UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

OPTIMISATION OF MICROCHANNELS AND MICROPIN-FIN HEAT SINKS WITH COMPUTATIONAL FLUID DYNAMICS IN COMBINATION WITH A MATHEMATICAL OPTIMISATION ALGORITHM

by

Fervent Urebho Ighalo

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

2010

Title: Optimisation of Microchannels and Micropin-fin Heat Sinks with Computational Fluid Dynamics in Combination with a Mathematical Optimisation Algorithm

Author: F U Ighalo

Supervisors: Dr T Bello-Ochende and Prof J P Meyer

Department: Mechanical and Aeronautical Engineering

University: University of Pretoria

Degree: Masters in Engineering (Mechanical Engineering)

ABSTRACT

In recent times, high power density trends and temperature constraints in integrated circuits have led to conventional cooling techniques not being sufficient to meet the thermal requirements. The ever-increasing desire to overcome this problem has led to worldwide interest in micro heat sink design of electronic components. It has been found that geometric configurations of micro heat sinks play a vital role in heat transfer performance. Therefore, an effective means of optimally designing these heat sinks is required. Experimentation has extensively been used in the past to understand the behaviour of these heat extraction devices. Computational fluid dynamics (CFD) has more recently provided a more cost-effective and less time-consuming means of achieving the same objective. However, in order to achieve optimal designs of micro heat sinks using CFD, the designer has to be well experienced and carry out a number of trial-and-error simulations. Unfortunately, this will still not always guarantee an accurate optimal design. In this dissertation, a design methodology which combines CFD with a mathematical optimisation algorithm (a leapfrog optimisation program and DYNAMIC-Q algorithm) is proposed. This automated process is applied to three design cases. In the first design case, the peak wall temperature of a microchannel embedded in a highly conductive solid is minimised. The second case involves the optimisation of a double row micropin-fin heat sink. In this case, the objective is to maximise the total rate of heat transfer with the effect of the thermal conductivity also being investigated. The third case extends the micropin-fin optimisation to a heat sink with three rows. In all three cases, fixed volume

constraint and manufacturing restraints are enforced to ensure industrial applicability. Lastly, the trends of the three cases are compared. It is concluded that optimal design can be achieved with a combination of CFD and mathematical optimisation.

Keywords: *Geometric configurations, computational fluid dynamics, mathematical optimisation, thermal conductivity, microchannel, micropin-fin, constraints*

ACKNOWLEDGEMENTS

I wish to express my gratitude to my supervisor, Dr T Bello-Ochende, for his guidance, support and friendship. It was a real privilege working under his mentorship, which has not been limited to my academic research.

I would also like to thank my co-supervisor, Prof J P Meyer, for his technical support, which has enabled the successful completion of this work.

I also wish to express my sincere gratitude to Prof J A Snyman, for his helpful insight into the numerical optimisation algorithm used in this dissertation. I also appreciate the efforts made by the Thermofluids Research Group to make this work more fulfilling.

I would like to acknowledge the financial support of the Advanced Engineering Centre of Excellence at the University of Pretoria, NRF, TESP, SOLAR Hub with the Stellenbosch University, EEDSM Hub and the CSIR.

I would like to thank my parents, Mr and Mrs Ighalo, family members and friends for their consistent encouragement and prayers during this period. Lastly, I want to thank the Almighty God without whom none of this would have been possible.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

A	Area, m^2
$\boldsymbol{A}$	Hessian matrix of the objective function
B	Channel width, m
$\boldsymbol{B}_i$	Hessian matrix of the inequality function
Be	Bejan number
Br	Brinkman number
C	Global thermal conductance
$\boldsymbol{C}_j$	Hessian matrix of the equality function
C_p	Specific heat capacity, $J.kg^{-1}$
a, b, c	Diagonals of Hessian matrices $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$
D	Pin diameter, m
D	Hydraulic diameter, m
D	Substantial derivative
$f(\boldsymbol{x})$	Objective function
$\tilde{f}(\boldsymbol{x})$	Objective approximate function
g	Gravity
$g_i(\boldsymbol{x})$	i-th inequality constraint function
$\tilde{g}_i(\boldsymbol{x})$	i-th inequality constraint approximate function
G	Computational domain width
$h_j(\boldsymbol{x})$	j-th equality constraint function
$\tilde{h}_j(\boldsymbol{x})$	j-th equality constraint approximate function
h	Enthalpy, $J.kg^{-1}$
h	Heat transfer coefficient, $W.m^{-2}K^{-1}$
H	Height, m
$\boldsymbol{I}$	Identity matrix
k	Thermal conductivity, $W.m^{-1}K^{-1}$
k_r	Conductivity ratio (k_{solid}/k_f)
L	Length, m
Nu	Nusselt number

P	Pressure, Pa
Po	Poiseuille number
Pr	Prandtl number
$P[k]$	Successive sub-problem
$p(\mathbf{x})$	Penalty function
q	Rate of heat transfer, W
q”	Heat flux, W.m^{-2}
$\tilde{q}$	Dimensionless heat transfer rate
$\tilde{Q}$	Heat transfer, W
$\mathbb{R}^n$	n-dimensional real space
R	Thermal resistance, K.W^{-1}
Re	Reynolds number
s	Interfin spacing, m
T	Temperature, K
t	Time, s
t_1	Half thickness of vertical solid, m
t_2	Channel base thickness, m
t_3	Channel base-to-height distance, m
U	Velocity, m.s^{-1}
u, v, w	Velocities in the x-, y-, z-directions, m.s^{-1}
V	Volume, m^3
$\mathbf{V}$	Velocity vector
W	Heat sink width, m
x,y,z	Cartesian coordinates
$\mathbf{x}^*$	Design variables
$\mathbf{x}^k$	Design points
m,n,l,k,r	Positive integer

Special characters

α	Thermal diffusivity, m^2.s^{-1}
α	Penalty function parameter for inequality constraint
β	Penalty function parameter for equality constraint
γ	Penalty function parameter for objective constraint

δ	Kronecker delta function
δ	Move limit
∂	Derivative
Δ	Difference
∇	Gradient function
ε	Value tolerance
λ	Vexing coefficient
ϕ	Volume fraction
Φ	Dissipation function
ρ	Density, $kg.m^{-3}$
ρ	Penalty function parameter
μ	Dynamic viscosity, $kg.m^{-1}.s^{-1}$
μ	Large positive value
Ω	Dimensionless temperature difference

Subscripts

0	Initial
1	First fin row
1	Phase 1
2	Second fin row
3	Third fin row
1,2,3,4,5	Design variable number
ave	Average
b	Base
best	Best
c	Channel
cond	Conduction
cons	Constant
conv	Convection
f	Fluid
f	Function
h	Hydraulic
inlet	Inlet

L	Length
max	Maximum
min	Minimum
norm	Normalised
i,j,k,l,n	Positive integers
opt	Optimum
solid	Solid
s	Surface
theoretical	Theoretical
w	Wall
x	Step size
∞	Free stream

Superscript

T	Transpose
k	Positive integer

1

CHAPTER 1: INTRODUCTION

1.1 BACKGROUND

Heat sinks are devices capable of removing heat from a system with which they are in direct contact by exchanging the extracted heat with another fluid or its surroundings. This is normally achieved by increasing the surface area significantly while also increasing the heat transfer coefficient. When the dimensions of heat sinks are smaller than 1 mm, they are referred to as micro heat sinks. These heat sinks are prevalent in compact electronic systems and the demand for micro heat sinks is growing daily with the advancement of the fabrication industry, which accommodates better manufacturing tolerances.

Today, heats sinks are usually applied to the thermal management of electronic devices and systems. In the past decade, tight packaging and the rapid development of integrated circuit technology have increased the thermal management requirements of electronic devices. Moore's law, depicted in Figure 1-1 [1], predicts that the number of transistors in an integrated circuit will double every 18 months due to the lowering of the minimum manufacturing cost per component each year. This comes with the problem of effective heat removal for these systems to operate without failure as the reliability of semi-conductor devices is inversely proportional to the square of its change in temperature.

Heat sinks work on the principle of conducting heat from the base where it is being generated and convecting it to another fluid or its surroundings. Therefore, it involves both conduction and convection heat transfer. The performance of heat sinks is measured by its thermal resistance R, which is given by the expression:

$$R = \frac{T_b - T_\infty}{q} \tag{1-1}$$

where q is the power dissipation of the integrated circuit.

Thermal resistance can be split further into the convective and conductive resistances as:

$$R_{\text{conv}} = \frac{1}{hA}$$
$$R_{\text{cond}} = \frac{H}{kA} \qquad (1\text{-}2)$$

with h and k being the convective heat transfer coefficient and thermal conductivity respectively.

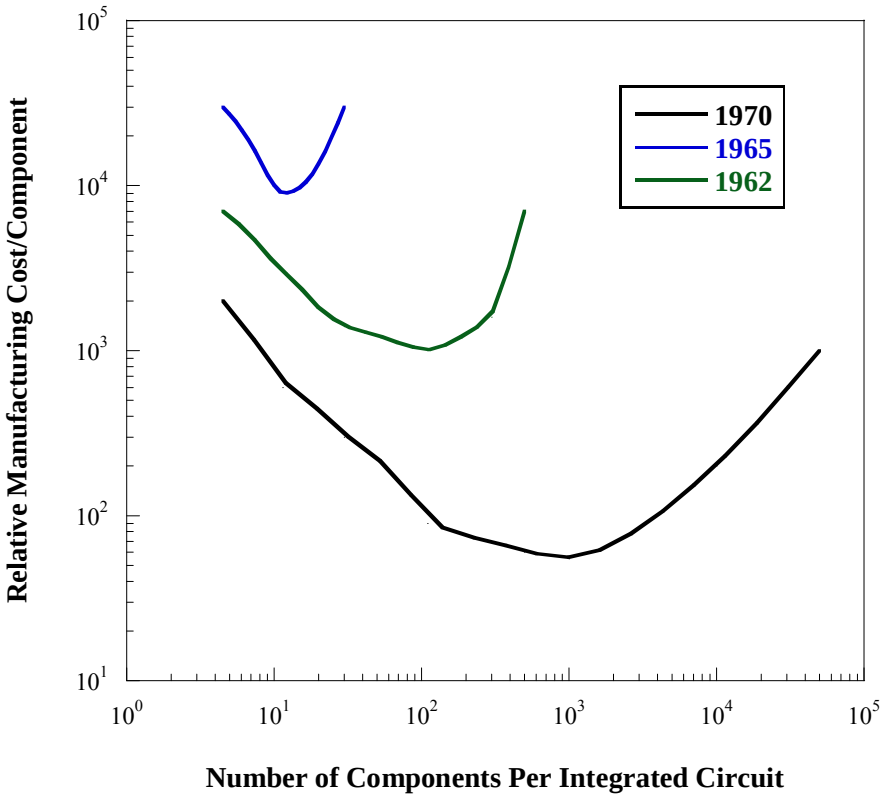


Figure 1-1: Manufacturing cost per component [1]

From Equation 1-2, it is evident that the geometry (area, A) of the heat sink plays a vital role in the performance of heat sinks. Other factors that also influence the performance of heat sinks are the aerodynamics, heat sink material selection and bonding techniques of the heat sink to the base material. In relation to heat sink design, its geometry is the most important factor over which a design engineer has

control. In general, for optimum thermal performance, multivariable optimisation of the various geometric parameters of the heat sink has to be considered. The fact is the impact of a single geometric parameter cannot be generalised without considering its consequence on the other parameters. For example, increasing the fin height will generally improve the overall thermal performance of a heat sink due to the increased surface area. However, for a fixed flow rate, an increase in the fin height will decrease its overall performance due to pressure drop effects and flow field patterns [2]. Therefore, it is safe to say that for heat sink design, multivariable optimisation is required as the geometric parameters are interdependent on each other for optimum thermal performance.

Theoretical mathematical expressions have previously [3-5] been used in the optimisation of heat sinks but these solutions have limited applications due to the various assumptions made when developing these expressions. This led to the advent of using a trial-and-error-based method [6-8] with computational fluid dynamics (CFD) simulations to find near-optimal solutions for various applications of heat sink designs. Others took the route of experimentation [9-11] to investigate the effects of various geometric parameters on the thermal performance of these heat exchangers. These approaches are not only expensive and/or time-consuming, but their results yield a limited range of operation.

The approach of coupling CFD to mathematical optimisation proves capable of producing optimal designs for any heat sink application within a reasonable computational time. With the availability of modern high-speed computers, an "automated" optimisation process, whereby a CFD software package is integrated into an optimisation algorithm for the optimal modification of various design parameters, is now made possible.

1.2 STATE OF THE ART

The consequence of sophisticated, compact, high-processing speed electronic devices and advances in semiconductor technology is the rising transistor density and switching speed of microprocessors. This, however, results in a drastic increase in the heat flux dissipation, which is anticipated to be in the excess of 100 W/cm^2 in the near future [12, 13]. Figure 1-2 gives an indication of the rapid advancement in the electronic industry over the years giving rise to greater clocking speeds and more compact devices.

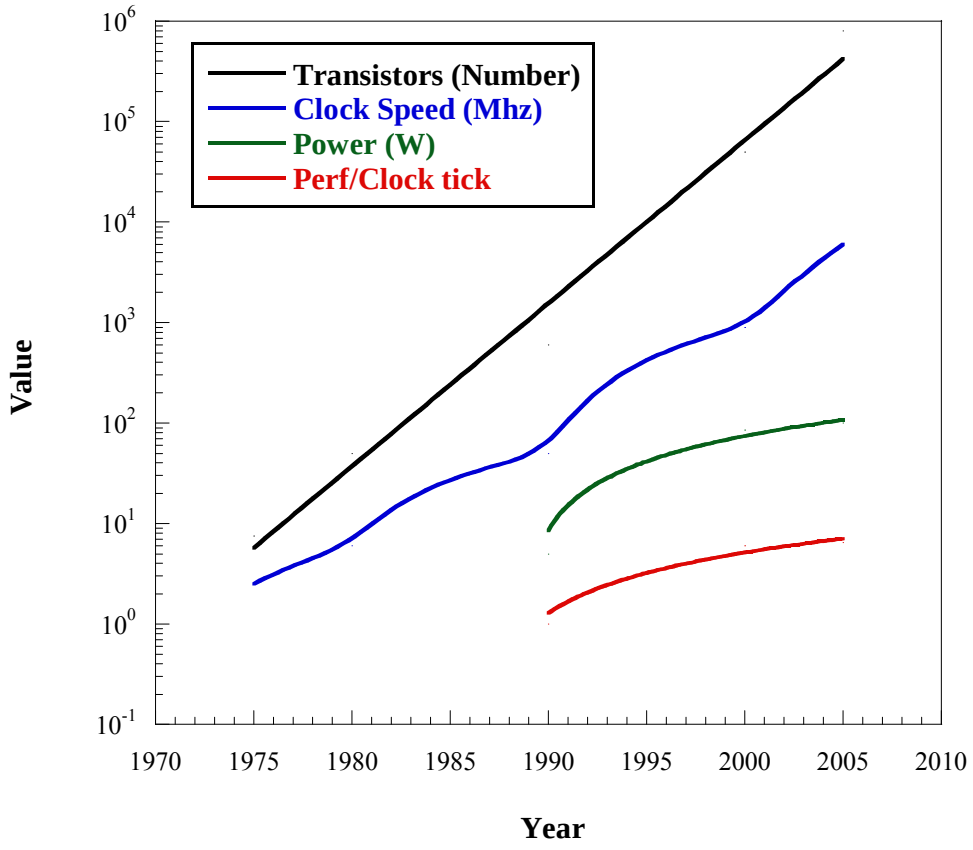


Figure 1-2: Microelectronic advancement prediction [14]

As the challenge for advanced cooling techniques toughens, microchannels have become of great interest and gained research popularity as they yield large heat transfer rates. Figure 1-3 shows the benefits micro heat exchangers possess in cooling over the currently used macro-scale (conventional) heat sinks.

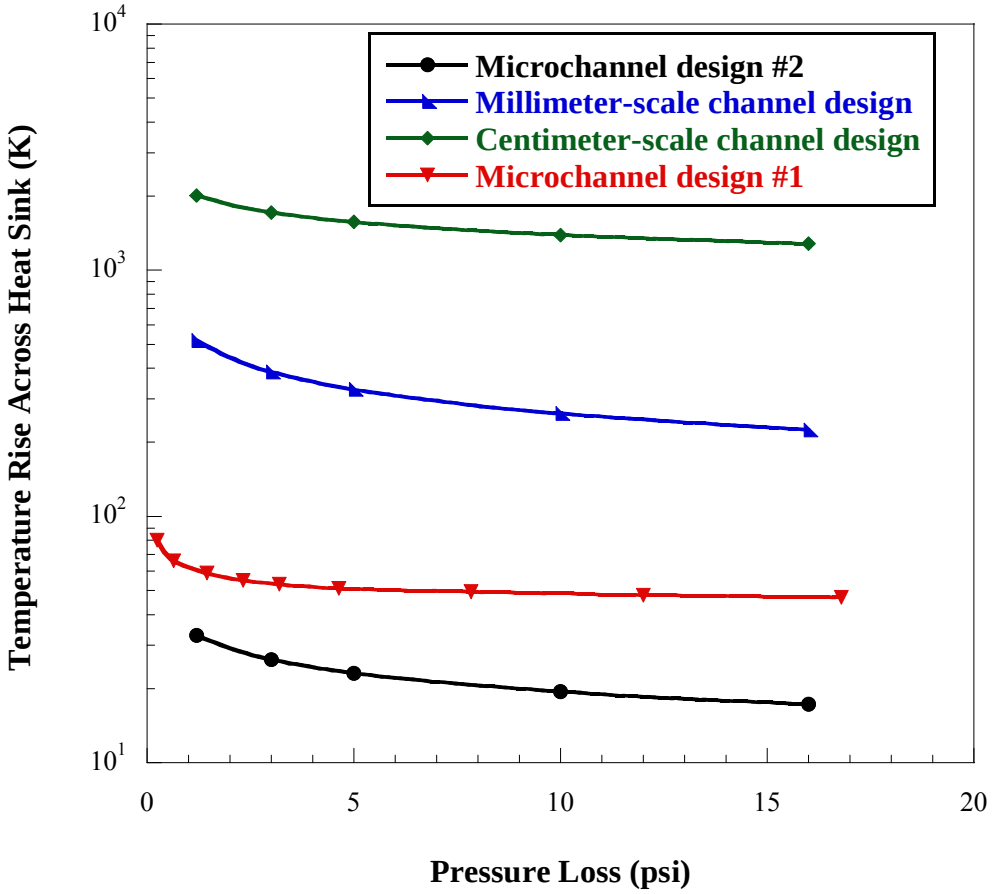


Figure 1-3: Graph showing cooling potential of microchannel heat sinks [15]

As far back as 1981, Tuckerman and Pease [16] proposed that single-phased microscopic heat exchangers using water as the coolant could achieve power density cooling of up to 1 000 W/cm^2 and with experimentation, the cooling water could dissipate a heat flux of about 790 W/cm^2. However, shape and various geometrical parameters such as the aspect ratio of a microchannel have a great influence on the heat transfer capabilities of these heat sinks [17]. Also, current research and development in microchannel cooling investigates the concept of two-layered microchannel heat sinks, which are simply two single-layered heat sinks stacked on top of each other with the flow in counterdirections. This technique possesses better cooling rates as increased convective heat transfer coefficients are achieved.

Studies into enhanced cooling techniques show that heat transfer enhancement can be achieved by the use of fins. Fins are generally known as extended surfaces used to

improve the rate of heat loss from a heated body. The basic phenomenon behind the enhanced heat transfer is the increased surface area created by the fins. The application of this cooling technique ranges from internal cooling of turbine blades to convectional heat exchanger design. Conventional scale pin-fin heat sinks have been widely used in industry but their application at micro-scale has been limited due to manufacturing restraints.

Figure 1-4 shows a conventional finned heat sink in use for cooling of electronic circuitry. However, advances in microfabrication technologies have allowed finned heat sinks to prevail in micro heat exchangers [18]. Micropin-fin heat sinks are the more dominant in the micro heat sink category as they prove to yield increased heat dissipation characteristics under severe space and acoustic restraints [19]. Nevertheless, design considerations, which include material selection, size and compactness, greatly influence the heat dissipation rates that can be achieved by these heat sinks.

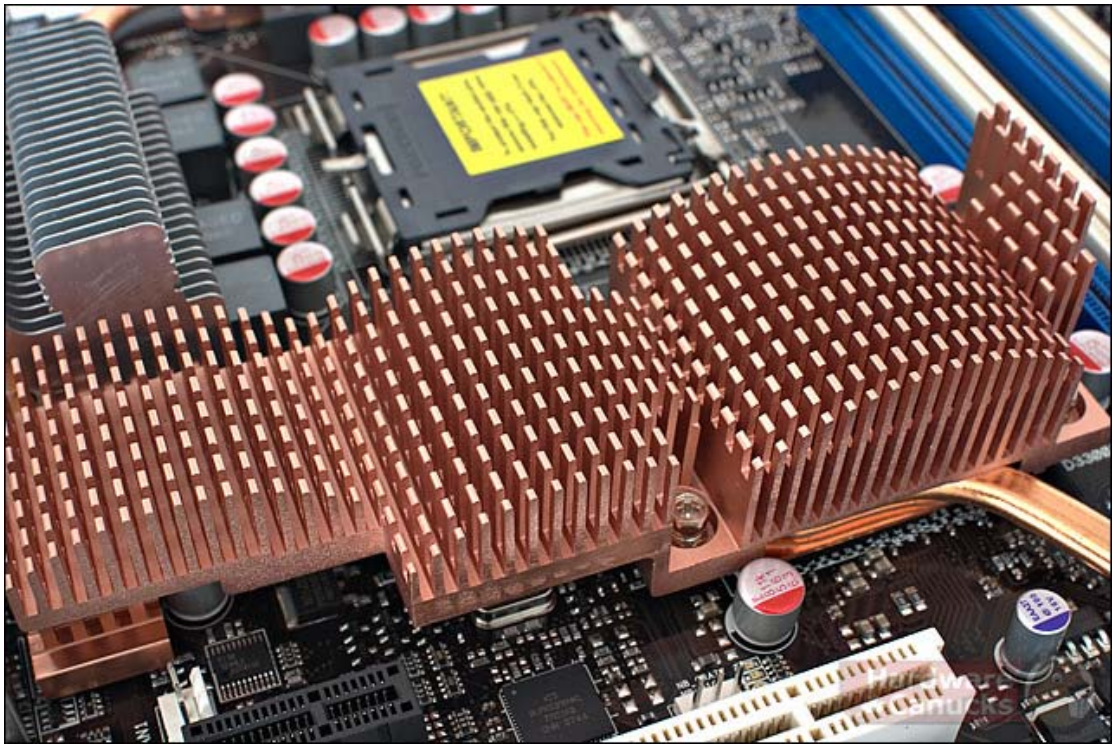


Figure 1-4: A typical finned heat sink [20]

Heat transfer in microchannel and finned heat sinks is mechanised by heat conduction and forced convection. Heat conduction, which involves the transfer of thermal

energy from the more energetic particles to its less energetic counterparts, is mainly influenced by the thermal conductivity of the material and is given by the expression:

$$\tilde{Q}_{cond} = kA\frac{\partial T}{\partial x} \tag{1-3}$$

where k is the thermal conductivity property of the material, which is dependent on the temperature and phase of the material.

Heat transfer by convection is made possible by the movement of fluid molecules and when this movement is facilitated by external forces such as fans and pumps, the term forced convection is used. This heat transfer phenomenon is given mathematically by Newton's law of cooling:

$$\tilde{Q}_{conv} = hA(T_s - T_\infty) \tag{1-4}$$

with the convective heat transfer coefficient h being the driving force of this heat transfer medium. The heat transfer coefficient is a function of various parameters such as the fluid velocity, flow geometry, geometric properties of the surroundings and fluid properties.

Material selection is a very critical component of heat sink design as a balance in the thermal properties, weight and material cost has to be achieved. Table 1-1 gives the material properties of heat sink materials. From the table, diamond proves to be best suited for heat sink design but its expensive cost makes it impractical for use in heat exchanger design. Copper, aluminium and in recent times silicon dominate the materials commonly used in heat sinks as they provide a good balance of thermal conductivity-to-density ratio.

Table 1-1: Material properties of typical heat sink materials

Material	**Thermal Conductivity [W/m°C]**	**Density [kg/m^3]**
Diamond	2 300	3 520
Copper	401	8 933
Aluminium	237	2 702
Silicon	148	2 330
Iron	80.2	7 870

In fluid mechanics and heat transfer today, CFD has proved to provide accurate predictions for flow velocity, temperature and various thermodynamic properties. With the advent of supercomputers over the last few decades, this technology has grown to various wide applications and has made research more cost-effective. CFD has also been useful in the design of thermal cooling systems whereby in a finite number of trial-and-error-based simulations, a near-optimal design is chosen based on the insight of the modeller. Numerical or mathematical optimisation is a systematic tool which searches for an optimal design based on certain specified criteria. This tool, when integrated into numerical modelling, enables optimal design to be achieved more effectively.

1.3 RESEARCH OBJECTIVES

The objectives of this study are:

- to geometrically optimise micro heat exchangers (microchannel heat sinks and micropin-fin heat sinks),
- to execute each optimisation problem by an automated code. This will couple the numerical computation to a mathematical optimisation algorithm.

1.4 SCOPE OF STUDY

In this dissertation, a multidisciplinary optimal design approach is employed to computationally and efficiently optimise the heat transfer capabilities of micro heat sinks using CFD and numerical optimisation. The flow is limited to the laminar flow regime. This study also takes an in-depth look at the optimisation of heat transfer objectives such as peak wall temperature of a microchannel heat sink and the total rate of heat transfer within a micropin-fin heat sink. An automated optimisation algorithm, which uses numerically approximated functions, is implemented for each design case. All the design cases are subjected to various constraints.

1.5 ORGANISATION OF THE DISSERTATION

The dissertation is divided into chapters for better organisation and ease of reading. The dissertation thereby consists of the following chapters:

- Chapter 2 gives an in-depth insight into relevant published work on microchannel heat sinks and micropin-fin heat sinks. This chapter also documents the role various geometric factors play in the optimisation of micro heat sinks.
- Chapter 3 gives appropriate literature pertaining to the numerical modelling of heat sinks. The mass, momentum and energy conservation equations governing the transport of mass and heat are discussed. The iterative method of coupling these governing equations is also highlighted.
- Chapter 4 deals with the subject of numerical optimisation focusing on the operation of the DYNAMIC-Q algorithm. The underlying principles and governing equations of the optimisation algorithm are given.
- Chapter 5 applies the numerical optimisation methodology developed in Chapter 4 to three optimisation case studies. Two of the case studies deal with micropin-fin heat sinks while the other deals with the geometric optimisation of a microchannel heat sink. In this chapter, the steps involved in linking the optimisation method to a commercial CFD code are also shown.
- Chapter 6 provides the summary, conclusions and recommendations for future work.

2

CHAPTER 2: LITERATURE STUDY

2.1 INTRODUCTION

This section deals with the literature pertaining to this dissertation giving an insight into micro heat sinks and the effect of geometry on the heat transfer capabilities of these heat sinks. The manufacturing constraints that will be enforced during the optimisation case studies are also discussed.

2.2 MICROCHANNEL HEAT SINKS

The need for more effective heat removal methods from modern electronic systems has resulted in worldwide research which aims at improving the removal capacity of heat from a silicon-etched substrate. Tuckerman and Pease's [16] early discovery paved the way for many more researchers to understand the transport characteristics of microchannels.

Dirker and Meyer [21] developed correlations that predict the cooling performance of heat-spreading layers in rectangular heat-generating electronic modules. They discovered that the thermal performance was dependent on the geometric size of the volume posed by the presence of thermal resistance.

Xu *et al.* [22], using different experimental methods, investigated the flow within a microchannel. They carried out their experiments with channels having hydraulic diameters ranging from 30 μm to 344 μm over Reynolds numbers ranging from 20 to 4 000. Their results were in agreement with values predicted by conventional classical correlations. This, however, contradicted experimental results obtained by Peng *et al*. [23]. These deviations were initially attributed to measurement errors rather than micro-scale effects [24].

More studies into the heat transfer characteristics of microchannels posed doubt over the reliability of results obtained [25, 26]. Various reasons such as invoking rarefaction, compressibility, dissipation effects, surface roughness and so on have been proposed to explain these deviations. More recently, it has been discovered that the geometric configuration, especially the aspect ratio of rectangular microchannels, greatly influences the heat transfer characteristics of heat sinks [27].

Wu and Cheng. [28] experimentally showed that though the hydraulic diameter of various microchannels may be the same, their friction factors may differ if their geometrical shapes are different. Also, they discovered that the friction factors increase as the aspect ratio of the microchannel is increased.

Koo and Kleinstreuer [29] investigated the effects of viscous dissipation on the evolution of temperature distributions employing scale analysis using numerical simulations. It was documented that for microchannels, viscous dissipation is a function of the channel aspect ratio, channel hydraulic diameter, Prandtl number, Reynolds number and Eckert number. Their work showed that as the aspect ratio deviated from unity, the dissipation effect increases.

Abbassi [30] used the entropy generation analysis method to investigate the effect of geometric parameters on the system performance of a microchannel heat sink. It was found that the thermal entropy rate decreases as the aspect ratio increases. He also investigated the frictional entropy generation by comparing the performance of various fluids. Water showed to have the minimum entropy generation rate. It was also documented that the pumping power increases as the group parameter (Br/Ω) increases.

Heat transfer in rectangular microchannels was analysed for volumetric heat generation due to an imposed magnetic field by Shevade and Rahman [31]. They conducted a thorough investigation for temperature and velocity distributions using water as the working fluid. It was found that as one moves from the symmetric boundary to the solid, the heat transfer coefficient decreases. They also emphasised that the solid-fluid interface heat transfer is higher in channels with smaller hydraulic diameters. They associated this phenomenon to the fact that a smaller hydraulic

diameter results in a larger fluid velocity, which gives rise to a larger rate of convective heat transfer.

According to Guo and Li [32], variations in the predominant factors such as hydraulic diameter and the wetted perimeter influence the importance of various phenomena on the heat transfer and fluid flow characteristics. They reaffirmed the fact that since microelectromechanical systems (MEMS) range from 1 μm to 1 mm, the flow continuum assumption made in numerical analysis is usually valid. They also stated that surface area-to-volume ratio affects the fluid flow and heat transfer properties in microchannels.

Chen [33] conducted an investigation into forced convection heat transfer within a microchannel and found that the heat transfer was influenced mainly by the aspect ratio and effective thermal conductivity of the heat sink. Also an increase in the aspect ratio resulted in an increase in the fluid temperature and the overall Nusselt number. However, the influence of the channel aspect ratio on the temperature of the solid was minimal.

Bejan and Sciubba [34] developed a means for calculating the optimal spacing of parallel plates under forced convection. In a bid to maximise the total heat transfer rate, an order-of-magnitude analysis, together with the intersecting of asymptotes, was employed to develop exact solutions for both isothermal and constant heat flux boundary conditions. It was learnt that the optimal spacing is proportional to the distance between the channels to the power of 1/2 and the pressure head maintained across the stack to the power of -1/4.

Muzychka [4] developed approximate expressions for the optimal geometry for various fundamental duct shapes. The approximate analysis method used by Bejan and Sciubba [34] was applied to determine the optimal size-to-length ratio in terms of Bejan number for several channel shapes. The less rigorous approximate analytical solution showed an excellent agreement with the exact solutions. It was also put forth that the dimension of an optimal duct is independent of its array structure.

Fisher and Torrance [35] implemented the complex variable boundary element method (CVBEM) to analyse the conjugate heat transfer for laminar flow. Total fixed pressure drop and pump work rate constraints were enforced in obtaining optimal duct shapes. Their results showed that channel width has a significant effect on the overall heat transfer with the optimal channel half-widths in the range of 75 μm to 225 μm. Their work showed that higher thermal conductivity results in a lower optimal fin thickness. In conclusion, they stated that increasing the channel curvature decreases the optimal distance between the parallel channels.

Bello-Ochende *et al.* [6] presented a three-dimensional geometric optimisation of microchannel heat sinks using scale analysis and an intersection of asymptotes method. They used the constructal design theory to determine optimal geometric configurations, which maximise the global thermal conductance in a dimensionless form. Their results showed an increase in the optimal aspect ratio for low pressure drop as opposed to a decrease in the optimised hydraulic diameter with an increase in pressure drop.

Experimental and numerical techniques [36-38] have been utilised to further investigate and maximise the cooling abilities of microchannel heat sinks with advances of modern process technology leading to new research ideas, which were virtually impossible in the past.

As microchannel cooling technology has advanced, so has its fabrication, with several techniques leading the way such as:

a) Micromechanical machining
b) Plasma etching prior to wafer bonding
c) Chemical etching
d) Silicon etching
e) Stereo lithography and other X-ray micromachining processes

Micromachining such as LIGA[1] (lithography, electroplating and moulding) technology in connection with X-ray lithography plays a vital role in the emerging of microtechnologies. This technique allows for the fabrication of microstructures with high aspect ratios without compromising their quality and surface roughness. The

[1]LIGA is a German acronym for *Lithographic, Galvanoformung, Abformung*

operating principle of X-ray lithography is by a flux of energetic X-ray photons passing through a lithographic mask and irradiating a polymer. The lithographic pattern is then obtained by shadow printing on a thin membrane. The irradiated structures are then developed by wet organic processing such as deep reactive ion etching (DRIE) [39].

Initially, microfabrication techniques for the integrated circuit community started at a 2D extruded geometry level but now 3D geometries, which were previously considered impossible to manufacture, are fabricated using this vast growing technique.

2.3 MICROPIN-FIN HEAT SINKS

Research into micro-scale cooling methods has extensively been limited to the concept of microchannel cooling due to fabrication limitations of other micro-scale cooling methods. Recent advancements in the fabrication industry have led to other cooling concepts such as the micropin-fin heat sinks getting relative sufficient attention.

Peles *et al*. [40] investigated the convective heat transfer and pressure drop phenomenon across a pin-fin micro heat sink by comparing its thermal resistance with that of a microchannel heat sink. They discovered that the thermo-hydraulic performance of a cylindrical micropin-fin heat sink is superior to that of a microchannel heat sink as very high heat fluxes can be dissipated with low wall temperature rises across the heat sink. Their results showed that for fin diameters larger than 50 μm, the thermal resistance is less sensitive to changes in the fin diameter and for increased efficiency short pins should be used.

A study into the laminar flow across a bank of low aspect ratio micropin-fins assessed the applicability of conventional scale correlations to micro-scale devices. The study concluded that conventional scale correlations do not accurately predict the pressure drop. Refined correlations accounting for fin density and end wall effects were developed for mico-scale configurations [41].

Khan *et al.* [19] optimised a fin heat sink by finding optimal geometric design parameters that minimise the entropy generation rate for both an in-line and staggered configuration. In-line arrangements gave lower entropy generation rates for both low and high thermal conductivity heat sink cases. In a further study by Khan and Yovanovich [42], the effects of geometric factors on the optimal design performance of pin-fin heat sinks were examined by using the entropy generation minimisation scheme. They found that the thermal resistance of these heat sinks increases with an increase in the side and top clearance ratios resulting in a decrease in the entropy generation rate. They also documented that the pin height has an effect on the optimal entropy generation rate of heat sinks.

Soodphakdee *et al.* [43] conducted a comparative study into the heat transfer performance of various fin geometries. The study consisted of fins having round, elliptical and plate cross-sections both for in-line and staggered configurations. They found that round geometries outperformed sharp-edged fin shapes with the circular fin shape yielding the highest Nusselt number and that of the parallel plate having the lowest Nusselt number for the $110 \leq Re \leq 1320$ range considered. It was also found that at lower pressure drops, elliptical fins provide the best heat transfer performance with the circular fins taking over at higher pressure drops. Parallel plates, however, offered the best performance in terms of pressure drop and pumping power requirement.

Similar work was carried out by Yang *et al.* [44], in which an experimental study was conducted on both an in-line and staggered configuration of circular, elliptical and square pin-fins. They compared the effect of fin density on the various configurations. They found that for the staggered configuration, the heat transfer coefficient increases with a rise in the fin density for all three cross-sectional types. However, in an in-line configuration, fin density plays no significant role with regard to the heat transfer performance for the square fin cross-section. Furthermore, it was found that the circular pin-fins possess the smallest thermal resistance, which was attributed to its flow characteristics.

Jiang and Xu [45] investigated the forced convection heat transfer and pressure drop characteristics of mini-fin structures using air and water as the fluid medium. Their experiments showed that with the same porosity, mini-channel structures give lower heat transfer coefficients than mini-fin structures. They also noted that with the same fin width and channel, in-line fin arrays offer less convective heat transfer coefficients as opposed to their staggered array counterpart.

Pressure drop measurements and prediction by Qu and Siu-Ho [46] highlighted the fact that for two-phase flow, micropin-fin heat sinks provide better flow stability than microchannels as their interconnecting flow passage nature promotes a more stable flow. They also noted that at the commencement of saturated flow boiling, the pressure drop increases immensely.

Tahat *et al.* [11] developed steady-state correlations predicting heat transfer performances of in-line and staggered pin-fins from which they found optimal designs. They revealed a dimensionless optimal pin-fin pitch in the span-wise and stream-wise direction of 0.135 and 0.173 respectively for the in-line arrangement. For the staggered arrangement, this dimensionless parameter was found to be 0.19 and 0.1 respectively.

Chiang and Chang [47] developed a response surface methodology to find the optimal design parameters of a pin-fin heat sink. They documented that the fin height and fin diameter are the main factors that affect the thermal resistance of the heat sink while the pitch influences its pressure drop requirements. The conclusion that the most important designing parameters affecting the thermal performance of pin-fin heat sink are the fin diameter and height was also supported by Chiang *et al.* [48]. Their (Chiang *et al.* [48]) work entailed an optimal design of pin-fin heat sink using a grey-fuzzy logic based on orthogonal arrays.

Pitchandi and Natarajan [49] calculated the entropy generation of pin-fins with circular and elliptical cross-sections and compared the performance with respect to their entropy generation. The entropy generation for both fin cross-sections was calculated with a mass constraint enforced to ensure equal material volume for both fin types. The results showed that the entropy generation for the circular pin-fins is

the same as that of the elliptical pin-fins if the circular ratio of the elliptical cross-sections is close to unity. However, for the same entropy generation, elliptical pin-fins offered lower optimal aspect ratios than those of their circular counterparts.

In a recent work, Yuan *et al.* [50] studied the sub-cooled flow boiling heat transfer performance of FC-72[+] from silicon chips fabricated with micropin-fins. The experimental results showed that all micropin-finned surfaces enhanced the heat transfer considerably compared with that of a smooth-surfaced chip. It was also discovered that for the finned surfaces, the flow boiling curves in the nucleate region are almost not affected by the velocity of the fluid.

In recent years, genetic algorithms and various other numerical optimisation schemes have been employed in the optimisation of pin-fin heat sinks, providing designers with a reference base for micro heat exchanger designs [51-53]. Research into various pin-fin configurations shows that elliptical pin-fins pose the best heat exchanger performance [54, 55]. However, circular pin-fins are the more viable option due to manufacturability. Fabrication of micropin-fin heat exchangers is made possible by the LIGA micromachining process [56].

2.4 CONCLUSION

This part of the dissertation provided some available literature on micro heat sink design and optimisation such as mircochannel heat sinks and micropin-fin heat sinks. The published work includes theoretical analysis, experimental procedures and numerical modelling, which are used to generate optimal correlations with regard to the thermal performance of different heat sink configurations. Various optimisation techniques used in the past to optimally design heat sinks were also discussed. Important geometric parameters, which influence the heat transfer abilities of these heat sinks, were also highlighted.

[+]Fluorinent liquid (usually used in low temperature heat transfer applications)

3

CHAPTER 3: NUMERICAL MODELLING

3.1 INTRODUCTION

This chapter gives an overview of the processes involved in numerical modelling and how these are applied to a typical commercial code. The equations that govern heat and mass transport are also discussed.

3.2 OVERVIEW OF NUMERICAL MODELLING

Numerical modelling in recent times have been made easy by the development of CFD codes structured around numerical algorithms, which solve fluid flow and heat transfer problems. These commercial codes consist of three processes, namely:

1. Pre-processing, which involves defining the computational domain, grid generation and selecting domain boundaries;
2. Solver execution, which involves integration, discretisation and solving of the governing equations over the computational domain;
3. Post-processing, which equips the modeller with visualisation tools such as grid displays, contour plots and particle tracking [57].

3.3 GRID GENERATION

Grid generation forms the major part of the pre-processing stage in a CFD analysis. This process involves dividing the computational domain into a finite number of discretised control volumes on which the governing equations can be solved.

This process has been made easier in recent times by the development of commercial automated grid generators in which a modeller can with the help of a graphical user interface (GUI), generate grids and meshes by mouse clicking. One of such grid

generators is called Geometry and Mesh Building Intelligent Toolkit (GAMBIT) [58]. GAMBIT has an added feature whereby the computational model can be set up and grid generated by a journal input file, which favours the automation of an optimisation process.

3.4 GOVERNING EQUATIONS

The governing equations are a set of non-linear partial differential equations describing the fluid hydrodynamics. The basic equations are the conservation of mass (continuity), momentum and energy.

3.4.1 Conservation of mass

In a Eulerian reference frame, the equation of continuity in the most general form for fluids is given by [59]:

$$\frac{D\rho}{Dt} + \rho \text{ div } \boldsymbol{V} = 0 \tag{3-1}$$

with ρ being the density of the fluid, t being time and $\boldsymbol{V}$ being the velocity vector of the fluid. For incompressible flow (constant density), Equation 3-1 reduces to:

$$\rho \text{ div } \boldsymbol{V} = 0 \tag{3-2}$$

3.4.2 Conservation of momentum

The momentum conservation equation is formally derived from Newton's second law, which relates the applied force to the resulting acceleration of a particle with mass. For Newtonian viscous fluids, Navier and Stokes fundamentally derived the following equation using the indicial notation:

$$\rho\frac{D\boldsymbol{V}}{Dt} = \rho\boldsymbol{g} - \nabla P + \frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right)\right] + \delta_{ij}\lambda \text{ div } \boldsymbol{V} \tag{3-3}$$

where $\boldsymbol{g}$ is the vector acceleration of gravity, P is the pressure, x is the spatial coordinate, μ is the coefficient of viscosity, v is the velocity component, δ_{ij} is the Kronecker delta function and λ is the vexing coefficient associated with volume expansion [59]. Using Stokes' hypothesis, $\lambda = -\frac{2}{3}\mu$.

