Towards Improved Understanding and Modelling of Compact Heat Exchangers

by

Andrew Buckrell

A Thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Mechanical Engineering

Waterloo, Ontario, Canada, 2021

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:
Dr. James Cotton
Professor,
Department of Mechanical Engineering, McMaster University

Supervisor:
Dr. Fue-Sang Lien
Professor
Department of Mechanical and Mechatronics Engineering, University of Waterloo

Internal Member:
Dr. Sean Peterson
Professor
Department of Mechanical and Mechatronics Engineering, University of Waterloo

Internal Member:
Dr. Chao Tan
Professor
Department of Mechanical and Mechatronics Engineering, University of Waterloo

Internal-External Member:
Dr. Luis Ricardez Sandoval
Associate Professor
Department of Chemical Engineering, University of Waterloo

Author's Declaration

Abstract

The present work pursues the ability to increase the fundamental understanding of and to improve numerical modelling capabilities for Compact Heat Exchangers (CHE). This goal is achieved by numerically and experimentally studying the small scales of flow, then incorporating these results into a novel reduced order model (ROM) of a full heat exchanger. The result is a numerically efficient modelling approach that significantly improves numerical modelling of CHE's.

The heat transfer enhancement surface of focus, the turbulizer, is studied in great detail with high resolution Computational Fluid Dynamics (CFD), using experimental flow visualisation and Laser Doppler Velocimetry (LDV) measurements to validate the results. The modelling process explores a variety of turbulence models and simulation methodologies, finding that a Large Eddy Simulation (LES) model with several modifications to the turbulizer geometry to replicate manufacturing process pressure drop predictions within 7% of experimental results and heat transfer within 15% of experimental results. Excellent correlation is also observed with predictions of transition to unsteady flow. Flow visualisation provides excellent correlation with predicted flow patterns at low Re. The validated turbulizer model is used to investigate flow conditions through a wide range of Re and flow incidence angles, which have not been previously studied.

Construction of the reduced order model leverages data obtained during the detailed simulation of the turbulizer under a variety of flow conditions, mapping the appropriate Nu and f_D to a porous media heat transfer framework. This framework is used to enforce the heat transfer and pressure drop calculated based on the detailed modelling phase. Model lookup performance is investigated using both an artificial neural network (ANN) and bi-linear interpolation. The ANN approach provides the best overall performance.

Implementation of the ROM and turbulizer flow data is undertaken within the framework of STAR-CCM+, using *field functions* and *user defined code* to interact with the proposed model. Heat transfer is validated against experimental test results of a heat exchanger design which has previously been problematic for analytical models to accommodate. The results indicate an approximate halving of the error in pressure drop and heat transfer predictions made by numerical and analytical models, respectively. This indicates that the proposed novel ROM methodology provides a significant increase in the numerical predictive capabilities of complex heat exchanger models under a wide variety of flow conditions.

Acknowledgements

I would like to thank my supervisor, Professor Fue-Sang Lien, for his guidance and support throughout the extended course of my research. I would also like to thank Dr. Michael Bardeleben for his continued support and mentorship of my professional and academic development, and Dr. Ben Kenney for his support and guidance through the development of my research.

Thank you as well to my committee members, Dr. Jim Cotton, Dr. Sean Peterson, Dr. Chao Tan and Dr. Luis Ricardez Sandoval, for their input during my research and suggestions for my thesis.

I would also like to thank my family and friends for their continued patience and support throughout the extended process of completing my Ph.D. research. Without this support, I would never have been able to make it to completion!

