("Fluid");
33            Boundary boundary_5 =
34      region_5.getBoundaryManager().getBoundary("Inlet");
35
36              VelocityMagnitudeProfile velocityMagnitudeProfile_0 =
```

```
37              boundary_5.getValues().get(VelocityMagnitudeProfile.class);
38          velocityMagnitudeProfile_0.getMethod(ConstantScalarProfileMethod.class).getQuan
39  tity().setValue(v);
40                  while (k <= 0.3) {
41                  double val = (A/r); // In order to scale for equal total heat flux, replace "1"
42  with r
43                  L = Math.sqrt(val);
44                  double As = L*0.0004;
45                  q = 0.5/As;
46  // Remove Parts
47                  Scene scene_3 =
48                      simulation_0.getSceneManager().getScene("Geometry Scene 1");
49                  //PartDisplayer partDisplayer_10 =
50                      //((PartDisplayer) scene_3.getHighlightDisplayer());
51                  CadPart cadPart_5 =
52                      ((CadPart)
53  simulation_0.get(SimulationPartManager.class).getPart("Fluid"));
54  //              PartSurface partSurface_14 =
55  //                  cadPart_5.getPartSurfaceManager().getPartSurface("Inlet");
56  //              PartSurface partSurface_15 =
57  //                  cadPart_5.getPartSurfaceManager().getPartSurface("Wall_fluid");
58  //              PartSurface partSurface_16 =
59  //                  cadPart_5.getPartSurfaceManager().getPartSurface("Outlet");
60  //              PartSurface partSurface_17 =
61  //                  cadPart_5.getPartSurfaceManager().getPartSurface("Symmetry");
62  //              partDisplayer_10.getParts().setObjects(partSurface_14, partSurface_15,
63  partSurface_16, partSurface_17);
64                  LeafMeshPart cadPart_6 =
65                      ((LeafMeshPart)
66  simulation_0.get(SimulationPartManager.class).getPart("Solid"));
67  //              PartSurface partSurface_18 =
68  //                  cadPart_6.getPartSurfaceManager().getPartSurface("Faces");
69  //              PartSurface partSurface_19 =
70  //                  cadPart_6.getPartSurfaceManager().getPartSurface("Sides");
71  //              PartSurface partSurface_20 =
72  //                  cadPart_6.getPartSurfaceManager().getPartSurface("Heatflux");
73  //              PartSurface partSurface_21 =
74  //                  cadPart_6.getPartSurfaceManager().getPartSurface("Wall_fluid");
75
76  //              PartSurface partSurface_22 =
```

```
77     //                      cadPart_6.getPartSurfaceManager().getPartSurface("Symmetry");
78     //                      partDisplayer_10.getParts().setObjects(partSurface_14, partSurface_15,
79     partSurface_16, partSurface_17, partSurface_18, partSurface_19, partSurface_20,
80     partSurface_21);
81                      simulation_0.get(SimulationPartManager.class).removeParts(new
82     NeoObjectVector(new Object[] {cadPart_5, cadPart_6}));
83
84     //                      partDisplayer_10.getParts().setObjects();
85     // Import Parts
86                      PartImportManager partImportManager_0 =
87                         simulation_0.get(PartImportManager.class);
88     partImportManager_0.importCadPart(resolvePath("C:\\Users\\User\\Documents\\Masters\
89     \StarCCM\\Bodies\\Complex\\Complex_Phi_new" + Phi +"_k-ratio_" + k +".x_b"),
90     "SharpEdges", 30.0, 2, true, true); // Complex_r_" + r + ".xb"
91                      simulation_0.getSceneManager().createGeometryScene("Geometry Scene",
92     "Outline", "Geometry", 1);
93                      Scene scene_1 =
94                         simulation_0.getSceneManager().getScene("Geometry Scene 2");
95                      scene_1.initializeAndWait();
96                      PartDisplayer partDisplayer_4 =
97                         ((PartDisplayer) scene_1.getCreatorDisplayer());
98                      partDisplayer_4.initialize();
99                      PartDisplayer partDisplayer_3 =
100                        ((PartDisplayer) scene_1.getDisplayerManager().getDisplayer("Outline
101    1"));
102                      partDisplayer_3.initialize();
103                      scene_1.open(true);
104                      PartDisplayer partDisplayer_5 =
105                        ((PartDisplayer) scene_1.getHighlightDisplayer());
106                      partDisplayer_5.initialize();
107                      PartDisplayer partDisplayer_6 =
108                        ((PartDisplayer)
109    scene_1.getDisplayerManager().getDisplayer("Geometry 1"));
110                      partDisplayer_6.initialize();
111                      CurrentView currentView_1 =
112                         scene_1.getCurrentView();
113                      currentView_1.setInput(new DoubleVector(new double[]
114    {0.07071067800000001, 0.014142135500000005, 0.002500000000000001}), new
115    DoubleVector(new double[] {0.07071067800000001, 0.014142135500000005,
```