For incompressible flow, the vexing coefficient λ and div $\boldsymbol{V}$ (due to the continuity relationship) vanish, simplifying Equation 3-3 to:

$$\rho\frac{D\boldsymbol{V}}{Dt} = \rho\boldsymbol{g} - \nabla P + \mu\nabla^2\boldsymbol{V} \tag{3-4}$$

3.4.3 Conservation of energy

The conservation equation is derived from the first law of thermodynamics, which states that an increase in energy is a result of work and heat added to the system. Neglecting radiative effects, the energy equation in its standard form can be written as:

$$\rho\frac{Dh}{Dt} = \frac{DP}{Dt} + \text{div}(k\ \nabla T) + \Phi \tag{3-5}$$

where h is the enthalpy of the fluid, k is its thermal conductivity, T is the temperature of the fluid and Φ represents the dissipation function expressed as:

$$\Phi = \mu\left[2\left(\frac{\partial u}{\partial x}\right)^2 + 2\left(\frac{\partial v}{\partial y}\right)^2 + 2\left(\frac{\partial w}{\partial z}\right)^2 + \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z}\right)^2 + \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right)^2\right] + \lambda\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right)^2 \tag{3-6}$$

For incompressible flow with constant thermal conductivity and low velocities, the viscous dissipation becomes negligible. Thus, Equation 3-5 can be simplified to:

$$\rho C_p\frac{DT}{Dt} = k\ \nabla^2 T \tag{3-7}$$

For steady conjugate heat transfer applications (combined conduction-convection problems), the energy equation given by Equation 3-7 is split into two different equations for both the fluid and solid mediums as given by Equations 3-8 and 3-9 respectively.

$$\rho C_p\left(U.\nabla T\right) = k_f\ \nabla^2 T \tag{3-8}$$

$$k_{solid}\ \nabla^2 T = 0 \tag{3-9}$$

with k_f and k_s representing the thermal conductivity of the fluid and solid respectively.

3.5 NUMERICAL SOLUTION SCHEME

In this section, the numerical scheme implemented by FLUENT [60] in solving the mass, momentum and energy conserving equations is discussed.

Firstly, the computational domain is divided into a finite number of discrete control volumes. Then there is the integration of the various governing equations on each control volume thereby constructing algebraic equations for the discrete dependent variables to be solved. Lastly, these discretised equations are linearised and the resulting system of linear equations is solved to yield updated values of the dependent variables.

These governing equations being non-linear and coupled are solved by segregating them from each other. This implies that before a converged solution is obtained, several iterations of the solution loop must be performed [60]. The flow chart in Figure 3-1 gives an overview of the steps of the iterative process.

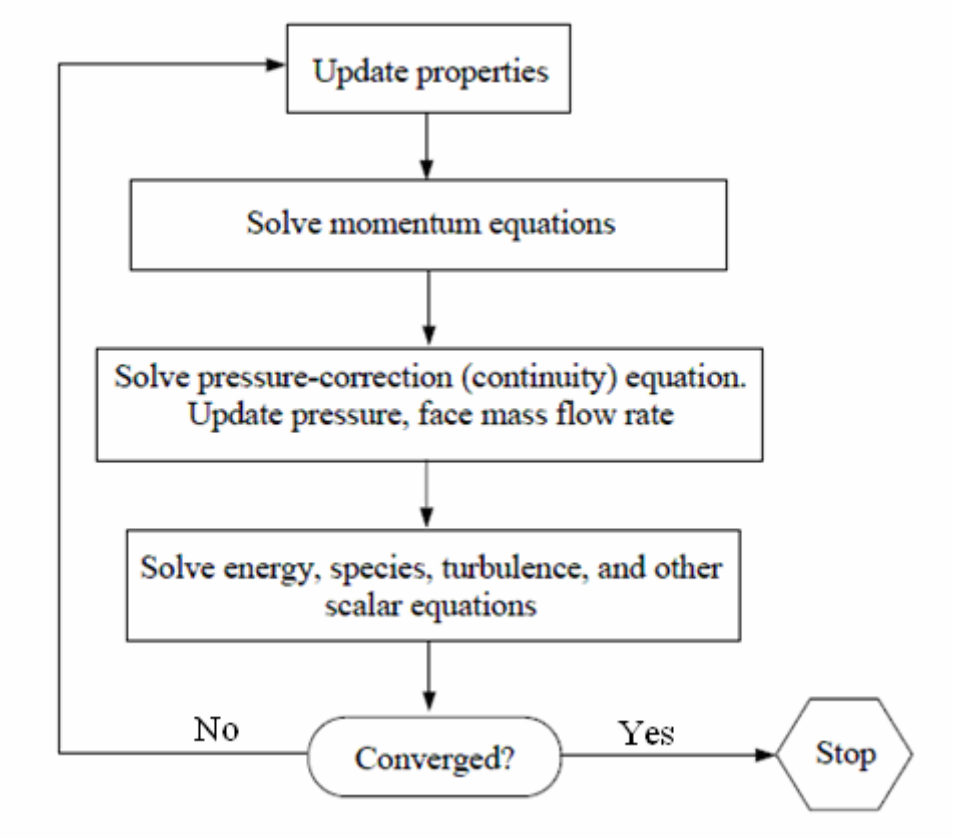


Figure 3-1: Overview of the segregated solution method [60]

3.6 CONCLUSION

This chapter presented an overview of the processes involved in solving fluid flow and heat transfer problems by using a typical commercial CFD code. A set of non-linear partial differential equations governing the transport of mass and heat was discussed. The numerical scheme implemented in solving the flow system within the micro heat sinks was also reviewed.

4

CHAPTER 4: NUMERICAL OPTIMISATION

4.1 INTRODUCTION

This section deals with the theory behind the optimisation algorithms used in this dissertation, together with the numerical modelling technique described in Chapter 3. An overview is given of the optimisation technique whereafter the technique is described.

4.2 NUMERICAL OPTIMISATION OVERVIEW

Numerical optimisation, also known as mathematical optimisation or non-linear programming, is the field that deals with determining the best solution to problems which can be expressed mathematically or numerically. In other words, it implies choosing the best element from a range of available alternatives. The history of this field dates back to the 1940s when the first optimisation technique called the steepest descent was developed for solving very simple problems [61].

4.3 NON-LINEAR CONSTRAINED OPTIMISATION

In numerical optimisation, the quantity to be minimised or maximised (optimised) is known as the objective or cost function $f(\boldsymbol{x})$, while the parameters to be changed in order to obtain this optimal solution are known as the design variables and they are usually represented by a vector $\boldsymbol{x}^*$. When certain constraints in the form of inequalities $g_i(\boldsymbol{x})$ or equalities $h_j(\boldsymbol{x})$ are introduced into the process, the research then has a constrained optimisation problem else the problem is unconstrained.

In general, a constrained optimisation problem is formally written in the form:

$$\begin{aligned} &\underset{\text{with respect to x}}{\text{minimise}}\ f(\boldsymbol{x}),\ \ \boldsymbol{x}=[x_1,x_2,..x_l..,x_n]^T, x_l \in \mathbb{R}^n \\ &\text{subject to the constraints:} \\ &g_i(\boldsymbol{x}) \le 0,\ \ i=1,2,...,m \\ &h_j(\boldsymbol{x}) = 0,\ \ j=1,2,...,r \end{aligned} \tag{4-1}$$

In the case where the objective function $f(\boldsymbol{x})$ is required to be maximised, the minimisation algorithm should still be applied but setting $f_{max}(\boldsymbol{x}) = -f(\boldsymbol{x})$. The plot in Figure 4-1 depicts how the maximisation problem is transformed into a minimisation problem [61].

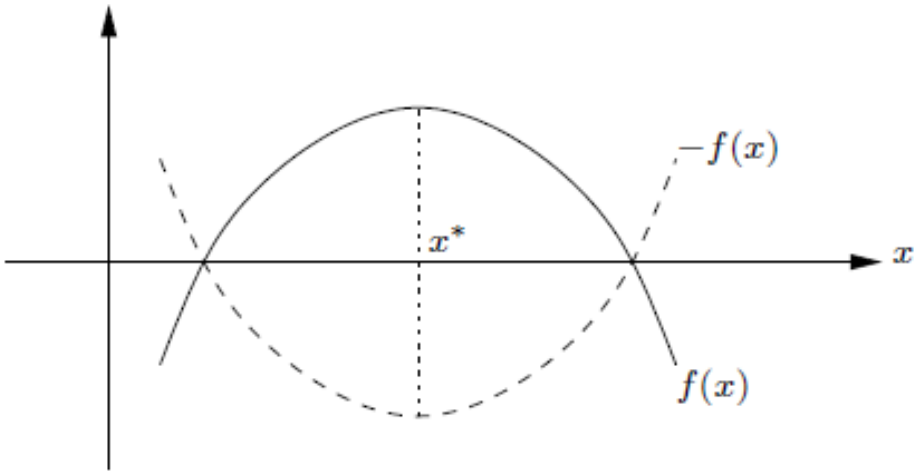


Figure 4-1: Graphical representation of a maximisation problem [61]

An optimisation problem is usually solved with developed algorithms of which some are commercially available. However, new methods are being developed and researched upon to solve certain inhibiting difficulties experienced with the available methods.

4.4 OPTIMISATION ALGORITHMS USED

The LFOPC (Leapfrog Optimisation Program for Constrained Problems) and DYNAMIC-Q algorithms were used as optimisation processes in this study. The LFOPC algorithm implements the penalty parameter in three stages, which increases the rate of finding an optimal design in limited time. The DYNAMIC-Q algorithm has minimal storage requirements, which makes it ideal for handling optimisation problems with a large number of variables. The DYNAMIC-Q is also computationally inexpensive as complex functions, which are expensive to compute numerically, are approximated using spherical quadratic approximate functions. Both algorithms are summarised in the subsequent sections.

4.4.1 Snyman's leapfrog optimisation program for constrained problems (LFOPC)

The LFOPC adapts the original LFOP [62, 63] to handle equality and inequality constraints by the formulation of a penalty function in three phases (Phases 0, 1 & 2) [64-67]. The penalty function is formulated as follows:

$$p(\boldsymbol{x}) = \gamma f(\boldsymbol{x}) + \sum_{i=1}^{m} \alpha_i g_i(\boldsymbol{x})^2 + \sum_{i=1}^{m} \beta_j h_j(\boldsymbol{x})^2$$

where: (4-2)

$$\alpha_i = \begin{cases} 0 \text{ if } g_i(\boldsymbol{x}) \le 0 \\ \rho_i \text{ if } g_i(\boldsymbol{x}) > 0 \end{cases}$$

In order to make the algorithm simpler, the penalty parameters ρ_i and β_j take on the same positive value μ and the higher the value of μ, the more accurate the solution will be. However, at high values of μ, the unconstrained optimisation problem becomes ill-conditioned. To solve this problem, the penalty parameter is increased gradually until it reaches the limit value of μ and it is then kept constant at this value until convergence is reached [64].

Phase 0 of the penalty formulation

In this phase, for a given initial guess of the design variables $\boldsymbol{x}_0$*, the penalty parameter is given a value μ_0 and the penalty function is minimised using the Leapfrog optimisation program (LFOP) with γ set to unity resulting in an optimum design variable vector $\boldsymbol{x}(\mu_o)$* at convergence. At this optimal point, the constraints are checked for violation and if there are no active constraints (constraints that are violated), this optimal point is then indeed the optimal minimum of the optimisation problem and the algorithm is then terminated.

Phase 1 of the penalty formulation

This phase is initialised if there are active constraints obtained from the solution of Phase 0. In this phase, the value of the penalty parameter μ is increased, γ is again set to unity and $\boldsymbol{x}(\mu_o)$* obtained from Phase 0 is used as the initial guess. The penalty parameter is then minimised and active constraints are then identified. If there are no active constraints, the optimisation algorithm is terminated and the solution $\boldsymbol{x}(\mu_1)$* becomes the optimal solution of the optimisation problem.

Phase 2 of the penalty formulation

In this phase, the optimal solution from the preceding phase is used as the starting guess and the LFOP is used on the penalty function with γ set to zero. The algorithm will then try to find the optimal solution which corresponds to the intersection of the active constraints. However, in the event that the active constraints do not intersect, the algorithm will then find the best probable solution, which is usually close enough to the actual solution with the lowest possible constraint violation.

4.4.2 Snyman's DYNAMIC-Q optimisation algorithm

The DYNAMIC-Q optimisation algorithm developed by Snyman and Hay at the University of Pretoria applies the dynamic trajectory LFOPC optimisation algorithm to successive quadratic approximations of the actual optimisation problem [65]. The DYNAMIC-Q algorithm is a gradient-based method, but it does not require any explicit cost function line search. DYNAMIC-Q poses a very robust optimisation

algorithm as it handles numerical analyses obtained from simulations such as CFD and FEM efficiently by handling all noise issues, which can be caused due to grid changes, convergence and numerical accuracy of the computer.

In this method, successive sub-problems $P[k]$, $k = 0,1,2,...$ are generated at successive design points $\boldsymbol{x}^k$ by constructing spherically quadratic approximations, which are used to approximate the objective functions or constraints (or both) if they are not analytically given or very expensive to compute numerically [66, 67]. These spherical quadratic approximations are given by:

$$\begin{aligned}
\tilde{f}(\boldsymbol{x}) &= f(\boldsymbol{x}^k) + \nabla^T f(\boldsymbol{x}^k)(\boldsymbol{x} - \boldsymbol{x}^k) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^k)\boldsymbol{A}(\boldsymbol{x} - \boldsymbol{x}^k) \\
\tilde{g}_i(\boldsymbol{x}) &= g_i(\boldsymbol{x}^k) + \nabla^T g_i(\boldsymbol{x}^k)(\boldsymbol{x} - \boldsymbol{x}^k) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^k)\boldsymbol{B}_i(\boldsymbol{x} - \boldsymbol{x}^k) \\
\tilde{h}_j(\boldsymbol{x}) &= h_j(\boldsymbol{x}^k) + \nabla^T h_j(\boldsymbol{x}^k)(\boldsymbol{x} - \boldsymbol{x}^k) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^k)\boldsymbol{C}_j(\boldsymbol{x} - \boldsymbol{x}^k)
\end{aligned} \tag{4-3}$$

where $\boldsymbol{A}$, $\boldsymbol{B}_i$ and $\boldsymbol{C}_j$ are Hessian matrices of the objective, inequality and equality functions respectively. These matrices are approximated by:

$$\begin{aligned}
\boldsymbol{A} &= diag(a, a, ..., a) = a\boldsymbol{I} \\
\boldsymbol{B}_i &= b_i\boldsymbol{I} \\
\boldsymbol{C}_j &= c_j\boldsymbol{I}
\end{aligned} \tag{4-4}$$

where $\boldsymbol{I}$ represents the identity matrix.

The gradient vectors $\nabla^T f$, $\nabla^T g_i$ and $\nabla^T h_j$ are approximated by means of a forward finite difference scheme if these vectors are not known analytically.

In order to achieve convergence in a stable and controlled form, move limits are used in the DYNAMIC-Q algorithm. The move limit δ_l takes on the form of a constraint by limiting the movement of each design variables $\boldsymbol{x}_l^k$ by not allowing the new

design point to move too far away from the current design point. This additional constraint is of the form:

$$\begin{aligned} \boldsymbol{x}_l - \boldsymbol{x}_l^k - \delta_l \le 0 \\ -\boldsymbol{x}_l + \boldsymbol{x}_l^k - \delta_l \le 0 \end{aligned} \quad ; l = 1, 2, ..., n \tag{4-5}$$

The DYNAMIC-Q algorithm terminates under the following criteria:

1. Step size:

$$\Delta x_{norm} = \frac{\| x^k - x^{k-1} \|}{1 + \| x^k \|} < \varepsilon_x \tag{4-6}$$

2. Function value:

$$\Delta f_{norm} = \frac{| f^k - f_{best} |}{1 + | f_{best} |} < \varepsilon_f \tag{4-7}$$

with ε_x and ε_f the step size and function value tolerances respectively.

4.5 FORWARD DIFFERENCING SCHEME FOR GRADIENT APPROXIMATION

For cases where the gradient functions of either the objective function or constraints (or both) are not analytically available, forward differencing will be used to approximate the gradient vector as follows:

$$\begin{aligned} \frac{\partial f(\boldsymbol{x})}{\partial x_l} &= \frac{f(\boldsymbol{x} + \Delta \boldsymbol{x}_l) - f(\boldsymbol{x})}{\Delta \boldsymbol{x}_l} \quad \forall\ l = 1, 2, ..., n \\ \frac{\partial g_i(\boldsymbol{x})}{\partial x_l} &= \frac{g_i(\boldsymbol{x} + \Delta \boldsymbol{x}_l) - g_i(\boldsymbol{x})}{\Delta \boldsymbol{x}_l} \quad \forall\ i = 1, 2, ..., m \\ \frac{\partial h_j(\boldsymbol{x})}{\partial x_l} &= \frac{h_j(\boldsymbol{x} + \Delta \boldsymbol{x}_l) - h_j(\boldsymbol{x})}{\Delta \boldsymbol{x}_l} \quad \forall\ j = 1, 2, ..., r \end{aligned} \tag{4-8}$$

with $\Delta \boldsymbol{x}_l = [0, 0, ..., \Delta x_l, ...0]^T$ being the differencing step size.

This, however, will increase the computational cost of the optimisation problem as a CFD simulation will be required to approximate each gradient. In order to reduce the computational cost, a constant differencing step size was assumed for each design

variable. This strategy was adequate as the variables are of the same order of magnitude.

4.6 EFFECT OF FORWARD DIFFERENCING STEP SIZE ON THE OPTIMISATION ALGORITHM

The size of the step Δx used in the differencing scheme is very crucial because if it is chosen wrongly it can result in erroneous results. Due to the fact that noise exists in any simulation, it is essential to choose a step size such that it eliminates the noise while giving an accurate representation of the gradient of the function.

Figure 4-2 shows how noise affects the selection of the step size of a function obtained from a simulation. Ideally, a very small step size should give an accurate approximation of the gradient of a function; however, in optimisation algorithms, the fact that noise exists limits how small a step size can be used. Figure 4-2(a) shows that if a very small step size is used, the gradients will be erroneous, therefore, it will be advantageous to use a large enough step size to eliminate the influence of the noise as selected in Figure 4-2(b). A dilemma therefore arises as choosing too large a step size will lead also to a wrong approximation of the gradients.

To ensure that the step size chosen is ideal, the optimisation problem should be done several times with different starting guesses and if the solution converges to the same value, then it can be concluded that the step size is sufficient, but if discrepancies are observed, the step size should be modified until the discrepancies in the results are eliminated.

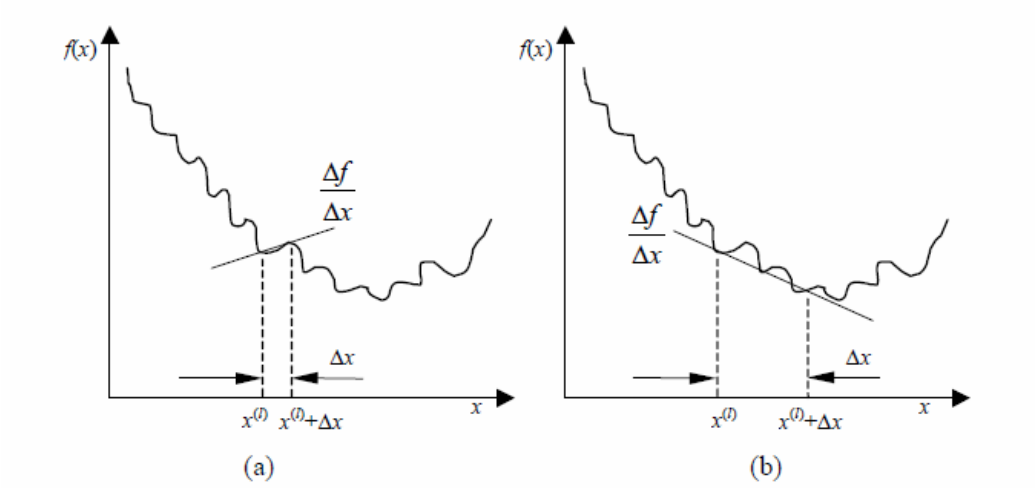


Figure 4-2: Graph depicting the effect of step size on gradient approximation [61]

4.7 CONCLUSION

This chapter focused on the optimisation algorithms used in this study, which are the LFOPC and DYNAMIC-Q algorithms. The DYNAMIC-Q, which builds on the LFOPC algorithm, presented an accurate, reliable and robust penalty method for solving practical constrained engineering problems and helps in optimal design of systems. The effect of numerical noise during simulation was discussed and an effective way of handling this problem was also proposed.

CHAPTER 5: OPTIMISATION OF MICROCHANNELS AND MICROPIN-FIN HEAT SINKS

5.1 INTRODUCTION

This section deals with the numerical approximation of the forced convective heat transfer within a microchannel heat sink and a micropin-fin heat sink with the use of CFD. It also applies the optimisation algorithm described in the preceding section to find the best geometry that enhances the heat transfer for three case studies. The first is the geometrical optimisation of a microchannel heat sink. The second case is the optimisation of a double row micropin-fin and the third case a triple row micropin-fin heat sink case study. The three cases are then compared before the chapter is concluded.

5.2 CASE STUDY 1: MICROCHANNEL EMBEDDED INSIDE A HIGH CONDUCTING SOLID

This case study builds on the research previously carried out by Bello-Ochende *et al.* [6], in which they maximised the global thermal conductance of a three-dimensional microchannel heat sink by using scale analysis and the intersection of the asymptotic method. In this study, heat will similarly be supplied to the bottom of a highly conductive silicon substrate and a computational unit cell will be modelled with the use of the symmetrical property of the heat sink. However, the DYNAMIC-Q optimisation algorithm will be used to find the optimal peak temperature by varying the geometric parameters of the heat sink subject to various constraints. The various heat transfer and optimisation results obtained will then be compared with those published in the preceding work.

5.2.1 The CFD model

Figure 5-1 shows the physical model and Figure 5-2 shows the unit cell computational domain of a microchannel heat sink. The computational domain is an elemental volume selected from a complete microchannel heat sink. Heat is supplied to a highly conductive silicon substrate with known thermal conductivity from a heating area located at the bottom of the heat sink. The heat is then removed by fluid flowing through a number of microchannels. The heat transfer in the elemental volume is a conjugate problem, which combines heat conduction in the solid and convective heat transfer in the liquid.

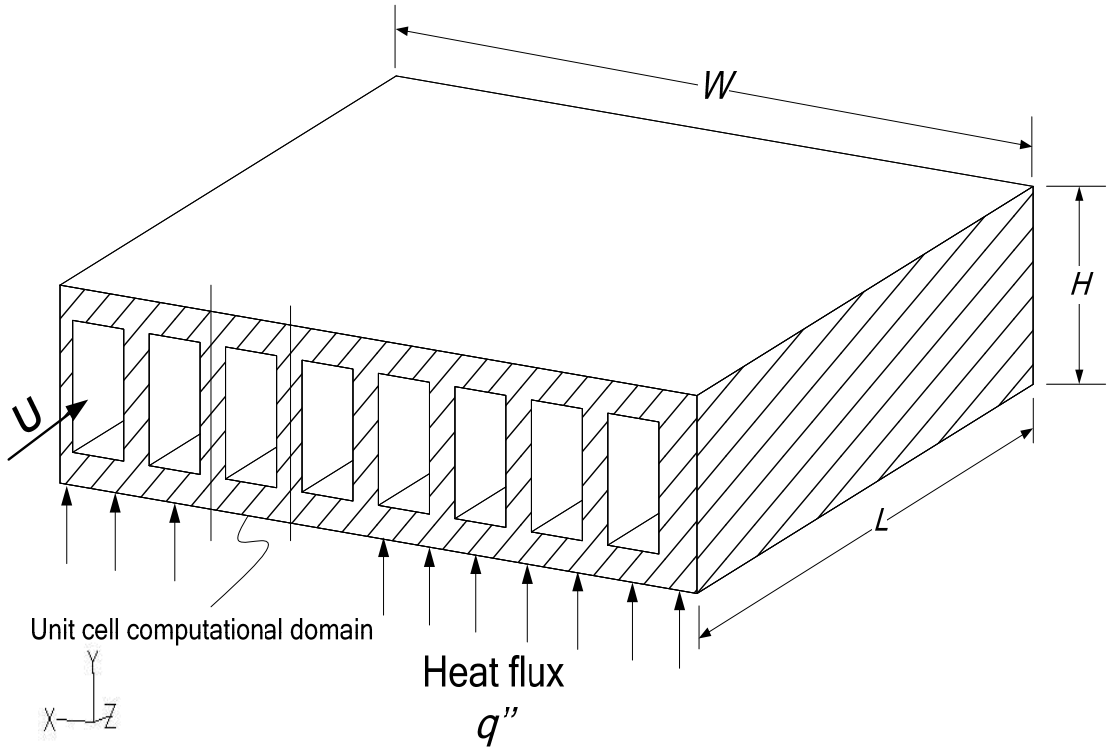


Figure 5-1: Physical model of a microchannel heat sink

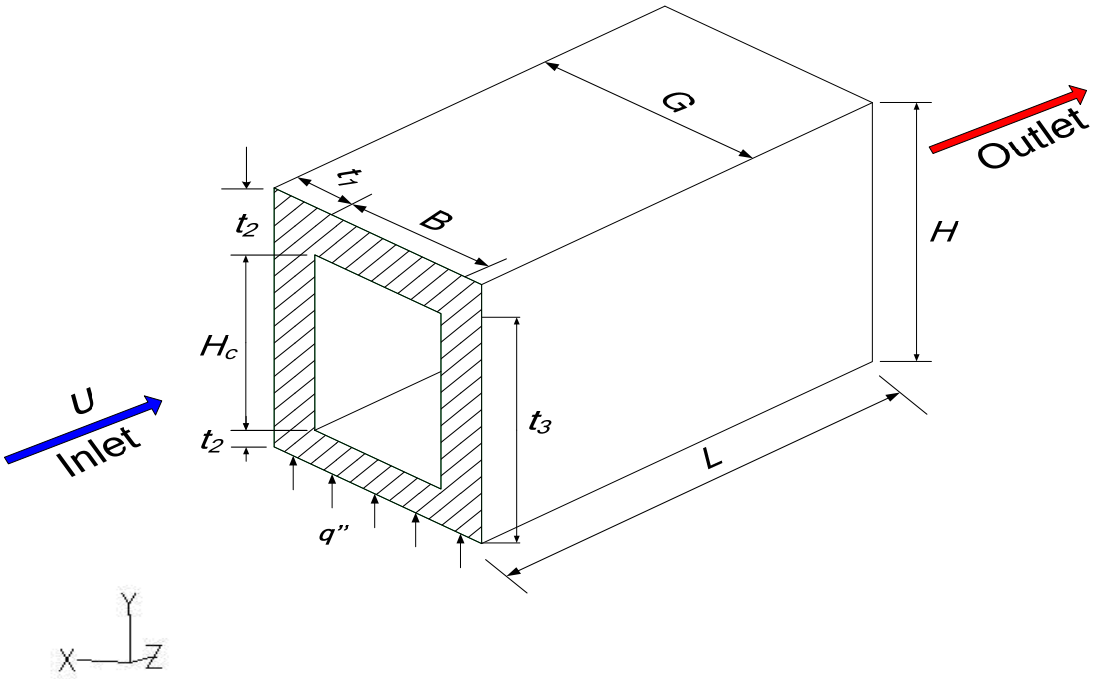


Figure 5-2: Unit cell computational domain for a microchannel heat sink

GAMBIT [58] was used to generate the computational model and grid meshing. Water at 20°C was supplied at the inlet with its properties given in Table 5-1. The bottom wall was supplied with a heat flux of 1 MW/m^2 and the coolant was pumped across the channel length. A constant pressure boundary condition was enforced at the inlet and a symmetrical boundary condition applied to the sides of the channel as depicted in Figure 5-3.

Table 5-1: Fluid properties of the water at the inlet of the microchannel heat sink [59]

Density (kg/m^3)	Specific Heat (J/kgK)	Thermal Conductivity (W/mK)	Viscosity (kg/ms)
998.2	4 182	0.6	0.001003

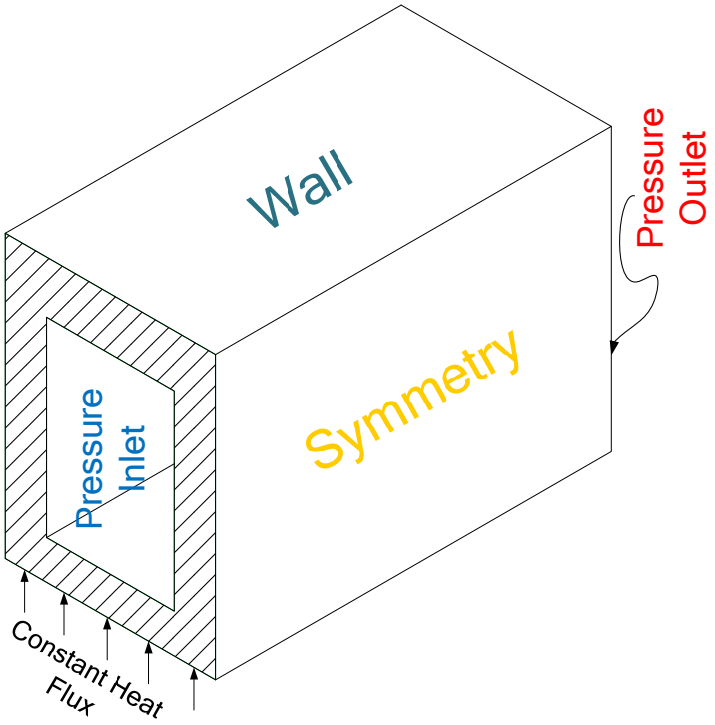


Figure 5-3: Boundary conditions enforced around the microchannel heat sink

FLUENT [60], a finite volume cell-centred commercial CFD code, was used to solve the continuity, momentum and energy equations using the above-stated boundary conditions. A second-order upwind scheme was used in discretising the momentum equation while the SIMPLE algorithm was used for the pressure velocity coupling. Convergence criteria were set to less than $1x10^{-4}$ for continuity and momentum residuals while the residual of energy was set to less than $1x10^{-7}$.

Other assumptions imposed on the model include steady flow, incompressible flow, laminar flow, constant fluid and material properties, negligible radiation and natural convection.

5.2.2 Validation of the CFD model

To ensure accurate results, mesh refinement was performed until a mesh size with negligible changes in thermal resistance was obtained. A grid dependence test was conducted using five different mesh sizes having 19 200, 25 920, 57 600, 88 000 and 110 880 grid cells. The computational volume, with dimensions given in Table 5-2, was used for the analysis. From the results given in Table 5-3 and Table 5-4 (for *Be* of 2×10^8 and 4×10^8 respectively), it follows that a mesh of 57 600 cells approximately ensures a change of smaller than 1% in the thermal resistance when the mesh size is increased. Thus a mesh having 16, 36 and 100 nodes in the *x*-, *y*- and *z*- directions respectively, resulting in a total of 57 600 cells, was chosen for the numerical simulation as it will guarantee results which are independent of the mesh size. Figure 5-4 shows the mesh as generated in GAMBIT [58].

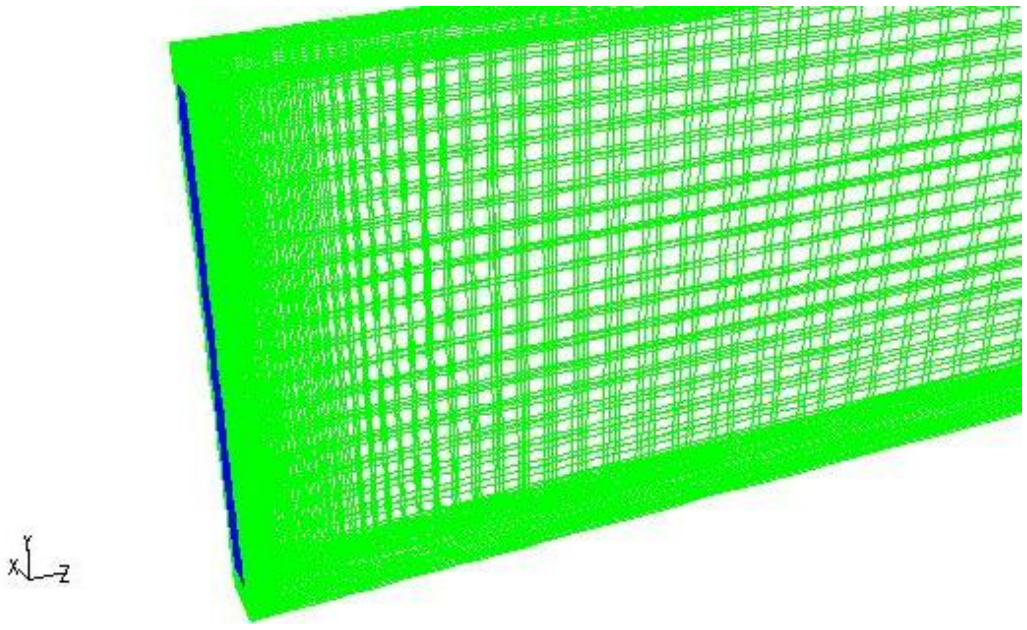


Figure 5-4: Mesh grid for microchannel heat sink numerical computation

Table 5-2: Dimensions of the microchannel heat sink for code validation

t_1 (mm)	t_2 (mm)	t_3 (mm)	B (mm)	H_c (mm)	G (mm)	H (mm)	L (mm)
0.02	0.21	0.69	0.06	0.48	0.1	0.9	10

Table 5-3: Grid independence test results at *Be* = 2×10^8

Number of Cells	Thermal Resistance ($K.cm^3/W$)	Difference
19 200	0.118	-
25 920	0.118	0.14%
57 600	0.122	1.33%
88 000	0.125	0.73%
110 880	0.126	0.29%

Table 5-4: Grid independence test results at *Be* = 4×10^8

Number of Cells	Thermal Resistance ($K.cm^3/W$)	Difference
19 200	0.0879	-
25 920	0.0881	0.09%
57 600	0.0916	1.20%
88 000	0.0920	0.16%
110 880	0.0924	0.13%

The numerical code was also evaluated by comparing the results generated with available widely accepted analytical results. Figures 5-5 and 5-6 show the numerical and analytical dimensionless velocity profile for fully developed flow within the microchannel along the *x*- and *y*- axes respectively. The velocity profile for the numerical solution was generated at the centre of the channel. The Shah and London [68] analytical solution was used against which to compare the numerical predictions obtained and an excellent agreement was found.

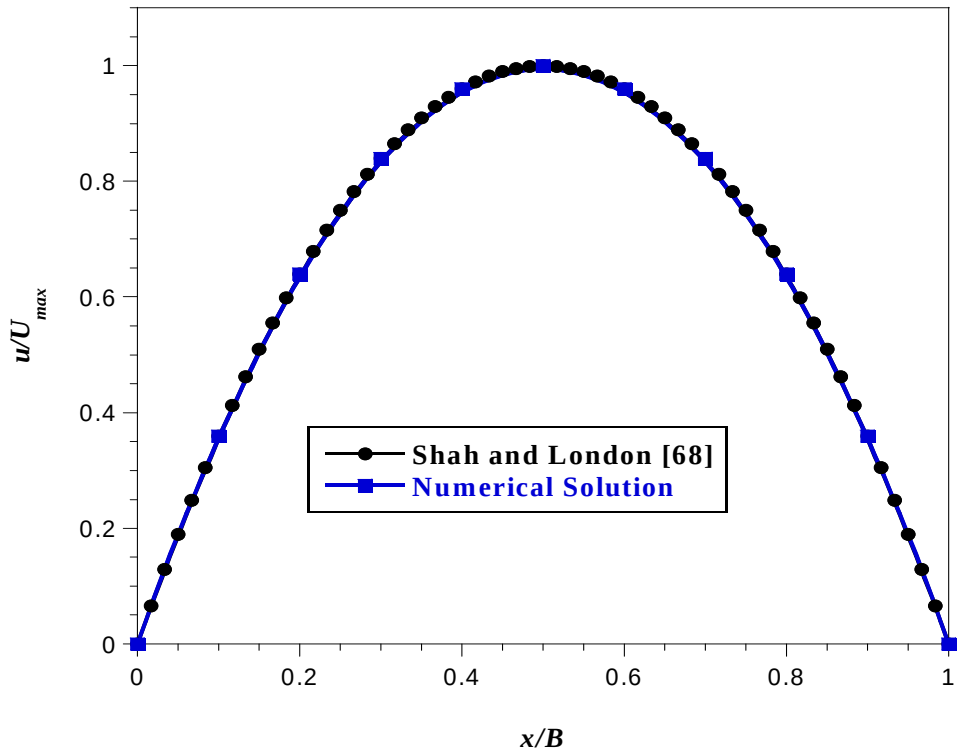


Figure 5-5: Comparison between numerical and analytical prediction for fully developed velocity profile along the *x*-axis

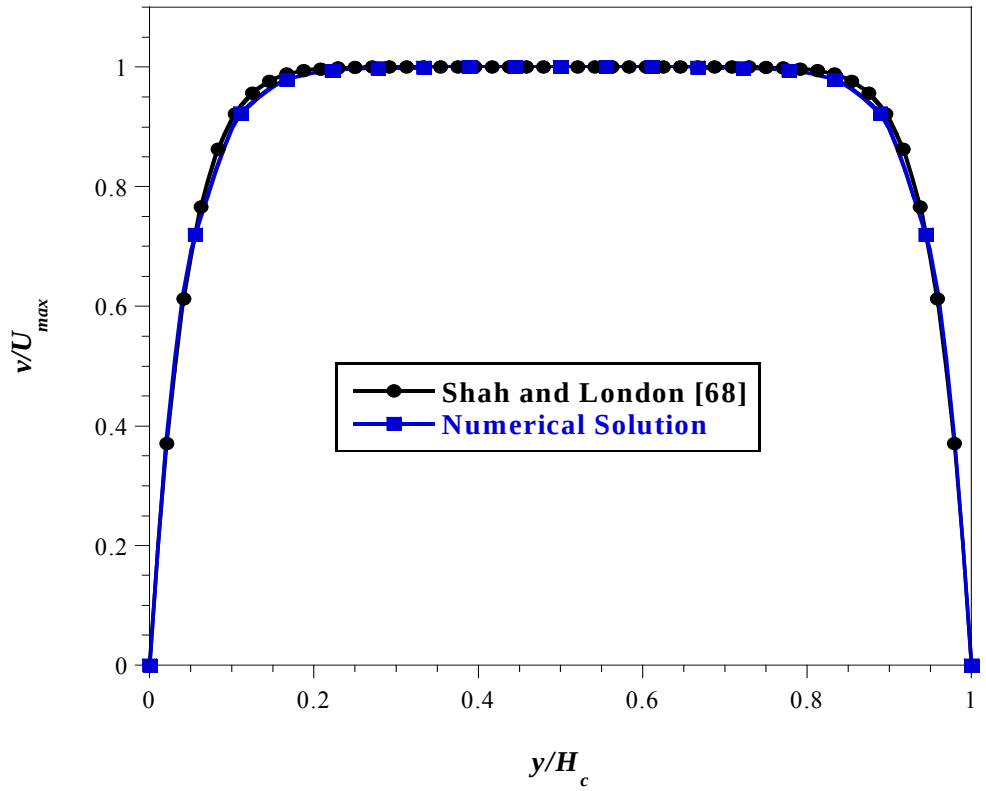


Figure 5-6: Comparison between numerical and analytical prediction for fully developed velocity profile along the *y*-axis

The energy equation was also verified by comparing the pure convective Nusselt number *Nu* with that given by Shah and London [68] and the comparison is shown in Figure 5-7. Using a constant longitudinal wall heat flux with uniform peripheral heat flux boundary condition, a high *Nu* is experienced at the entrance region but converges to the analytical *Nu* of 2.94 once the velocity and thermal boundary layers are fully developed, which happens for laminar flow at $0.05ReD_h$ and $0.05RePrD_h$ respectively. This is the reason why in Figure 5-7, the *z*-axis of the graph, which is the non-dimensionalised axial length starts at $z/L = 0.5$

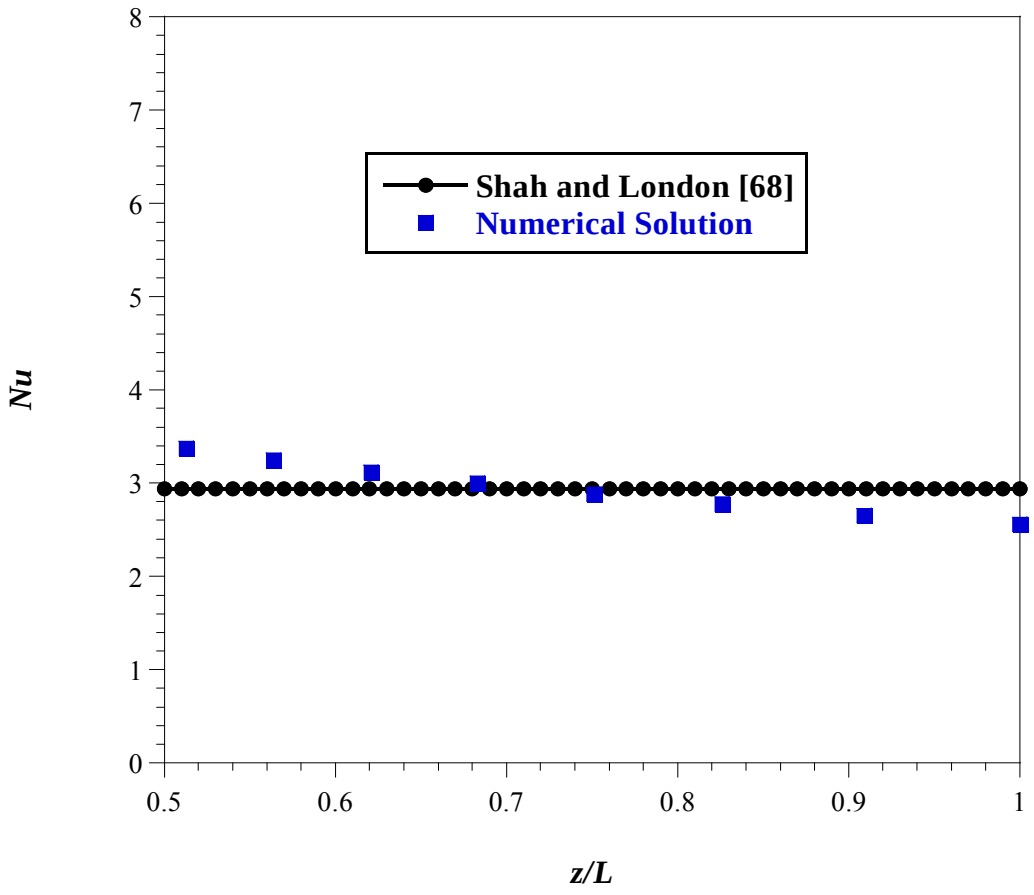


Figure 5-7: Comparison between numerical and analytical prediction of *Nu* profile along the channel length

5.2.3 Mathematical formulation of the optimisation problem

Objective (cost) function

In this problem, it is the aim to find an optimum maximum peak temperature at the wall of the microchannel heat sink. Therefore, the objective function is the maximum wall temperature at the walls and the optimisation problem is to minimise this maximum wall temperature. This function is not available analytically but it is obtained via a CFD simulation using FLUENT [60].

Design variables

The design variables for any optimisation problem are those variables which a designer has control over. As this case is a geometric optimisation problem, it allows for the design variables to be chosen as the geometric parameters t_1, t_2, t_3, H and G as depicted in Figure 5-2. According to literature [17, 27, 30], these variables have a significant influence on the performance and cooling ability of heat sinks.

Constraints

1. Solid volume fraction: As defined by Bello-Ochende *et al.* [6], the solid volume fraction ϕ is the ratio of the solid volume to the total volume of the heat sink.

$$\phi = \frac{V_{solid}}{V} = \frac{A_{solid}L}{AL} = \frac{A_{solid}}{A} \tag{5-1}$$

   From Equation 5-1, it follows that the solid volume fraction is only dependent on the cross-sectional area of the heat sink. For the optimisation problem, the volume ratio was made to vary between 0.3 and 0.8.