Contents

List of Figures

List of Tables

Nomenclature

Roman Symbols

B	Body force
C	Wilson coefficient, heat capacity rate
c_p	Specific heat capacity at constant pressure
f	Fanning, Darcy friction factors
h	Heat transfer coefficient
j	Colburn j factor
k	Thermal conductivity, turbulent kinetic energy
L	Flow length scale
m	Reynolds number exponent
n	Prandtl number exponent
p	Pressure
q	Heat transfer
T	Temperature
u	Velocity
UA	Overall heat transfer coefficient multiplied by area

Greek Symbols

ε	Turbulent eddy dissipation, effectiveness
κ	Viscous component, Darcy law (also Darcy-Forchheimer Law)
κ_1	Inertial Component, Darcy-Forchheimer Law
μ	Dynamic viscosity
ρ	Density
τ	Shear stress, residual stress tensor
ψ	Generic scalar

Non-dimensional Numbers

Pr	Prandtl Number
Re	Reynolds number
Sc	Schmidt number
St	Stanton Number
Nu	Nusselt number

Subscripts and Superscripts

eff	effective
i, j	Einstein notation, directional component or fluid index
in	Average inlet value
min	Minimum value
max	Maximum value
out	Average outlet value
w	Wall
$1, 2$	Heat transfer medium 1 or 2

Other Notation

$\overline{\psi}$	Reynolds average of ψ
ψ'	Fluctuation about mean

Acronyms

ANN	Artificial Neural Network
ATF	Automatic Transmission Fluid
CAD	Computer Aided Design
CHE	Compact Heat Exchanger
CFD	Computational Fluids Dynamics
DNS	Direct Numerical Simulation
HPD	High Pressure Drop
JCGM	Joint Committee for Guides in Metrology
LES	Large Eddy Simulation
LMTD	Logarithmic Mean Temperature Difference
LPD	Low Pressure Drop
NTU	Number of Transfer Units
OSF	Offset Strip Fin
PIV	Particle Image Velocimetry
RANS	Reynolds Averaged Navier-Stokes
RI	Refractive Index
ROM	Reduced Order Modelling
RSS	Root Sum Square
SGS	Subgrid Scale
TKE	Turbulent Kinetic Energy

Chapter 1

Introduction

The role of heat transfer in the automotive industry has been progressing steadily forward to aid in the improved power density and efficiency of traditional internal combustion engines. This has resulted in the application of heat exchangers to tasks ranging from improving engine warm-up times/reducing emissions via heating coolant or gear oil, to cooling compressed air in turbocharged or supercharged engine applications, to the more traditional radiators or oil coolers used to keep the engine operating within its design load conditions. With the pursuit of power density and downsizing of engines, the need to provide heat exchangers that have a smaller size and weight but improved heat transfer and pressure drop has forced the automotive industry to search for new design and optimisation tools. Furthermore, the recent proliferation of electric vehicles has spurred on the development of large surface area, but low temperature differential battery thermal management systems, as well as extremely small, high power density power electronics coolers. Both of these styles of heat exchangers require significant design and development to ensure proper operation under their intended operating conditions.

The march towards improved heat exchangers, and the development of many key tools that are still in use today, began in earnest in the 1950s. One of the key works on heat exchanger development of the period was published by Kays and London [1] which provided a great volume of information on the performance of various heat transfer surfaces and modelling techniques. Publication of this text drove forward the understanding and development of modern heat exchangers and provided a stepping stone for design engineers, which spawned many other works that would further expand on the subject.

Although there has been a more recent focus on design tools that can be integrated into a discretised numerical model, such as CFD (computational fluid dynamics), there still lacks a predictive model that can be used early in the design phase to assess and calculate heat transfer performance, before any physical test data are available for use or validation. The current state-of-the-art CFD models are typically limited to a fixed thermal resistance interface, isotropic porous media assumptions, or tabulated test data based on temperatures and flow rates. None of these provide the ability to determine the effectiveness of new, and complex heat transfer enhancement surfaces and largely rely on validation with known results to use for any predictive tasking. The non-linear, highly anisotropic performance of typical heat transfer enhancement surfaces precludes the confident use of these models for heat exchanger development.