```
116    0.28128299403815615}), new DoubleVector(new double[] {0.0, 1.0, 0.0}),
117    0.07277696533501173, 0);
118                    CadPart cadPart_1 =
119                        ((CadPart)
120    simulation_0.get(SimulationPartManager.class).getPart("Body 1"));
121    //            PartSurface partSurface_0 =
122    //                cadPart_1.getPartSurfaceManager().getPartSurface("Sides");
123    //            PartSurface partSurface_1 =
124    //                cadPart_1.getPartSurfaceManager().getPartSurface("Heatflux");
125    //            PartSurface partSurface_2 =
126    //                cadPart_1.getPartSurfaceManager().getPartSurface("Faces");
127    //            partDisplayer_5.getParts().setObjects(partSurface_0, partSurface_1,
128    partSurface_2);
129                    CadPart cadPart_2 =
130                        ((CadPart)
131    simulation_0.get(SimulationPartManager.class).getPart("Fluid"));
132    //            PartSurface partSurface_3 =
133    //                cadPart_2.getPartSurfaceManager().getPartSurface("Inlet");
134    //            PartSurface partSurface_4 =
135    //                cadPart_2.getPartSurfaceManager().getPartSurface("Wall_fluid");
136    //            PartSurface partSurface_5 =
137    //                cadPart_2.getPartSurfaceManager().getPartSurface("Outlet");
138    //            PartSurface partSurface_6 =
139    //                cadPart_2.getPartSurfaceManager().getPartSurface("Faces");
140    //            partDisplayer_5.getParts().setObjects(partSurface_3, partSurface_4,
141    partSurface_5, partSurface_6);
142    //            partDisplayer_5.getParts().setObjects(partSurface_0, partSurface_1,
143    partSurface_2);
144    //            partDisplayer_5.getParts().setObjects(partSurface_3, partSurface_4,
145    partSurface_5, partSurface_6, partSurface_0, partSurface_1, partSurface_2);
146    // Generate Subtract boolean & rename
147                    MeshActionManager meshActionManager_0 =
148                        simulation_0.get(MeshActionManager.class);
149                    LeafMeshPart leafMeshPart_3 =
150                      (LeafMeshPart) meshActionManager_0.subtractParts(new
151    NeoObjectVector(new Object[] {cadPart_1, cadPart_2}), cadPart_1, "Discrete");
152    //            partDisplayer_5.getParts().setObjects(partSurface_0, partSurface_1,
153    partSurface_2);
154                    simulation_0.get(SimulationPartManager.class).removeParts(new
155    NeoObjectVector(new Object[] {cadPart_1}));
```

```
156    //              partDisplayer_5.getParts().setObjects();
157    //              partDisplayer_5.getParts().setObjects(partSurface_3, partSurface_4,
158    partSurface_5, partSurface_6);
159                    Region region_0 =
160                        simulation_0.getRegionManager().getRegion("Fluid");
161                    cadPart_2.setRegion(region_0);
162    //              PartSurface partSurface_7 =
163    //                  cadPart_3.getPartSurfaceManager().getPartSurface("Faces");
164    //              PartSurface partSurface_8 =
165    //                  cadPart_3.getPartSurfaceManager().getPartSurface("Sides");
166    //              PartSurface partSurface_9 =
167    //                  cadPart_3.getPartSurfaceManager().getPartSurface("Heatflux");
168    //              PartSurface partSurface_10 =
169    //                  cadPart_3.getPartSurfaceManager().getPartSurface("Wall_fluid");
170    //              partDisplayer_5.getParts().setObjects(partSurface_7, partSurface_8,
171    partSurface_9, partSurface_10);
172                    leafMeshPart_3.setPresentationName("Solid");
173                    Region region_1 =
174                        simulation_0.getRegionManager().getRegion("Solid");
175                    leafMeshPart_3.setRegion(region_1);
176    // Create volume control
177                    MeshContinuum meshContinuum_0 =
178                        ((MeshContinuum)
179    simulation_0.getContinuumManager().getContinuum("Mesh 1"));
180                    VolumeSource volumeSource_0 =
181                        ((VolumeSource)
182    meshContinuum_0.getVolumeSources().getObject("Volumetric Control Fluid"));
183                    CadPart cadPart_4 =
184                        ((CadPart)
185    simulation_0.get(SimulationPartManager.class).getPart("Fluid"));
186                    volumeSource_0.getPartGroup().setObjects(cadPart_4);
187                    //VolumeSource volumeSource_1 =
188                        //((VolumeSource)
189    meshContinuum_0.getVolumeSources().getObject("Volumetric Control Solid"));
190    //              LeafMeshPart leafMeshPart_5 =
191    //                  ((LeafMeshPart)
192    simulation_0.get(SimulationPartManager.class).getPart("Solid"));
193    //              volumeSource_1.getPartGroup().setObjects(leafMeshPart_5);
194    // Mesh
195                    MeshPipelineController meshPipelineController_0 =
```