2. Manufacturing constraints: Microchannels can currently only be manufactured with an aspect ratio (H_c/B) of up to 20:1 and 6:1 using DRIE (deep reactive ion etching) and potassium hydroxide (KOH) wet etching fabrication techniques respectively. Also fabrication techniques limit the thickness of the top and bottom wall to $50\mu m$ [69, 70]. Therefore, assuming the DRIE technique [71, 72] was used in manufacturing the heat sink, the following constraints were imposed (refer to Figure 5.2).

$$\frac{H_c}{B} \leq 20 \tag{5-2}$$

$$t_2 \geq 50\ \mu m \tag{5-3}$$

$$H - t_3 \geq 50\ \mu m \tag{5-4}$$

3. Total volume constraint: In order to ensure that the optimisation problem is valid, the computational volume is kept constant.

$$\therefore V = GHL = const \tag{5-5}$$

Considering that the length L is constant (as L = 10 mm), Equation 5-5 can be reduced to:

$$\therefore \frac{V}{L} = A = GH = const = 0.09\ mm^2 \tag{5-6}$$

Scaling of design variables

Scaling of the design variables proves necessary if the design variables are of different orders (for example, k is of order 1 000 while L is of order 1). Using the variables without adequately scaling them will lead to instabilities when choosing step sizes. Therefore, adequate formal variables were chosen in order to scale the design parameters to vary from zero to one.

The variables are chosen as:

$$\begin{aligned} x_1 &= 10t_1 \\ x_2 &= t_2 \\ x_3 &= t_3 \\ x_4 &= H \\ x_5 &= G \end{aligned} \tag{5-7}$$

Substituting Equation 5-7 into Equations 5-1, 5-3 and 5-4, results in the objective and constraints functions given in Equation 5-8. The inequality functions g_1(x) and g_2(x) are derived from the volume fraction constraint of Equation 5-1 while g_3(x) and g_4(x) are derived with reference to the manufacturing constraints of Equations 5-3 and 5-4 respectively.

$$
\begin{aligned}
&f(x) = T_{\max} \\
&g_1(x) = x_5x_2 - 0.2x_1x_2 + x_4x_5 - x_5x_3 + 0.2x_1x_3 - 0.072 \le 0 \\
&g_2(x) = -x_5x_2 + 0.2x_1x_2 - x_4x_5 + x_5x_3 - 0.2x_1x_3 + 0.027 \le 0 \\
&g_3(x) = \frac{x_3 - x_2}{20(x_5 - 0.2x_1)} - 1 \le 0 \\
&g_4(x) = 1 - \frac{x_4 - x_3}{0.05} \le 0
\end{aligned}
\tag{5-8}
$$

5.2.4 Formal mathematical statement of the optimisation problem

The formal mathematical statement of the optimisation problem (with reference to Equation 5-8) now becomes

minimise $f(x) = T_{max}$

such that

$$
\begin{aligned}
&g_1(x) = x_5x_2 - 0.2x_1x_2 + x_4x_5 - x_5x_3 + 0.2x_1x_3 - 0.072 \le 0 \\
&g_2(x) = -x_5x_2 + 0.2x_1x_2 - x_4x_5 + x_5x_3 - 0.2x_1x_3 + 0.027 \le 0 \\
&g_3(x) = \frac{x_3 - x_2}{20(x_5 - 0.2x_1)} - 1 \le 0 \\
&g_4(x) = 1 - \frac{x_4 - x_3}{0.05} \le 0
\end{aligned}
\tag{5-9}
$$

5.2.5 Automation of the optimisation problem

The optimisation problem was done automatically in a MATLAB [73] environment while using GAMBIT [58] and FLUENT [60] simultaneously for mesh generation and flow modelling respectively. This was made possible by the assistance of both GAMBIT [58] and FLUENT [60] journal files, which were executed in MATLAB [73] by Windows executable files.

The optimisation algorithm was initiated by a starting guess of the design variables. A GAMBIT [58] journal file *(Design_variables.jou)* was then written and executed in MATLAB [73]. Another GAMBIT [58] journal file *(Micro1.jou)* was executed to generate the computational unit geometry mesh while using the geometrical parameters declared by the previous GAMBIT [58] operation. The mesh created was

imported by FLUENT [60] where post-processing was carried out by another journal file *(Micro_fluent.jou)*, after which a temperature data file *(Temp_data.dta)* was written with all the temperatures at the various computational cells. This data file was then read into MATLAB [73], where the maximum temperature was found and equated to the objective function. The DYNAMIC-Q optimisation algorithm (supplied in Appendix A) written in MATLAB [73] was then used to find a better design variable vector. This cycle continued until convergence occured with the step size and function value convergence tolerances set at 1×10^{-5} and 1×10^{-8} respectively. Figure 5-8 provides a flow chart of the automated optimisation process.

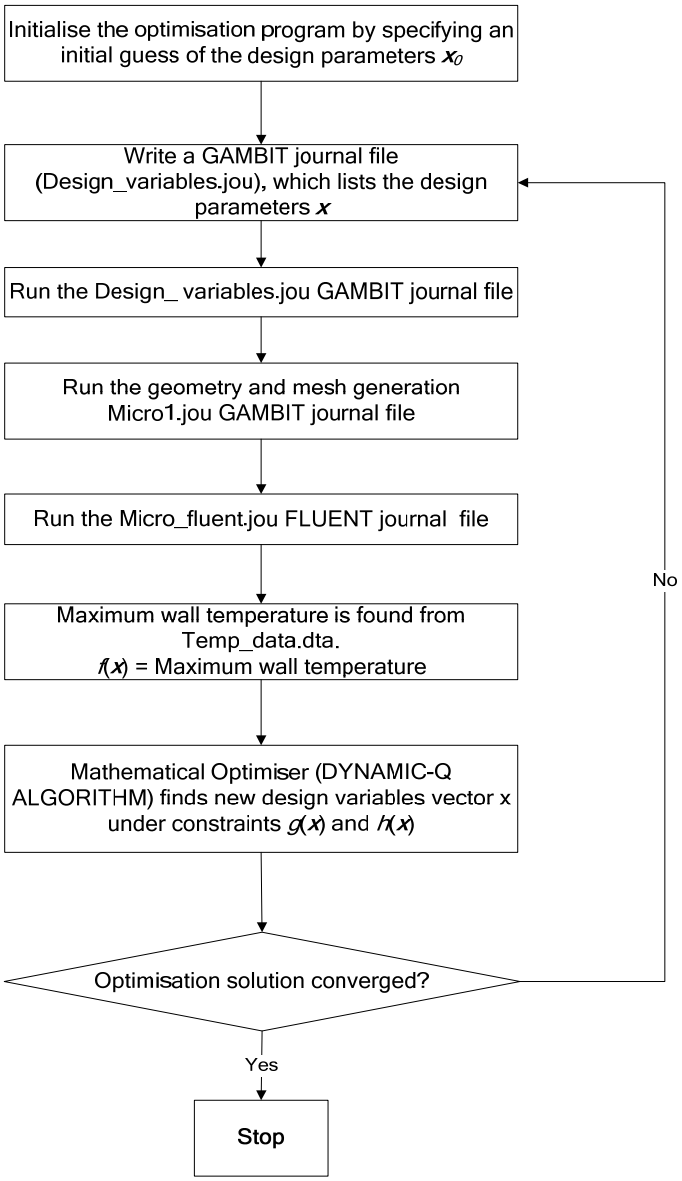


Figure 5-8: The optimisation process flow chart for the microchannel embedded inside a high conducting solid

5.2.6 Selection of appropriate forward differencing step size

In order to find an appropriate step size for the forward differencing scheme, numerical values for the maximum temperature at the walls were obtained as a function of the design parameter t_1 (half thickness of the vertical solid) for different step sizes. The step size that gave a smooth function of maximum temperature as a function of t_1 was selected as the candidate step size. This candidate step size was then verified by running the optimisation program with various starting guesses and checking for any discrepancies in the final solution. Figures 5-9 to 5-14 show that a step size of 1×10^{-4} gives a smooth continuous function of maximum temperature and it indeed proved to be an ideal forward differencing scheme step size for all design variables.

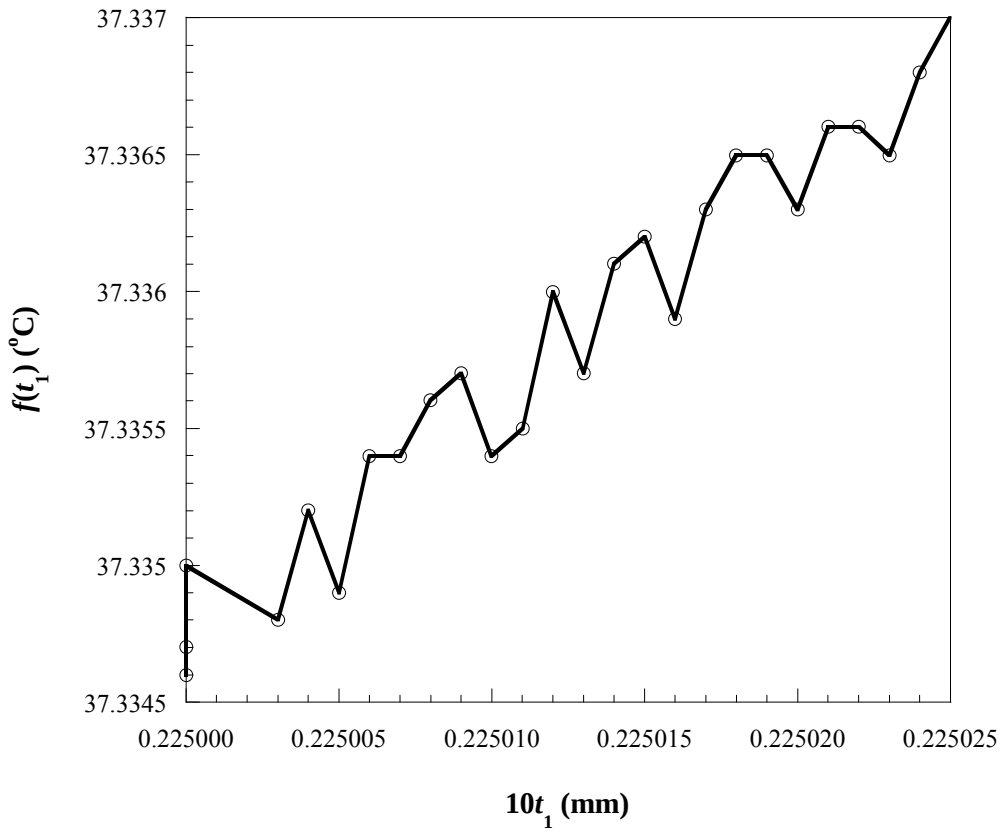


Figure 5-9: Plot of temperature at different t_1 values for a step size of 1E-6°C

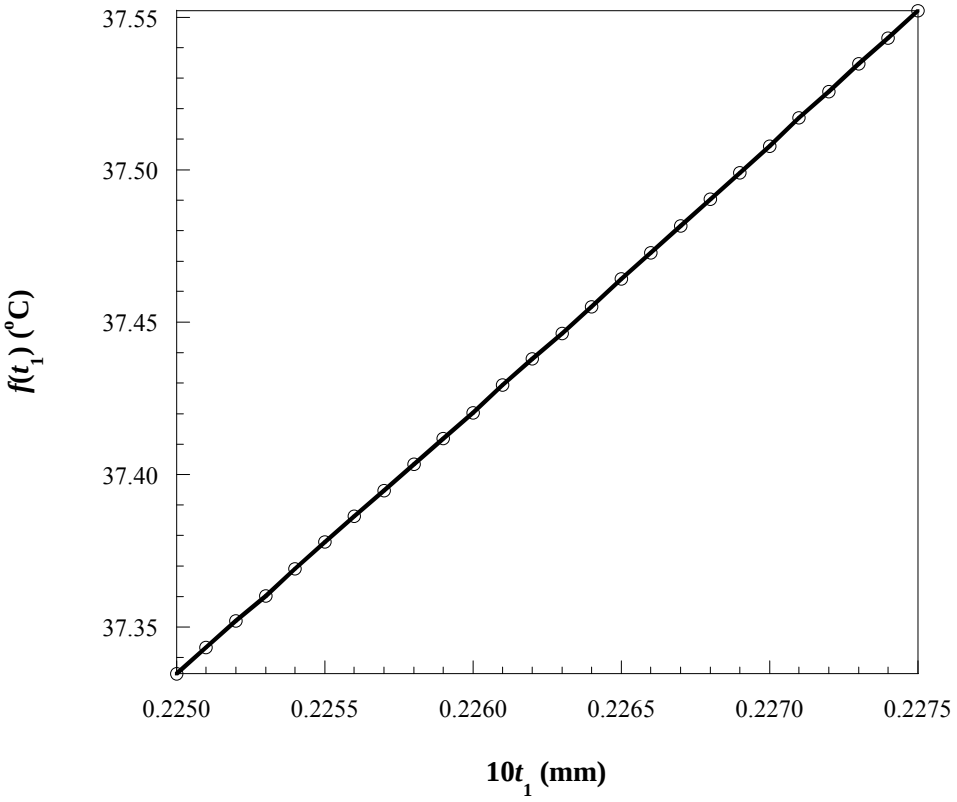


Figure 5-10: Plot of temperature at different t_1 values for a step size of 1E-4°C

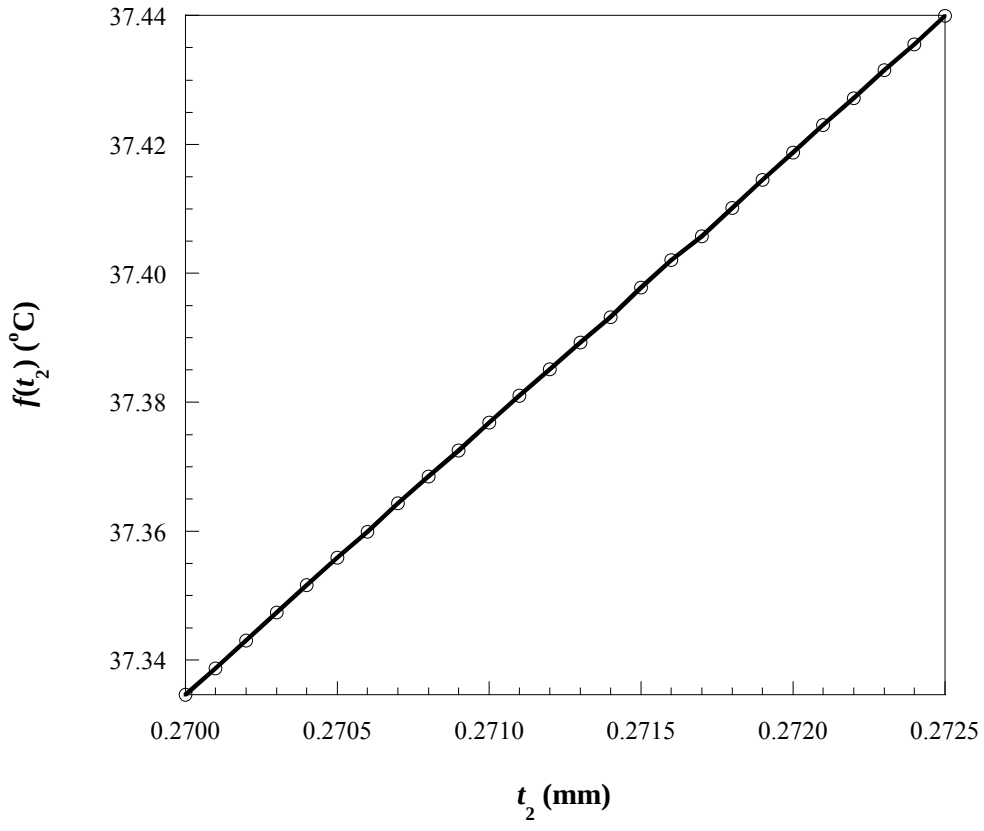


Figure 5-11: Plot of temperature at different t_2 values for a step size of 1E-4°C

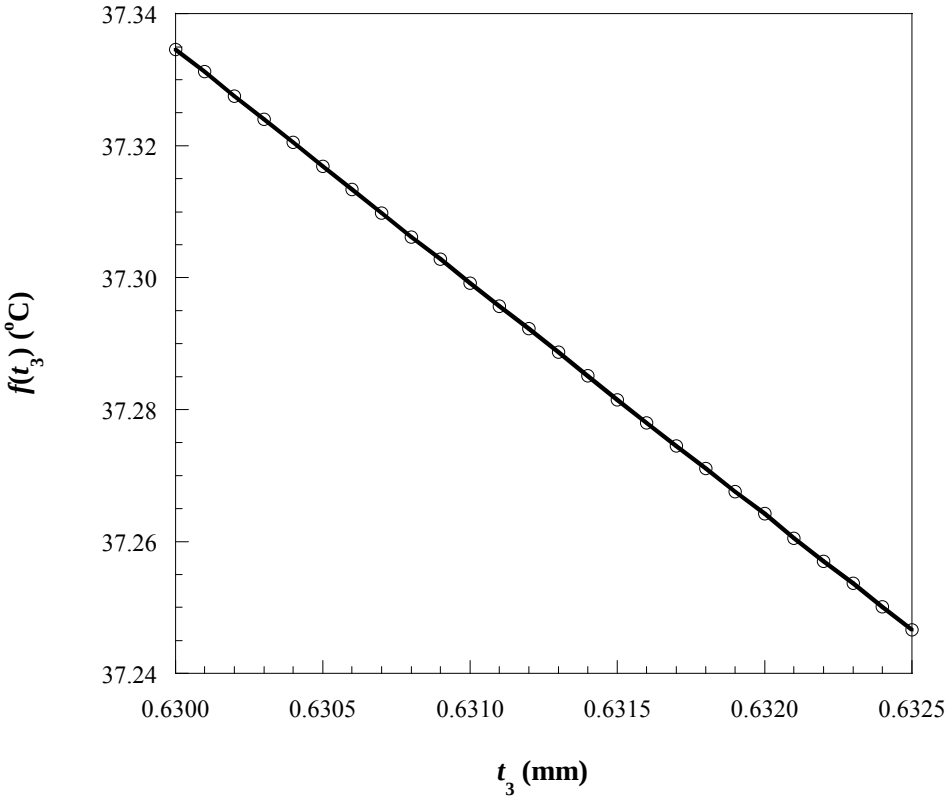


Figure 5-12: Plot of temperature at different t_3 values for a step size of 1E-4°C

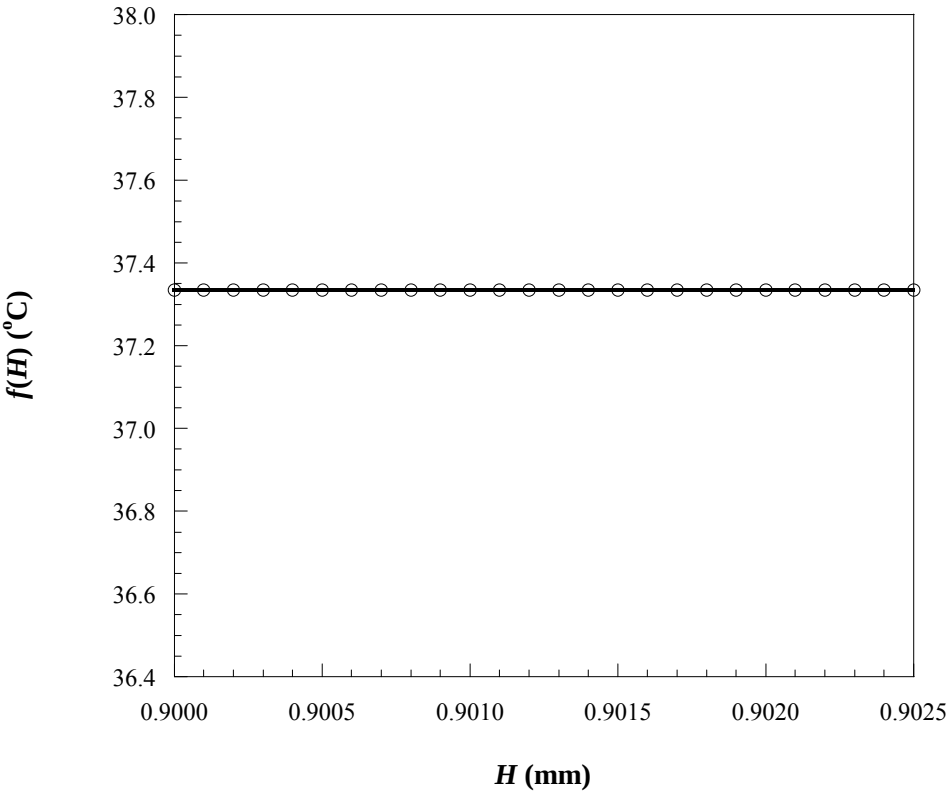


Figure 5-13: Plot of temperature at different *H* values for a step size of 1E-4°C

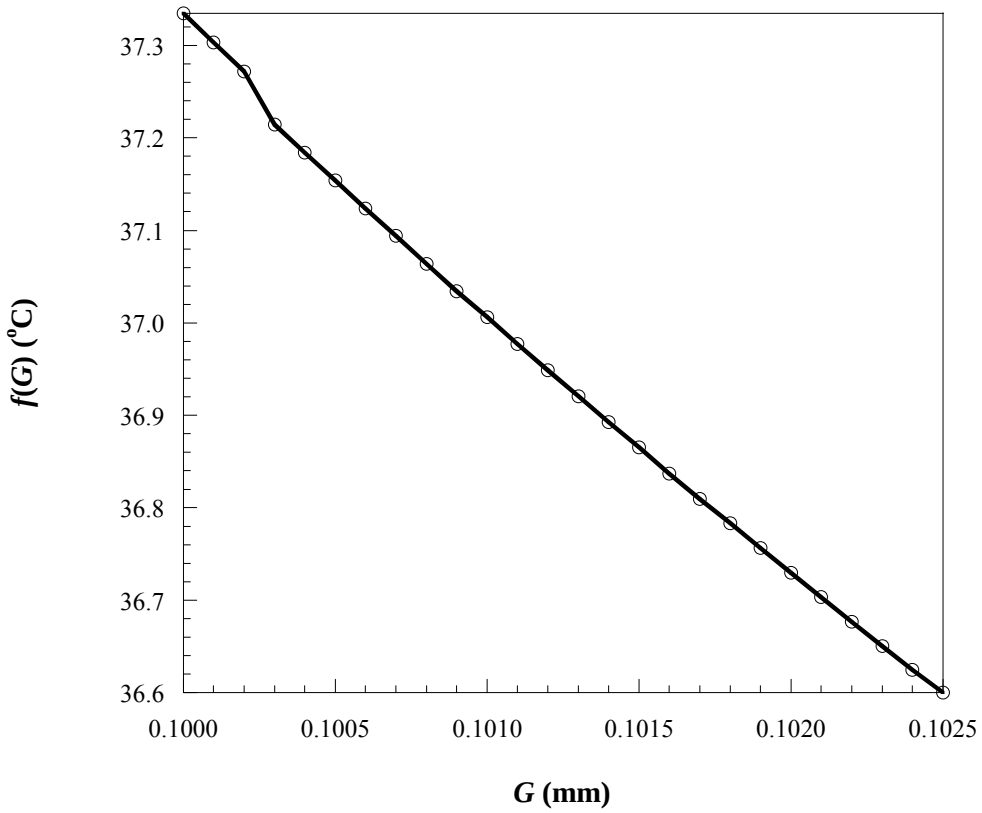


Figure 5-14: Plot of temperature at different *G* values for a step size of 1E-4°C

5.2.7 Results

5.2.7.1 Optimisation results

Figures 5-15 to 5-19 show the search history of the objective function and the convergence history of the design variables during the optimisation process for $Be = 3.2\times10^{8}$ across the heat sink. The dimensionless pressure drop parameter is the Bejan number (Be) and is defined as

$$Be = \frac{\Delta P V^{2/3}}{\alpha\mu} \tag{5-10}$$

After the three phases of the optimisation process, an optimal maximum wall temperature was obtained but it should be noted that due to the nature of the function, all the constraints did not intersect at Phase 2, thus a small constraint violation (inequality constraint function slightly greater than zero) was incurred, which was however, negligible.

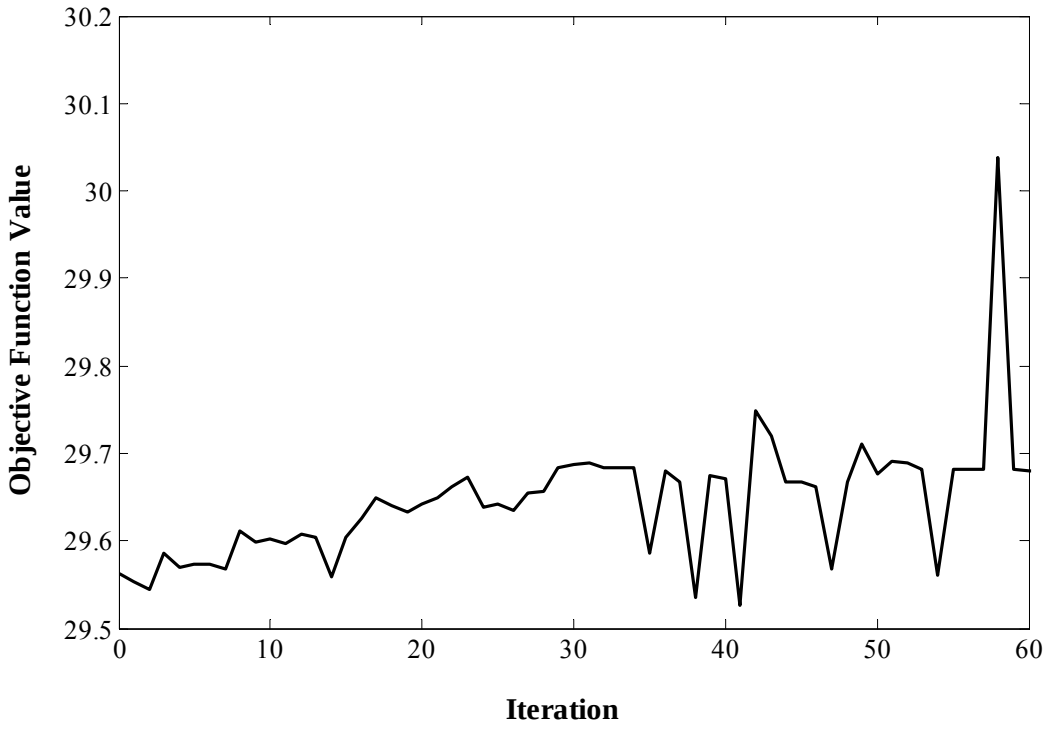


Figure 5-15: Search history of the objective function

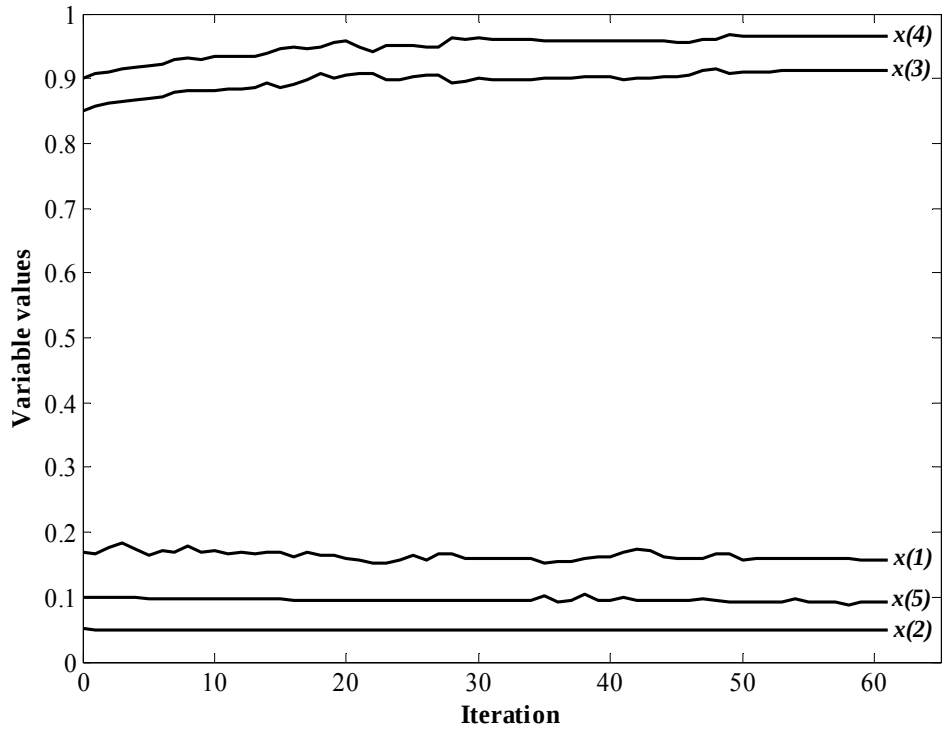


Figure 5-16: Convergence history of the design variables

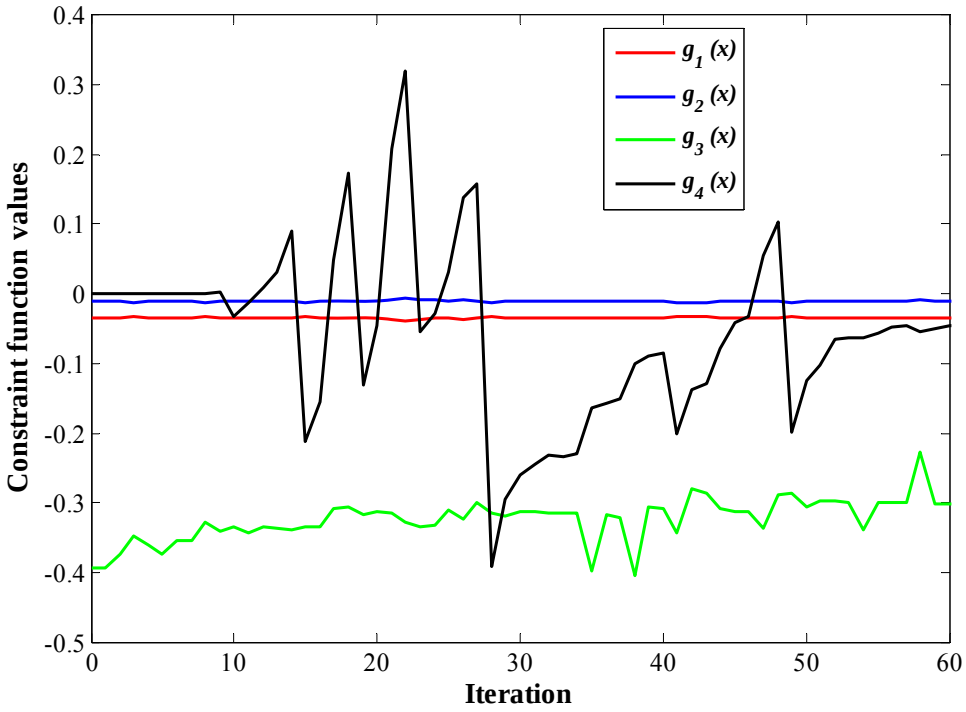


Figure 5-17: Convergence history of inequality constraints

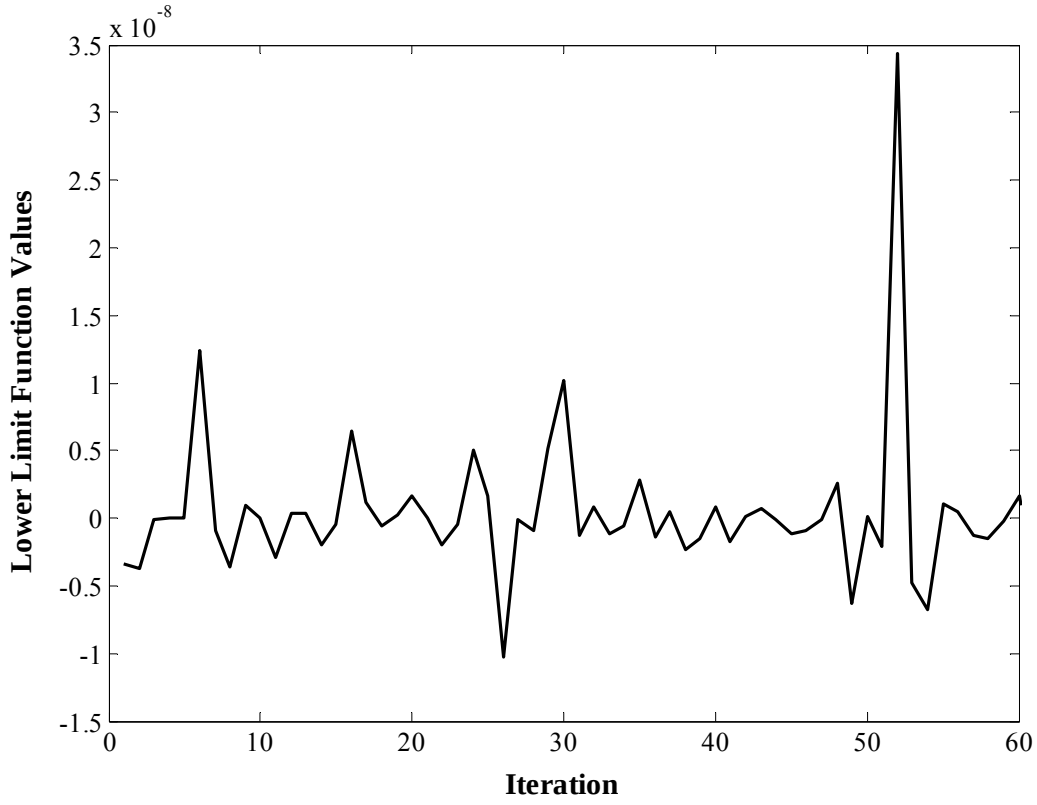


Figure 5-18: History of lower limit inequality constraint

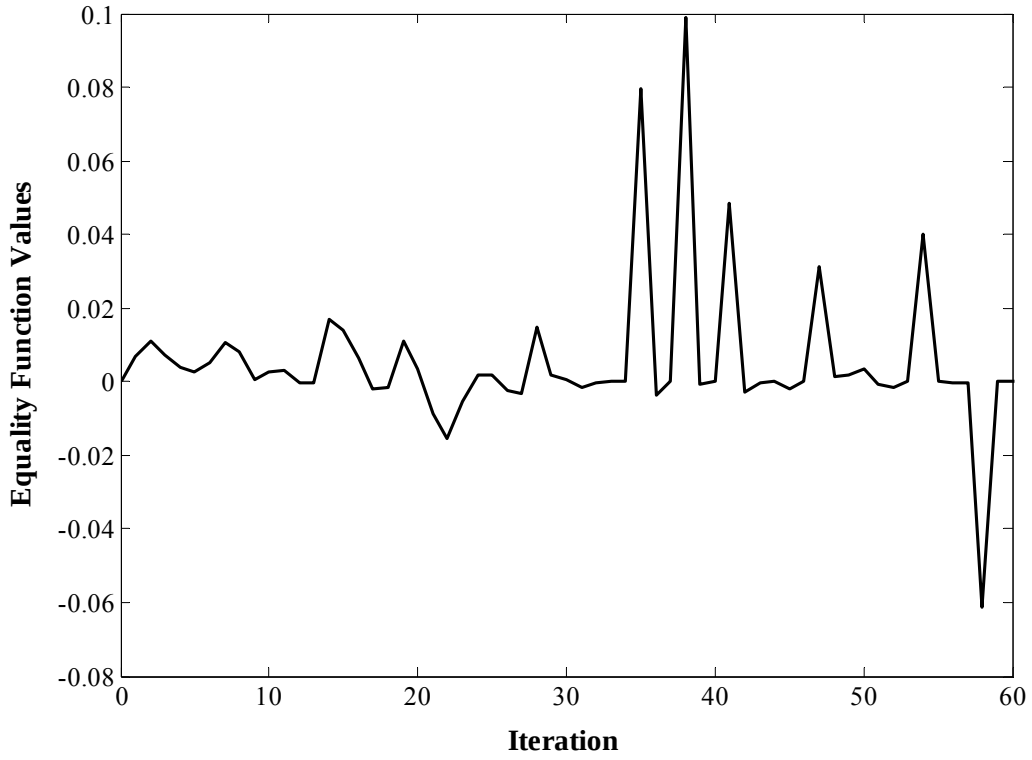


Figure 5-19: Convergence history of equality constraint

5.2.7.2 Optimal heat transfer results

The solution of the conjugate heat transfer problem showed a gradual increase in temperature across the channel length from the fluid inlet to its outlet as is shown in the plot of the temperature profile along the length of the channel in Figure 5-20. Figure 5-21 also shows that the fluid gets heated up as it passes through the channel. Furthermore, the rate at which the fluid is being heated is higher at the entrance region due to the growing thermal boundary layer till fully developed flow is reached (approximately at 2 mm from the inlet). As one moves along the length of the channel, a decrease in the temperature difference between the fluid and the solid is evident.

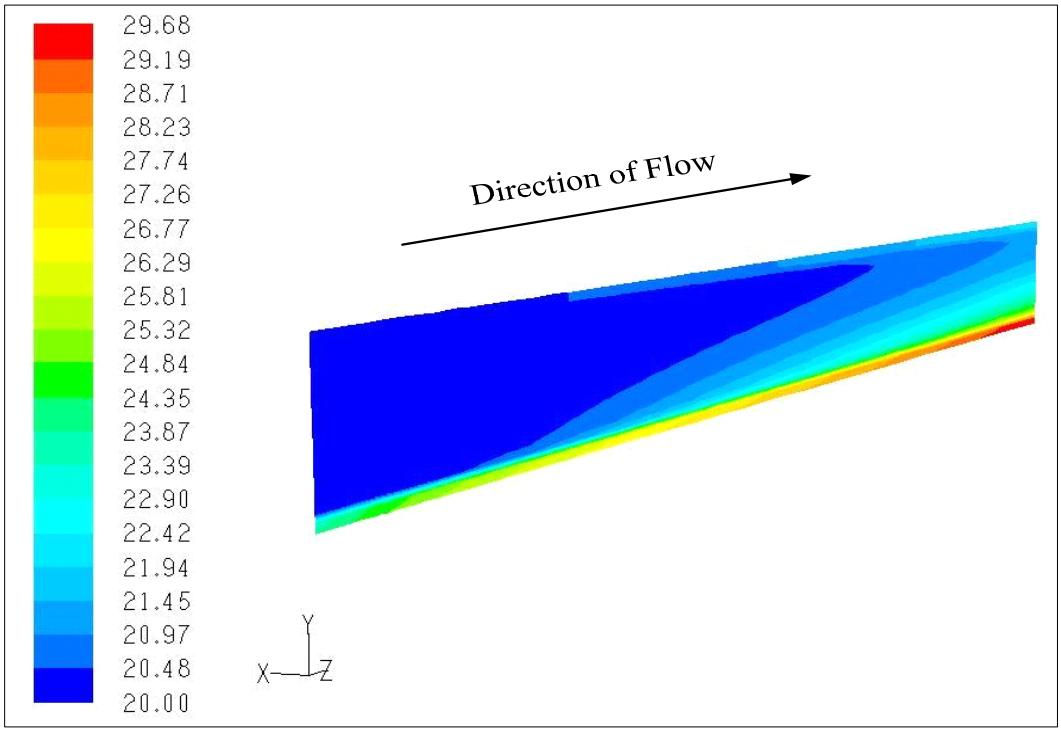


Figure 5-20: Temperature contour (in °C) of the optimised microchannel heat sink for $Be = 3.2\times10^{8}$

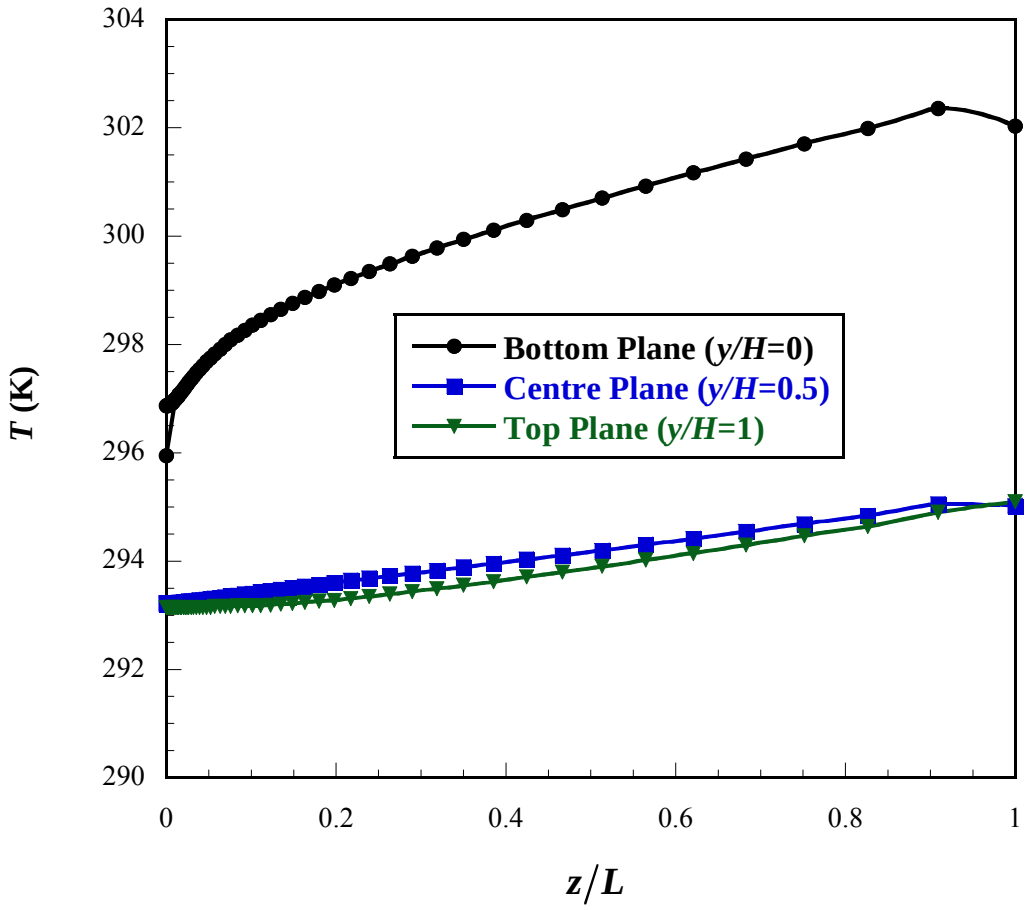


Figure 5-21: Plot of the temperature along the channel length

The highest temperature (hot spot) is encountered at the bottom wall of the heat sink ($y/H = 0$) at the region of the fluid outlet. This is due to the fact that at this point the heat transfer is minimal as the fluid temperature has increased in the longitudinal direction. The temperature distribution in the transverse axis (Figure 5-22) shows a decrease in the solid temperature with an increase in height with the converse applying to the fluid temperature.