Given the existing model limitations, it is the focus of this work to develop and demonstrate efficacy of a new modelling technique which is able to incorporate arbitrary flow conditions on complex heat transfer enhancement surfaces, without making *a priori* assumptions based on flow direction, Reynolds number (Re) or fluid properties (Pr, μ, ρ, etc.). Prediction of both heat transfer (local Nu) and friction factor (f_D) are calculated directly and in a simplified manner, and integrated directly into the CFD calculations, without the need for a noticeable increase in computational effort beyond existing models. This allows for simplification of *global* CFD models, i.e. incorporation of the entire heat exchanger, for a single simulation of both the overall and local heat transfer as well as pressure drop.

1.1 Compact Heat Exchangers

The concept of the compact heat exchanger revolves around the idea of using specially designed heat transfer surfaces to yield a higher level of heat transfer for a given package volume. In many automotive applications, minimising volume and weight of the heat exchanger is critical in providing an acceptable design. Obtaining this high performance density relies on the utilisation of specific heat transfer enhancement surfaces: surfaces that are designed to extract and transfer as much heat as possible between the two fluid media, or occasionally even between a fluid and a solid, such as in battery cooling or power electronics applications. These heat transfer enhancement surfaces have two primary objectives: provide as much surface area as possible into the CHE and to provide a strong mixing environment which suitably enhances heat transfer for a given surface area. Inevitably, introduction of increased surface area and more intense mixing of the fluid yields a higher pressure drop, which is typically undesirable. By considering the properties of both fluid streams in the heat exchanger, the most appropriate heat transfer enhancement surfaces can be selected from a variety of available geometries which best balance the requirements for low pressure drop and high heat transfer. An example of several different heat transfer enhancement surfaces is shown in Fig. 1.1. Balancing these objectives, namely maximising heat transfer while minimising pressure drop, along with considerations of manufacturing cost and durability, define the challenge of CHE design. Highlighting and isolating an "optimal" design relies on the use of design tools which can accurately provide an *a priori* prediction of heat transfer and pressure drop to aid in the early stages of design and development without requiring expensive and time consuming prototyping processes.

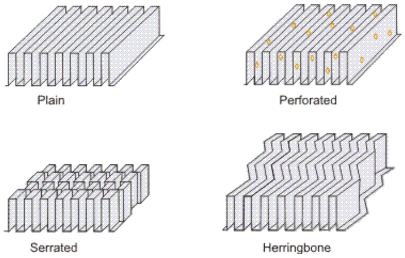


Figure 1.1: Plate fin surface types [2]

As a traditionally accepted metric for comparison, the "compactness" of a CHE can be defined by comparing the ratio of surface area to total heat exchanger packaging volume. This is illustrated in Fig. 1.2, with designs on the right hand side illustrating increased compactness. In this figure, it can also be seen that increasing surface area density subsequently results in the decrease of the hydraulic diameter. This reduction of hydraulic diameter brings about both benefits and drawbacks. Reducing the hydraulic diameter tends to increase pressure drop of the heat exchanger, while shifting the flow from the fully developed turbulent regime towards the laminar regime by reducing flow Re, which subsequently decreases heat transfer performance. Thus, increased compactness comes at a price.

More recent applications have called for perhaps a slight revision to the traditional definition of compactness. While the heat transfer requirements certainly continue to play a large role, there is increased emphasis on the ability of a heat exchanger to conform to non-traditional form factors, often weaving between other "higher priority" items in vehicle design, requiring more flexibility and development efforts in order to meet requirements. Furthermore, applications are now often requiring a reduced or tailored thermal resistance profile for electronics thermal management, which typically have a fixed thermal loading profile and are more dependent on the ability to manage temperature distribution and thermal stress profiles. Thus, compactness can be viewed in more holistic measures than purely the ratio of heat transfer to volume, but rather desired performance to available performance to resource consumption, where the resource could be available volume, mass, etc.