```
196                 simulation_0.get(MeshPipelineController.class);
197
198                 meshPipelineController_0.generateVolumeMesh();
199     // Stopping Criterion
200                 StepStoppingCriterion stepStoppingCriterion_0 =
201                     ((StepStoppingCriterion)
202     simulation_0.getSolverStoppingCriterionManager().getSolverStoppingCriterion("Maximum
203     Steps"));
204                 stepStoppingCriterion_0.setMaximumNumberSteps(1500);
205     // Set Heatflux
206                 Scene scene_2 =
207                     simulation_0.getSceneManager().getScene("Scalar Scene 1");
208                 PartDisplayer partDisplayer_7 =
209                     ((PartDisplayer) scene_2.getHighlightDisplayer());
210                 partDisplayer_7.getParts().setObjects();
211                 Boundary boundary_0 =
212                     region_1.getBoundaryManager().getBoundary("Heatflux");
213                 HeatFluxProfile heatFluxProfile_0 =
214                     boundary_0.getValues().get(HeatFluxProfile.class);
215     heatFluxProfile_0.getMethod(ConstantScalarProfileMethod.class).getQuantity().setValue(q)
216     ;
217     // Clear and Run Solution
218                 Solution solution_0 =
219                     simulation_0.getSolution();
220                 solution_0.clearSolution();
221                 ResidualPlot residualPlot_0 =
222                     ((ResidualPlot)
223     simulation_0.getPlotManager().getObject("Residuals"));
224                 residualPlot_0.setTitleFont(new java.awt.Font("SansSerif", 0, 12));
225                 simulation_0.getSimulationIterator().run();
226                 scene_1.close(true);
227                 simulation_0.getSceneManager().deleteScenes(new NeoObjectVector(new
228     Object[] {scene_1}));
229                 scene_2.close(true);
230                 // Writing reports to .csv
231                 try {
232      // Collecting the simualtion file name
233                     String simulationName = simulation_0.getPresentationName();
234         simulation_0.println("Simulation Name:" + simulationName);
235     // Open Buffered Input and Output Readers
```

```java
236    // Creating file with name "<sim_file_name>+report.csv"
237                        bwout = new BufferedWriter(new
238    FileWriter(resolvePath(simulationName + "_Phi_" + Phi + "v_0.1-
239    0.03_NEWPHI_k.csv"),true));
240                        bwout.write("Phi " + Phi + " k " + k + " v " + v + ",    Value,      Unit,
241    \n");          // k value listed
242                        Collection<Report> reportCollection =
243    simulation_0.getReportManager().getObjects();
244    for (Report thisReport : reportCollection){
245    String  fieldLocationName = thisReport.getPresentationName();
246    Double fieldValue = thisReport.getReportMonitorValue();
247    String fieldUnits = thisReport.getUnits().toString();
248    // Printing to chek in output window
249    simulation_0.println("Field Location :" + fieldLocationName);
250    simulation_0.println("    Field Value :" + fieldValue);
251    simulation_0.println("    Field Units :" + fieldUnits);
252    simulation_0.println("");
253    // Write Output file as "sim file name"+report.csv
254                        bwout.write( fieldLocationName + ",   " +fieldValue + ",   " +
255    fieldUnits +"\n");
256                        }
257                        bwout.close();
258                } catch (IOException iOException) {
259                }
260    // New r/k value
261                k = k + 0.05; //r = r + 0.1;
262                double knew = k*1000;
263                knew = Math.round(knew);
264                k = knew/1000;
265                }
266    // New Velocity
267                v = v - 0.01;
268                double vnew = v*10000;
269                vnew = Math.round(vnew);
270                v = vnew/10000;
271                k = 0.7;  // Reset r/k
272        }
273    }
```