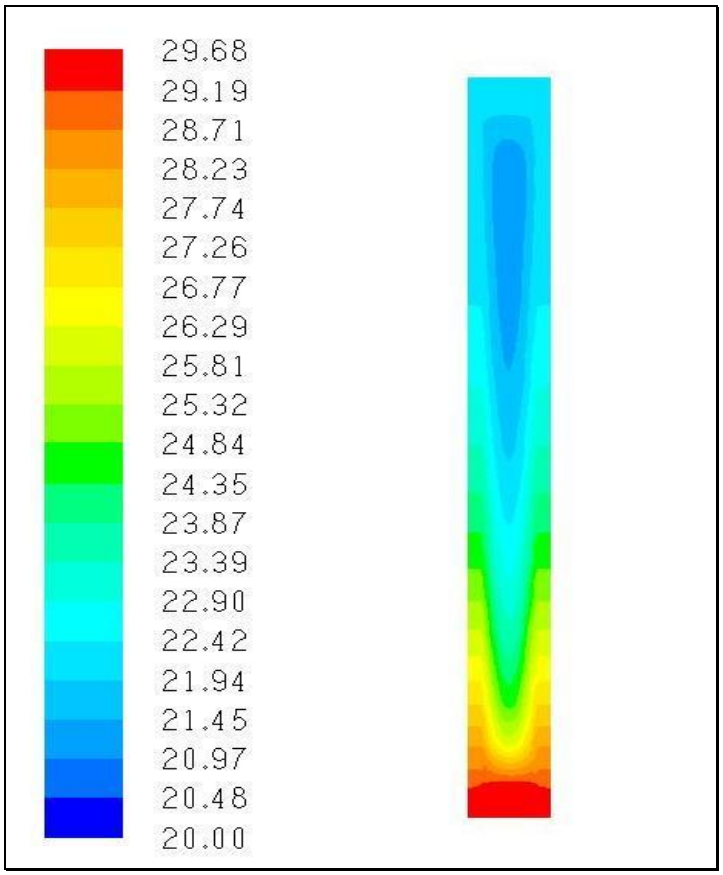


Figure 5-22: Temperature distribution (in °C) of the optimised microchannel heat sink along the transverse axis for $Be = 3.2\times10^{8}$

Figure 5-23 describes the trend of the minimised microchannel wall peak temperature difference in relation to different dimensionless pressure drops across the channel. The peak temperature difference decreases linearly with an increase in pressure drop. This trend correlates as:

$$2.85\times10^{4}\,Be^{-0.41} \tag{5-11}$$

As shown in Figure 5-24, the optimal aspect ratio of the microchannel shows a varying relationship with a change in *Be*. Initially, an increase in the dimensionless pressure drop parameter results in an increase in the optimal aspect ratio until approximately $Be = 2\times10^{8}$, where a decrease in the aspect ratio is observed with any further increase in *Be*. This trend correlates accurately with results already published [6, 69, 74].

Figure 5-25 shows the effect of the *Be* on the optimal solid volume fraction ϕ_{opt}. In general, an increase in *Be* results in an increase in ϕ_{opt}. An approximate linear relationship exists between ϕ_{opt} and the dimensionless pressure drop with the optimal solid volume fraction range being between 0.32 and 0.44, which agrees with the results published by Bello-Ochende *et al.* [6].

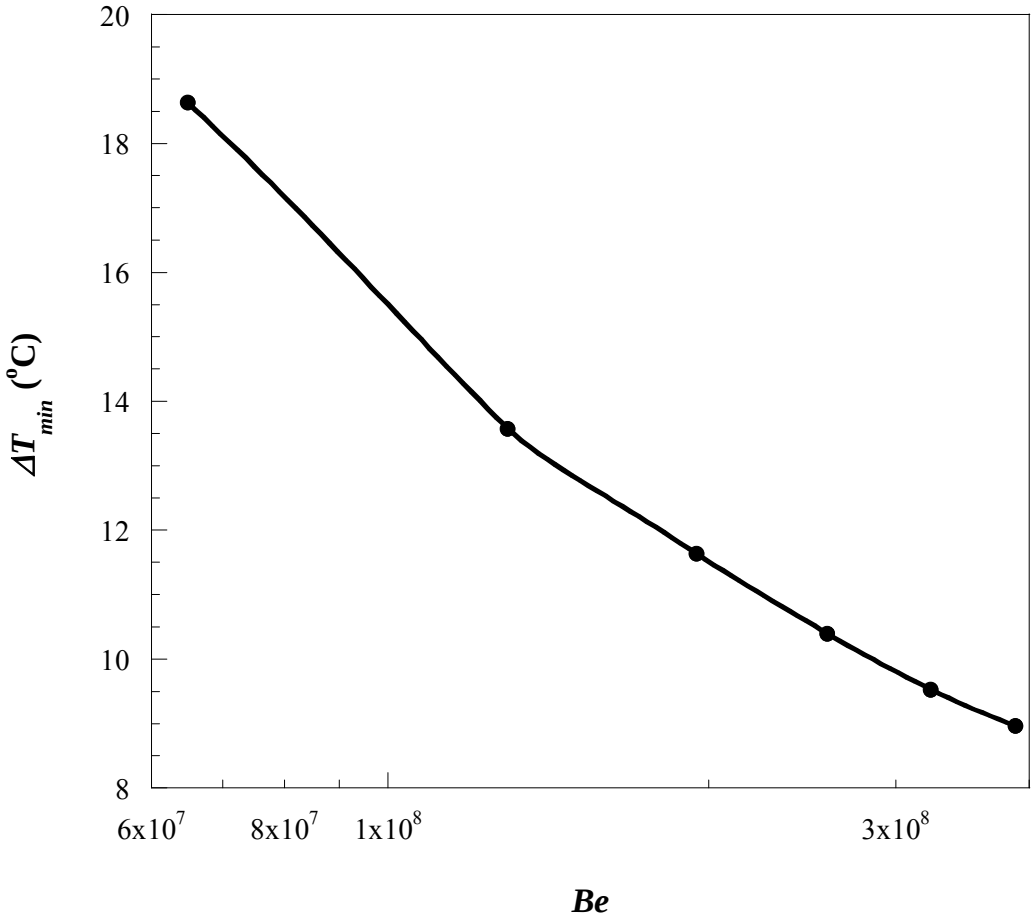


Figure 5-23: The influence of the dimensionless pressure drop parameter on the optimal peak wall temperature difference

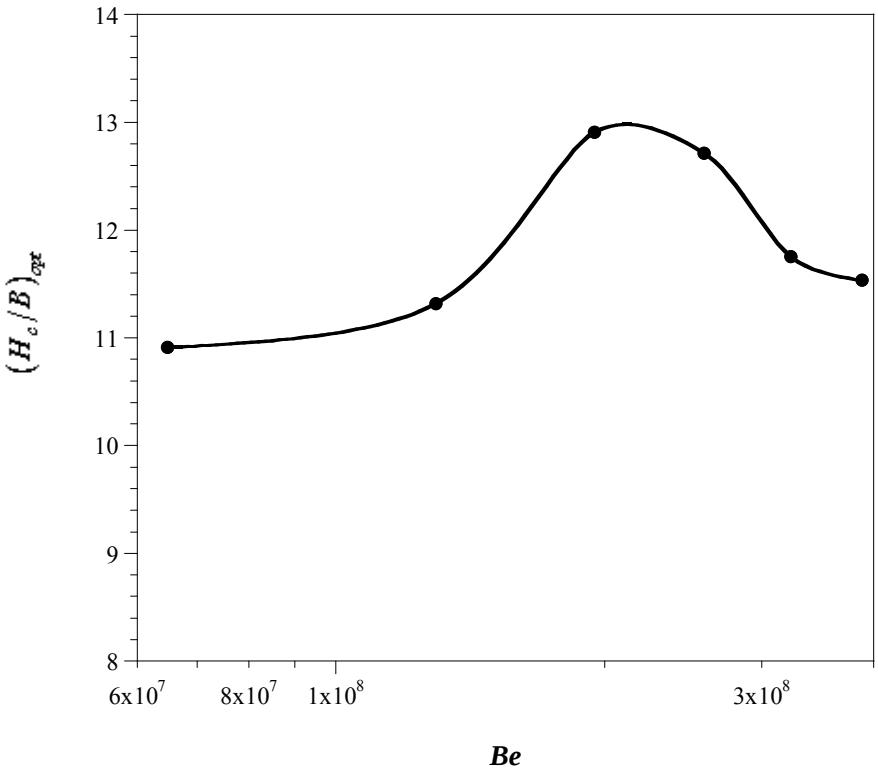


Figure 5-24: The effect of the change in *Be* on the optimal channel aspect ratio

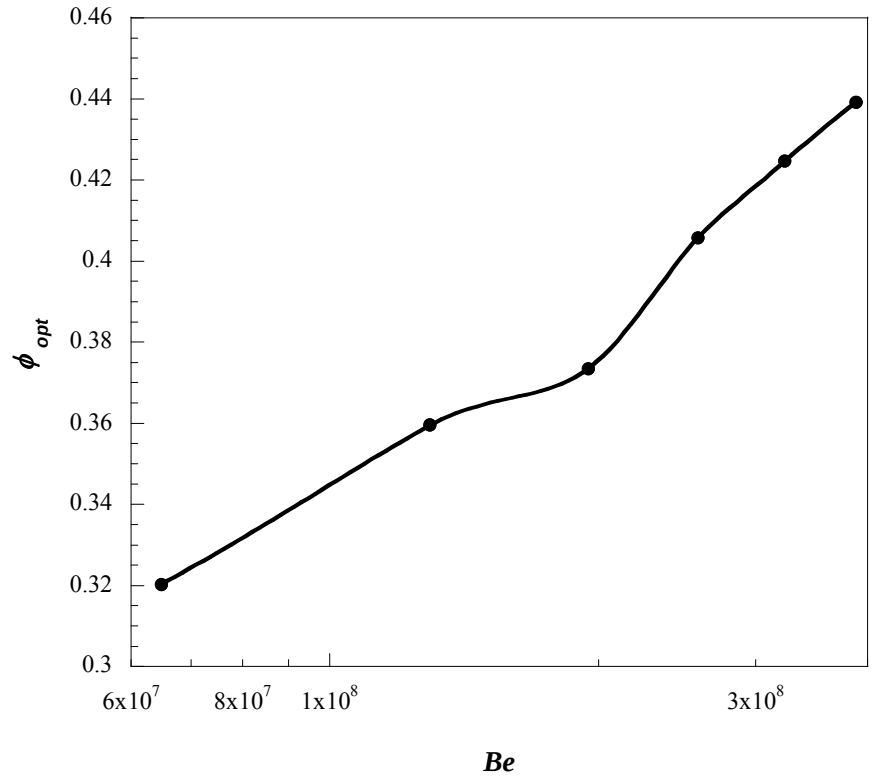


Figure 5-25: The effect of the change in *Be* on the optimal solid volume fraction

Figure 5-26 shows a direct proportional relationship between the maximised dimensionless global thermal conductance and the dimensionless pressure drop where the global thermal conductance C is a dimensionless ratio of the heat transfer rate to the peak wall temperature difference of a heat sink and is expressed as

$$C_{\max} = \frac{q"L}{k\Delta T_{\min}} \tag{5-12}$$

Figure 5-26 shows that the maximum global conductance increases with an increase in the dimensionless pressure drop (Bejan number). Using the constructal theory, Bello-Ochende *et al.* [6] derived an expression for the theoretical global conductance as:

$$C_{\max,theoretical} = 0.864 \frac{Be^{1/2}}{Po^{1/4}} \tag{5-13}$$

with the Poiseuille number *Po* defined as:

$$Po = \frac{12}{\left(1+\frac{B}{H_c}\right)^2 \left[1-\frac{192}{\pi^5}\frac{B}{H_c}\tanh\left(\frac{\pi}{2}\frac{H_c}{B}\right)\right]} \tag{5-14}$$

When comparing this derived conductance with that obtained from the mathematical optimisation, similar trends were found as shown in Figure 5-27. However, deviations, which are attributed to simplifying assumptions made in the formulation of the theoretical global conductance, were experienced. Figure 5-27 also shows similar trends when comparing the maximised global thermal conductance with the numerical prediction of Bello-Ochende *et al.* [6].

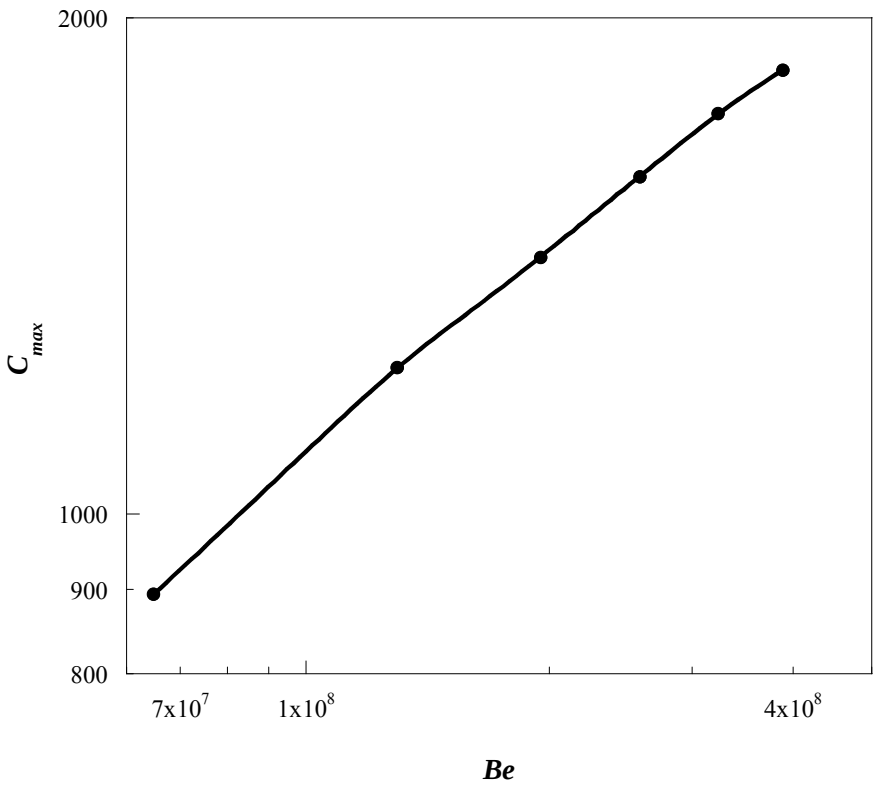


Figure 5-26: The influence of the dimensionless pressure drop on the maximised global thermal conductance of a microchannel heat sink

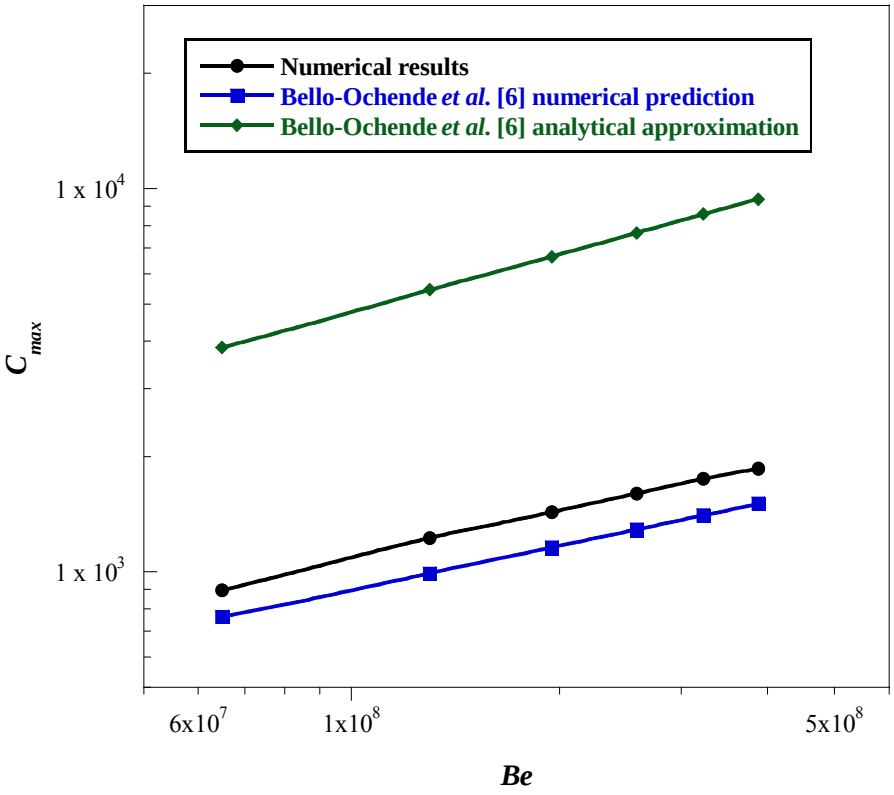


Figure 5-27: A comparison between the theoretical and numerical maximised global thermal conductance [6] with the numerically maximised conductance obtained in this study

The optimal hydraulic diameter trend for *Be* range between 6.5×10^7 to 4×10^8 is shown in Figure 5-28. A decrease in the optimal hydraulic diameter of the heat sink is observed with an increase in *Be*. This decrease continues until the hydraulic diameter is such that the cooling fluid being pumped in is not sufficient to cause the desired cooling. The optimal hydraulic diameter ranges from 120 µm to 140 µm, which does not violate the assumption made when the computational model was defined.

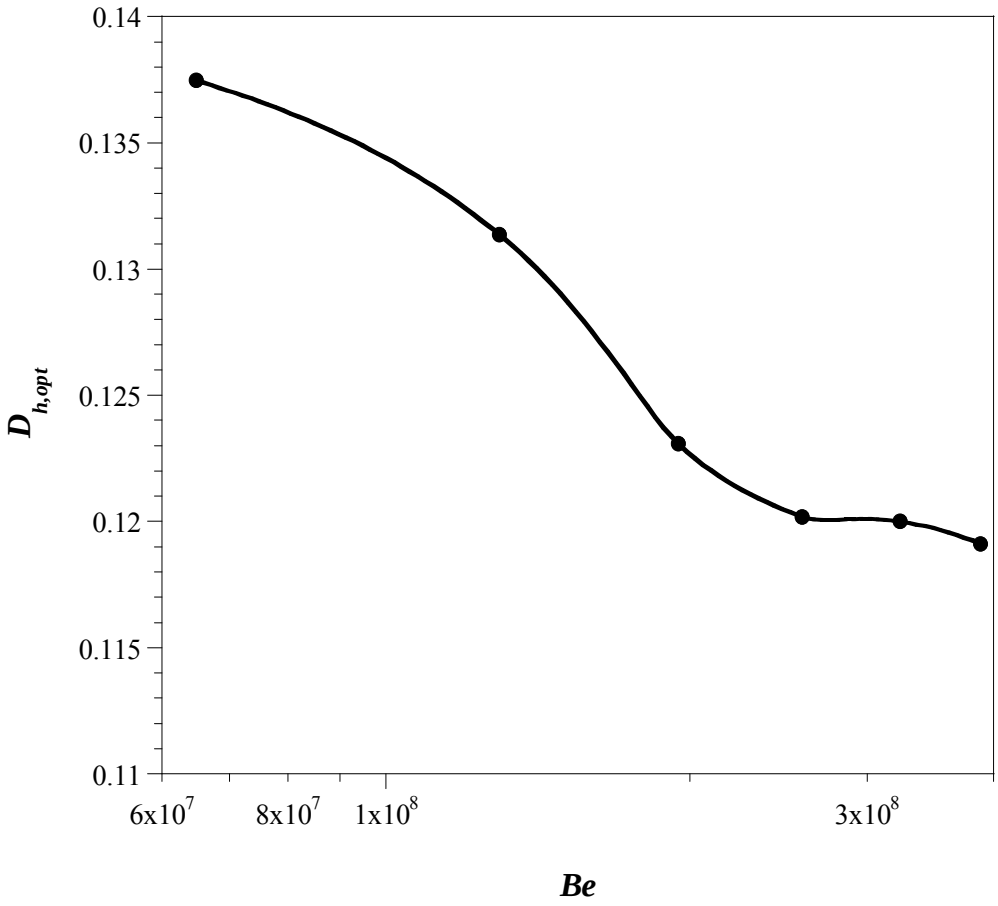


Figure 5-28: The effect of changes in the dimensionless pressure drop parameter on the optimal hydraulic diameter

The volume constraint was relaxed and then decreased gradually from the initial set volume of 0.9 mm^3 to investigate the influence of the computational volume on the heat sink optimal dimensions. Table 5-5 gives the design results for a range of constant computational volumes for $Be = 3.2\times10^8$. These results show a decrease in the minimised wall peak temperature with an increase in the heat sink volume as the heat flux generated within the volume increases when the heat sink volume is decreased. The table also shows that the optimised volume fraction ϕ_{opt} and hydraulic diameter increase as the volume increases.

Table 5-5: Optimal design results for various computational volumes

Volume (mm^3)	Minimised Peak Temperature (^{o}C)	Optimised Aspect Ratio $(H_c/B)_{opt}$	Optimised Volume Fraction ϕ_{opt}	$(D_h)opt$ (mm)
0.9	29.53	11.752	0.425	0.122
0.8	29.79	10.069	0.425	0.123
0.7	30.12	10.359	0.386	0.118

The optimisation process was then executed with the length not fixed to 10 mm but relaxed, increasing the degree of freedom of the heat sink thereby obtaining an optimal length. It proved to offer better optimal cooling effects at lower pressure drops with more than a 3°C decrease in the optimal peak wall temperature difference at $Be = 7\times10^7$ as shown in Figure 5-29. Table 5-6 documents the optimal design parameters for the heat sink when the axial length is relaxed. The results show a linear increasing trend of the optimised aspect ratio as a function of Be with the ratio of solid volume to total volume between 0.38 and 0.44. This optimal configuration provides improved heat transfer capabilities with an increased maximised global thermal conductance of the heat sink of up to 20% at low pressure drops.

Table 5-6: Optimal design results when the axial length is relaxed

Be	Optimised Aspect Ratio $(H_c/B)_{opt}$	Optimised Volume Fraction ϕ_{opt}	$(D_h)_{opt}$ (mm)	C_{max}
3.9×10^8	11.8	0.440	0.126	1884
3.2×10^8	11.3	0.439	0.131	1791
2.6×10^8	10.8	0.440	0.139	1683
1.9×10^8	10.1	0.429	0.149	1544
1.3×10^8	9.17	0.407	0.161	1355
6.5×10^7	7.87	0.382	0.188	1082

\

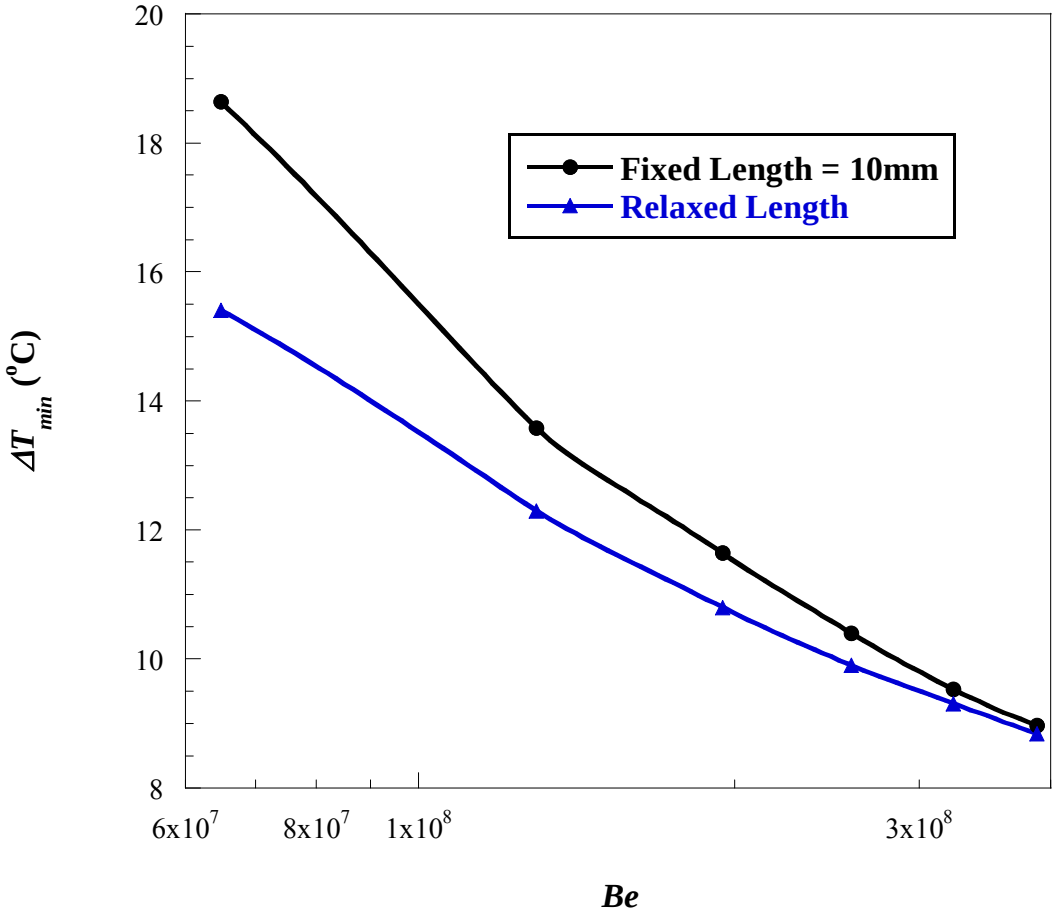


Figure 5-29: The effect of the relaxation of the axial length as compared with the fixed length optimal peak wall temperature difference

Figure 5-30 gives a relationship between the optimised length and the dimensionless pressure drop. The results show that as the pressure drop is increased, the resulting optimal channel configuration will be of a longer but slender nature.

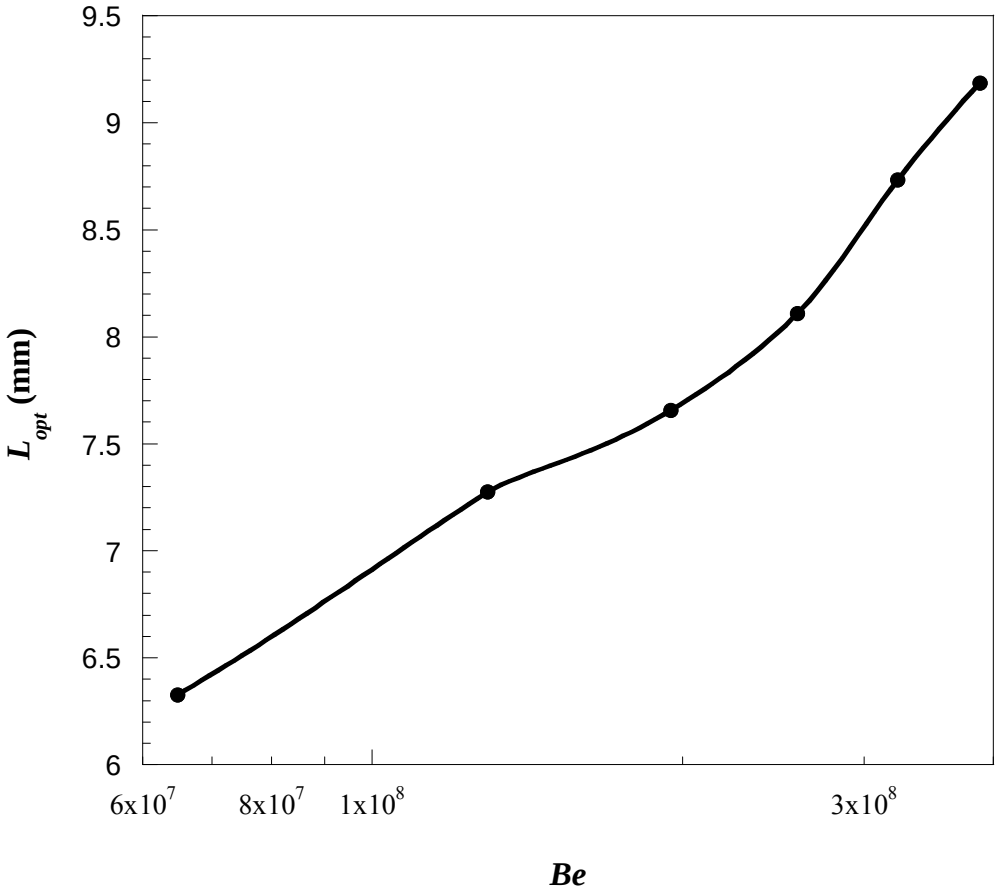


Figure 5-30: The optimal axial length as a function of the *Be*

5.3 CASE STUDY 2: DOUBLE ROW MICROPIN-FIN CONFIGURATION

This part of the study considers two rows of a micropin-fin (cylindrical) heat sink. The geometric design of a heat sink which will result in the best heat transfer rate is the main consideration. The resulting heat transfer across the cylindrical micropin-fin is by laminar forced convection of a uniform isothermal free stream. The optimisation process is carried out numerically under fixed constraints.

5.3.1 The CFD model

Figure 5-31 gives the physical model of a double row micropin-fin geometry and its unit cell computational domain is given in Figure 5-32. The two fins of varying diameters D_1 and D_2 and respective heights H_1 and H_2 spaced at a distance s from each other aim to enhance the extraction of heat supplied to the base of the thermal conductive material at a temperature T_w. The distance between the leading edge and the first fin is $s/2$. Air, which is uniform and isothermal, is driven across the heat sink of fixed flow length L and width G. A computational domain with overall dimensions of 1mm × 0.6mm × 1mm is used for this analysis

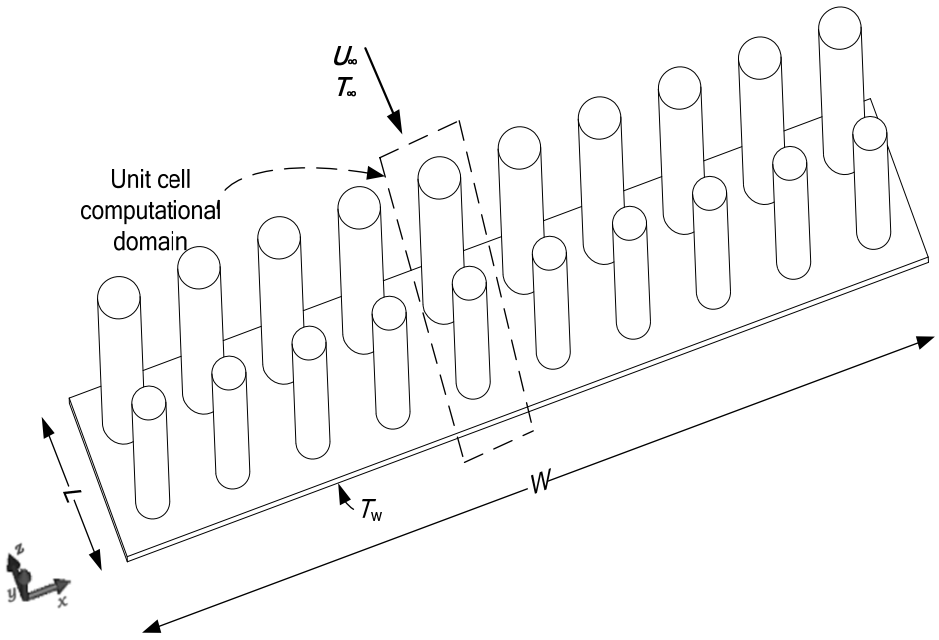


Figure 5-31: Physical model of a double row finned heat sink

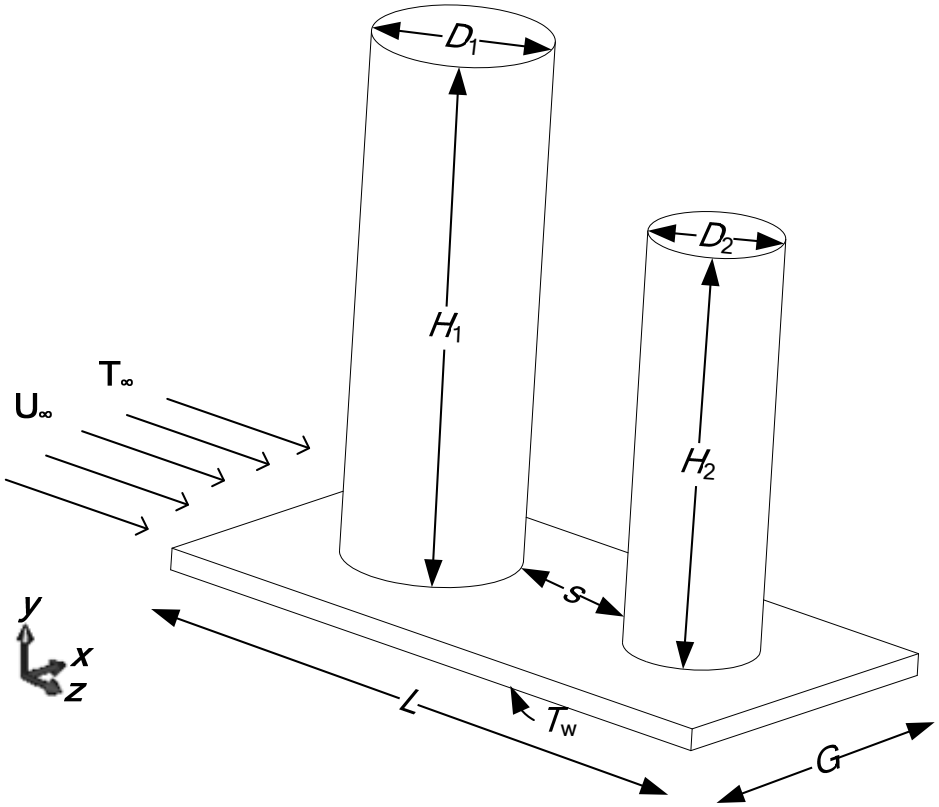


Figure 5-32: Unit cell computational domain of a micropin-fin heat sink

The mass, momentum and energy conservation equations were solved over the discretised domain shown in Figure 5-33 using the finite volume CFD code FLUENT [60]. GAMBIT [58] and FLUENT [60] journal files for the micropin-fin are supplied in Appendix B and C respectively.

No-slip, no-penetration boundary conditions were enforced on the fin and wall surfaces, no flow was allowed at symmetry planes with outflow allowed only at the top plane and the outlet. A constant wall temperature of 100°C was applied to the base wall. The flow was assumed to be steady, laminar and incompressible with all fluid and material properties assumed to be constant. Figure 5-34 gives a schematic diagram of the boundary conditions enforced around the pin-fin heat sink.

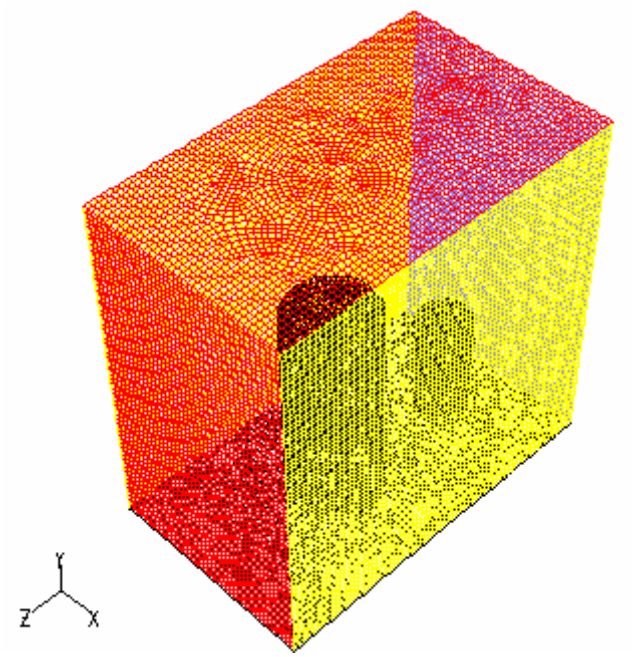


Figure 5-33: Double row pin-fin mesh grid

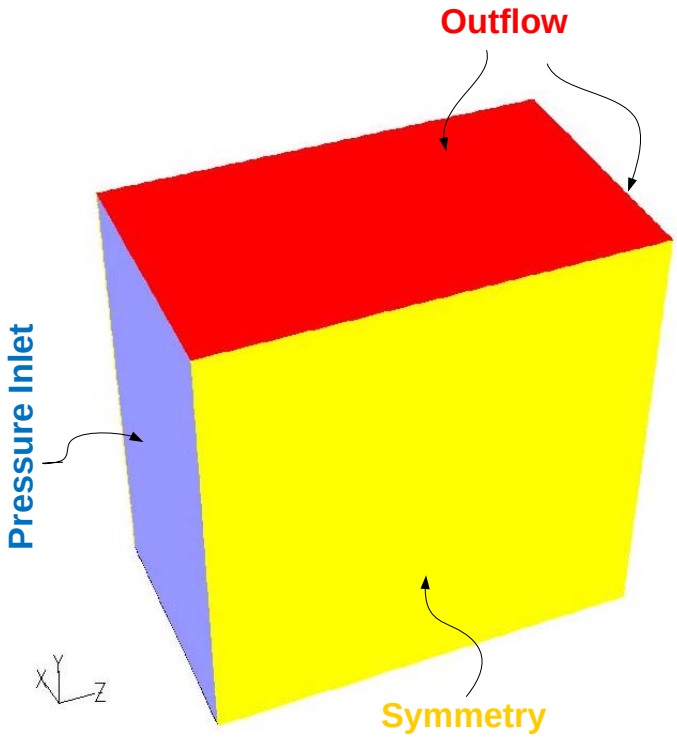


Figure 5-34: Boundary conditions enforced around the micropin-fin heat sink

5.3.2 Verification of the model

In order to verify the numerical model developed, grid independence tests were carried out on the pin-fin heat sink, whose dimensions are given in Table 5-7. The dimensionless measure of the rate of heat transfer is given by:

$$\tilde{q} = \frac{q}{Lk_f\left(T_w - T_\infty\right)} \tag{5-15}$$

where q is the overall rate of heat transfer, and T_w and T_∞ are the wall and free-stream temperatures respectively.

The tests were conducted for various control volume mesh sizes until the deviation in dimensionless heat transfer rate $\tilde{q}$ was negligible as shown in Figure 5-35 with the finest mesh consisting of 615 000 cells. The maximum average difference of $\tilde{q}$ encountered when using a mesh having greater than 159 768 cells was 2.2%, giving the confidence that the simulations carried out based on a 178 488-celled mesh provide satisfactory numerical accuracy.

Table 5-7: Heat sink dimensions used for the code validation process

D_1 (mm)	D_2 (mm)	H_1 (mm)	H_2 (mm)	s (mm)	G (mm)	L (mm)
0.15	0.25	0.3	0.6	0.2	0.6	1

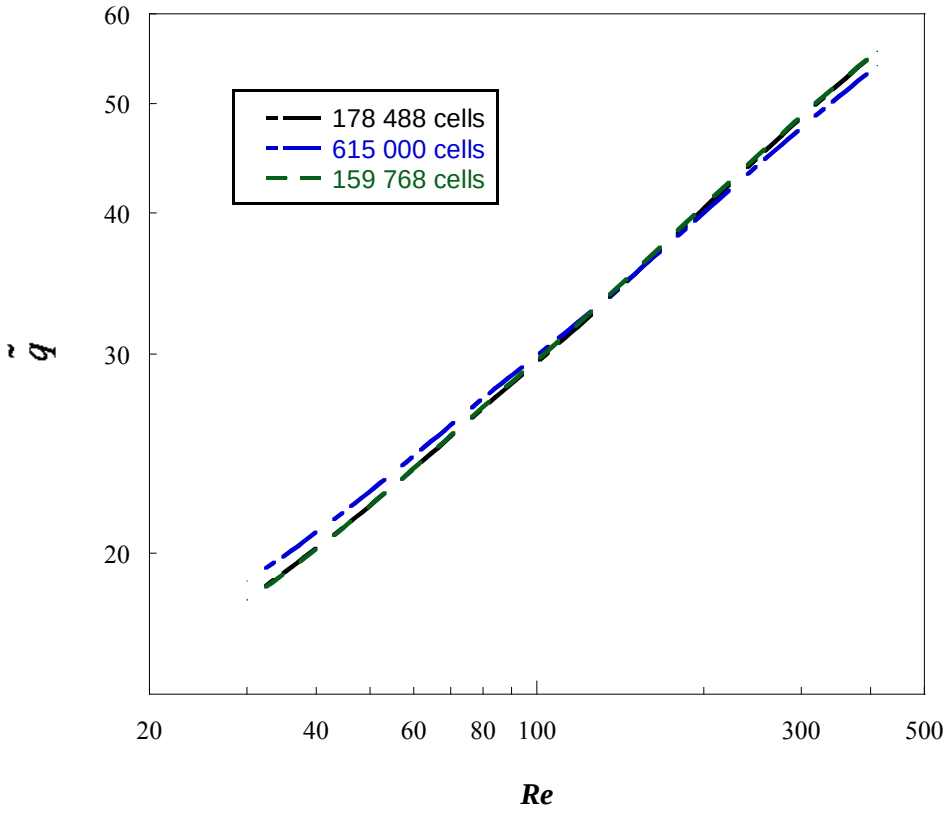


Figure 5-35: Grid independence test for the double row finned heat sink meshed grid

5.3.3 Mathematical formulation of the optimisation problem

Objective function and design variables

The objective of the optimisation problem is to find the best configuration of the geometric ratios $\frac{D_2}{D_1}$, $\frac{H_2}{H_1}$ and interspacing s that will maximise the rate of heat transfer from the solid to the fluid under fixed constraints. Therefore, the objective function is the rate of heat transfer and the design variables are the parameters D_1, D_2, H_1, H_2 and s.

Constraints

1. Total fin volume constraint: In heat sink design, weight and material cost of fins are limiting factors. Therefore, the total volume of the cylindrical fins is fixed to a constant value.

$$\therefore V_1 + V_2 = \text{Constant}$$
$$\frac{\pi D_1^2}{4} H_1 + \frac{\pi D_2^2}{4} H_2 = C \tag{5-16}$$
$$D_1^2 H_1 + D_2^2 H_2 = \frac{4C}{\pi}$$

where V_1 and V_2 are the volume of the fins.

2. Manufacturing constraint: Pin-fin manufacturing and size constraint allows for typical aspect ratios in the range of 0.5 and 4 [75, 76]. Considering fabrication techniques, interfin spacing is limited to 50 microns [69, 70], therefore the manufacturing constraint for this problem is expressed as

$$0.5 \le \frac{H_1}{D_1} \le 4$$
$$0.5 \le \frac{H_2}{D_2} \le 4 \tag{5-17}$$
$$s \ge 50\ \mu m$$

5.3.4 Formal mathematical statement of optimisation problem

Choosing the design variables as:

$$x_1 = D_1$$
$$x_2 = D_2$$
$$x_3 = H_1 \tag{5-18}$$
$$x_4 = H_2$$
$$x_5 = s$$

and substituting Equation 5-18 into Equations 5-16 and 5-17 results in the objective and constraints functions given in Equation 5-19. The inequality functions $g_1(x)$ to $g_4(x)$ are derived from the manufacturing constraint of Equations 5-17 while $h_1(x)$ is derived with reference to the total fin volume constraints of Equation 5-16.

Therefore, the formal mathematical optimisation problem can be written as:

Maximise $f(x)=q$

such that

$$g_1(x)=\frac{x_3}{4x_1}-1\le 0$$
$$g_2(x)=1-\frac{2x_3}{x_1}\le 0$$
$$g_3(x)=\frac{x_4}{4x_2}-1\le 0 \qquad (5\text{-}19)$$
$$g_4(x)=1-\frac{2x_4}{x_2}\le 0$$
$$h_1(x)=x_1^{\,2}x_3+x_2^{\,2}x_4-4\frac{C}{\pi}=0$$

An automated optimisation process similar to that shown in the flow chart given in Figure 5-8 was carried out with noise analysis resulting in an appropriate forward differencing step size of 1×10^{-4}.