In the search for improved compactness, it was noticed that the performance density of a heat exchanger, in comparison to the traditional shell-and-tube design, could often be improved by stacking alternating fluid channels on top of each other to form a heat exchanger. This stacked form of heat exchanger is shown in a cutaway in Fig. 1.3. In this configuration, two separate fluid volumes are stacked in alternating layers, separated by thin plates, to provide a maximum amount of heat transfer area and short conduction paths to transfer heat between the two fluid streams. The stacked channels are then brazed together in a specialised furnace. The brazing process melts a thin layer of alloy on the heat exchanger surfaces which subsequently solidifies to form a single, sealed part. In this construction, each of the two alternating channel designs may be optimised for its specific fluid and flow conditions, which could be comprised of air-water, air-oil or water-oil. Manifolds are provided at the inlet and outlet of each channel to keep the two fluids separate from each other, but at the same time ensure that each respective channel receives the desired flow volume.

1.2 Predictive Design Tools

In early heat exchangers, specifically shell-and-tube style industrial heat exchangers, engineers would typically predict performance and base their designs based on an accepted "fudge factor", usually 60% of ideal performance, to account for the deviation from ideal due to flow non-uniformities, baffle position and leakage (flow bypassing baffles) [4]. This

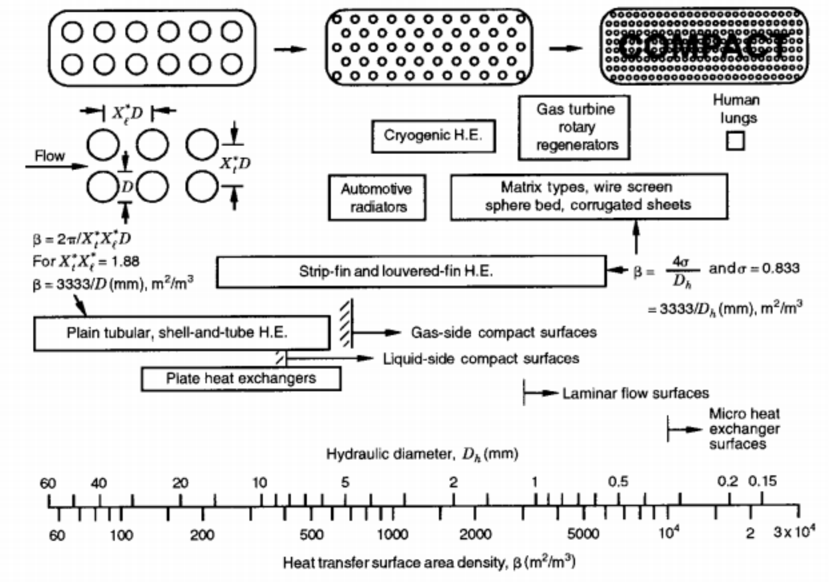

COMPACT
X_t^*D
Flow
D
X_ℓ^*D
$\beta = 2\pi/X_t^*X_\ell^*D$
For $X_t^*X_\ell^* = 1.88$
$\beta = 3333/D$ (mm), m^2/m^3
Cryogenic H.E.
Gas turbine rotary regenerators
Human lungs
Automotive radiators
Matrix types, wire screen sphere bed, corrugated sheets
Strip-fin and louvered-fin H.E.
$\beta = \frac{4\sigma}{D_h}$ and $\sigma = 0.833$
$= 3333/D_h$ (mm), m^2/m^3
Plain tubular, shell-and-tube H.E.
Gas-side compact surfaces
Liquid-side compact surfaces
Plate heat exchangers
Laminar flow surfaces
Micro heat exchanger surfaces
Hydraulic diameter, D_h (mm)
60 40 20 10 5 2 1 0.5 0.2 0.15
60 100 200 500 1000 2000 5000 10^4 2 3×10^4
Heat transfer surface area density, β (m^2/m^3)

Figure 1.2: Heat exchanger compactness [3]

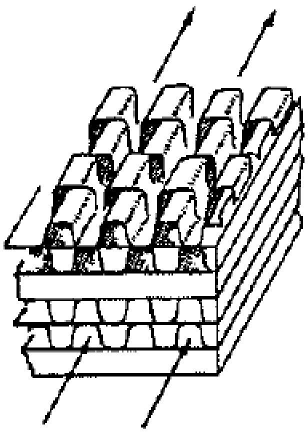


Figure 1.3: Typical stacked CHE, layers alternate between two fluids, separated by solid plates [1]

simplistic design approach worked well for industrial applications, where limited design variables were present, however, lacked flexibility when large deviations from the "ideal" design were present. This problem was largely addressed in the 1955 text by Kays and London [1], which presented a new way for approaching the design of heat exchangers: the ϵ-NTU (effectiveness - Number of Transfer Units) method. This approach, often credited to A.L. London [5], provided one of the most practical design methodologies, allowing engineers to accurately size heat exchangers of various layouts with any number of different heat transfer enhancement surfaces. In addition to the introduction of the ϵ-NTU method, the Kays and London text provides a large collection of experimental heat transfer and pressure drop data for a variety of different heat transfer enhancement surfaces [1]. When the ϵ-NTU method is combined with other methodologies for the interpretation of test data, i.e. the modified Wilson plot method [6] or the Colburn j factor [3], the combination becomes a very powerful, yet simple suite of tools which is still in widespread use within the heat transfer field [3, 5, 7].

The transition of the industry away from solely using empirical or analytical models in search of increased design flexibility and accuracy has brought forth the use of more in-depth, discretised numerical methods for predicting heat transfer. While some of the earliest models relied on the 1-D discretisation of the heat exchanger system [8], other attempts took a more sophisticated route and tackled the numerical modelling of a two dimensional flowfield around an offset strip fin (OSF) [9]. Modelling has since progressed to much more complex simulations, including full 3-D computational fluid dynamics (CFD) simulation of heat exchangers, using turbulence models such as Reynolds-averaged Navier-Stokes (RANS) based k-ϵ [10], or the more computationally intensive method of Large Eddy Simulation (LES) [11, 12]. Unfortunately, many of the more advanced, CFD based numerical models require significant computational expense for a full solution, and as a result, the majority of global heat exchanger studies rely on some sort of simplification method, typically the usage of a porous media assumption for the heat transfer enhancement surface [13, 14].

The tools currently available for the design of heat exchangers go a long way to reducing the uncertainty in the design process. Proven data reduction and correlation techniques for experimental testing allow for reliable performance analysis. High fidelity numerical models of small scale heat transfer enhancement surfaces provide relatively accurate predictions of heat transfer performance. Large, full scale, CFD models take advantage of the numerical simplicity of the porous media approach, yielding a model that can be used to optimise designs. However, while predictions of pressure drop using the porous media assumptions have been successful in full scale models, unfortunately, there is currently no reliable method that is able to accurately incorporate heat transfer data from either experimental testing or high fidelity, small scale CFD models into these full scale models.

As previously stated, it is the goal of this project to develop a model able to bridge this gap in predictive capabilities and fully incorporate accurate small scale flow and heat transfer detail into a computationally efficient global CFD model, while at the same time better understand the flow phenomena present in the small scale features of compact heat exchangers. During the development of this model, focus was also placed on the elucidation and characterisation of some of the small scale flow features, such as prediction of the transition to statistically unsteady or turbulent flow and the heat transfer performance therein.

1.3 Scope of Work

Due to the inherent complexity of the heat transfer enhancement surface, the scales of flow occurring within a modern automotive CHE range from the very small, with flow structures on the order of 10^{-4}m, to several orders of magnitude larger, on the order of 10^{-1}m. This separation of scales results in particular challenges when performing numerical flow and heat transfer analysis of the heat exchanger, namely some compromise in terms of either the detail and scale of resolved flow structures or computational efficiency of the model.

Throughout this project, several areas of study relating to the improved understanding of CHE performance and predictive modelling are pursued in a depth sufficient to provide a better understanding of the basics of the flow, while developing a relatively simple model to provide the end-user a tool capable of accurately predicting performance with minimal additional computational expense. The successful completion of this research has not only resulted in a model capable of providing the desired predictive, but has also highlighted some of the challenges and issues in using typical computer generated models of flow surfaces, which are not necessarily reflective of the mass-production representation of a heat exchanger.