5.3.5 Results

The optimisation procedure was carried out for Reynolds numbers ranging from 30 to 411 with the effect of Reynolds number on the pin-fin geometric optimal configuration and heat transfer capabilities investigated. Figure 5-36 establishes the fact that the optimal rate of heat transfer increases with an increase in Reynolds number. The results in Figure 5-36 were for a conductivity ratio of 100, which is the ratio of the solid's thermal conductivity to that of the fluids. This relationship between the maximum (optimal) dimensionless rate of total heat transfer $\tilde{q}_{\max}$ and Reynolds number *Re* can be given by the expression:

$$\tilde{q}_{\max} \propto Re_{\mathrm{L}}^{\,0.323} \qquad (5\text{-}20)$$

$$\therefore \tilde{q}_{\max} = \mathrm{C}Re_{\mathrm{L}}^{\,0.323} \qquad (5\text{-}21)$$

where C is a constant that depends on the thermal conductivity ratio k_r and scale effects. For a thermal conductivity ratio of 100 for micro-scale applications, the constant C was found to be 9.78. Equation 5-21 correlates within an error of less than 1% to the CFD results produced and it is in agreement with the work published recently by Bello-Ochende *et al.* [77].

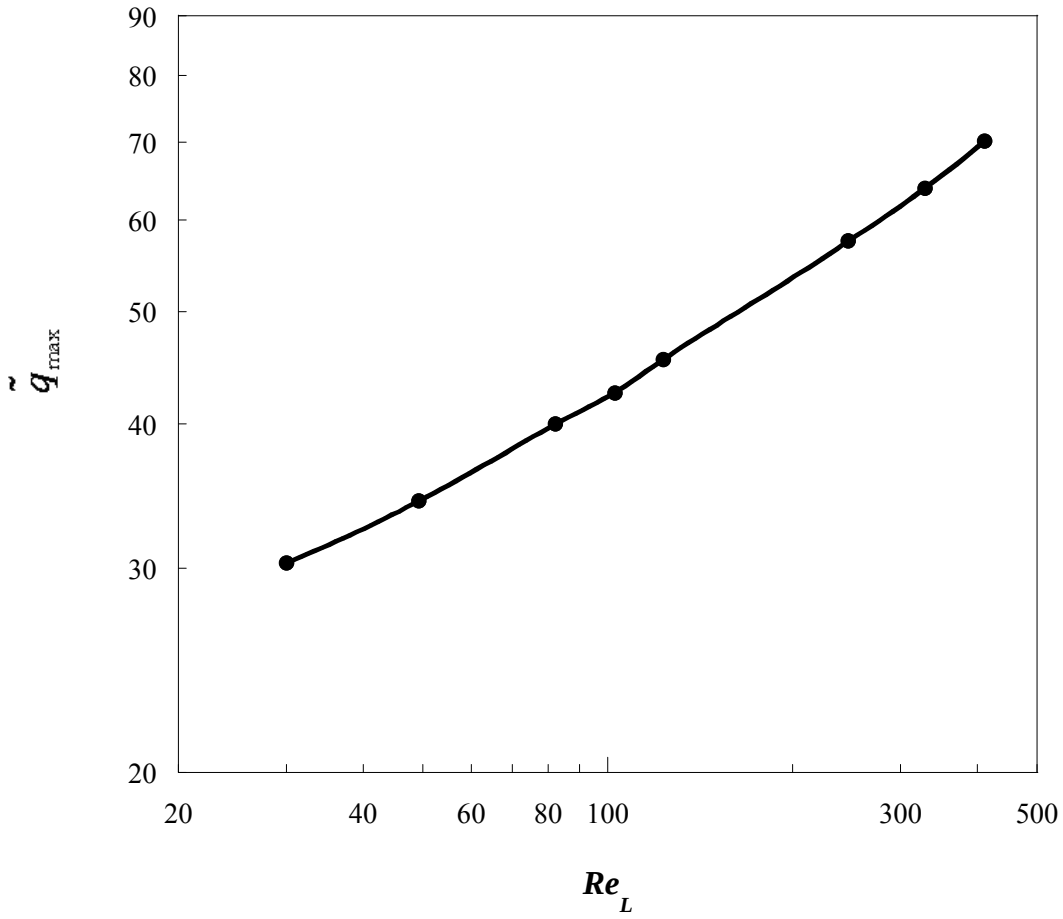


Figure 5-36: The maximised rate of heat transfer as a function of Reynolds number with the conductivity ratio (k_r) equal to 100

Figure 5-37 shows that the optimal fin-height ratio is generally independent of Reynolds number (and thus free stream velocity) and thermal conductivity. This is evident in the insignificant change of the optimal fin-height ratio over the Re_L range and change in the thermal conductivity ratio. It can therefore be deduced that $(H_2/H_1)_{opt}$ = 0.925. This implies that for maximum heat transfer, the pin-fins in the first row should be slightly higher than the fins in the next row.

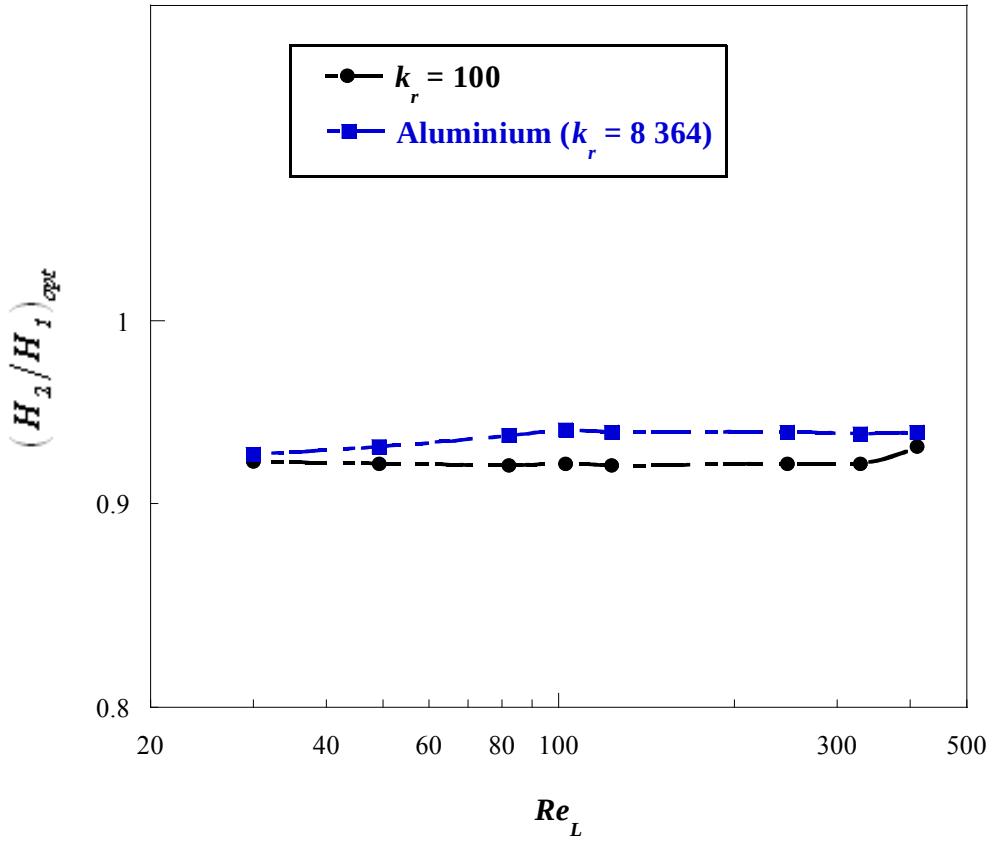


Figure 5-37: The influence of Reynolds number on the optimal height ratio

The optimal spacing s_{opt} between the pin-fins remains unchanged regardless of the Reynolds number (based on the length of the control volume) as shown in Figure 5-38. This constant value coincides with the allowable spacing due to manufacturing constraints, which is 50 μm. This implies that the closer the fins are to one another, the more effective the heat transfer rate will be. The more advanced microfabrication techniques become, the more closer this spacing can become, which will result in improved cooling abilities of heat sinks.

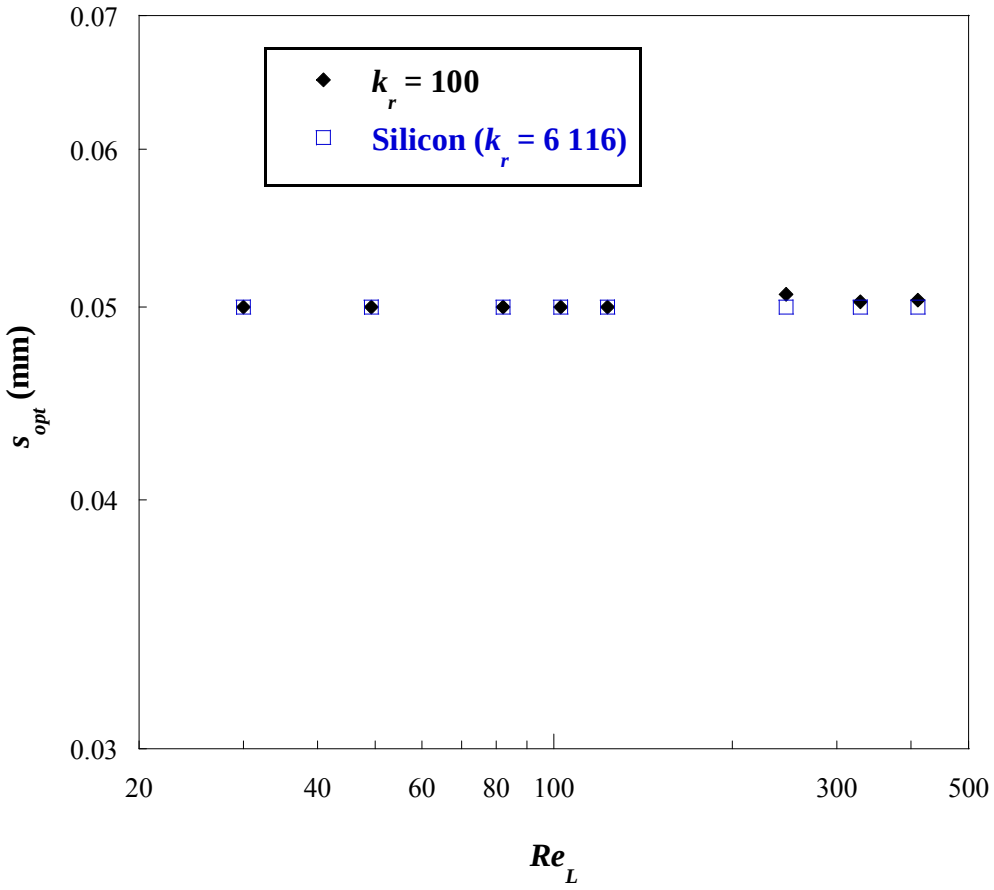


Figure 5-38: The effect of flow velocity on the optimal interfin spacing

Figure 5-39 shows an increase in the optimal fin-diameter ratio $(D_2/D_1)_{opt}$ with Reynolds number. With an error of less than 1%, the CFD results can be correlated as:

$$\left(D_2/D_1\right)_{\text{opt}} = 0.464 Re^{0.0314} \tag{5-22}$$

The results further imply that the non-uniformity of the diameters of fins in the various rows plays a vital role in the heat transfer rate of pin-fin heat sinks. Furthermore, the results show that at lower Reynolds numbers, the diameter of the pin-fins in the first row should be about twice the diameter of those in the second row in order to achieve the maximum heat transfer rate, while at higher Reynolds numbers it should be about 1.8 times the diameter of those in the second row.

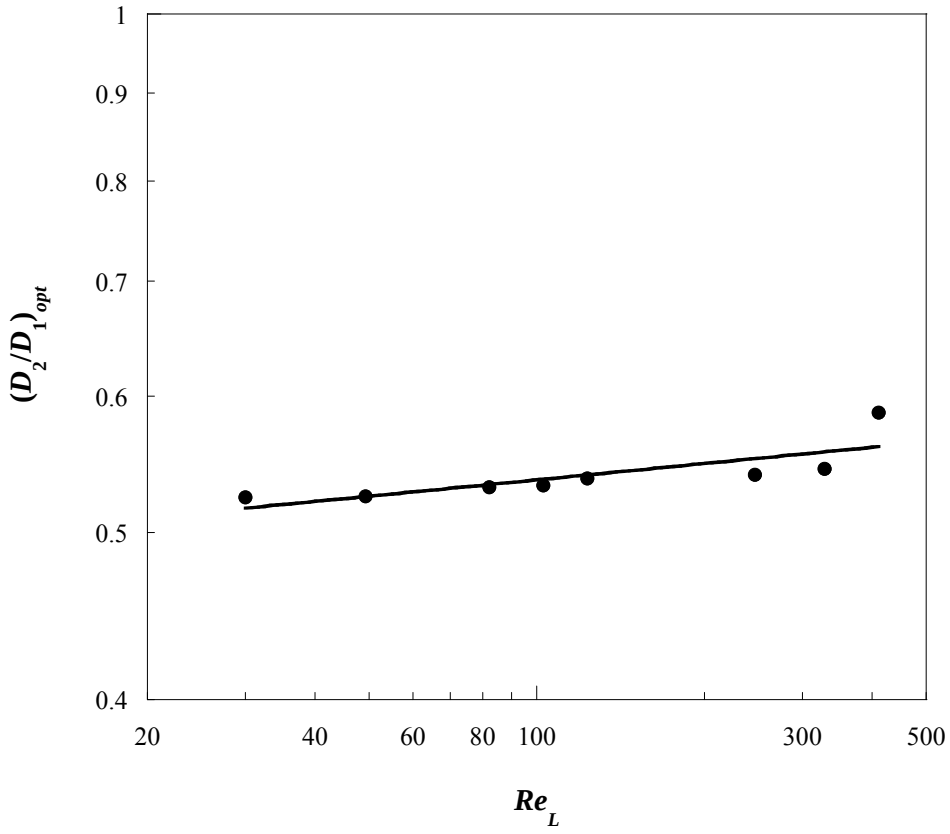


Figure 5-39: The relationship between Reynolds number and the optimal diameter ratio for a thermal conductivity ratio of 100

The effect of the thermal conductivity ratio k_r on the maximised rate of heat transfer and the geometrical ratio of the pin-fins was investigated. This effect was investigated at a Reynolds number of 123. Figure 5-40 shows that the maximised rate of heat transfer increases as the thermal conductivity ratio increases. However, at higher thermal conductivities (k_r > 1 000), the rate of heat transfer approximately reaches a plateau even though the thermal conductivity ratio increases. This is due to the fact that convection rather than conduction is the more dominant medium thus rendering the thermal conductivity property of little importance.

Figure 5-41 gives a plot of the maximised dimensionless rate of heat transfer as a function of Reynolds number for various materials. It shows that at high thermal conductivity, its effect on the heat transfer rate is minimal.

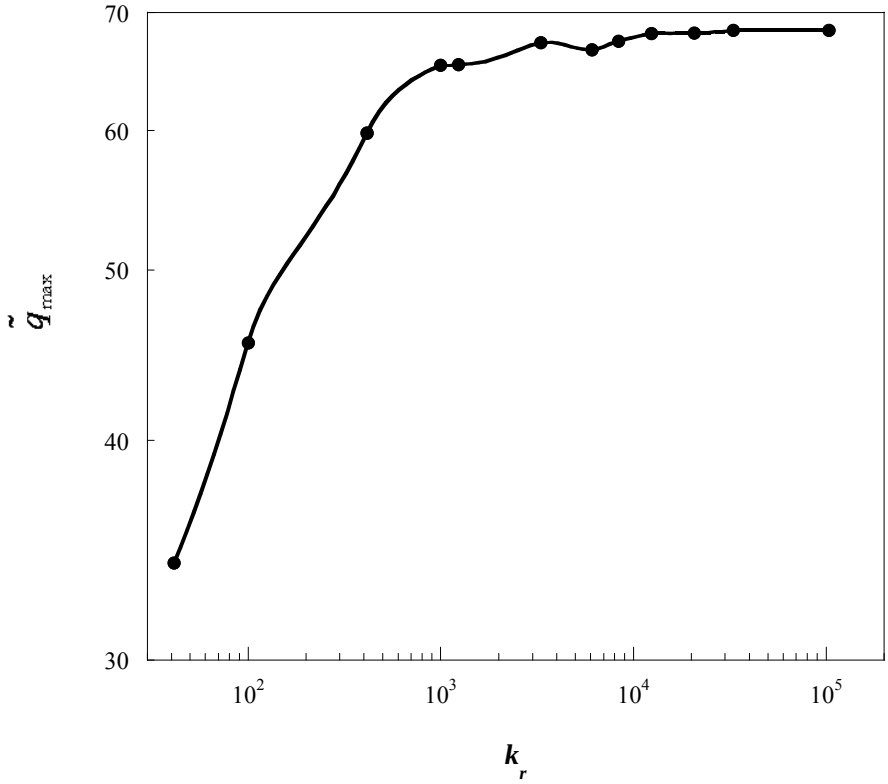


Figure 5-40: The effect of the thermal conductivity ratio on the maximised rate of heat transfer at a Reynolds number of 123

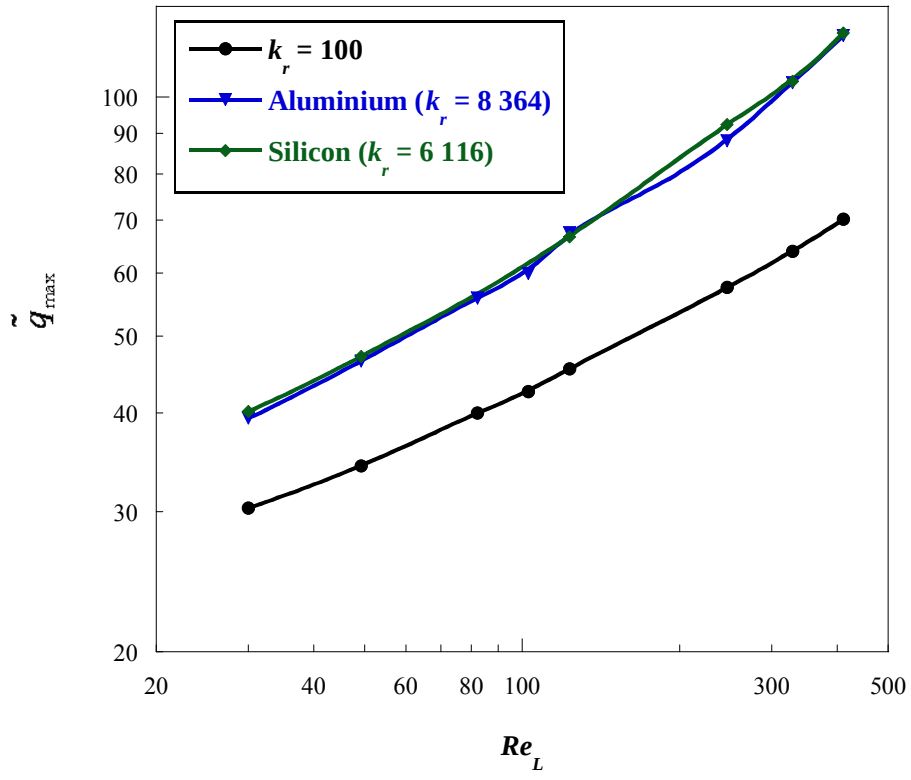


Figure 5-41: Heat transfer rate comparisons for various heat sink materials

Figure 5-42 also shows that the thermal conductivity ratio has no effect on the pin-fin-diameter ratio and height. Results shown in Figure 5-37 further highlight the fact that the pin-fin-height ratio is not significantly influenced by the thermal conductivity parameter.

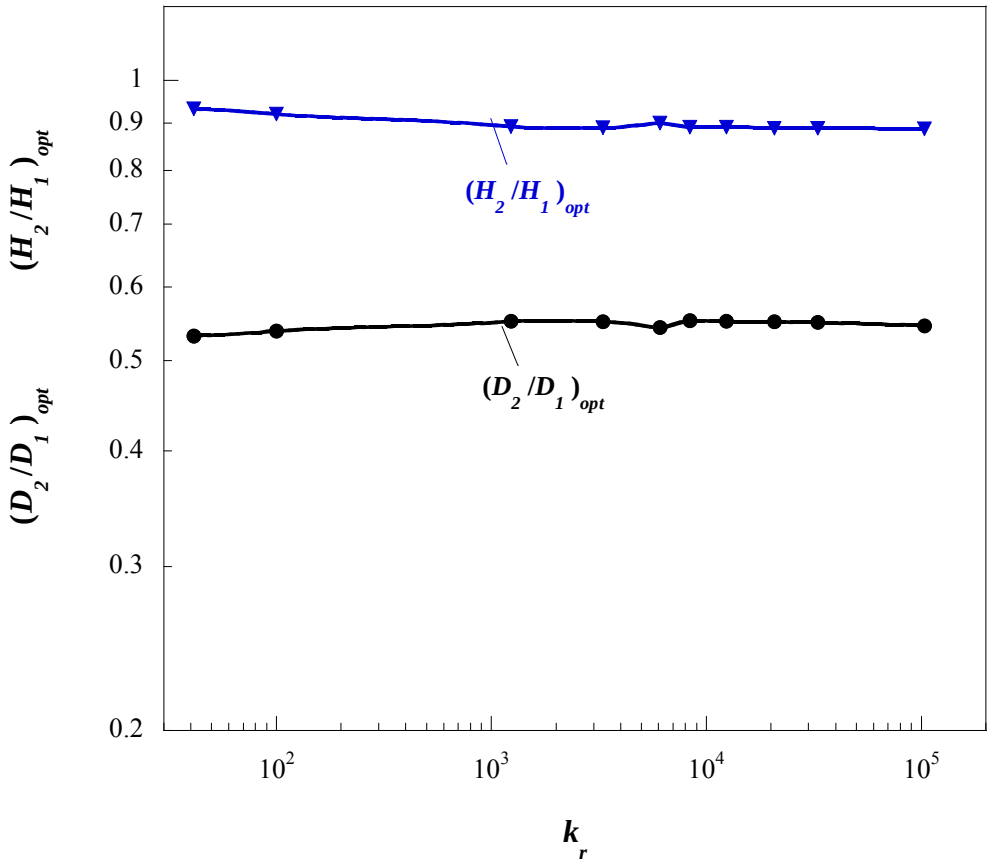
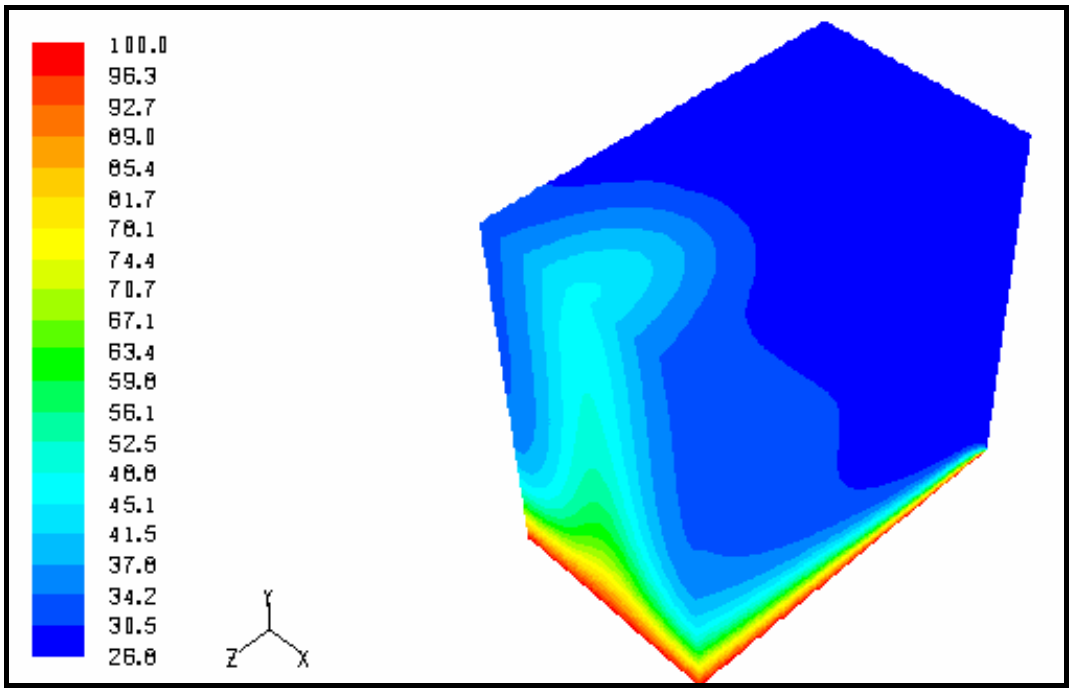
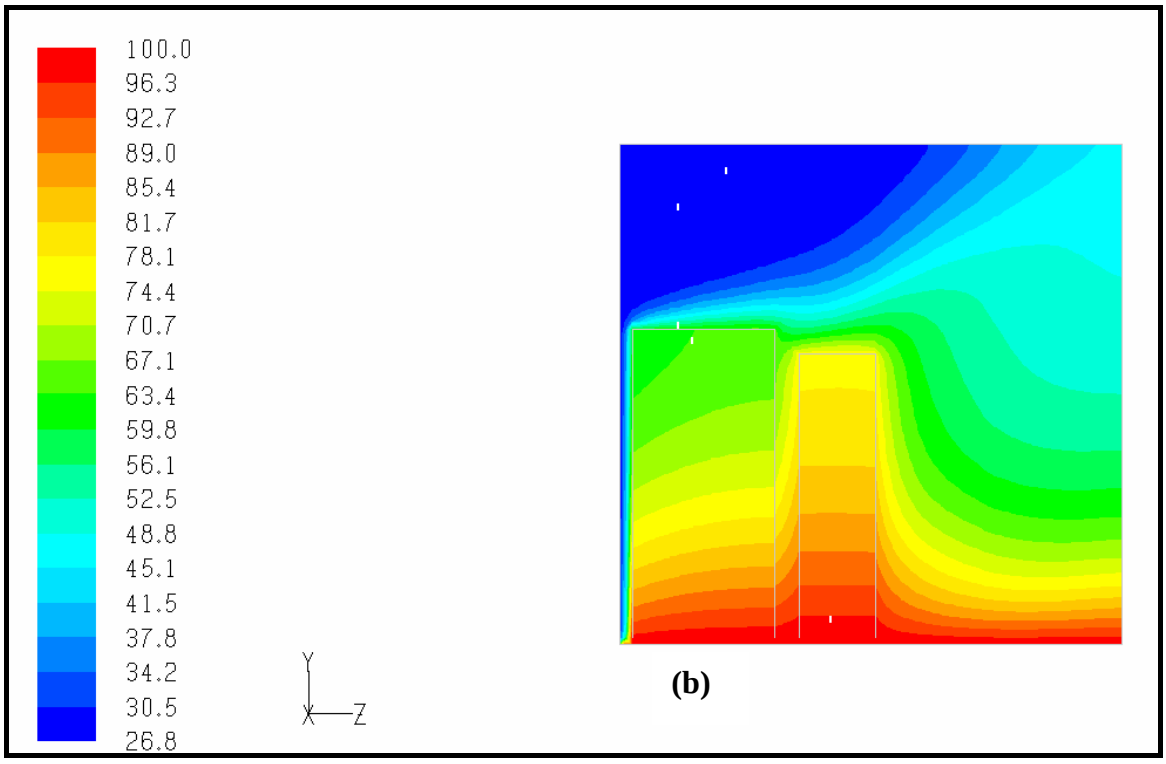


Figure 5-42: The effect of the thermal conductivity ratio on the optimised geometric configuration of a double row finned heat sink at a Reynolds number of 123

Figure 5-43 shows the optimal temperature contour distribution of the micropin-fin heat sink at a Reynolds number of 123 with a thermal conductivity ratio k_r of 100. Figure 5-43(a) gives a temperature distribution of the entire control volume in an isometric view while Figure 5-43(b) shows the temperature distribution in the centre plane of the control volume. The red colour at the base of the heat sink emphasises the constant wall temperature applied at the position y = 0. Major colour changes are visible in the region where the pin-fins are present implying that the major heat transfer occurs between the pin-fins and the fluid, which causes the cooling enhancement of such heat sinks.



(a) Temperature distribution across the entire control volume



(b) Temperature contour plot across the centre plane of the heat sink

Figure 5-43: Temperature distribution (in °C) of the optimally designed double row micropin-fin heat sink

5.4 CASE STUDY 3: TRIPLE ROW MICROPIN-FIN CONFIGURATION

This optimisation case builds upon the previous case of double micropin-fin configuration. In this case, a third row is added while applying the deduced results from Case Study 2. The optimisation process enables the development of a correlation between the maximised rate of heat transfer as a function of Reynolds number.

5.4.1 The CFD model

Figure 5-44 shows the physical model and Figure 5-45 the unit cell computational model of the triple row micropin-fin heat sink configuration. The vertically arranged pin-fins form part of a three-row-finned array with row-specific diameters D_1, D_2, D_3 respectively. The various rows are spaced by a distance s_1 and s_2 as depicted in Figure 5-45. The distance between the leading edge and the first fin is $s/2$. Results from the preceding case suggest that a uniform row height assumption can be made as the optimal height ratio was found to be close to unity. Therefore, it was assumed that $H_1 = H_2 = H_3$. The heat sink with fixed length L and width G is supplied with heat from the bottom of the enclosure. An overall dimensions of 1mm × 0.6mm × 1mm is used for this analysis

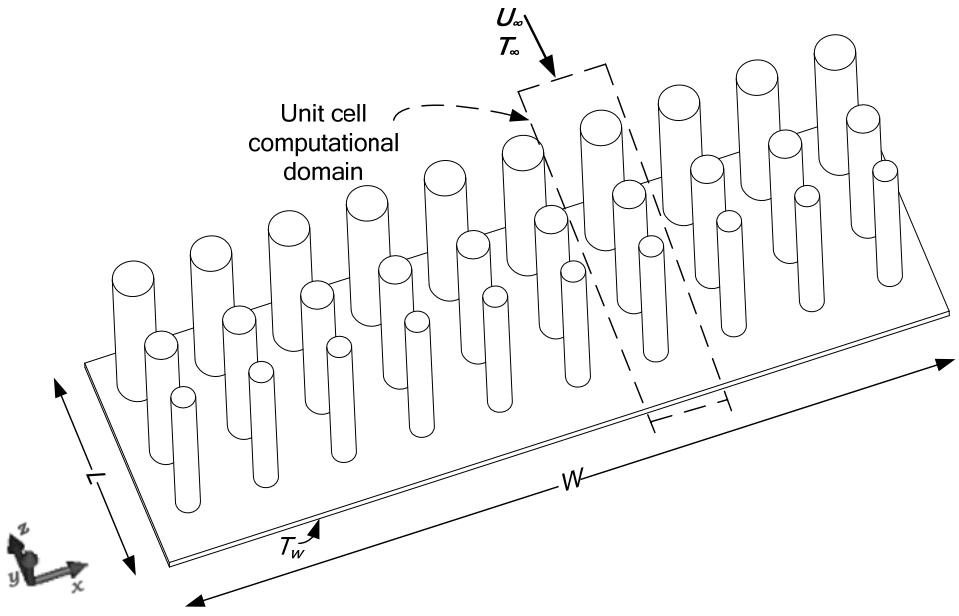


Figure 5-44: Physical model of a triple variable row micropin-fin heat sink

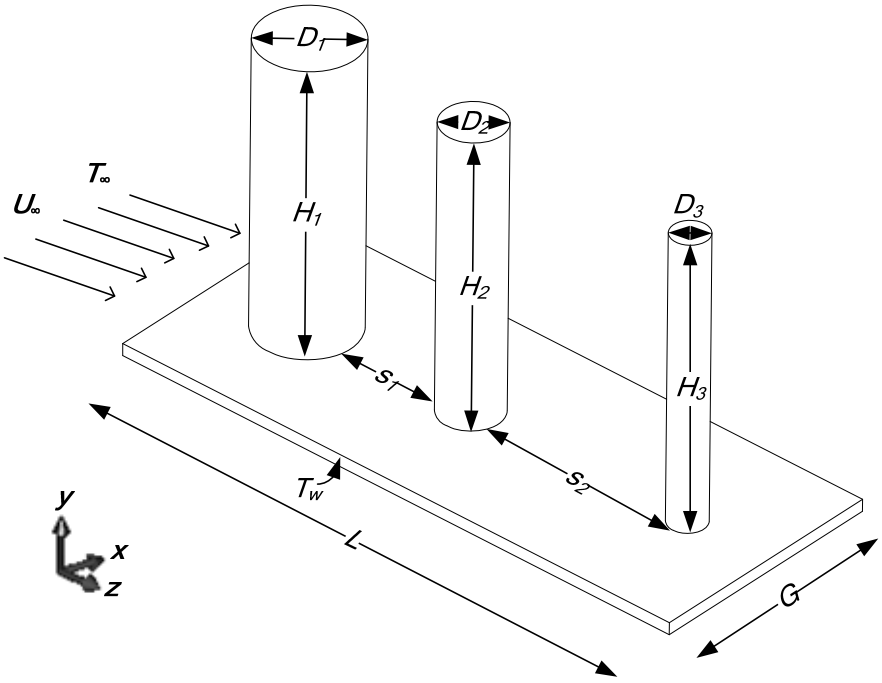


Figure 5-45: Unit cell computational domain of a triple micropin-fin heat sink

The conservation equations discussed in Chapter 3 are solved over the fully discretised domain, which is shown in Figure 5-46. A one-dimensional uniform velocity with constant temperature is assumed at the inlet:

$$\begin{aligned} u(x,y,0) &= v(x,y,0) = 0 \\ w(x,y,0) &= U_\infty \\ T(x,y,0) &= T_{inlet} \end{aligned} \tag{5-23}$$

At the outlet it is assumed that the velocity gradients are zero:

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial x} = \frac{\partial w}{\partial x} = 0 \tag{5-24}$$

No-slip, no-penetration boundary conditions are enforced on the fin and wall surfaces. Symmetry boundary conditions are applied to the vertical ends of the domain to reasonably represent the physical and geometric characteristics of flow through pin-fin arrays. The schematic diagram in Figure 5-34 also explains the boundary condition applied in this case study. A constant wall temperature completes the thermal boundary condition. Uniform isothermal free stream (air) is used as the working fluid.

Other flow-related assumptions implemented include steady flow, laminar flow, incompressibility, and constant fluid and material properties.

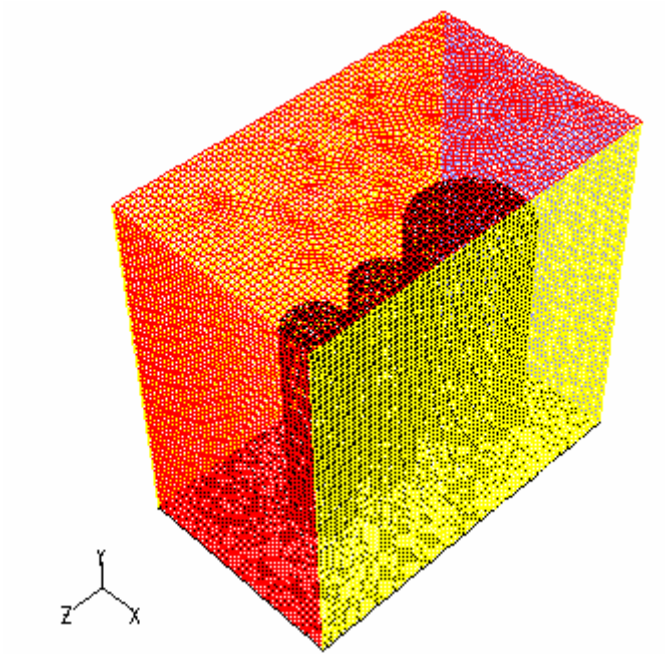


Figure 5-46: Meshed computational grid of the triple micropin-fin heat sink

5.4.2 Verification of the model

Grid independence checks were utilised to verify the numerical code. Three mesh sizes of 147 546 cells, 182 358 cells and 605 300 cells respectively were used for the verification procedure. As shown by the results presented in Figure 5-47, it was found that the maximum difference in the dimensionless rate of heat transfer $\tilde{q}$ between the three mesh sizes is <1%. This gives confidence that a mesh with 182 358 cells will give satisfactory accuracy in the prediction of the heat transfer across the fin array.

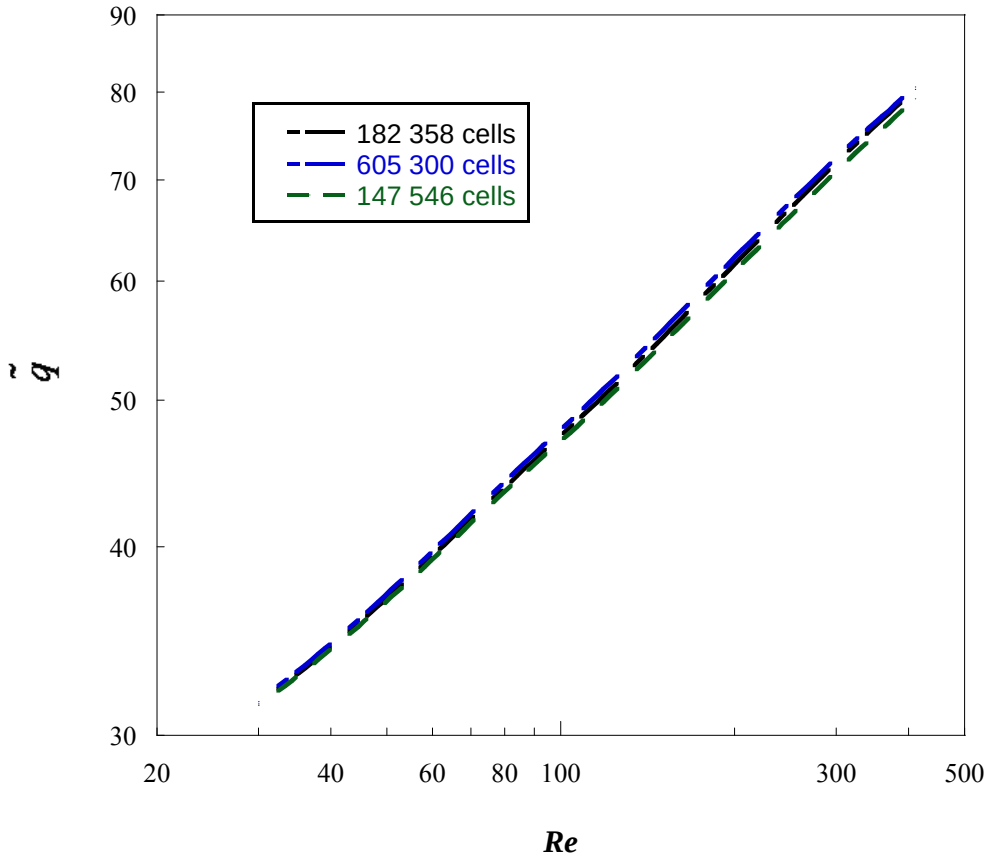


Figure 5-47: Plot of the dimensionless heat transfer rate for different mesh sizes

5.4.3 Mathematical formulation of the optimisation problem

The objective is quite similar to that of the previous case, which is the maximisation of the total heat transfer rate across the fin array. With the height of the fins unified, the design variables are the geometric parameters D_1, D_2, D_3, s_1 and s_2.

A total fin volume constraint was implemented, which ensures that the material cost for the fins is fixed. Mathematically, this constraint is given by:

$$\begin{aligned} &\therefore V_1 + V_2 + V_3 = \text{Constant} \\ &\frac{\pi D_1^{\,2}}{4} H + \frac{\pi D_2^{\,2}}{4} H + \frac{\pi D_3^{\,2}}{4} H = C \\ &D_1^{\,2} + D_2^{\,2} + D_3^{\,2} = \frac{4C}{\pi H} \end{aligned} \qquad (5\text{-}25)$$

Side constraints were enforced on the diameters and spacing to ensure $1/2 \leq H/D \leq 4$ and the minimum allowable spacing of 50 microns was adhered to.

5.4.4 Selection of the adequate differencing step size

In order to ensure accurate representation of the function gradient, an adequate step size is required such that noise within the simulation is eliminated. In Figures 5-48 to 5-50 step sizes were chosen with analysis conducted to check which one gives a smooth representation of the heat transfer rate. As shown in Figure 5-50, a step size of 10^{-2} gives a smooth curve with respect to the heat transfer rate, which will provide a good gradient representation of the function.

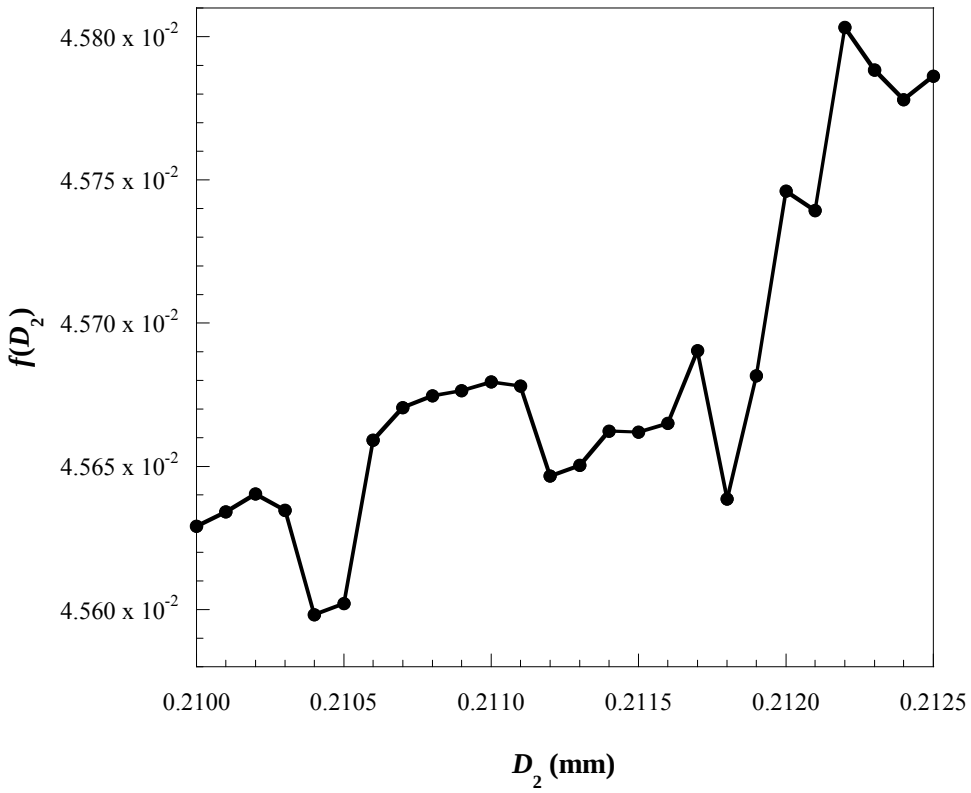


Figure 5-48: Plot of the heat transfer rate for small increments of 10^{-4}

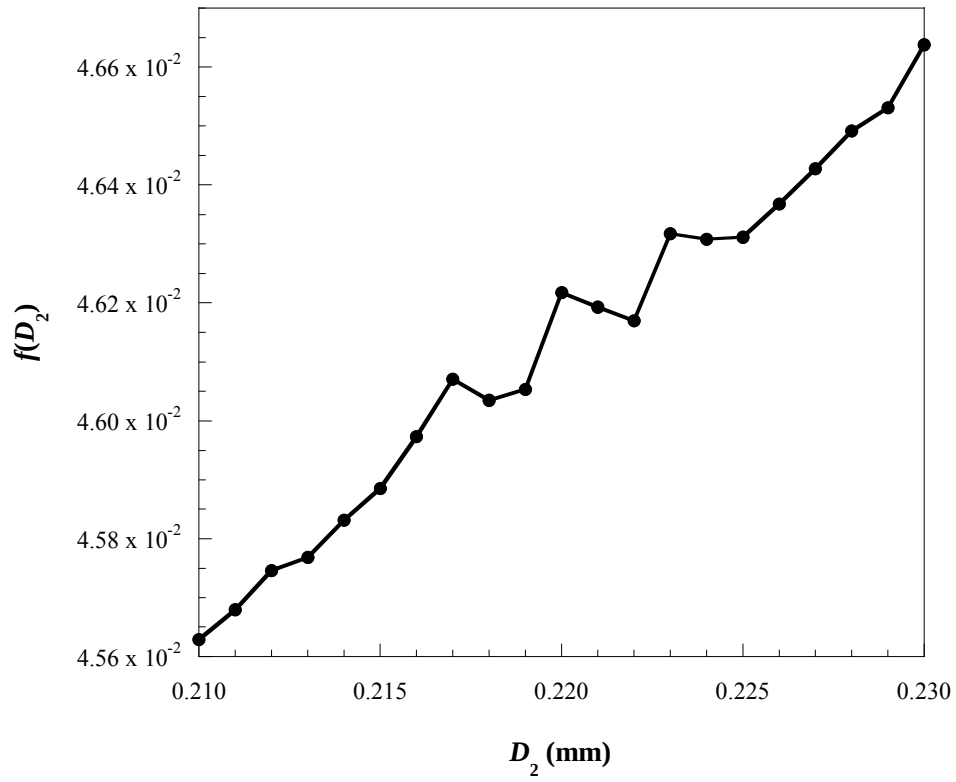


Figure 5-49: Plot of the heat transfer rate for small increments of 10^{-3}

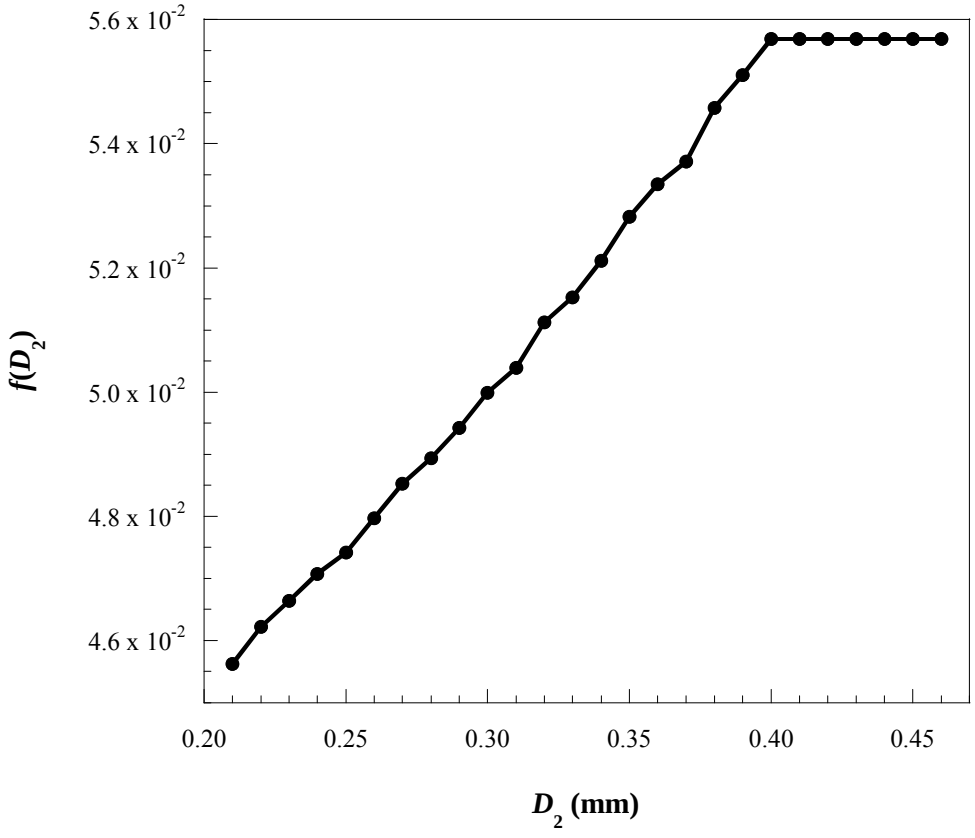


Figure 5-50: Plot of the heat transfer rate for small increments of 10^{-2}

5.4.5 Results

The result of the optimisation process was the maximisation of the total rate of heat transfer along the finned array. This was achieved by an optimal search of the geometric parameters of the heat sink by means of a mathematical algorithm.

Figure 5-51 depicts an almost linear increase (on a log-log scale) in the dimensionless heat transfer rate as a function of Reynolds number. This expected result hails from the fact that the convective heat transfer coefficient is a strong function of the fluid velocity. For a thermal conductivity ratio of 100, the relationship (within a 1% error) between Reynolds number and the maximal rate of heat transfer can be correlated as:

$$\tilde{q}_{\max} = 8.45\,\mathrm{Re}^{0.375} \tag{5-26}$$

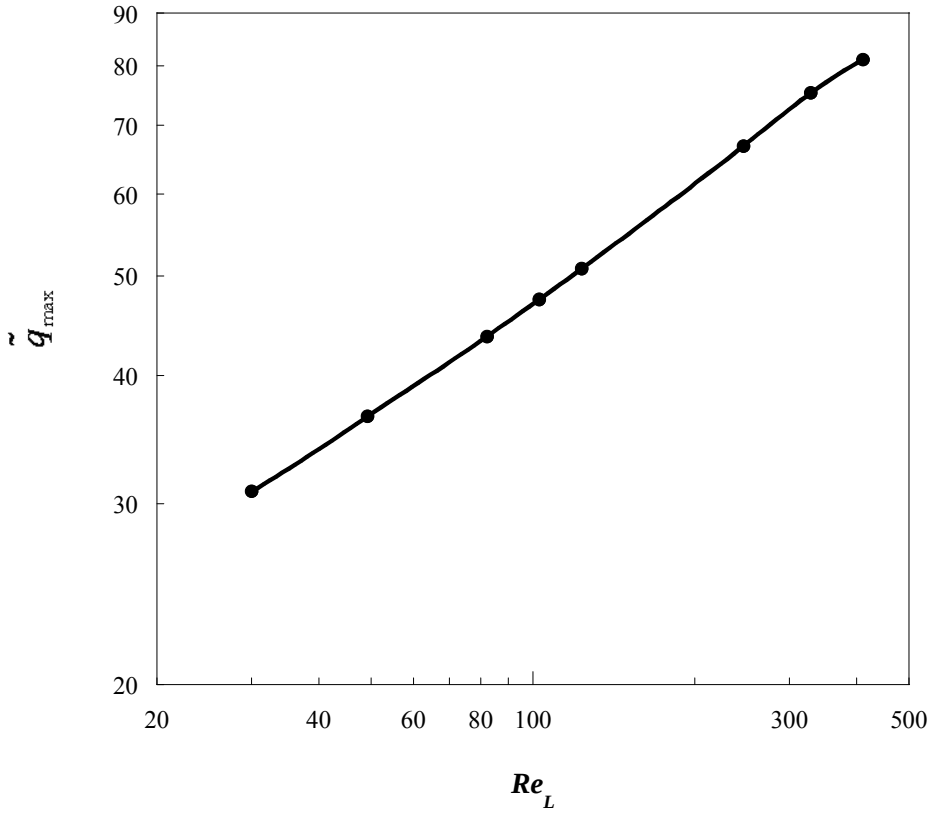


Figure 5-51: The relationship between the optimal dimensionless rate of heat transfer and Reynolds number for a triple row heat sink for a thermal conductivity ratio of 100

The optimised geometric parameters (Figure 5-52) predict that pin-fins in the first row D_1 should be larger than the pin-fins in the next row with this decreasing diameter trend continuing to the third row. It further shows that the optimal diameters D_1, D_2 and D_3 change slightly as the Reynolds number across the finned array increases. The results indicate that as the Reynolds number increases, the pin diameter of the fins in the first row decreases while the diameter of the fins in the third row increases. The pin diameters in the penultimate row show independence with regard to an increasing *Re*.

Table 5-8 shows the optimal diameter ratios of the fins separated by spacing s_1 and s_2. An increasing trend in the diameter ratio is evident as the velocity of the fluid is gradually increased.

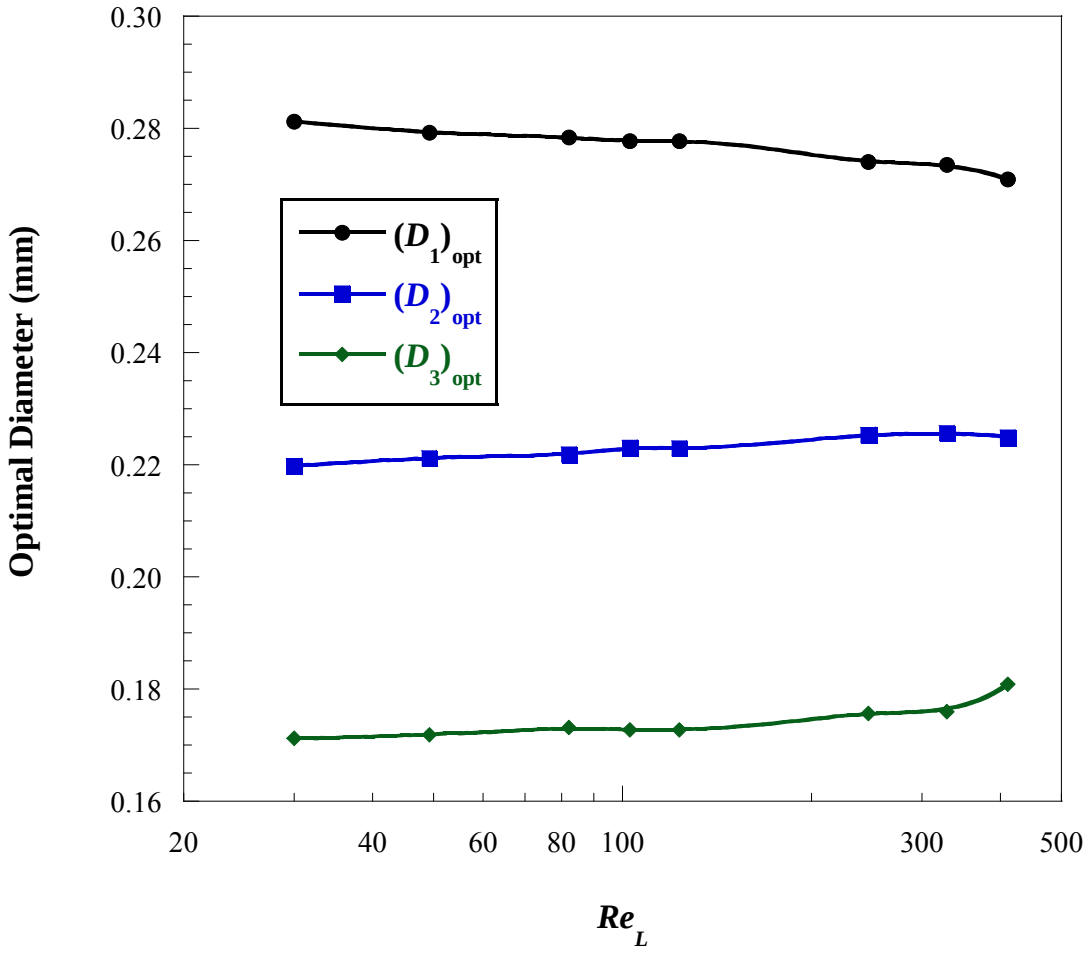


Figure 5-52: The relationship between the optimal diameters for each fin row as a function of Reynolds number

Table 5-8: Optimal diameter ratios for various Reynolds numbers

Re	$\left(D_2/D_1\right)_{opt}$	$\left(D_3/D_2\right)_{opt}$	$\left(D_3/D_1\right)_{opt}$
30	0.782	0.779	0.609
49	0.792	0.777	0.616
82	0.797	0.780	0.622
103	0.803	0.774	0.622
123	0.803	0.775	0.622
246	0.822	0.780	0.641
329	0.825	0.780	0.644
411	0.831	0.804	0.668

The effect of various materials on the maximised rate of heat transfer of the heat sink was also investigated. Figure 5-53 shows that an increase in the thermal conductivity ratio k_r, causes an increase in the maximal heat transfer rate. However, varying gradients of the dimensionless heat transfer rate as a function of *Re* are experienced with higher positive gradients experienced at lower k_r (less than 500) and almost zero gradients for conductivity ratios greater than 6 000. The results suggest that a heat sink designed to operate within a medium where the conductivity ratio is about 400 will perform very well and increasing the conductivity ratio will not significantly increase the dimensionless heat transfer rate.

Figure 5-54 shows that the thermal conductivity ratio does not affect the optimal geometric configuration of a triple row micropin-fin heat sink. For a Reynolds number of 123, the pin diameters for each row stay constant with an increase in the conductivity ratio k_r. This result implies that the solid-fluid medium combination is insignificant with regard to the geometric design of such heat sinks. In addition, it is intuitive that the minimum allowable spacing due to manufacturing constraints of 50 μm is the optimal spacing separating the pin-fins in the various rows. This optimal spacing also shows solid-fluid medium independence; in other words, the optimal spacing remains constant regardless of an increase in the thermal conductivity ratio.

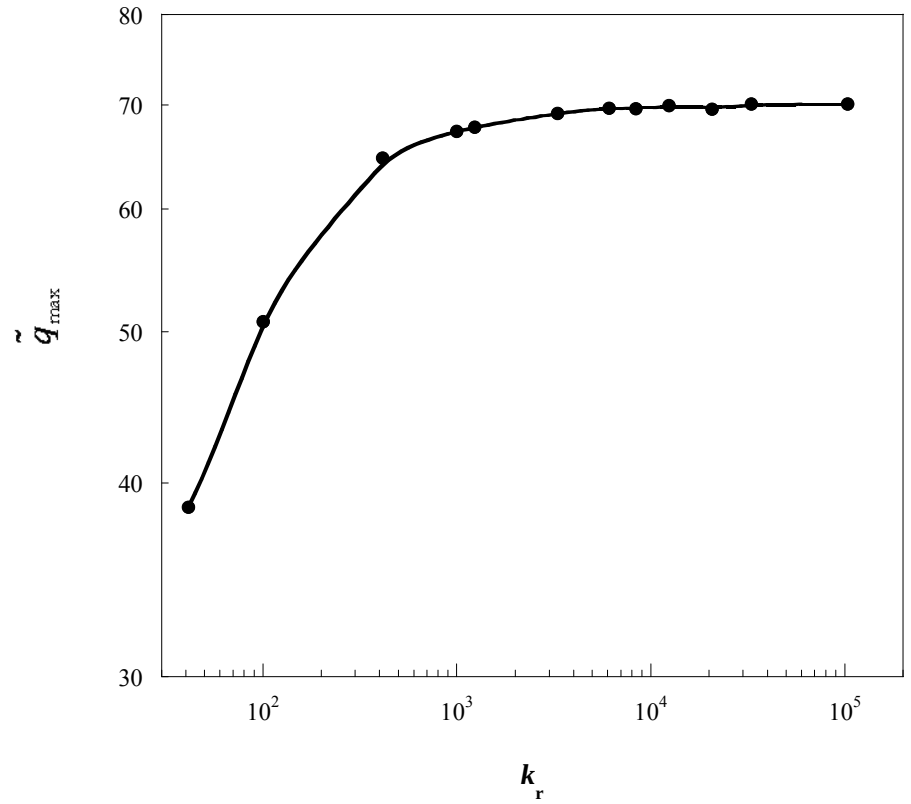


Figure 5-53: The effect of the conductivity ratio on the maximised heat transfer rate for a triple row micro heat sink for a Reynolds number of 123

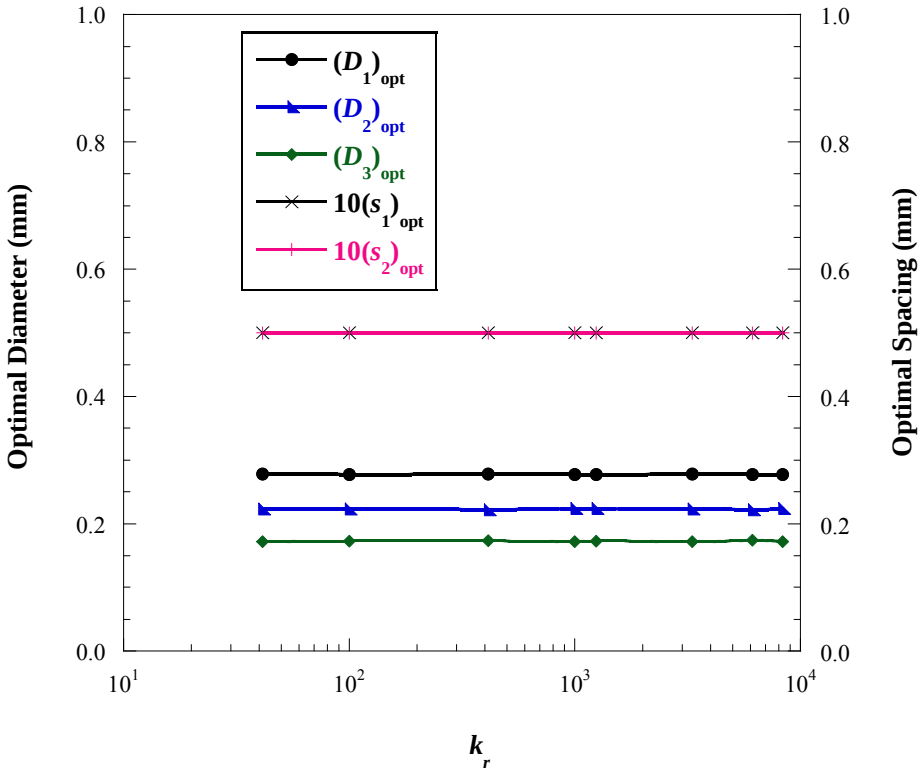
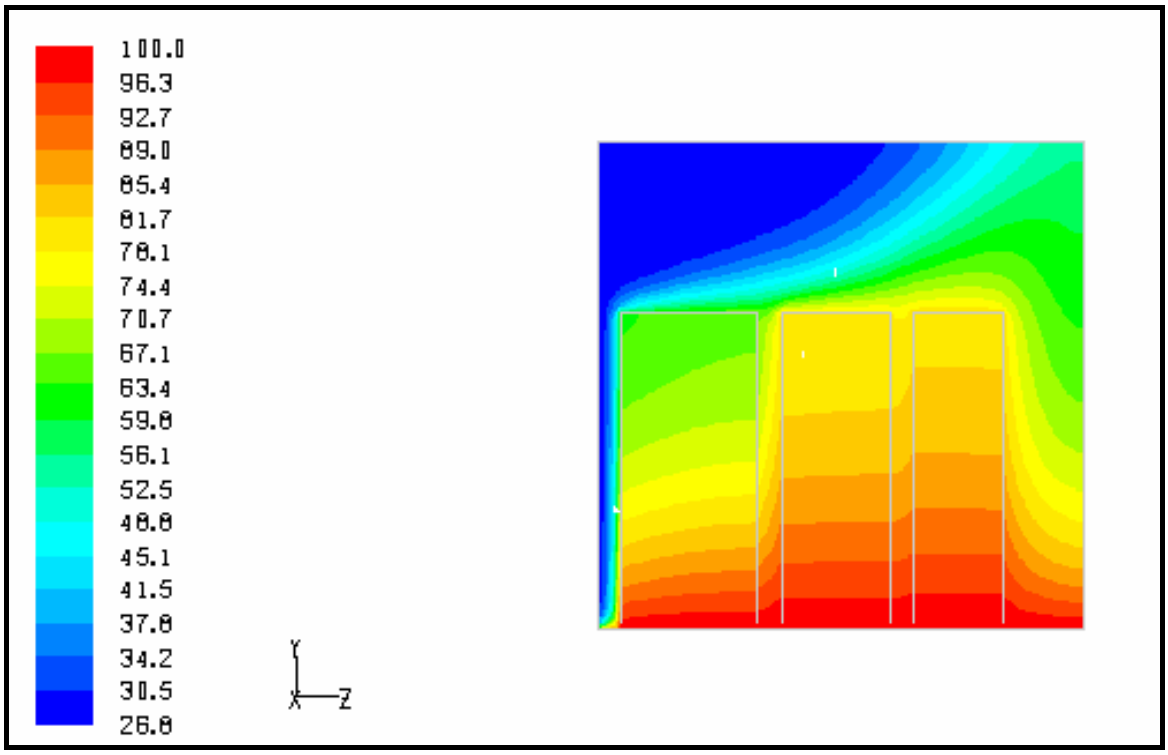


Figure 5-54: The influence of a change in the thermal conductivity ratio on the optimal geometric parameters of the heat sink for a Reynolds number of 123

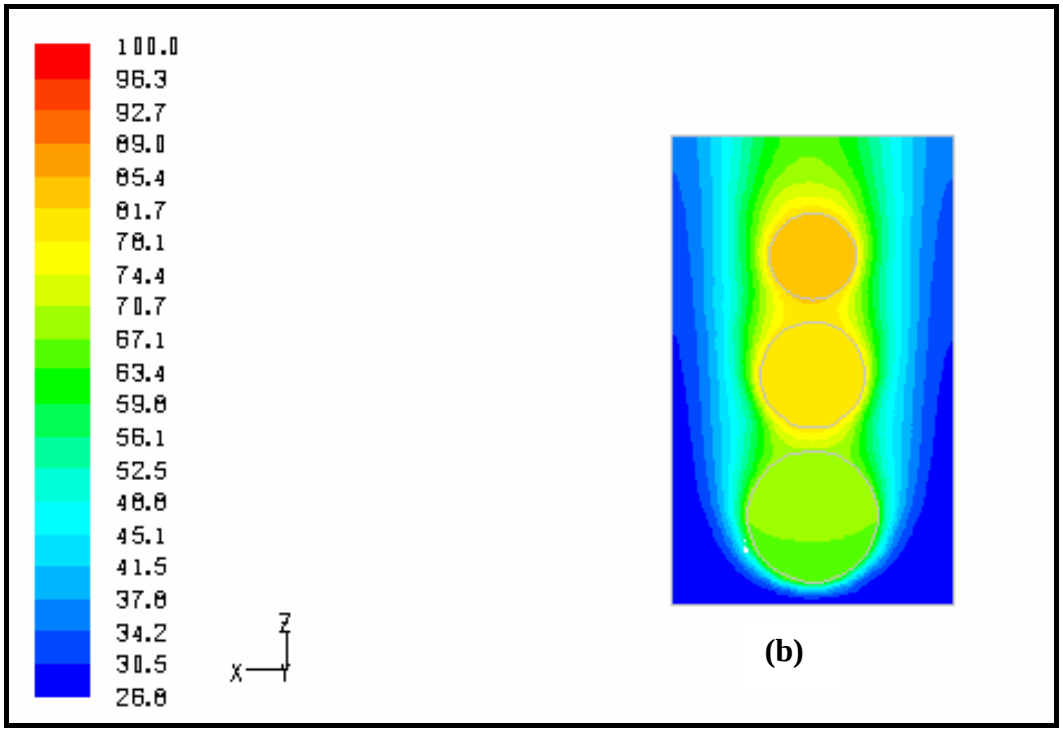
Figure 5-55 shows the temperature field corresponding to a Reynolds number of 123 with a thermal conductivity ratio of 100. Figure 5-55(a) shows the distribution of the centre plane across the micropin-fins. Visible at the base is the enforced constant wall temperature boundary condition of 100°C. The effect of the heat transfer along the fins is evident with the major colour change experienced in the pin-fin region. The temperature profile shows that the third row of fins experiences the hottest temperatures due to the fluid being the warmest at that region of the heat sink.

Figure 5-55(b) gives a plan view temperature contour of the heat sink. It is evident from the colour maps that temperatures upstream of both the fluid and solid are lower than the temperatures downstream. At each region, the fins have the highest temperatures as they act as a heat sink drawing heat from the base wall and dissipating it to the fluid, which is warmed as it flows downstream.

Figure 5-56 gives a velocity vector representation of the flow field within the micropin-fin heat sink. As can be expected, it shows higher velocities at the side walls than those close to the pin-fins. Figure 5-57 shows the pressure distribution across the pin-fins with high pressure, low velocity experienced at the entry region, which is typical of flow over blunt bodies. Flow separation caused by increasing fluid velocity accounts for the colour changes at the sides of the pin fins.



(a) Temperature distribution across the centre plane of the micropin-fins



(b) Temperature contour plot of the plan view of the finned heat sink

Figure 5-55: Temperature profile of the triple row micropin-fin heat sink

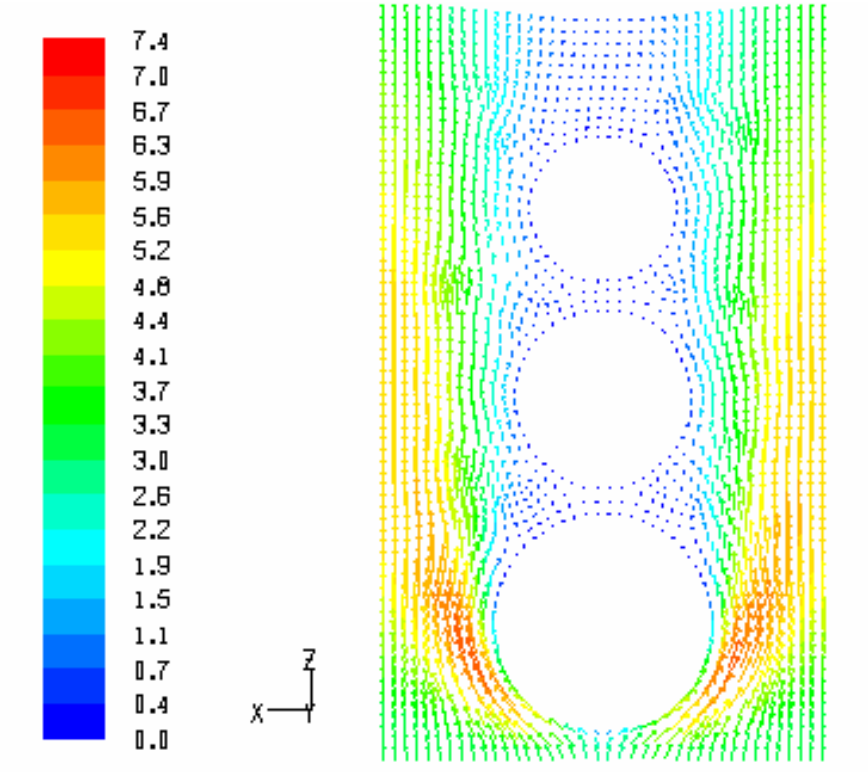


Figure 5-56: Velocity vector representing the flow field within the micropin-fin heat sink

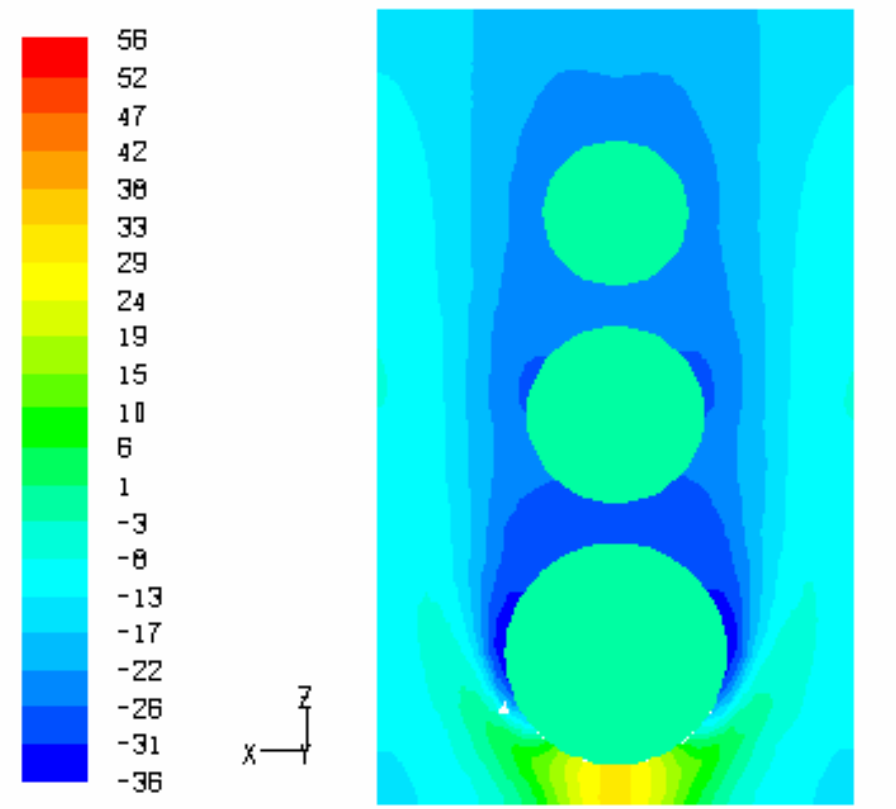


Figure 5-57: Pressure contour along the length of the micropin-fin heat sink

5.5 SUMMARISED TRENDS OF THE THREE CASE STUDIES

In the preceding sections, the maximal thermal performance of a microchannel, double row pin-fin and triple row pin-fin heat sinks was determined by optimising their geometrical parameters. In this section, a summarised view is given of all three optimisation cases.

Figure 5-59 shows the maximised thermal performance of the microchannel, double row pin-fin and triple row pin-fin heat sinks as a function of Reynolds number under various thermal boundary conditions. A constant heat flux boundary condition was used for the microchannel case study while a constant wall temperature boundary was used for the double and triple row pin-fin heat sink case study. Due to the varying boundary conditions, actual comparisons of the three case studies will be inconclusive. As shown in Figure 5-59, all three cases show a similar increasing trend in their thermal performances with an increase in Reynolds number. The optimised microchannel heat sink shows a linear increase in the maximised global thermal conductance $C_{\max}$ as with increasing Reynolds number based on the optimised hydraulic diameter. The Reynolds number (based on the hydraulic diameter) is defined mathematically as

$$Re = \frac{u_{\text{ave}} \rho D_{\text{h,opt}}}{\mu} \tag{5-27}$$

For a similar Reynolds number range, an increase in the dimensionless rate of heat transfer is observed with an increase in Reynolds number (based on the axial length). Heat transfer enhancement is also evident from Figure 5-58 as a result of the added row of pin-fins in the triple row pin-fin heat sink. This enhancement is due to the increased heat transfer surface created by the third row of fins.

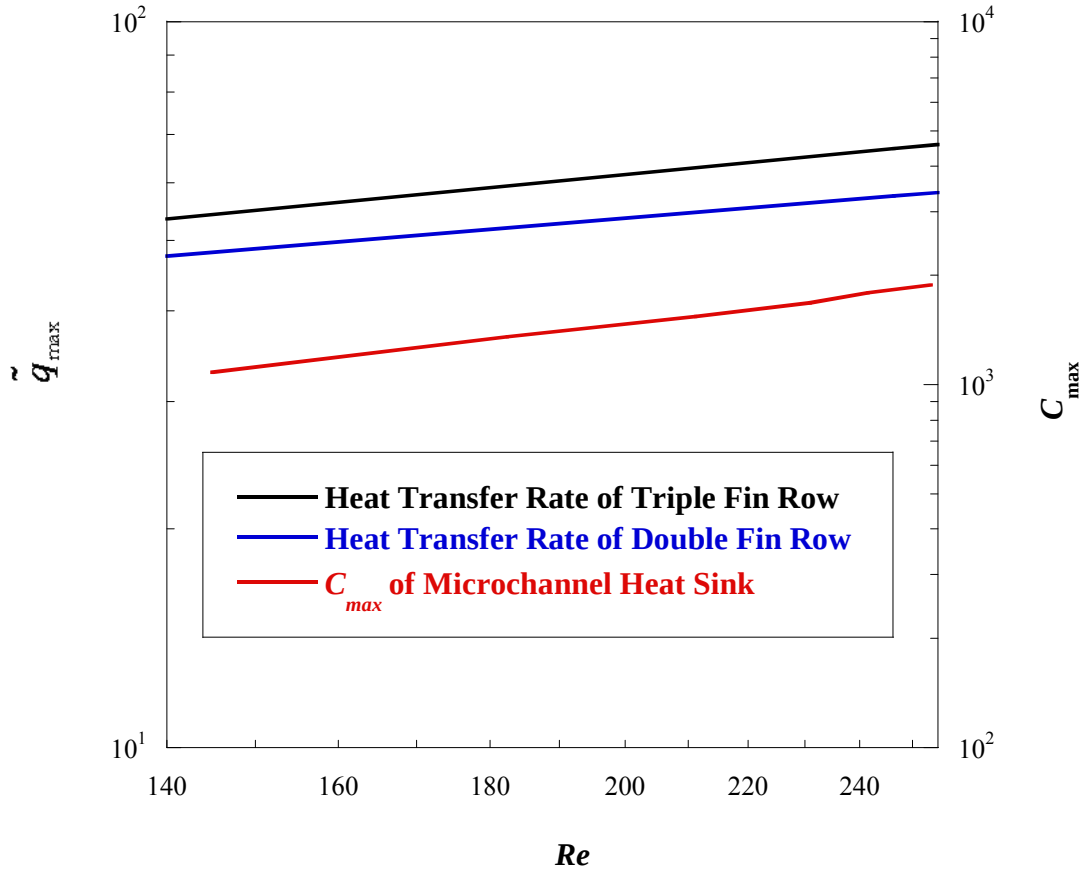


Figure 5-58: A summarised look at the thermal performance of the microchannel and micropin-fin heat sinks

5.6 CONCLUSION

In this chapter, a numerical optimisation methodology was applied to three geometric optimisation case designs. In the first case, a microchannel embedded in a highly conductive solid material was optimised. The objective of this case was to minimise the peak wall temperature of the heat sink in order to achieve lowered thermal resistances. Under a fixed volume and other material constraints, relationships between various optimal geometric parameters and the dimensionless pressure drop were developed. It was also revealed that increasing the number of design variables will result in a better optimum as up to a 20% increase in the global thermal conductance was obtained when the axial length was relaxed in the optimisation process.

In Case 2, the optimisation problem resulted in the maximisation of the total rate of heat transfer for a double row micropin-fin heat sink. Also, the effect of various material combinations on the optimal parameters was reported. It was also found that the influence of a non-uniform height to the thermal performance of the heat sink is quite negligible. It was also seen that at higher thermal conductivity ratios, the rate of heat transfer has a deceasing trend.

The third case built on the foundation created in the second case extending the problem to a third row of micropin-fins. The optimisation returned optimal geometric parameters which are independent of solid-fluid combination. It was also proved that the cooling abilities of micropin-fins will be greatly enhanced by better manufacturing techniques as the allowable optimal spacing was the manufacturing limitation. It was found that by adding a third row of pin-fins, the rate of heat transfer is enhanced with enhancement greater than 10% achievable for $Re > 100$. However, the enhancement rate decreases at higher thermal conductivity ratio kr.

The temperature distribution of the different cases was analysed. Sensitivity analysis was carried out to ensure CFD noise did not affect the optimal solutions. This highlights the importance of correct formulation and design set-up for effective and accurate optimisation. The various case designs emphasised the fact that for micro heat sink design, material cost and pressure drop considerations are vital elements in achieving efficient optimal designs.

6

CHAPTER 6: SUMMARY, CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

6.1 SUMMARY

The continued increases in the functionality and compactness of microelectronics coupled with the stringent operational temperature requirement have led to the thermal management of these devices being a challenge. Minimising the peak wall temperatures and the temperature gradients within these devices has been the main aim of thermal management systems. Various techniques such as heat pipes and impinging jets have been applied to achieve effective heat removal. However, these techniques in recent times have proved incapable of handling these extreme temperatures until the era of micro heat sink sprung into life. A continuous drive to further understand the flow dynamics, mass and heat transfer of micro heat sinks, has led to the publication of many research papers in the last decade. The research papers of which some have been discussed in Chapter 2 range from fluid flow within heat sinks, numerical and experimental heat transfer and pressure drop measurements to optimisation of these heat sinks. Available literature also shows that the geometrical configuration of micro heat sinks ultimately plays a vital role in their thermal performance. This has drawn much attention and has resulted in various methods of optimisation such as genetic algorithms and entropy minimisation schemes being employed to help develop optimal designs.

This dissertation dealt with the geometric optimisation of micro heat sinks (microchannel and micropin-fin) using a combined CFD and mathematical optimisation. Fundamentals of the micro heat sink operation were given in Chapter 1. In Chapter 2, a literature survey was presented to give a clear insight into the effect of various geometric parameters on the heat-removal abilities of such heat sinks.

Chapters 3 and 4 presented the relevant literature pertaining to the computational fluid dynamics modelling of a micro heat sink and the mathematical optimisation algorithms used in this dissertation respectively. The numerical modelling section stated the governing equations that describe the transport of fluid and heat within the heat sink. In Chapter 4, the DYNAMIC-Q algorithm and its application to practical engineering problems were explained.

The methodology developed in the preceding chapters was applied to three design cases. In the first case, an optimal geometry for a microchannel heat sink was numerically determined, which minimises the peak wall temperature using mathematical optimisation and constructal design theory. In the second case, the geometric parameters of a double row micropin-fin were optimised such that they result in the maximal heat transfer rate. In Case 3, a third row of pin-fins was added building on the model created in Case 2. In all three cases, the optimisation process was carried out numerically under total fixed volume and manufacturing constraints.

6.2 CONCLUSIONS

In Case 1, it was seen that the optimal peak wall temperature decreased exponentially with an increase in pressure. It was shown that a unique optimal geometric configuration exists for a given pressure drop applied across a channel that will result in a minimised peak wall temperature. Furthermore, taking more design parameters into account will result in even better cooling capabilities of microchannel heat sinks as up to a 20% increase in the global thermal conductance was obtained when the axial length was relaxed in the optimisation process. In Case 2, it was found that the influence of non-uniform fin-height to the optimal solution is negligible. Results from Case 3 proved that the thermal conductivity ratio does not influence the optimal geometrical configuration in the laminar regime of micropin-fin heat sinks. In all three design cases, it was concluded that for micro heat sink design, material selection and pressure drop considerations play vital roles in the achievement of efficient optimal designs.

The automation process, which allows the incorporation of numerical simulations within an optimisation algorithm, provides a framework for optimisation in a thermal

system. However, for actual optimal solutions to be achieved, correct formulation of the optimisation problem is essential coupled with a relevant understanding of the influence of CFD noise on the solution.

This work demonstrated the effectiveness optimisation has on improving the heat-removal capabilities of micro heat sinks.

6.3 RECOMMENDATIONS AND FUTURE WORK

The following areas were identified as potential aspects for further research and investigation.

6.3.1 Modelling improvement

In this study as discussed in previous chapters, the microchannel heat sink was modelled using a constant pressure inlet boundary condition. This boundary condition can be replaced by a pre-determined profiled inlet condition in a future study to investigate the effect it will have on the peak wall temperature as compared with using a constant pressure boundary condition. This profiled inlet condition can be obtained by first modelling a long adiabatic channel with a constant pressure inlet boundary condition and using the velocity profile obtained at the outlet as the inlet condition for the actual heat sink model.

6.3.2 Application of methodology to staggered pin-fin arrays

In the latter part of this work, the optimisation methodology was applied to pin-fins with emphasis on the in-line configuration of the fins. The optimisation can be extended to finned arrays of a staggered configuration. This will enable a comparative study on the maximised heat transfer capabilities of both configurations.

REFERENCES

[1] G.E. Moore, 'Cramming more components onto integrated circuits', *Electronics,* Vol. 38, No. 8, 1965.

[2] S. Lee, 'Optimum design and selection of heat sinks', *IEEE Transactions on Components, Packaging and Manufacturing Technology – Part A*, Vol. 18, No. 4, pp. 812-817, 1995.

[3] A. Poulikakos, A. Bejan, 'Fin geometry for minimum entropy generation in forced convection', *ASME Journal of Heat Transfer,* Vol. 104, pp. 616-623, 1982.

[4] Y.S. Muzychka, 'Constructal design of forced convection cooled microchannel heat sinks and heat exchangers', *International Journal of Heat and Mass Transfer,* Vol. 48, pp. 3119–3127, 2005.

[5] W.W. Lin, D.J. Lee, 'Second-law analysis on a pin fin array under crossflow', *International Journal of Heat and Mass Transfer,* Vol. 40, No. 8, pp. 1937-1945, 1997.

[6] T. Bello-Ochende, L. Liebenberg, J.P. Meyer, 'Constructal cooling channels for micro-channel heat sinks', *International Journal of Heat and Mass Transfer,* Vol. 50, pp. 4141-4150, 2007.

[7] K.K. Ambatipidi, M.M. Rahman, 'Analysis of conjugate heat transfer in mircochannel heat sinks', *Numerical Heat Transfer, Part A: Applications*, Vol. 37, No. 7, pp. 711-731, 2000.

[8] G. Stanescu, A.J. Fowler, A. Bejan, 'The optimal spacing of cylinders in free-stream crossflow forced convection', *International Journal of Heat and Mass Transfer,* Vol. 39, No. 2, pp. 311-317, 1996.

[9] B. Sahin, A. Demir, 'Thermal performance analysis and optimum design parameters of heat exchanger having perforated pin fins', *Energy Conversion and Management,* Vol. 49, pp. 1684-1695, 2008.

[10] B.A. Jubran, M.A. Hamdan, R.M. Abdullah, 'Enhanced heat transfer, missing pin, and optimisation for cylindrical pin fin arrays', *ASME Journal of Heat Transfer,* Vol. 115, pp. 576-583, 1993.

[11] M. Tahat, Z.H. Kodah, B.A. Jarrah, S.D. Probert, 'Heat transfers from pin-fin arrays experiencing forced convection', *Applied Energy,* Vol. 67, pp. 419-442, 2000.

[12] F.J. Hong, P. Cheng, H. Ge, G.T. Joo, 'Conjugate heat transfer in fractal-shaped microchannel network heat sink for integrated micoelectronic cooling application', *International Journal of Heat and Mass Transfer*, Vol. 50, pp. 4986-4998, 2007.

[13] K.C. Toh, X.Y. Chen, 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.

[14] Science and Technology Magazine 2008, Los Alamos National Security, LLC, viewed 28 September, 2009, <http://www.lanl.gov/news/index.php/fuseaction/1663.article/d/20085/id/13277>.

[15] S.A. Solovitz, 'Microchannels take heatsinks to the next level', *Power Electronics Technology,* pp. 14-20, 2006.

[16] D.B. Tuckerman, R.F.W. Pease, 'High performance heat sinking for VLSI', *IEE Electron Device Letters*, EDL-2, pp. 126-129, 1981.