1.3.1 Research Outline

The breakdown of this research can be largely separated into two categories:

1. **Detailed Study and Validation of the performance of the Turbulizer**
2. **Development and Implementation of a Reduced Order Model for Turbulizers in a Global Heat Exchanger Model**

The first stage of the present work provides a focus directly on the turbulizer. The turbulizer, which is a common industry name for the style of heat transfer enhancement surface primarily focused on for the duration of the work, is a small scale heat transfer enhancement surface providing a large increase in active surface area and an extremely strong mixing effect, promoting a high level of heat transfer. Very little published data exist around details of the flow, and the individual performance of these surfaces remains largely a guarded industry secret. This study provides a glimpse at the small scale flows present around these surfaces under a variety of different conditions, in particular, investigating:

- **Small scale, unsteady flow and heat transfer around the turbulizer**
- **Measurement and comparison of CAD data to as-manufactured geometry**
- **Experimental validation of numerical modelling techniques used to predict heat transfer and pressure drop data**

Throughout the first stage of the work, a detailed numerical analysis of a typical turbulizer geometry is presented, tested and validated at the smallest practically simulated flow scales. An understanding of the detailed pressure drop and heat transfer characteristics is provided as a general deliverable, however, a further dive into the understanding of the formation of coherent flow structures, transition to unsteady laminar flow, and eventually fully turbulent flow is calculated, analysed and validated against various scale models and test data using a mixture of flow visualisation using dye injection (or other appropriate means) [15] and particle image velocimetry (PIV) [9]. Furthermore, using X-ray Computed Tomography (XCT) and physical sectioning and reconstruction methods, a numerical model of the exact geometry present in a heat exchanger, following both the forming and brazing steps, is analysed and presented with an effort to numerically model and compare the results to nominal CAD data. These data are further used to provide guidelines to approximate the final as-manufactured form of a turbulizer prior to manufacturing ever taking place.

Following the successful development and characterisation of the numerical model of the turbulizer, the focus of study is placed on the second phase: development of a reduced order model (ROM), capable of integrating known performance details of a turbulizer into a global heat exchanger model with minimal added computational cost. In particular, the goals of the second phase are:

- **Development of a turbulizer flow "library" comprising different flow angles and** Re
- **Development of a ROM using library results for prediction**
- **Testing, validation and benchmarking of ROM output**

Building on the learnings of the initial phase of study, a flow library is constructed for various flow conditions to represent a range of possibilities that may be encountered in a typical heat exchanger. These results are, through analysis and reduction, used to construct a basis for an advanced model that dynamically references known flow conditions to determine local heat transfer and pressure drop data for a ROM. These data are then fed to the calculation in order to predict performance based on calculated local conditions, rather than relying on a catch-all general correlation for heat transfer prediction. Validation and benchmarking of the model are performed against a known heat exchanger design, using experimental test data, CFD data generated using existing predictive methods (a hybrid analytical and numerical approach), and the newly developed predictive model.

For the duration of the study, all numerical results, as well as the proposed ROM methodology are obtained and developed using an interface intended for use with commercial CFD codes, particularly CD-Adapco/Siemens' Star-CCM+, at the request of Dana Canada, sponsors of the research project. The development of the model has provided a methodology that allows for a relatively easy of implementation into a pre-existing framework of end-user programming, rather than integration into the core kernel of the CFD software. It is, however, the belief of the author that integration of this model directly into the core framework of a CFD code will improve the widespread adoption of the model and increase its efficacy and ease of use for specific end-user requirements.

1.3.2 Novelty of Current Work

The present study aims to fill gaps in both the understanding of the flow in compact heat exchangers, as well as the ability to provide a numerically efficient and accurate model that can be used to solve the heat transfer and pressure drop of a CHE without requiring *a priori* testing knowledge of the design. It follows that this will allow an extension of the predictive abilities of CFD modelling to provide improved capabilities extending beyond what is currently possible, on a scale that can allow for detailed thermal and pressure drop modelling with sufficient resolution that it may be used for design iteration and optimisation.