[17] H.Y. Wu, P. Cheng, 'An experimental study of convective heat transfer in silicon microchannels with different surface conditions', *International Journal of Heat and Mass Transfer*, Vol. 46, pp. 2547–2556, 2003.

[18] B.A. Jasperson, Y. Jeon, K.T. Turner, F.E. Pfefferkorn, W. Qu, 'Comparison of micro-pin-fin and microchannel heat sinks considering thermal-hydraulic performance and manufacturability', *IEEE Transactions on Components and Packaging Technology,* pp. 1-13, 2009.

[19] W.A. Khan, J.R. Culham, M.M. Yovanovich, 'Optimisation of pin-fin heat sinks using entropy generation minimization', *IEEE Transactions on Components and Packaging Technology,* Vol. 28, No. 2, pp. 1-13, 2005.

[20] Hardware Canucks 2008, GTO Media Inc, viewed 10 March, 2010, <http://www.hardwarecanucks.com/forum/hardware-canucks-reviews/7054-asus-rampage-formula-x48-motherboard-review-5.html>.

[21] J. Dirker, J.P. Meyer, 'Thermal characterisation of embedded heat spreading layers in rectangular heat-generating electronic modules', *International Journal of Heat and Mass Transfer,* Vol. 52, pp. 1374-1384, 2009.

[22] B. Xu, K.T. Ooi, N.T. Wong, W.K. Choi, 'Experimental investigation of flow friction for liquid flow in microchannels', *International Communications in Heat and Mass Transfer,* Vol. 27, No. 8, pp. 1165-1176, 2000.

[23] X.F. Peng, G.P. Peterson, 'Convective heat transfer and flow friction for water flow in microchannel structures', *International Journal of Heat and Mass Transfer,* Vol. 39, pp. 2599-2608, 1996.

[24] P. Lee, S.V. Garimella, D. Liu, 'Investigation of heat transfer in rectangular microchannels', *International Journal of Heat and Mass Transfer,* Vol. 48, pp. 1688-1704, 2005.

[25] T.M. Harms, M.J. Kazmierczak, F.M. Gerner, 'Developing convective heat transfer in deep rectangular microchannels', *International Journal of Heat and Fluid Flow,* Vol. 20, 149-157, 1999.

[26] J. Judy, D. Maynes, B.W. Webb, 'Characterization of frictional pressure drop for liquid flows through microchannels', *International Journal of Heat and Mass Transfer,* Vol. 45, pp. 3477-3489, 2002.

[27] G.L. Morini, 'Single-phase convective heat transfer in microchannels: a review of experimental results', *International Journal of Thermal Sciences,* Vol. 43, pp. 631-651, 2004.

[28] H.Y. Wu, P. Cheng, 'Friction factors in smooth trapezoidal silicon microchannels with different aspect ratios', *International Journal of Heat and Mass Transfer*, Vol. 46, pp. 2519-2525, 2003.

[29] J. Koo, C. Kleinstreuer, 'Viscous dissipation effects in microtubes and microchannels', *International Journal of Heat and Mass Transfer,* Vol. 47, pp. 3159–3169, 2004.

[30] H. Abbassi, 'Entropy generation analysis in a uniformly heated microchannel heat sink', *Energy,* Vol. 32, pp. 1932-1947, 2007.

[31] S.S. Shevade, M.M. Rahman, 'Heat transfer in rectangular microchannels during volumetric heating of the substrate', *International Communications in Heat and Mass Transfer*, Vol. 34, pp. 661–672, 2007.

[32] Z. Guo, Z. Li, 'Size effect on single-phase channel flow and heat transfer at microscale', *International Journal of Heat and Fluid Flow*, Vol. 24, pp. 284–298, 2003.

[33] C. Chen, 'Forced convection heat transfer in microchannel heat sinks', *International Journal of Heat and Mass Transfer,* Vol. 50, pp. 2182–2189, 2007.

[34] A. Bejan, E. Sciubba, 'The optimal spacing of parallel plates cooled by forced convection', *International Journal of Heat and Mass Transfer,* Vol. 35, No. 12, pp. 3259-3264, 1992.

[35] T.S. Fisher, K.E. Torrance, 'Constrained optimal duct shapes for conjugate laminar forced convection', *International Journal of Heat and Mass Transfer,* Vol. 43, pp. 113–126, 2000.

[36] G. Gamrat, M. Favre-Marinet, D. Asendrych, 'Conduction and entrance effects on laminar liquid flow and heat transfer in rectangular microchannels', *International Journal of Heat and Mass Transfer,* Vol. 48, pp. 2943–2954, 2005.

[37] Z. Guo, Z. Li, 'Size effects on single-phase channel flow and heat transfer at microscale', *International Journal of Heat and Fluid Flow,* Vol. 24, pp. 284–298, 2003.

[38] W. Qu, I. Mudawar, 'Analysis of three-dimensional heat transfer in micro-channel heat sinks', *International Journal of Heat and Mass Transfer,* Vol. 45, pp. 3973–3985, 2002.

[39] R.K. Kupka, F. Bouamrane, C. Cremers, S. Megtert, 'Microfabrication: LIGA-X and applications', *Applied Surface Science,* Vol. 164, pp. 97-110, 2000.

[40] Y. Peles, A. Koşar, C. Mishra, C. Kuo, B. Schneider, 'Forced convective heat transfer across a pin fin micro heat sink', *International Journal of Heat and Mass Transfer,* Vol. 48, pp. 3615–3627, 2005.

[41] A. Koşar, C. Mishra, Y. Peles, 'Laminar flow across a bank of low aspect ratio micro pin fins', *Journal of Fluids Engineering,* Vol. 127, pp. 419-430, 2005.

[42] W.A. Khan, M.M. Yovanovich, 'Optimisation of pin-fin heat sinks in bypass flow using entropy generation minimization method', *Proceedings of IPACK 2007,* Vancouver, Canada, 2007.

[43] D. Soodphakdee, M. Behnia, D.W. Copeland, 'A comparison of fin geometries for heatsinks in laminar forced convection: Part I - round,

elliptical, and plate fins in staggered and in-line configurations', *The International Journal of Microcircuits and Electronic Packaging,* Vol. 24, No. 1, pp. 68-76, 2001.

[44] K. Yang, W. Chu, I. Chen, C. Wang, 'A comparative study of the airside performance of heat sinks having pin fin configurations', *International Journal of Heat and Mass Transfer,* Vol. 50, pp. 4661-4667, 2007.

[45] P. Jiang, R. Xu, 'Heat transfer and pressure drop characteristics of mini-fin structures', *International Journal of Heat and Fluid Flow,* Vol. 28, pp. 1167-1177, 2007.

[46] W. Qu, A. Siu-Ho, 'Measurement and prediction of pressure drop in a two-phase micro-pin-fin heat sink', *International Journal of Heat and Mass Transfer,* Vol. 52, pp. 5173–5184, 2009.

[47] K. Chiang, F. Chang, 'Application of response surface methodology in the parametric optimisation of a pin-fin type heat sink', *International Communications in Heat and Mass Transfer,* Vol. 33, pp. 836-845, 2006.

[48] K. Chiang, F. Chang, T. Tsai, 'Optimum design parameters of pin-fin heat sink using the grey-fuzzy logic based on the orthogonal arrays', *International Communications in Heat and Mass Transfer,* Vol. 33, pp.744-752, 2006.

[49] K. Pitchandi, E. Natarajan, 'Entropy generation in pin fins of circular and elliptical cross-sections in forced convection with air', *International Journal of Thermodynamics,* Vol. 11, No. 4, pp. 161-171, 2008.

[50] M. Yuan, J. Wei, Y. Xue, J. Fang, 'Subcooled flow boiling heat transfer of FC-72 from silicon chips fabricated with micro-pin-fins', *International Journal of Thermal Sciences,* Vol. 48, pp. 1416-1422, 2009.

[51] Y. Wang, Y. Li, D. Liu, 'The application of genetic algorithm for pin-fin heat sink optimisation design', *IEEE Transactions on Components and Packaging Technologies,* pp. 2816-2821, 2009.

[52] F. Bobaru, S. Rachakonda, 'Optimal shape profiles for cooling fins of high and low conductivity', *International Journal of Heat and Mass Transfer,* Vol. 47, pp. 4953–4966, 2004.

[53] K. Lee, W. Kim, J. Si, 'Optimal shape and arrangement of staggered pins in the channel of a plate heat exchanger', *International Journal of Heat and Mass Transfer,* Vol. 44, pp. 3223–3231, 2001.

[54] N. Sahiti, F. Durst, P. Geremia, 'Selection and optimisation of pin cross-sections for electronics cooling', *Applied Thermal Engineering,* Vol. 27, pp. 111-119, 2007.

[55] N. Sahiti, A. Lemouedda, D. Stojkovic, F. Durst, E. Franz, 'Performance comparison of pin fin in-duct flow arrays with various pin cross-sections', *Applied Thermal Engineering,* Vol. 26, pp. 1176-1192, 2006.

[56] C. Marques, K.W. Kelly, 'Fabrication and performance of a pin fin micro heat exchanger', *Transactions of the ASME,* Vol. 126, pp. 434-444, 2004.

[57] H.K. Versteeg, W. Malalasekra, *An introduction to computational fluid dynamics: the finite volume method*, 2nd Edition, Prentice Hall, England, 2007.

[58] Fluent Inc., *Gambit Version 6 Manuals*, Centerra Resource Park, 10 Cavendish Court, Lebanon, New Hampshire, USA, 2001 (www.fluent.com).

[59] F.M. White, *Viscous Fluid Flow,* 2nd Edition, McGraw-Hill International Editions, Singapore, 1991.

[60] Fluent Inc., *Fluent Version 6 Manuals*, Centerra Resource Park, 10 Cavendish Court, Lebanon, New Hampshire, USA, 2001 (www.fluent.com).

[61] J.A. Snyman, *Practical mathematical optimisation: an introduction to basic optimisation theory and classical and new gradient-based algorithms,* Springer, New York, 2005.

[62] J.A. Snyman, 'A new and dynamic method for unconstrained minimization', *Applied. Mathematical Modelling*, Vol. 6, pp. 449-462, 1982.

[63] J.A. Snyman, 'An improved version of the original leap-frog dynamic method for unconstrained minimization: LFOP1(b)', *Applied. Mathematical Modelling*, Vol. 7, pp. 216-218, 1983.

[64] J.A. Snyman, N. Stander, W.J. Roux, 'A dynamic penalty function method for the solution of structural optimisation problems', *Applied. Mathematical Modelling*, Vol. 18, pp. 453-460, 1994.

[65] J.A. Snyman, 'The LFOPC Leap-frog algorithm for constrained optimisation', *Computer and Mathematics with Applications*, Vol. 40, pp. 1085-1096, 2000.

[66] J.A. Snyman, A.M. Hay, 'The DYNAMIC-Q optimisation method: an alternative to SQP?', *Computer and Mathematics with Applications*, Vol. 44, pp. 1589-1598, 2002.

[67] D.J. de Kock, Optimal tundish methodology in a continuous casting process. PhD Thesis, Department of Mechanical and Aeronautical Engineering, University of Pretoria, 2005.

[68] R. K. Shah, A. L. London, Laminar flow forced convection in ducts: a source book for compact heat exchanger analytical data, Supl. 1. Academic Press, New York, 1978.

[69] A. Husain, K. Kim, 'Shape optimisation of micro-channel heat sink for micro-electronic cooling', *IEEE Transactions on Components and Packaging Technologies,* Vol. 31, No. 2, pp. 322-330, 2008.

[70] J. Li, G. P. Peterson, 'Geometric optimisation of a micro heat sink with liquid flow', *IEEE Transactions on Components and Packaging Technologies,* Vol. 29, No. 1, pp. 145-154, 2006.

[71] F. Laermer, A. Urban, 'Challenges, developments and application of silicon deep reactive ion etching', *Microelectronic Engineering*, Vol. 67-68, pp. 349-355, 2003.

[72] M.J. Madou, "MEMS Fabrication," in *MEMS Handbook,* M. Gad-el-Hak, Ed., Boca Raton, FL: CRC, 2002.

[73] The MathWorks, Inc., *MATLAB & Simulink Release Notes for R2008a,* 3 Apple Hill Drive, Natick, MA, 2008 (www.mathworks.com).

[74] A. Husain, K. Kim, 'Multiobjective optimisation of a microchannel heat sink using evolutionary algorithm', *Journal of Heat Transfer,* Vol. 130, pp. 1-3, 2008.

[75] V.S. Achanta, An experimental study of endwall heat transfer enhancement for flow past staggered non-conducting pin fin arays. PhD Thesis, Department of Mechanical Engineering, Texas A & M University, 2003.

[76] M.E. Lyall, Heat transfer from low aspect ratio pin fins. PhD Thesis, Department of Mechanical Engineering, Virginia Polytechnic Institute and State University, 2006.

[77] T. Bello-Ochende, J.P. Meyer, A. Bejan, 'Constructal multi-scale pin-fins', *International Journal of Heat and Mass Transfer,* Vol. 53, pp. 2773-2779, 2010.

PUBLICATIONS IN JOURNALS AND CONFERENCE PAPERS

The following article and conference papers were produced during this study.

Article Published

1. T. Bello-Ochende, J.P. Meyer, F.U. Ighalo, 'Combined numerical optimisation and constructal theory for the design of micro-channel heat sinks', *Numerical Heat Transfer, Part A: Applications,* Vol. 11, pp. 882-899, 2010.

Conference Papers Published

1. F.U. Ighalo, T. Bello-Ochende, J.P. Meyer, 'Mathematical optimisation: Application to the design of optimal micro-channel heat sinks', *Proceedings of the Third Southern Conference on Computational Modelling,* Rio Grande, RS, Brazil, 23-25 November, 2009.

2. F.U. Ighalo, T. Bello-Ochende, J.P. Meyer, 'Designed micro-channel heat sinks using mathematical optimisation with variable axial length', *Proceedings of the 7th International Conference on Heat Transfer, Fluid Mechanics and Thermodynamics,* Antalya, Turkey, pp. 1345-1350, 19-21 July, 2010.

Conference Paper Accepted

1. F.U. Ighalo, T. Bello-Ochende, J.P. Meyer, 'Geometric optimisation of multiple-arrays of micropin-fins', *Proceedings of ASME/JSME 2011 8th Thermal Engineering Joint Conference,* AJTEC2011-44285, Honolulu, Hawaii, USA, 14-17 March, 2011, Accepted on 28 October, 2010.

2. F.U. Ighalo, T. Bello-Ochende, J.P. Meyer, 'Design of multiple row micropin-fins using mathematical optimization', *Proceedings of the 2nd AfriCOMP Conference,* Paper Number 62, Cape Town, South Africa, 5-8 January, 2011, Accepted on 3 November, 2010.

Article Submitted

1. F.U. Ighalo, T. Bello-Ochende, J.P. Meyer, 'Maximum heat transfer from rows of micro pin-fins with non-uniform configurations', *International Journal of Thermal Sciences,* THESCI-D-11-00025, Submitted.

A

APPENDIX A: DYNAMIC-Q OPTIMISATION ALGORITHM

A.1 DYNQ.M

```
function [X,F]=dynq(x0,varargin);
tic
%
%         DYNAMIC-Q ALGORITHM FOR CONSTRAINED OPTIMISATION
%              GENERAL MATHEMATICAL PROGRAMMING CODE
%              -------------------------------------
%
% This code is based on the Dynamic-Q method of Snyman documented
% in the paper "THE DYNAMIC-Q OPTIMISATION METHOD: AN ALTERNATIVE
% TO SQP?" by J.A. Snyman and A.M. Hay. Technical Report, Dept Mech.
% Eng., UP.
%
%                MATLAB implementation by A.M. HAY
%        Multidisciplinary Design Optimisation Group (MDOG)
%  Department of Mechanical Engineering, University of Pretoria
%                       August 2002
%
%                   UPDATED : 23 August 2002
%
%                      BRIEF DESCRIPTION
%                      -----------------
%

%  Dynamic-Q solves inequality and equality constrained optimisation
%  problems of the form:
%
%                   minimise F(X)  ,  X={X(1),X(2),...,X(N)}
%    such that
%                    Cp(X) <= 0        p=1,2,...,NP
%    and
%                    Hq(X) =  0        q=1,2,...,NQ
%    with lower bounds
%        CLi(X) = V_LOWER(i)-X(NLV(i)) <= 0   i=1,2,...,NL
%    and upper bounds
%        CUj(X) = X(NUV(j))-V_UPPER(j) <= 0   j=1,2,...,NU
%
% This is a completely general code - the objective function and the
% constraints may be linear or non-linear. The code therefore solves
% LP, QP and NLP problems.
%
%                      -----------------
%
% User specified functions:
%
% The objective function F and constraint functions C and H must be
```

```
% specified by the user in function FCH. Expressions for the
respective
% gradient vectors must be specified in function GRADFCH.
%
% {The user may compute gradients by finite differences if necessary
%  - see example code in GradFCH}
%
% Side constraints should not be included as inequality constraints
% in the above subroutines, but passed to the dynq function as
% input arguments LO and UP. (Described below)
%
% In addition to FCH and GRADFCH the following functions are called
% by DYNQ and should not be altered:
%       DQLFOPC,DQFUN,DQCONIN,DQCONEQ,DQGRADF,DQGRADC,DQGRADH
%
% In addition the script HISTPLOT.m plots various optimisation
% histories. To suppress automatic plotting set PRNCONST=0 below.
%
%                       -----------------
%
%   synopsis:
%
%           [X,F] = dynq(x0,lo,up,dml,xtol,ftol,clim,np,nq,kloop);
%
%   outputs:
%       X  = optimal solution (1xN)
%       F  = optimal function value
%
%   inputs:
%       x0 = starting point (1xN)
%       lo = NLx2 matrix associated with lower limits on the
variables
%               containing variable index NLV(i) in the first column
and
%               associated value V_LOWER of that limit in the second
column
%               (optional, otherwise assumed no lower side
constraints)
%       up = NUx2 matrix associated with lower limits on the
variables
%               containing variable index NUV(i) in the first column
and
%               associated value V_UPPER of that limit in the second
column
%               (optional, otherwise assumed no upper side
constraints)
%      dml = the move limit which should be approximately the same
order
%               of magnitude as the "radius of the region of
interest"
%               = sqrt(n)*max-variable-range (optional, default =1)
%     xtol = convergence tolerance on the step size (optional,
default =1e-5)
%     ftol = convergence tolerance on the function value (optional,
default =1e-8)
%     clim = tolerance for determining whether constraints are
violated
%               (optional, default =ftol*1e2)
%       np = number of inequality constraints (optional)
%       nq = number of equality constraints (optional)
```

```
%               Note: Both np and nq are optional and determined
automatically
%               if not specified, but at the cost of an extra
function evalution.
%    kloop = maximum number of iterations (optional, default = 100)
%
%     NOTE: use [] to activate default inputs, for example
%
%  [X,F]=dynq(x0,[],[],2); uses dml=2 but default values for all
other inputs.
%
%       See FCH and GRADFCH for an example problem.
%
%   ---- This program is for educational purposes only ----


%*****PLOT OPTIMISATION HISTORIES AT END OF
PROGRAM?******************
%           YES: 1        OR            NO: 0
%
PRNCONST=1;
%*****************************************************************
**

clc;

N=length(x0);   % Determine number of variables
X=x0;

[dum,D]=size(varargin);
vars=cell(1,9);
vars(1:D)=varargin;

LO=vars{1};
UP=vars{2};
DML=vars{3};
XTOL=vars{4};
FTOL=vars{5};
CLIM=vars{6};
NP=vars{7};
NQ=vars{8};
KLOOPMAX=vars{9};

% default values
[NL,dum]=size(LO);
if NL>0
    NLV=LO(:,1)';
    V_LOWER=LO(:,2)';
else
    NLV=[];
    V_LOWER=[];
end
[NU,dum]=size(UP);
if NU>0
    NUV=UP(:,1)';
    V_UPPER=UP(:,2)';
else
    NUV=[];
    V_UPPER=[];
end
```

```
if isempty(DML)
    DML=1; end
if isempty(XTOL)
    XTOL=1e-5; end
if isempty(FTOL)
    FTOL=1e-8; end
if isempty(CLIM)
    CLIM=FTOL*1e2; end
if isempty(NP)|isempty(NQ)
    [F,C,H]=fch(X);
    NP=length(C);
    if isempty(C)
        NP=0;
    end
    NQ=length(H);
    if isempty(H)
        NQ=0;
    end
end
if isempty(KLOOPMAX)
    KLOOPMAX=100; end

%######################################################################
##C
%**********************************************************************
**C
%     MAIN PROGRAM FOLLOWS: Do not alter!!!!
%**********************************************************************
**C
%######################################################################
##C

%*****OPEN OUPUT
FILES************************************************C
%
fidA=fopen('Approx.out','wt+');
fidD=fopen('DynamicQ.out','wt+');
fidH=fopen('History.out','wt+');
%
%*****SPECIFY INITIAL APPROXIMATION
CURVATURES************************C
%
ACURV=0.D0;
BCURV=zeros(1,NP);
if NP==0
    BCURV=[];
end
CCURV=zeros(1,NQ);
if NQ==0
    CCURV=[];
end
%
%
%
%*****INITIALIZE
OUTPUT*********************************************C
FEASIBLE=0;

fprintf(fidA,' DYNAMICQ OUTPUT FILE \n');
fprintf(fidA,' -------------------- \n');
fprintf(fidA,' Number of variables [N]= %i \n',N);
```

```
fprintf(fidA,' Number of inequality constraints [NP]= %i \n',NP);
fprintf(fidA,' Number of equality constraints [NQ]= %i \n',NQ);
fprintf(fidA,' Move limit= %12.8e \n',DML);

fprintf(1,'\n DYNAMICQ OPTIMISATION ALGORITHM \n');
fprintf(1,' ------------------------------ \n');
% (MAXX=Maximum number of X-values to be displayed on screen)
MAXX=4;
if N<=MAXX
    fprintf(1,' Iter Function value  ? XNORM      RFD        ');
    fprintf(1,'X(%i)       ',1:N);
    fprintf(1,'\n ----------------------------------------');
    for I=1:N
        fprintf(1,'------------',1:N);
    end
    fprintf(1,'\n');
else
    fprintf(1,' Iter Function value  ? XNORM      RFD ');
    fprintf(1,'\n ----------------------------------------------\n');
end

fprintf(fidD,' DYNAMICQ OPTIMISATION ALGORITHM\n');
fprintf(fidD,' ------------------------------\n');
fprintf(fidD,' Iter Function value       ? XNORM      RFD        ');
fprintf(fidD,'X(%i)           ',1:N);
fprintf(fidD,'\n');

fprintf(fidD,' ------------------------------------------------');
for i=1:N
    fprintf(fidD,'---------------');
end
fprintf(fidD,'\n');

% Initialize outer loop counter
KLOOP=0;

% Arbitrary large values to prevent premature termination
F_LOW=1.D6;
RFD=1.D6;
RELXNORM=1.D6;

C_A=zeros(1,NP+NL+NU+1);

%*****START OF OUTER OPTIMISATION
LOOP*******************************C

while KLOOP<=KLOOPMAX

%*****APPROXIMATE
FUNCTIONS*****************************************C

% Determine function values
    [F,C,H]=fch(X);

% Calculate relative step size
    if KLOOP>0
        DELXNORM=sqrt((X_H(KLOOP,:)-X)*(X_H(KLOOP,:)-X)');
        XNORM=sqrt(X*X');
        RELXNORM=DELXNORM/(1+XNORM);
```

```
    end

% Determine lowest feasible function value so far
    if KLOOP>0
    FEASIBLE=1;
        check=find(C<CLIM);
        if isempty(check)&NP>0;
            FEASIBLE=0;
        end
        check=find(abs(H)<CLIM);
        if isempty(check)&NQ>0;
            FEASIBLE=0;
        end
        for I=1:NL
            if C_A(I+NP)>CLIM
                FEASIBLE=0;
            end
        end
        for I=1:NU
            if C_A(I+NP+NL)>CLIM
                FEASIBLE=0;
            end
        end
    end

% Calculate relative function difference
    if F_LOW~=1.D6&FEASIBLE==1
        RFD=abs(F-F_LOW)/(1+abs(F));
    end

    if FEASIBLE==1&F<F_LOW
        F_LOW=F;
    end

% Store function values
    X_H(KLOOP+1,:)=X;   % Need to adjust from Fortran version since
    F_H(KLOOP+1)=F;     % Matlab does not accept 0 as a matrix index
    if NP>0
        C_H(KLOOP+1,1:NP)=C;
    end
    if NL>0
        C_H(KLOOP+1,NP+1:NP+NL)=C_A(NP+1:NP+NL);
    end
    if NU>0
        C_H(KLOOP+1,NP+NL+1:NP+NL+NU)=C_A(NP+NL+1:NP+NL+NU);
    end
    C_H(KLOOP+1,NP+NL+NU+1)=C_A(NP+NL+NU+1);
    if NQ>0
        H_H(KLOOP+1,:)=H;
    end

% Determine gradients
    [GF,GC,GH]=gradfch(X);

% Calculate curvatures
    if KLOOP>0
        DELX=X_H(KLOOP,:)-X_H(KLOOP+1,:);
        DELXNORM=DELX*DELX';

% Calculate curvature ACURV
```

```
        DP=GF*DELX';
        ACURV=2.*(F_H(KLOOP)-F_H(KLOOP+1)-GF*DELX')/DELXNORM;

        for J=1:NP
            DP=GC(J,:)*DELX';
% Calculate corresponding curvature BCURV(J)
            BCURV(J)=2.*(C_H(KLOOP,J)-C_H(KLOOP+1,J)-
GC(J,:)*DELX')/DELXNORM;
        end

        for J=1:NQ
            DP=GH(J,:)*DELX';
% Calculate corresponding curvature CCURV(J)
            CCURV(J)=2.*(H_H(KLOOP,J)-H_H(KLOOP+1,J)-
GH(J,:)*DELX')/DELXNORM;
        end
    end

%*****RECORD PARAMETERS FOR THE
ITERATION*****************************C

% Write approximation constants to Approx.out
    fprintf(fidA,' Iteration %i \n',KLOOP);
    fprintf(fidA,' --------------\n');
    fprintf(fidA,' X=\n');
    for I=1:N
        fprintf(fidA,' %12.8f ',X(I));
    end
    fprintf(fidA,'\n F= %15.8e\n',F);
    for I=1:NP
        fprintf(fidA,' C(%i)=%15.8e',I,C(I));
    end
    for I=1:NQ
         fprintf(fidA,' H(%i)=%15.8e',I,H(I));
    end

    fprintf(fidA,' Acurv=%15.8e',ACURV);
    for I=1:NP
        fprintf(fidA,' Bcurv(%i)=%15.8e',I,BCURV(I));
    end
    for I=1:NQ
        fprintf(fidA,' Ccurv(%i)=%15.8e',I,CCURV(I));
    end

% Write solution to file
    if KLOOP==0
        fprintf(fidD,' %4i %+19.12e %i
',KLOOP,F,FEASIBLE);
    else
        if RFD~=1.D6
            fprintf(fidD,' %4i %+19.12e %i %9.3e
%9.3e',KLOOP,F,FEASIBLE,RELXNORM,RFD);
        else
            fprintf(fidD,' %4i %+19.12e %i %9.3e
',KLOOP,F,FEASIBLE,RELXNORM);
        end
    end
    fprintf(fidD,' %+13.6e',X);
    fprintf(fidD,'\n');
```

```
% Write solution to screen
    if KLOOP==0
        if N<=MAXX
            fprintf(1,' %4i %+14.7e %i
',KLOOP,F,FEASIBLE);
            fprintf(1,' %+9.2e',X);
            fprintf(1,'\n');
        else
            fprintf(1,' %4i %+14.7e %i\n',KLOOP,F,FEASIBLE);
        end
     else
        if N<=MAXX
            if RFD~=1.D6&FEASIBLE==1
                fprintf(1,' %4i %+14.7e %i %9.3e
%9.3e',KLOOP,F,FEASIBLE,RELXNORM,RFD);
            else
                fprintf(1,' %4i %+14.7e %i %9.3e
',KLOOP,F,FEASIBLE,RELXNORM);
            end
            fprintf(1,' %+9.2e',X);
            fprintf(1,'\n');
        else
            if RFD~=1.D6&FEASIBLE==1
                fprintf(1,' %4i %+14.7e %i %9.3e
%9.3e\n',KLOOP,F,FEASIBLE,RELXNORM,RFD);
            else
                fprintf(1,' %4i %+14.7e %i
%9.3e\n',KLOOP,F,FEASIBLE,RELXNORM);
            end
        end
    end

% Exit do loop here on final iteration
    if KLOOP==KLOOPMAX|RFD<FTOL|RELXNORM<XTOL
        if KLOOP==KLOOPMAX
            fprintf(1,' Terminated on max number of steps\n');
            fprintf(fidD,' Terminated on max number of steps\n');
        end
        if RFD<FTOL
            fprintf(1,' Terminated on function value\n');
            fprintf(fidD,' Terminated on function value\n');
        end
        if RELXNORM<XTOL
            fprintf(1,' Terminated on step size\n');
            fprintf(fidD,' Terminated on step size\n');
        end
        fprintf(1,'\n');
        fprintf(fidD,'\n');
        break;
    end

%*****SOLVE THE APPROXIMATED
SUBPROBLEM*******************************C

[X,F_A,C_A,H_A]=dqlfopc(X,NP,NQ,F,C,H,GF,GC,GH,ACURV,BCURV,CCURV,DML.
..
    ,NL,NU,NLV,NUV,V_LOWER,V_UPPER,XTOL,KLOOP);

% Record solution to approximated problem
```

```
    fprintf(fidA,'Solution of approximated problem:\n');
    fprintf(fidA,'X=\n');
    for I=1:N
        fprintf(fidA,' %12.8f\n',X(I));
    end
    fprintf(fidA,' F_A=%15.8e\n',F_A);
    for I=1:NP+NL+NU+1
        fprintf(fidA,'C_A(%i)=%15.8e\n',I,C_A(I));
    end
    for I=1:NQ
        fprintf(fidA,'H_A(%i)=%15.8e\n',I,H_A(I));
    end

% Increment outer loop counter
    KLOOP=KLOOP+1;
end

% Write final constraint values to file

if NP>0
    fprintf(fidD,' Final inequality constraint function values:\n');
    for I=1:NP
        fprintf(fidD,'  C(%i)=%15.8e\n',I,C(I));
    end
end
if NQ>0
    fprintf(fidD,' Final equality constraint function values:\n');
    for I=1:NQ
        fprintf(fidD,'  H(%i)=%15.8e\n',I,H(I));
    end
end
if NL>0
    fprintf(fidD,' Final side (lower) constraint function
values:\n');
    for I=1:NL
        fprintf(fidD,'  C(X(%i))=%15.8e\n',NLV(I),C_A(NP+I));
    end
end
if NU>0
    fprintf(fidD,' Final side (upper) constraint function
values:\n');
    for I=1:NU
        fprintf(fidD,'  C(X(%i))=%15.8e\n',NUV(I),C_A(NP+NL+I));
    end
end

% Write final constraint values to screen
fprintf(1,' Constraint values follow:\n\n')
if NP>0
    fprintf(1,' Final inequality constraint function values:\n');
    for I=1:NP
        fprintf(1,'  C(%i)=%15.8e\n',I,C(I));
    end
end
if NQ>0
    fprintf(1,' Final equality constraint function values:\n');
    for I=1:NQ
        fprintf(1,'  H(%i)=%15.8e\n',I,H(I));
    end
end
```

```
if NL>0
    fprintf(1,' Final side (lower) constraint function values:\n');
    for I=1:NL
        fprintf(1,'  C(X(%i))=%15.8e\n',NLV(I),C_A(NP+I));
    end
end
if NU>0
    fprintf(1,' Final side (upper) constraint function values:\n');
    for I=1:NU
        fprintf(1,'  C(X(%i))=%15.8e\n',NUV(I),C_A(NP+NL+I));
    end
end

% Write history vectors

fprintf(fidH,' %3i%3i%3i%3i%3i%3i\n', KLOOP,N,NP,NL,NU,NQ);
for I=1:KLOOP+1
    fprintf(fidH,' %3i %15.8e',I-1,F_H(I));
    for J=1:N
        fprintf(fidH,' %15.8e',X_H(I,J));
    end
    fprintf(fidH,'\n');
end
if NP>0
    for I=1:KLOOP+1
        fprintf(fidH,' %3i',I-1);
        for J=1:NP
            fprintf(fidH,' %15.8e',C_H(I,J));
        end
        fprintf(fidH,'\n');
    end
end
if NL>0
    for I=1:KLOOP+1
        fprintf(fidH,' %3i',I-1);
        for J=NP+1:NP+NL
            fprintf(fidH,' %15.8e',C_H(I,J));
        end
        fprintf(fidH,'\n');
    end
end
if NU>0
    for I=1:KLOOP+1
        fprintf(fidH,' %3i',I-1);
        for J=NP+NL+1:NP+NL+NU
            fprintf(fidH,' %15.8e',C_H(I,J));
        end
        fprintf(fidH,'\n');
    end
end
if NQ>0
    for I=1:KLOOP+1
        fprintf(fidH,' %3i',I-1);
        for J=1:NQ
            fprintf(fidH,' %15.8e',H_H(I,J));
        end
        fprintf(fidH,'\n');
    end
end

fclose(fidD);
```

```
fclose(fidH);
fclose(fidA);

if PRNCONST
    histplot;
%    disp('Press a key to continue');
%    pause;
%    close all;
end
toc
```

A.2 FCH.M

```
function [F,C,H]=fch(X);
% Objective and constraint function evaluation for DYNAMIC-Q
%           (USER SPECIFIED)
%
%   synopsis:
%
%       [F,C,H]=fch(X);
%
%   outputs:
%       F  = objective function value
%       C  = vector of inequality constraint functions (1xNP)
%       H  = vector of equality constraint functions (1xNQ)
%
%   inputs:
%       X = design vector (1xN)
%
%                       -----------------
%
% The application of the code is illustrated here for the very simple
% but general example problem (Hock 71):
%
%      minimise  F(X) = X(1)*X(4)*(X(1)+X(2)+X(3))+X(3)
% such that
%                 C(X) = 25-X(1)*X(2)*X(3)*X(4) <= 0
%      and
%                 H(X) = X(1)^2+X(2)^2+X(3)^2+X(4)^2-40 = 0
%
%     and side constraints
%
%                 1 <= X(I) <= 5 , I=1,2,3,4
%
%   Starting point is (1,5,5,1)
%
%   Solution of this problem is accomplished by:
%       (with FCH and GRADFCH unaltered)
%
%       x0=[1,5,5,1] % Specify starting point
%       lo=[1:4;1,1,1,1]' % Specify lower limits
%       up=[1:4;5,5,5,5]' % Specify upper limits
%       [X,F]=dynq(x0,lo,up); % Solve using Dynamic-Q
%
%   NOTE: This function should return C=[]; H=[]; if these are
%         not defined.
%
%   See also DYNQ and GRADFCH
%

%Objective Function
%Load Design Variables

%Get the Total Heat transfer

F = -LL4{2};

%Inequality Constraints
C(1)=(X(3)/(4*X(1)))-1;
C(2)=1-(2*X(3)/X(1));
```

```
C(3)=(X(4)/(4*X(2)))-1;
C(4)=1-(2*X(4)/X(2));

Volu = 0.05;
%Equality Constraints
H(1)=(X(1)^2*X(3))+(X(2)^2*X(4))-(4*Volu/pi);

% To eliminate error messages
% Do not delete

if ~exist('C')
    C=[];
end
if ~exist('H')
    H=[];
end
```

A.3 GRADFCH.M

```
function [GF,GC,GH]=gradfch(X);
% Objective and constraint function GRADIENT evaluation for DYNAMIC-Q
%           (USER SPECIFIED)
%
%   synopsis:
%
%       [GF,GC,GH]=gradfch(X);
%
%   outputs: Partial derivatives wrt variables X(I) of
%      GF  = objective function (1xN)
%      GC  = inequality constraint functions (NPxN)
%      GH  = equality constraint functions (NQxN)
%
%   inputs:
%       X = design vector (1xN)
%
%     COMPUTE THE GRADIENT VECTORS OF THE OBJECTIVE FUNCTION F,
%     INEQUALITY CONSTRAINTS C, AND EQUALITY CONSTRAINTS H
%     W.R.T. THE VARIABLES X(I):
%              GF(I),I=1,N
%              GC(J,I), J=1,NP I=1,N
%              GH(J,I), J=1,NQ I=1,N
%
%   NOTE: This function should return GC=[]; GH=[]; if these are
%         not defined.
%
%   See also DYNQ, FCH
%

% Determine gradients by finite difference
FDFLAG=1;

if FDFLAG
    DELTX=1.D-4;    % Finite difference interval
    [F,C,H]=fch(X);
    N=length(X);
    for I=1:N
        DX=X;
        DX(I)=X(I)+DELTX;
        [F_D,C_D,H_D]=fch(DX);
        GF(I)=(F_D-F)/DELTX;
        if ~isempty(C)
            GC(1,1)=-X(3)/(4*X(1)^2);
            GC(1,2)=0;
            GC(1,3)=1/(4*X(1));
            GC(1,4)=0;
            GC(1,5)=0;
            GC(2,1)=2*X(3)/(X(1)^2);
            GC(2,2)=0;
            GC(2,3)=-2/X(1);
            GC(2,4)=0;
            GC(2,5)=0;
            GC(3,1)=0;
            GC(3,2)=-X(4)/(4*X(2)^2);
            GC(3,3)=0;
            GC(3,4)=1/(4*X(2));
            GC(3,5)=0;
            GC(4,1)=0;
```

```
            GC(4,2)=2*X(4)/(X(2)^2);
            GC(4,3)=0;
            GC(4,4)=-2/X(2);
            GC(4,5)=0;
        end
        if ~isempty(H)
        GH(1,1)=2*X(1)*X(3);
        GH(1,2)=2*X(2)*X(4);
        GH(1,3)=X(1)^2;
        GH(1,4)=X(2)^2;
        GH(1,5)=0;
        end
    end
end

% To eliminate error messages
% Do not erase
if ~exist('GC')
    GC=[];
end
if ~exist('GH')
    GH=[];
end
```

A.4 EXECUTE_FINSIM.M

```
%This program initiates DYNQ.M
clear all
clc
close all
x0=[+2.824638e-001 +1.513331e-001 +6.310029e-001 +5.814793e-001
+5.0000e-002];
lo=[1 0.05
    2 0.05
    5 0.05];
up=[3 0.95
    4 0.95];
dml=0.0005;
xtol=[];
ftol=[];
clim=[];
np=4;
nq=1;
kloop=[];
[X,F] = dynq(x0,lo,up,dml,xtol,ftol,clim,np,nq,kloop);
```