The novelty in this study is present in two distinct areas, which are briefly introduced and discussed below.

Firstly, the small scale study of flow structures in and around a turbulizer-like geometry have not been studied in great depth either numerically or experimentally to date. Furthermore, the transitions between steady and unsteady flow, and laminar and turbulent flow have not been characterised for similar geometries. It is desired to quantify these *Re* regimes in order to better model and analyse heat transfer and pressure drop results. Additionally, these studies include a digitised as-manufactured turbulizer section, yielding a comparison of the flowfields formed for nominal CAD and as-manufactured geometries. Flow conditions for the numerical testing of these geometries are also extended to a range of different mean flow angles, which highlights previously unseen performance and characteristics of turbulizer and similar geometries operating outside of their nominal design and test conditions. Performance in these conditions, which has previously been assumed to be a blending of orthogonal components, is shown to be a non-linear, complex combination that could not have been predicted without such in-depth investigation. The further understanding of these flow interactions may subsequently be used to further design and develop a new generation of heat transfer enhancement surfaces which are optimised for off-angle performance.

Secondly, there currently does not exist a comprehensive reduced order modelling technique that is able to account for arbitrary or developmental heat transfer enhancement surfaces under a variety of flow conditions while directly incorporating the detailed results into a global heat exchanger model. The development and introduction of this model provides a new analysis tool that expands the abilities of predictive performance modelling without *a priori* manufacturing and testing data, thus speeding the development cycle as well as significantly reducing prototyping costs by avoiding the costly manufacture of manufacturing tooling. The proposed model allows for a relative ease of implementation by a design engineer without having to rely on specialist knowledge or experience in development or implementation model. Upon testing, the reduced order model developed herein provides a significant reduction to both the heat transfer and pressure drop error observed through experimental validation of a known geometry.

Upon completion of the present work, the improvement in understanding of both the flow characteristics of a turbulizer-like geometry provides new insight into the flow development within these geometries, experimental validation of the phenomena observed numerically, and a method to incorporate these data into a more complete global heat exchanger model. This work has been completed not only to address the needs of Dana Canada in the design and development of compact heat exchangers, but is expected to provide a step forward in the ability of CFD codes to be able to accurately and consistently predict heat exchanger performance for a wide range of industrial flows with a variety of heat transfer enhancement surfaces.

Chapter 2

Literature Review

The following chapter provides a thorough review of the relevant literature related to the scope of the present research: heat transfer theory and its application to compact heat exchangers. This review is focused on several key areas: the use of non-dimensional numbers in heat transfer analysis, test methods used for characterisation of heat exchangers and heat exchanger flows, the design and development of heat transfer enhancement surfaces and numerical modelling techniques.

2.1 Non-Dimensional Numbers

Prior to embarking on a discussion of the theory behind construction and testing of CHE's, it is crucial to define several non-dimensional units that are commonly referred to when analysing heat exchangers.

Perhaps the most well known of all fluid dynamics related non-dimensional numbers is the Reynolds number (Re). The Reynolds number is an expression of the ratio of inertial forces to viscous forces. When Re is large, inertial forces, such as momentum, dominate the flow, while a small Re is characteristic of flow dominated by viscous forces, such as a creeping flow. In many instances, the onset of turbulence is predicted by analysis of the flow Re, calculated by the expression:

$$Re = \frac{\rho u L}{\mu}, \tag{2.1}$$

where ρ is fluid density, u is the reference velocity of the fluid, L is some characteristic length scale dependent on the flow geometry, and μ is the dynamic viscosity of the fluid. When performing analysis of standard geometric configurations, such as flow over a cylinder, the Re value takes on a very definite meaning, with certain transitions occurring over a small Re range, however, for less commonly studied flows, the choice of length scale is somewhat arbitrary, provided it is proportional to some representative characteristic length of the flow. The definition of hydraulic diameter for the present turbulizer remains proprietary information and cannot be disclosed in the present document This selection is based on the preference of Dana Canada so that the present data correspond to their measurement system.