B

APPENDIX B: GAMBIT JOURNAL FILE FOR GRID CREATION AND MESHING

B.1 MICROCHANNEL HEAT SINK JOURNAL FILE

```
/
/ File opened for write Tue Apr 19 08:46:13 2005.
$L1=$X1/10
$L2=$X5-($X1/10)
$L3=$X5
/Height
$H1=$X2
$H2=$X3
$H3=$X4
/
/Axial Direction
$Z1= 10
/
vertex create coordinates 0   0  0
vertex create coordinates $L1 0  0
vertex create coordinates $L2 0  0
vertex create coordinates $L3 0  0
/
vertex create coordinates 0   $H1 0
vertex create coordinates $L1 $H1 0
vertex create coordinates $L2 $H1 0
vertex create coordinates $L3 $H1 0
/
vertex create coordinates 0   $H2 0
vertex create coordinates $L1 $H2 0
vertex create coordinates $L2 $H2 0
vertex create coordinates $L3 $H2 0
/
vertex create coordinates 0   $H3 0
vertex create coordinates $L1 $H3 0
vertex create coordinates $L2 $H3 0
vertex create coordinates $L3 $H3 0
/
/
vertex create coordinates 0   0  $Z1
vertex create coordinates $L1 0  $Z1
vertex create coordinates $L2 0  $Z1
vertex create coordinates $L3 0  $Z1
/
vertex create coordinates 0   $H1 $Z1
vertex create coordinates $L1 $H1 $Z1
vertex create coordinates $L2 $H1 $Z1
vertex create coordinates $L3 $H1 $Z1
/
vertex create coordinates 0   $H2 $Z1
```

```
vertex create coordinates $L1 $H2 $Z1
vertex create coordinates $L2 $H2 $Z1
vertex create coordinates $L3 $H2 $Z1
/
vertex create coordinates 0   $H3 $Z1
vertex create coordinates $L1 $H3 $Z1
vertex create coordinates $L2 $H3 $Z1
vertex create coordinates $L3 $H3 $Z1
/
edge create straight "vertex.1" "vertex.5" "vertex.9" "vertex.13"
edge create straight "vertex.2" "vertex.6" "vertex.10" "vertex.14"
edge create straight "vertex.3" "vertex.7" "vertex.11" "vertex.15"
edge create straight "vertex.4" "vertex.8" "vertex.12" "vertex.16"
edge create straight "vertex.1" "vertex.2" "vertex.3" "vertex.4"
edge create straight "vertex.5" "vertex.6" "vertex.7" "vertex.8"
edge create straight "vertex.9" "vertex.10" "vertex.11" "vertex.12"
edge create straight "vertex.13" "vertex.14" "vertex.15" "vertex.16"
edge create straight "vertex.17" "vertex.21" "vertex.25" "vertex.29"
edge create straight "vertex.18" "vertex.22" "vertex.26" "vertex.30"
edge create straight "vertex.19" "vertex.23" "vertex.27" "vertex.31"
edge create straight "vertex.20" "vertex.24" "vertex.28" "vertex.32"
edge create straight "vertex.17" "vertex.18" "vertex.19" "vertex.20"
edge create straight "vertex.21" "vertex.22" "vertex.23" "vertex.24"
edge create straight "vertex.25" "vertex.26" "vertex.27" "vertex.28"
edge create straight "vertex.29" "vertex.30" "vertex.31" "vertex.32"
edge create straight "vertex.1" "vertex.17"
edge create straight "vertex.5" "vertex.21"
edge create straight "vertex.9" "vertex.25"
edge create straight "vertex.13" "vertex.29"
edge create straight "vertex.2" "vertex.18"
edge create straight "vertex.6" "vertex.22"
edge create straight "vertex.10" "vertex.26"
edge create straight "vertex.14" "vertex.30"
edge create straight "vertex.3" "vertex.19"
edge create straight "vertex.7" "vertex.23"
edge create straight "vertex.11" "vertex.27"
edge create straight "vertex.15" "vertex.31"
edge create straight "vertex.4" "vertex.20"
edge create straight "vertex.8" "vertex.24"
edge create straight "vertex.12" "vertex.28"
edge create straight "vertex.16" "vertex.32"
/
face create wireframe "edge.3" "edge.22" "edge.6" "edge.19" real
face create wireframe "edge.6" "edge.23" "edge.20" "edge.9" real
face create wireframe "edge.9" "edge.24" "edge.12" "edge.21" real
face create wireframe "edge.2" "edge.19" "edge.5" "edge.16" real
face create wireframe "edge.5" "edge.20" "edge.8" "edge.17" real
face create wireframe "edge.8" "edge.21" "edge.11" "edge.18" real
face create wireframe "edge.1" "edge.16" "edge.4" "edge.13" real
face create wireframe "edge.4" "edge.17" "edge.7" "edge.14" real
face create wireframe "edge.7" "edge.18" "edge.10" "edge.15" real
face create wireframe "edge.27" "edge.46" "edge.30" "edge.43" real
face create wireframe "edge.30" "edge.47" "edge.33" "edge.44" real
face create wireframe "edge.33" "edge.48" "edge.36" "edge.45" real
face create wireframe "edge.26" "edge.43" "edge.29" "edge.40" real
face create wireframe "edge.29" "edge.44" "edge.32" "edge.41" real
face create wireframe "edge.32" "edge.45" "edge.35" "edge.42" real
face create wireframe "edge.25" "edge.40" "edge.28" "edge.37" real
face create wireframe "edge.28" "edge.41" "edge.31" "edge.38" real
face create wireframe "edge.31" "edge.42" "edge.34" "edge.39" real
face create wireframe "edge.52" "edge.3" "edge.51" "edge.27" real
```

```
face create wireframe "edge.51" "edge.2" "edge.50" "edge.26" real
face create wireframe "edge.50" "edge.1" "edge.49" "edge.25" real
face create wireframe "edge.30" "edge.56" "edge.6" "edge.55" real
face create wireframe "edge.55" "edge.5" "edge.54" "edge.29" real
face create wireframe "edge.54" "edge.4" "edge.53" "edge.28" real
face create wireframe "edge.33" "edge.60" "edge.9" "edge.59" real
face create wireframe "edge.59" "edge.8" "edge.58" "edge.32" real
face create wireframe "edge.58" "edge.7" "edge.57" "edge.31" real
face create wireframe "edge.64" "edge.12" "edge.63" "edge.36" real
face create wireframe "edge.63" "edge.11" "edge.62" "edge.35" real
face create wireframe "edge.62" "edge.10" "edge.61" "edge.34" real
face create wireframe "edge.52" "edge.56" "edge.22" "edge.46" real
face create wireframe "edge.56" "edge.23" "edge.60" "edge.47" real
face create wireframe "edge.60" "edge.24" "edge.64" "edge.48" real
face create wireframe "edge.51" "edge.55" "edge.19" "edge.43" real
face create wireframe "edge.44" "edge.55" "edge.20" "edge.59" real
face create wireframe "edge.59" "edge.21" "edge.63" "edge.45" real
face create wireframe "edge.50" "edge.16" "edge.54" "edge.40" real
face create wireframe "edge.41" "edge.54" "edge.17" "edge.58" real
face create wireframe "edge.58" "edge.18" "edge.62" "edge.42" real
face create wireframe "edge.49" "edge.13" "edge.53" "edge.37" real
face create wireframe "edge.53" "edge.14" "edge.57" "edge.38" real
face create wireframe "edge.57" "edge.15" "edge.61" "edge.39" real
/
volume create stitch "face.19" "face.22" "face.1" "face.10"
"face.31" \
   "face.34" real
volume create stitch "face.22" "face.25" "face.2" "face.11" "face.32"
\
  "face.35" real
volume create stitch "face.25" "face.28" "face.33" "face.36"
"face.12" \
  "face.3" real
volume create stitch "face.13" "face.4" "face.20" "face.23" "face.34"
\
  "face.37" real
volume create stitch "face.35" "face.38" "face.23" "face.26"
"face.14" \
  "face.5" real
volume create stitch "face.15" "face.6" "face.26" "face.29" "face.36"
\
  "face.39" real
volume create stitch "face.16" "face.7" "face.21" "face.24" "face.37"
\
  "face.40" real
volume create stitch "face.17" "face.8" "face.24" "face.27" "face.38"
\
  "face.41" real
volume create stitch "face.18" "face.9" "face.30" "face.27" "face.39"
\
  "face.42" real
//////////////ZZ1
edge picklink "edge.61" "edge.57" "edge.53" "edge.49" "edge.62"
"edge.58" \
  "edge.54" "edge.50" "edge.63" "edge.59" "edge.55" "edge.51"
"edge.64" \
  "edge.60" "edge.56" "edge.52"
edge mesh "edge.64" "edge.60" "edge.56" "edge.52" "edge.61" "edge.57"
\
  "edge.53" "edge.49" "edge.62" "edge.58" "edge.54" "edge.50"
"edge.63" \
```

```
  "edge.59" "edge.55" "edge.51" successive ratio1 1.10 intervals 100
//////////////////YY1
edge picklink "edge.10" "edge.7" "edge.4" "edge.1" "edge.34"
"edge.31" \
  "edge.28" "edge.25"
edge mesh "edge.25" "edge.28" "edge.31" "edge.34" "edge.1" "edge.4"
"edge.7" \
  "edge.10" successive ratio1 1 ratio2 1 intervals 9
//////////YY2
edge picklink "edge.11" "edge.8" "edge.5" "edge.2" "edge.35"
"edge.32" \
  "edge.29" "edge.26"
edge mesh "edge.26" "edge.29" "edge.32" "edge.35" "edge.2" "edge.5"
"edge.8" \
  "edge.11" successive ratio1 1.0 ratio2 1.0 intervals 18
////////YY3
edge picklink "edge.12" "edge.9" "edge.6" "edge.3" "edge.36"
"edge.33" \
  "edge.30" "edge.27"
edge mesh "edge.27" "edge.12" "edge.9" "edge.6" "edge.3" "edge.36"
"edge.33" \
  "edge.30" successive ratio1 1 ratio2 1 intervals 9
//////XX1
edge picklink "edge.39" "edge.37" "edge.42" "edge.40" "edge.45"
"edge.43" \
  "edge.48" "edge.46" "edge.15" "edge.13" "edge.18" "edge.16"
"edge.21" \
  "edge.24" "edge.19" "edge.22"
edge mesh "edge.22" "edge.19" "edge.24" "edge.21" "edge.16" "edge.18"
\
  "edge.13" "edge.15" "edge.46" "edge.48" "edge.43" "edge.45"
"edge.40" \
  "edge.42" "edge.37" "edge.39" successive ratio1 1 ratio2 1
intervals 3
///////XX2
edge picklink "edge.38" "edge.41" "edge.44" "edge.47" "edge.14"
"edge.17" \
  "edge.20" "edge.23"
edge mesh "edge.23" "edge.20" "edge.17" "edge.14" "edge.47" "edge.44"
\
  "edge.41" "edge.38" successive ratio1 1.00 ratio2 1.00 intervals 10
///////
volume mesh "volume.1" "volume.2" "volume.3" "volume.4" "volume.5"
"volume.6" \
  "volume.7" "volume.8" "volume.9" map size 1
//////
physics create "inlet" btype "PRESSURE_INLET" face "face.5"
physics create "Inlet_wall" btype "WALL" face "face.3" "face.2"
"face.1" \
  "face.4" "face.7" "face.8" "face.9" "face.6"
physics create "Top_Wall" btype "WALL" face "face.31" "face.32"
"face.33"
physics create "bottomH_wall" btype "WALL" face "face.40" "face.41"
"face.42"
physics create "SYMM1" btype "SYMMETRY" face "face.19" "face.20"
"face.21"
physics create "SYMM2" btype "SYMMETRY" face "face.28" "face.29"
"face.30"
physics create "Inner_wall" btype "WALL" face "face.35" "face.23"
"face.26" \
  "face.38"
```

```
physics create "outlet" btype "PRESSURE_OUTLET" face "face.14"
physics create "outlet_wall" btype "WALL" face "face.10" "face.11"
"face.12" \
  "face.15" "face.18" "face.17" "face.16" "face.13"
physics create "fluid1" ctype "FLUID" volume "volume.5"
physics create "solid1" ctype "SOLID" volume "volume.1" "volume.2"
"volume.3" \
  "volume.6" "volume.9" "volume.8" "volume.7" "volume.4"
export fluent5 "Micro1222.msh"
abort
```

B.2 DOUBLE ROW MICROPIN-FIN HEAT SINK JOURNAL FILE

```
/Parameter
/////
$D1 = $X1
$D2 = $X2
$H1 = $X3
$H2 = $X4
$S = $X5
$L = 1
$L1 = 0.6
$HT = 1
////
$X1 = 0.5*($L1 - $D2)
$X2 = 0.5*($L1 - $D1)
$X3 = 0.5*$L1
$X4 = $X3 + 0.5*$D1
$X5 = $X3 + 0.5*$D2
$X6 = $L1
$Y1 = $H1
$Y2 = $H2
$Y3 = $HT
///
$Z1 = $S/2
$Z2 = ($S/2) + 0.5*$D1
$Z3 = ($S/2) + $D1
$Z4 = $S + $Z3
$Z5 = $Z4 + 0.5*$D2
$Z6 = $Z4 + $D2
$Z7 = $L
//
//
vertex create coordinates 0   0   0
vertex create coordinates 0   $Y3 0
vertex create coordinates $X6 0   0
vertex create coordinates $X6 $Y3 0
///
vertex create coordinates $X3 0   $Z1
vertex create coordinates $X3 $Y1 $Z1
vertex create coordinates $X3 $Y3 $Z1
///
vertex create coordinates 0   0   $Z2
vertex create coordinates 0   $Y1 $Z2
vertex create coordinates 0   $Y3 $Z2
vertex create coordinates $X2 0   $Z2
vertex create coordinates $X2 $Y1 $Z2
vertex create coordinates $X2 $Y3 $Z2
vertex create coordinates $X3 0   $Z2
vertex create coordinates $X3 $Y1 $Z2
vertex create coordinates $X3 $Y3 $Z2
vertex create coordinates $X4 0   $Z2
vertex create coordinates $X4 $Y1 $Z2
vertex create coordinates $X4 $Y3 $Z2
vertex create coordinates $X6 0   $Z2
vertex create coordinates $X6 $Y1 $Z2
vertex create coordinates $X6 $Y3 $Z2
////
vertex create coordinates $X3 0   $Z3
vertex create coordinates $X3 $Y1 $Z3
vertex create coordinates $X3 $Y3 $Z3
```

```
////
vertex create coordinates $X3 0   $Z4
vertex create coordinates $X3 $Y2 $Z4
vertex create coordinates $X3 $Y3 $Z4
/////
vertex create coordinates 0   0   $Z5
vertex create coordinates 0   $Y2 $Z5
vertex create coordinates 0   $Y3 $Z5
vertex create coordinates $X1 0   $Z5
vertex create coordinates $X1 $Y2 $Z5
vertex create coordinates $X1 $Y3 $Z5
vertex create coordinates $X3 0   $Z5
vertex create coordinates $X3 $Y2 $Z5
vertex create coordinates $X3 $Y3 $Z5
vertex create coordinates $X5 0   $Z5
vertex create coordinates $X5 $Y2 $Z5
vertex create coordinates $X5 $Y3 $Z5
vertex create coordinates $X6 0   $Z5
vertex create coordinates $X6 $Y2 $Z5
vertex create coordinates $X6 $Y3 $Z5
/////
vertex create coordinates $X3 0   $Z6
vertex create coordinates $X3 $Y2 $Z6
vertex create coordinates $X3 $Y3 $Z6
////
vertex create coordinates 0   0   $Z7
vertex create coordinates 0   $Y3 $Z7
vertex create coordinates $X6 0   $Z7
vertex create coordinates $X6 $Y3 $Z7
////
edge create straight "vertex.1" "vertex.2"
edge create straight "vertex.1" "vertex.3"
edge create straight "vertex.2" "vertex.4"
edge create straight "vertex.4" "vertex.3"
edge create straight "vertex.48" "vertex.50"
edge create straight "vertex.47" "vertex.49"
edge create straight "vertex.48" "vertex.47"
edge create straight "vertex.50" "vertex.49"
edge create straight "vertex.3" "vertex.20" "vertex.41" "vertex.49"
edge create straight "vertex.1" "vertex.8" "vertex.29" "vertex.47"
edge create straight "vertex.2" "vertex.10" "vertex.31" "vertex.48"
edge create straight "vertex.4" "vertex.22" "vertex.43" "vertex.50"
edge create straight "vertex.43" "vertex.42" "vertex.41"
edge create straight "vertex.22" "vertex.21" "vertex.20"
edge create straight "vertex.31" "vertex.30" "vertex.29"
edge create straight "vertex.10" "vertex.9" "vertex.8"
edge create straight "vertex.46" "vertex.45" "vertex.44"
edge create straight "vertex.34" "vertex.33" "vertex.32"
edge create straight "vertex.28" "vertex.27" "vertex.26"
edge create straight "vertex.40" "vertex.39" "vertex.38"
edge create straight "vertex.7" "vertex.5" "vertex.6"
edge create straight "vertex.25" "vertex.24" "vertex.23"
edge create straight "vertex.13" "vertex.12" "vertex.11"
edge create straight "vertex.19" "vertex.18" "vertex.17"
edge create center2points "vertex.14" "vertex.5" "vertex.17" minarc
arc
edge create center2points "vertex.14" "vertex.17" "vertex.23" minarc
arc
edge create center2points "vertex.14" "vertex.23" "vertex.11" minarc
arc
```

```
edge create center2points "vertex.14" "vertex.11" "vertex.5" minarc
arc
edge create center2points "vertex.15" "vertex.6" "vertex.18" minarc
arc
edge create center2points "vertex.15" "vertex.18" "vertex.24" minarc
arc
edge create center2points "vertex.15" "vertex.24" "vertex.12" minarc
arc
edge create center2points "vertex.15" "vertex.12" "vertex.6" minarc
arc
edge create center2points "vertex.16" "vertex.7" "vertex.19" minarc
arc
edge create center2points "vertex.16" "vertex.19" "vertex.25" minarc
arc
edge create center2points "vertex.16" "vertex.25" "vertex.13" minarc
arc
edge create center2points "vertex.16" "vertex.13" "vertex.7" minarc
arc
edge create center2points "vertex.35" "vertex.26" "vertex.38" minarc
arc
edge create center2points "vertex.35" "vertex.38" "vertex.44" minarc
arc
edge create center2points "vertex.35" "vertex.44" "vertex.32" minarc
arc
edge create center2points "vertex.35" "vertex.32" "vertex.26" minarc
arc
edge create center2points "vertex.36" "vertex.27" "vertex.39" minarc
arc
edge create center2points "vertex.36" "vertex.39" "vertex.45" minarc
arc
edge create center2points "vertex.36" "vertex.45" "vertex.33" minarc
arc
edge create center2points "vertex.36" "vertex.33" "vertex.27" minarc
arc
edge create center2points "vertex.37" "vertex.28" "vertex.40" minarc
arc
edge create center2points "vertex.37" "vertex.40" "vertex.46" minarc
arc
edge create center2points "vertex.37" "vertex.46" "vertex.34" minarc
arc
edge create center2points "vertex.37" "vertex.34" "vertex.28" minarc
arc
edge create straight "vertex.20" "vertex.17"
edge create straight "vertex.11" "vertex.8"
edge create straight "vertex.41" "vertex.38"
edge create straight "vertex.32" "vertex.29"
edge create straight "vertex.39" "vertex.42"
edge create straight "vertex.33" "vertex.30"
edge create straight "vertex.34" "vertex.31"
edge create straight "vertex.40" "vertex.43"
edge create straight "vertex.12" "vertex.9"
edge create straight "vertex.18" "vertex.21"
edge create straight "vertex.19" "vertex.22"
edge create straight "vertex.13" "vertex.10"
face create wireframe "edge.3" "edge.1" "edge.2" "edge.4" real
face create wireframe "edge.7" "edge.5" "edge.8" "edge.6" real
face create wireframe "edge.17" "edge.25" "edge.26" "edge.14"
"edge.7" real
face create wireframe "edge.16" "edge.27" "edge.28" "edge.13"
"edge.26" \
  "edge.25" real
```

```
face create wireframe "edge.28" "edge.27" "edge.15" "edge.1"
"edge.12" real
face create wireframe "edge.8" "edge.20" "edge.21" "edge.22"
"edge.11" real
face create wireframe "edge.21" "edge.19" "edge.23" "edge.24"
"edge.10" \
  "edge.22" real
face create wireframe "edge.23" "edge.18" "edge.4" "edge.9" "edge.24"
real
face create wireframe "edge.76" "edge.66" "edge.67" "edge.75"
"edge.17" \
  "edge.5" "edge.20" real
face create wireframe "edge.75" "edge.16" "edge.80" "edge.55"
"edge.54" \
  "edge.79" "edge.19" "edge.76" "edge.65" "edge.68" real
face create wireframe "edge.80" "edge.15" "edge.3" "edge.18"
"edge.79" \
  "edge.53" "edge.56" real
face create wireframe "edge.14" "edge.72" "edge.59" "edge.58"
"edge.71" \
  "edge.11" "edge.6" real
face create wireframe "edge.72" "edge.13" "edge.70" "edge.47"
"edge.46" \
  "edge.69" "edge.10" "edge.71" "edge.57" "edge.60" real
face create wireframe "edge.70" "edge.12" "edge.2" "edge.9" "edge.69"
\
  "edge.45" "edge.48" real
face create wireframe "edge.73" "edge.36" "edge.22" "edge.71" real
face create wireframe "edge.36" "edge.62" "edge.30" "edge.58" real
face create wireframe "edge.59" "edge.30" "edge.63" "edge.32" real
face create wireframe "edge.32" "edge.74" "edge.26" "edge.72" real
face create wireframe "edge.21" "edge.76" "edge.35" "edge.73" real
face create wireframe "edge.62" "edge.35" "edge.66" "edge.29" real
face create wireframe "edge.63" "edge.29" "edge.67" "edge.31" real
face create wireframe "edge.31" "edge.75" "edge.25" "edge.74" real
face create wireframe "edge.36" "edge.61" "edge.34" "edge.57" real
face create wireframe "edge.60" "edge.32" "edge.64" "edge.34" real
face create wireframe "edge.64" "edge.31" "edge.68" "edge.33" real
face create wireframe "edge.67" "edge.68" "edge.65" "edge.66" real
face create wireframe "edge.63" "edge.64" "edge.61" "edge.62" real
face create wireframe "edge.60" "edge.57" "edge.58" "edge.59" real
face create wireframe "edge.69" "edge.24" "edge.78" "edge.44" real
face create wireframe "edge.44" "edge.50" "edge.40" "edge.46" real
face create wireframe "edge.47" "edge.40" "edge.51" "edge.42" real
face create wireframe "edge.70" "edge.28" "edge.77" "edge.42" real
face create wireframe "edge.78" "edge.23" "edge.79" "edge.43" real
face create wireframe "edge.43" "edge.50" "edge.39" "edge.54" real
face create wireframe "edge.51" "edge.39" "edge.55" "edge.41" real
face create wireframe "edge.77" "edge.41" "edge.80" "edge.27" real
edge delete "edge.37" lowertopology
edge delete "edge.38" lowertopology
edge create straight "vertex.5" "vertex.6" "vertex.7"
face create wireframe "edge.45" "edge.44" "edge.49" "edge.81" real
face create wireframe "edge.52" "edge.81" "edge.48" "edge.42" real
face create wireframe "edge.43" "edge.49" "edge.82" "edge.53" real
face create wireframe "edge.52" "edge.41" "edge.56" "edge.82" real
face create wireframe "edge.47" "edge.48" "edge.45" "edge.46" real
face create wireframe "edge.51" "edge.52" "edge.49" "edge.50" real
face create wireframe "edge.55" "edge.56" "edge.53" "edge.54" real
volume create stitch "face.28" "face.27" "face.16" "face.23"
"face.24" \
```

```
  "face.17" real
face create wireframe "edge.65" "edge.33" "edge.61" "edge.35" real
volume create stitch "face.44" "face.25" "face.21" "face.20"
"face.27" \
  "face.26" real
volume create stitch "face.41" "face.42" "face.37" "face.30"
"face.31" \
  "face.38" real
volume create stitch "face.43" "face.42" "face.35" "face.40"
"face.39" \
  "face.34" real
volume create stitch "face.8" "face.5" "face.1" "face.11" "face.14"
"face.29" \
  "face.33" "face.36" "face.32" "face.37" "face.39" "face.40"
"face.38" real
volume create stitch "face.36" "face.32" "face.33" "face.29"
"face.15" \
  "face.19" "face.22" "face.18" "face.25" "face.24" "face.44"
"face.23" \
  "face.35" "face.31" "face.34" "face.30" "face.4" "face.7" "face.13"
\
  "face.10" real
volume create stitch "face.18" "face.22" "face.21" "face.17"
"face.20" \
  "face.16" "face.19" "face.15" "face.6" "face.3" "face.9" "face.12"
"face.2" \
  real
/
face mesh "face.14" pave size 0.015
volume mesh "volume.5" cooper source "face.11" "face.14" size 0.015
/
face mesh "face.42" "face.41" pave size 0.015
volume mesh "volume.3" cooper source "face.42" "face.41" size 0.015
/
volume mesh "volume.4" cooper source "face.42" "face.43" size 0.015
/
/
face mesh "face.13" pave size 0.015
volume mesh "volume.6" cooper source "face.13" "face.10" size 0.015
face mesh "face.27" pave size 0.015
/
volume mesh "volume.1" cooper source "face.27" "face.28" size 0.015
volume mesh "volume.2" cooper source "face.27" "face.26" size 0.015
/
face mesh "face.12" pave size 0.015
volume mesh "volume.7" cooper source "face.9" "face.12" size 0.015
/
physics create "Inlet" btype "VELOCITY_INLET" face "face.1"
physics create "Outlet" btype "OUTFLOW" face "face.2"
physics create "Symmetry1" btype "SYMMETRY" face "face.5" "face.4"
"face.3"
physics create "Symmetry2" btype "SYMMETRY" face "face.8" "face.7"
"face.6"
physics create "Top_wall" btype "WALL" face "face.11" "face.10"
"face.9" \
  "face.26" "face.43"
physics create "Hot_Base" btype "WALL" face "face.14" "face.13"
"face.12" "face.28" "face.41"
physics create "Hot_Fins1" btype "WALL" face "face.30" "face.31" \
  "face.38" "face.37""face.42"
```

```
physics create "Hot_Fins2" btype "WALL" face "face.27" "face.17"
"face.16" \
  "face.23" "face.24"
physics create "fluid1" ctype "FLUID" volume "volume.5" "volume.4"
"volume.6" \
  "volume.2" "volume.7"
physics create "solids" ctype "SOLID" volume "volume.3" "volume.1"
export fluent5 "fin1.msh"
abort
```

B.3 TRIPLE ROW MICROPIN-FIN HEAT SINK JOURNAL FILE

```
/Parameter
/////
$D1 = $X1
$D2 = $X2
$D3 = $X3
$H1 = 0.65
$H2 = 0.65
$H3 = 0.65
$S = 0.05
$S1 = 0.05
$S2 = 0.05
$L = 1
$L1 = 0.6
$HT = 1
////
$XX1 = 0.5*($L1 - $D3)
$XX2 = 0.5*($L1 - $D2)
$XX3 = 0.5*($L1 - $D1)
$XX4 = 0.5*$L1
$XX5 = $XX4 + 0.5*$D1
$XX6 = $XX4 + 0.5*$D2
$XX7 = $XX4 + 0.5*$D3
$XX8 = $L1
$Y1 = $H1
$Y2 = $H2
$Y3 = $H3
$Y4 = $HT
///
$Z1 = $S
$Z2 = $S + 0.5*$D1
$Z3 = $S + $D1
$Z4 = $S1 + $Z3
$Z5 = $Z4 + 0.5*$D2
$Z6 = $Z4 + $D2
$Z7 = $S2 + $Z6
$Z8 = $Z7 + 0.5*$D3
$Z9 = $Z7 + $D3
$Z10 = $L
//
//
vertex create coordinates 0   0   0
vertex create coordinates 0   $Y4 0
vertex create coordinates $XX8 0   0
vertex create coordinates $XX8 $Y4 0
///
vertex create coordinates $XX4 0   $Z1
vertex create coordinates $XX4 $Y1 $Z1
vertex create coordinates $XX4 $Y4 $Z1
///
vertex create coordinates 0   0   $Z2
vertex create coordinates 0   $Y1 $Z2
vertex create coordinates 0   $Y4 $Z2
vertex create coordinates $XX3 0   $Z2
vertex create coordinates $XX3 $Y1 $Z2
vertex create coordinates $XX3 $Y4 $Z2
vertex create coordinates $XX4 0   $Z2
vertex create coordinates $XX4 $Y1 $Z2
```

```
vertex create coordinates $XX4 $Y4 $Z2
vertex create coordinates $XX5 0   $Z2
vertex create coordinates $XX5 $Y1 $Z2
vertex create coordinates $XX5 $Y4 $Z2
vertex create coordinates $XX8 0   $Z2
vertex create coordinates $XX8 $Y1 $Z2
vertex create coordinates $XX8 $Y4 $Z2
////
vertex create coordinates $XX4 0   $Z3
vertex create coordinates $XX4 $Y1 $Z3
vertex create coordinates $XX4 $Y4 $Z3
////
vertex create coordinates $XX4 0   $Z4
vertex create coordinates $XX4 $Y2 $Z4
vertex create coordinates $XX4 $Y4 $Z4
/////
vertex create coordinates 0   0   $Z5
vertex create coordinates 0   $Y2 $Z5
vertex create coordinates 0   $Y4 $Z5
vertex create coordinates $XX2 0   $Z5
vertex create coordinates $XX2 $Y2 $Z5
vertex create coordinates $XX2 $Y4 $Z5
vertex create coordinates $XX4 0   $Z5
vertex create coordinates $XX4 $Y2 $Z5
vertex create coordinates $XX4 $Y4 $Z5
vertex create coordinates $XX6 0   $Z5
vertex create coordinates $XX6 $Y2 $Z5
vertex create coordinates $XX6 $Y4 $Z5
vertex create coordinates $XX8 0   $Z5
vertex create coordinates $XX8 $Y2 $Z5
vertex create coordinates $XX8 $Y4 $Z5
/////
vertex create coordinates $XX4 0   $Z6
vertex create coordinates $XX4 $Y2 $Z6
vertex create coordinates $XX4 $Y4 $Z6
////
vertex create coordinates $XX4 0   $Z7
vertex create coordinates $XX4 $Y3 $Z7
vertex create coordinates $XX4 $Y4 $Z7
/////
vertex create coordinates 0   0   $Z8
vertex create coordinates 0   $Y3 $Z8
vertex create coordinates 0   $Y4 $Z8
vertex create coordinates $XX1 0   $Z8
vertex create coordinates $XX1 $Y3 $Z8
vertex create coordinates $XX1 $Y4 $Z8
vertex create coordinates $XX4 0   $Z8
vertex create coordinates $XX4 $Y3 $Z8
vertex create coordinates $XX4 $Y4 $Z8
vertex create coordinates $XX7 0   $Z8
vertex create coordinates $XX7 $Y3 $Z8
vertex create coordinates $XX7 $Y4 $Z8
vertex create coordinates $XX8 0   $Z8
vertex create coordinates $XX8 $Y3 $Z8
vertex create coordinates $XX8 $Y4 $Z8
/////
vertex create coordinates $XX4 0   $Z9
vertex create coordinates $XX4 $Y3 $Z9
vertex create coordinates $XX4 $Y4 $Z9
////
vertex create coordinates 0   0   $Z10
```

```
vertex create coordinates 0   $Y4 $Z10
vertex create coordinates $XX8 0   $Z10
vertex create coordinates $XX8 $Y4 $Z10
////
edge create straight "vertex.1" "vertex.2"
edge create straight "vertex.1" "vertex.3"
edge create straight "vertex.2" "vertex.4"
edge create straight "vertex.4" "vertex.3"
edge create straight "vertex.69" "vertex.71"
edge create straight "vertex.68" "vertex.70"
edge create straight "vertex.69" "vertex.68"
edge create straight "vertex.71" "vertex.70"
edge create straight "vertex.3" "vertex.20" "vertex.41" "vertex.62"
"vertex.70"
edge create straight "vertex.1" "vertex.8" "vertex.29" "vertex.50"
"vertex.68"
edge create straight "vertex.2" "vertex.10" "vertex.31" "vertex.52"
"vertex.69"
edge create straight "vertex.4" "vertex.22" "vertex.43" "vertex.64"
"vertex.71"
edge create straight "vertex.62" "vertex.63" "vertex.64"
edge create straight "vertex.41" "vertex.42" "vertex.43"
edge create straight "vertex.22" "vertex.21" "vertex.20"
edge create straight "vertex.50" "vertex.51" "vertex.52"
edge create straight "vertex.29" "vertex.30" "vertex.31"
edge create straight "vertex.10" "vertex.9" "vertex.8"
edge create straight "vertex.65" "vertex.66" "vertex.67"
edge create straight "vertex.59" "vertex.60" "vertex.61"
edge create straight "vertex.47" "vertex.48" "vertex.49"
edge create straight "vertex.53" "vertex.54" "vertex.55"
edge create straight "vertex.46" "vertex.45" "vertex.44"
edge create straight "vertex.34" "vertex.33" "vertex.32"
edge create straight "vertex.28" "vertex.27" "vertex.26"
edge create straight "vertex.40" "vertex.39" "vertex.38"
edge create straight "vertex.7" "vertex.6" "vertex.5"
edge create straight "vertex.25" "vertex.24" "vertex.23"
edge create straight "vertex.13" "vertex.12" "vertex.11"
edge create straight "vertex.19" "vertex.18" "vertex.17"
edge create center2points "vertex.14" "vertex.5" "vertex.17" minarc
arc
edge create center2points "vertex.14" "vertex.17" "vertex.23" minarc
arc
edge create center2points "vertex.14" "vertex.23" "vertex.11" minarc
arc
edge create center2points "vertex.14" "vertex.11" "vertex.5" minarc
arc
edge create center2points "vertex.15" "vertex.6" "vertex.18" minarc
arc
edge create center2points "vertex.15" "vertex.18" "vertex.24" minarc
arc
edge create center2points "vertex.15" "vertex.24" "vertex.12" minarc
arc
edge create center2points "vertex.15" "vertex.12" "vertex.6" minarc
arc
edge create center2points "vertex.16" "vertex.7" "vertex.19" minarc
arc
edge create center2points "vertex.16" "vertex.19" "vertex.25" minarc
arc
edge create center2points "vertex.16" "vertex.25" "vertex.13" minarc
arc
```

```
edge create center2points "vertex.16" "vertex.13" "vertex.7" minarc
arc
edge create center2points "vertex.35" "vertex.26" "vertex.38" minarc
arc
edge create center2points "vertex.35" "vertex.38" "vertex.44" minarc
arc
edge create center2points "vertex.35" "vertex.44" "vertex.32" minarc
arc
edge create center2points "vertex.35" "vertex.32" "vertex.26" minarc
arc
edge create center2points "vertex.36" "vertex.27" "vertex.39" minarc
arc
edge create center2points "vertex.36" "vertex.39" "vertex.45" minarc
arc
edge create center2points "vertex.36" "vertex.45" "vertex.33" minarc
arc
edge create center2points "vertex.36" "vertex.33" "vertex.27" minarc
arc
edge create center2points "vertex.37" "vertex.28" "vertex.40" minarc
arc
edge create center2points "vertex.37" "vertex.40" "vertex.46" minarc
arc
edge create center2points "vertex.37" "vertex.46" "vertex.34" minarc
arc
edge create center2points "vertex.37" "vertex.34" "vertex.28" minarc
arc
edge create center2points "vertex.56" "vertex.65" "vertex.53" minarc
arc
edge create center2points "vertex.56" "vertex.53" "vertex.47" minarc
arc
edge create center2points "vertex.56" "vertex.47" "vertex.59" minarc
arc
edge create center2points "vertex.56" "vertex.59" "vertex.65" minarc
arc
edge create center2points "vertex.57" "vertex.66" "vertex.54" minarc
arc
edge create center2points "vertex.57" "vertex.54" "vertex.48" minarc
arc
edge create center2points "vertex.57" "vertex.48" "vertex.60" minarc
arc
edge create center2points "vertex.57" "vertex.60" "vertex.66" minarc
arc
edge create center2points "vertex.58" "vertex.67" "vertex.55" minarc
arc
edge create center2points "vertex.58" "vertex.55" "vertex.49" minarc
arc
edge create center2points "vertex.58" "vertex.49" "vertex.61" minarc
arc
edge create center2points "vertex.58" "vertex.61" "vertex.67" minarc
arc
edge create straight "vertex.20" "vertex.17"
edge create straight "vertex.11" "vertex.8"
edge create straight "vertex.41" "vertex.38"
edge create straight "vertex.32" "vertex.29"
edge create straight "vertex.39" "vertex.42"
edge create straight "vertex.33" "vertex.30"
edge create straight "vertex.34" "vertex.31"
edge create straight "vertex.40" "vertex.43"
edge create straight "vertex.12" "vertex.9"
edge create straight "vertex.18" "vertex.21"
edge create straight "vertex.19" "vertex.22"
```

```
edge create straight "vertex.13" "vertex.10"
edge create straight "vertex.50" "vertex.53"
edge create straight "vertex.51" "vertex.54"
edge create straight "vertex.52" "vertex.55"
edge create straight "vertex.59" "vertex.62"
edge create straight "vertex.60" "vertex.63"
edge create straight "vertex.61" "vertex.64"
face create wireframe "edge.3" "edge.1" "edge.2" "edge.4" real
face create wireframe "edge.7" "edge.5" "edge.8" "edge.6" real
face create wireframe "edge.7" "edge.20" "edge.31" "edge.16"
"edge.32" real
face create wireframe "edge.19" "edge.31" "edge.33" "edge.15"
"edge.34" \
  "edge.32" real
face create wireframe "edge.34" "edge.33" "edge.14" "edge.35"
"edge.36" \
  "edge.18" real
face create wireframe "edge.17" "edge.1" "edge.13" "edge.35"
"edge.36" real
face create wireframe "edge.8" "edge.24" "edge.25" "edge.26"
"edge.12" real
face create wireframe "edge.25" "edge.26" "edge.23" "edge.28"
"edge.27" \
  "edge.11" real
face create wireframe "edge.28" "edge.27" "edge.10" "edge.30"
"edge.29" \
  "edge.22" real
face create wireframe "edge.29" "edge.30" "edge.9" "edge.4" "edge.21"
real
face create wireframe "edge.20" "edge.5" "edge.24" "edge.114"
"edge.96" \
  "edge.93" "edge.111" real
face create wireframe "edge.19" "edge.111" "edge.94" "edge.95"
"edge.114" \
  "edge.23" "edge.104" "edge.82" "edge.83" "edge.103" real
face create wireframe "edge.18" "edge.103" "edge.84" "edge.81"
"edge.104" \
  "edge.22" "edge.107" "edge.70" "edge.71" "edge.108" real
face create wireframe "edge.17" "edge.108" "edge.72" "edge.69"
"edge.107" \
  "edge.21" "edge.3" real
face create wireframe "edge.16" "edge.6" "edge.12" "edge.112"
"edge.88" \
  "edge.85" "edge.109" real
face create wireframe "edge.15" "edge.109" "edge.86" "edge.87"
"edge.112" \
  "edge.11" "edge.99" "edge.74" "edge.75" "edge.100" real
face create wireframe "edge.14" "edge.100" "edge.76" "edge.73"
"edge.99" \
  "edge.10" "edge.97" "edge.62" "edge.63" "edge.98" real
face create wireframe "edge.13" "edge.98" "edge.64" "edge.61"
"edge.97" \
  "edge.9" "edge.2" real
face create wireframe "edge.112" "edge.25" "edge.113" "edge.39" real
face create wireframe "edge.39" "edge.88" "edge.37" "edge.92" real
face create wireframe "edge.85" "edge.43" "edge.37" "edge.89" real
face create wireframe "edge.109" "edge.43" "edge.110" "edge.31" real
face create wireframe "edge.113" "edge.26" "edge.114" "edge.40" real
face create wireframe "edge.40" "edge.96" "edge.92" "edge.38" real
face create wireframe "edge.38" "edge.93" "edge.44" "edge.89" real
face create wireframe "edge.32" "edge.110" "edge.111" "edge.44" real
```

```
face create wireframe "edge.87" "edge.39" "edge.41" "edge.91" real
face create wireframe "edge.86" "edge.41" "edge.43" "edge.90" real
face create wireframe "edge.44" "edge.90" "edge.94" "edge.42" real
face create wireframe "edge.42" "edge.95" "edge.91" "edge.40" real
face create wireframe "edge.94" "edge.93" "edge.95" "edge.96" real
face create wireframe "edge.89" "edge.90" "edge.91" "edge.92" real
face create wireframe "edge.85" "edge.86" "edge.87" "edge.88" real
face create wireframe "edge.99" "edge.27" "edge.52" "edge.101" real
face create wireframe "edge.52" "edge.74" "edge.46" "edge.78" real
face create wireframe "edge.79" "edge.46" "edge.75" "edge.48" real
face create wireframe "edge.33" "edge.100" "edge.48" "edge.102" real
face create wireframe "edge.101" "edge.28" "edge.104" "edge.51" real
face create wireframe "edge.51" "edge.82" "edge.45" "edge.78" real
face create wireframe "edge.83" "edge.45" "edge.47" "edge.79" real
face create wireframe "edge.103" "edge.34" "edge.102" "edge.47" real
face create wireframe "edge.52" "edge.73" "edge.77" "edge.50" real
face create wireframe "edge.80" "edge.48" "edge.76" "edge.50" real
face create wireframe "edge.84" "edge.49" "edge.80" "edge.47" real
face create wireframe "edge.81" "edge.51" "edge.77" "edge.49" real
face create wireframe "edge.81" "edge.82" "edge.83" "edge.84" real
face create wireframe "edge.77" "edge.78" "edge.79" "edge.80" real
face create wireframe "edge.73" "edge.74" "edge.75" "edge.76" real
face create wireframe "edge.30" "edge.97" "edge.60" "edge.106" real
face create wireframe "edge.62" "edge.60" "edge.66" "edge.56" real
face create wireframe "edge.63" "edge.58" "edge.67" "edge.56" real
face create wireframe "edge.36" "edge.98" "edge.105" "edge.58" real
face create wireframe "edge.106" "edge.29" "edge.107" "edge.59" real
face create wireframe "edge.70" "edge.59" "edge.66" "edge.55" real
face create wireframe "edge.57" "edge.71" "edge.55" "edge.67" real
face create wireframe "edge.108" "edge.57" "edge.35" "edge.105" real
face create wireframe "edge.54" "edge.65" "edge.60" "edge.61" real
face create wireframe "edge.68" "edge.58" "edge.64" "edge.54" real
face create wireframe "edge.72" "edge.57" "edge.68" "edge.53" real
face create wireframe "edge.69" "edge.53" "edge.65" "edge.59" real
face create wireframe "edge.69" "edge.70" "edge.71" "edge.72" real
face create wireframe "edge.65" "edge.66" "edge.67" "edge.68" real
face create wireframe "edge.61" "edge.62" "edge.63" "edge.64" real
volume create stitch "face.20" "face.21" "face.28" "face.27"
"face.32" \
  "face.33" real
volume create stitch "face.25" "face.31" "face.32" "face.24"
"face.30" \
  "face.29" real
volume create stitch "face.35" "face.36" "face.42" "face.43"
"face.47" \
  "face.48" real
volume create stitch "face.47" "face.44" "face.45" "face.39"
"face.40" \
  "face.46" real
volume create stitch "face.50" "face.51" "face.57" "face.58"
"face.62" \
  "face.63" real
volume create stitch "face.62" "face.54" "face.55" "face.59"
"face.60" \
  "face.61" real
volume create stitch "face.1" "face.14" "face.10" "face.53" "face.49"
"face.6" \
  "face.56" "face.18" "face.60" "face.59" "face.58" "face.57"
"face.52" real
volume create stitch "face.5" "face.13" "face.56" "face.41" "face.9"
\
```

```
  "face.49" "face.53" "face.17" "face.45" "face.44" "face.37"
"face.43" \
  "face.42" "face.34" "face.38" "face.54" "face.55" "face.50"
"face.51" \
  "face.52" real
volume create stitch "face.4" "face.12" "face.26" "face.22" "face.23"
\
  "face.30" "face.27" "face.16" "face.35" "face.34" "face.39"
"face.8" \
  "face.38" "face.37" "face.41" "face.28" "face.36" "face.40"
"face.29" \
  "face.19" real
volume create stitch "face.2" "face.11" "face.3" "face.7" "face.15"
"face.19" \
  "face.20" "face.21" "face.24" "face.25" "face.26" "face.22"
"face.23" real
/
face mesh "face.18" pave size 0.015
volume mesh "volume.7" cooper source "face.18" "face.14" size 0.015
face mesh "face.62" "face.63" pave size 0.015
volume mesh "volume.5" cooper source "face.62" "face.63" size 0.015
volume mesh "volume.6" cooper source "face.62" "face.61" size 0.015
/
face mesh "face.17" pave size 0.015
volume mesh "volume.8" cooper source "face.17" "face.13" size 0.015
face mesh "face.47" "face.48" pave size 0.015
volume mesh "volume.3" cooper source "face.47" "face.48" size 0.015
volume mesh "volume.4" cooper source "face.47" "face.46" size 0.015
/
face mesh "face.16" pave size 0.015
volume mesh "volume.9" cooper source "face.16" "face.12" size 0.015
face mesh "face.32" "face.33" pave size 0.015
volume mesh "volume.1" cooper source "face.32" "face.33" size 0.015
volume mesh "volume.2" cooper source "face.32" "face.31" size 0.015
/
face mesh "face.15" pave size 0.015
volume mesh "volume.10" cooper source "face.15" "face.11" size 0.015
/
physics create "Inlet" btype "VELOCITY_INLET" face "face.1"
physics create "Outlet" btype "OUTFLOW" face "face.2"
physics create "Symmetry1" btype "SYMMETRY" face "face.6" "face.5"
"face.4" "face.3"
physics create "Symmetry2" btype "SYMMETRY" face "face.10" "face.9"
"face.8" "face.7"
physics create "Top_wall" btype "WALL" face "face.11" "face.12"
"face.13" \
  "face.14" "face.31" "face.46" "face.61"
physics create "Hot_Base" btype "WALL" face "face.15" "face.16"
"face.17" "face.18" "face.33" "face.48" "face.63"
physics create "Hot_Fins1" btype "WALL" face "face.50" "face.51" \
  "face.57" "face.58""face.62"
physics create "Hot_Fins2" btype "WALL" face "face.35" "face.36"
"face.42" \
  "face.43" "face.47"
physics create "Hot_Fins3" btype "WALL" face "face.20" "face.21"
"face.27" \
  "face.28" "face.32"
physics create "fluid1" ctype "FLUID" volume "volume.7" "volume.6"
"volume.8" \
  "volume.4" "volume.9" "volume.2" "volume.10"
```

```
physics create "solids" ctype "SOLID" volume "volume.3" "volume.1"
"volume.5"
export fluent5 "fin2.msh"
abort
```

C

APPENDIX C: FLUENT JOURNAL FILE FOR NUMERICAL SIMULATION OF MICRO HEAT SINK

```
;; Read Mesh and Scale Mesh
file/set-batch-options yes yes yes no
file/read-case fin2.msh
grid/scale 0.001 0.001 0.001
;; Read Boundary Conditions
file/read-bc flow2r5
;; Define Models and Units
define/models/energy yes no no no yes
define/models/viscous/laminar yes
define/units temperature c
;; Monitors
solve/monitors/residual/plot yes
solve/monitors/residual/print yes
solve/monitors/residual/convergence-criteria 1e-3 5e-4 5e-4 5e-4 1e-7
;; Initialize and Solve
solve/initialize/compute-defaults all-zones
solve/initialize/initialize-flow
solve/iterate 55
;; Post Processing
report/fluxes/heat-transfer no hot_base hot_base:001 hot_fins1
hot_fins2 hot_fins3 () yes heat_tran.dta no yes
;; Finalizing
file/write-case-data Fin_data1.cas.gz
exit
```