The Prandtl number, Pr, is defined as the ratio of viscous diffusion to thermal diffusion. In high viscosity fluids, such as cold engine oil or pitch, Pr can assume very large values, while at the opposite end of the spectrum, liquid metals can yield very low Pr. For reference, air at ambient conditions (standard temperature and pressure) has a Pr of approximately 0.7, while water is on the order of 7. Pr is calculated by:

$$Pr = \frac{c_p \mu}{k}, \tag{2.2}$$

where c_p is the fluid specific heat capacity, μ is dynamic viscosity and k is thermal conductivity of the fluid.

The Nusselt number, Nu, is a measure of the heat transfer occurring at a boundary. It is defined as the ratio between the convective heat transfer normal to the surface and the conductive heat transfer normal to the surface, i.e.:

$$Nu = \frac{hL}{k}. \tag{2.3}$$

In this equation, h is the heat transfer coefficient, L is a characteristic length, and k is the thermal conductivity of the fluid. The value of Nu varies depending on the geometric and flow conditions encountered. In fully developed laminar flow, Nu is typically a constant value; however, when the flow becomes turbulent, Nu often depends on both Re and Pr.

The Stanton number, St, is a measure of the ratio between heat transfer into a fluid and the thermal capacity of the fluid. This metric is often used as a measure of convective heat transfer in a fluid, and is defined as:

$$St = \frac{h}{\rho u c_p}. \tag{2.4}$$

Additional non-dimensional numbers will be discussed as they are introduced.

2.2 Experimental Test Methods

Although the study of thermal energy dates back to the 19th century with the likes of Joule, Carnot and Clausius, the study of heat exchangers did not begin in earnest until the early 20th century, with the works of Wilhelm Nusselt [16] and E.E. Wilson [6], which opened the doors for the study of heat transfer as it relates to heat exchangers. These works, among others, are the basis of many of the experimental characterisation techniques still in use today. Despite the fact that the format and technology applied to heat exchangers has undergone a variety of large changes, the analysis techniques remain as relevant and practical as when they were first introduced, which is a testament to their practicality. This section discusses some of these techniques and how they are applied.

2.2.1 Wilson Plot and Modifications

Obtaining detailed experimental data from the testing of a heat exchanger can pose a significant challenge. Often, local heat transfer data are nearly impossible to measure by direct means, and must be obtained by taking the average heat flux over a known area. Compounding this difficulty is the fact that many heat exchangers rely on extremely small scale heat transfer enhancement surfaces, which may have a large variation in heat transfer over a small spatial distance.

One of the first practical techniques used to overcome this difficulty was introduced by Wilson [6]. By making several assumptions about the heat transfer surface, one can generalise the performance and extract several key characteristics without *a priori* knowledge of the enhancement surface. This method is outlined below.

In a two fluid heat exchanger, the overall thermal resistance can be expressed as the sum of the thermal convection resistance for each of the two fluids and the conduction resistance of the wall:

$$R_{tot} = R_1 + R_w + R_2, \tag{2.5}$$

which can be further expressed as:

$$\frac{1}{UA} = \frac{1}{h_1 A_1} + \frac{t}{k} + \frac{1}{h_2 A_2}. \tag{2.6}$$

In this equation, U represents the overall heat transfer coefficient and A represents surface area of the heat exchanger. For practical purposes, these are typically grouped into what is effectively a single term: UA. The remaining terms in this equation are: t, the wall thickness, k, the wall thermal conductivity, and h, the heat transfer coefficient.

Although the expression in Eq. 2.6 contains 3 separate terms, maintaining one of the fluid streams at a constant flow rate and temperature allows all changes in UA to be attributed to only the change in a single term in Eq. 2.6. As A_i remains constant for all tests, linear-regression can be performed against the results to determine a best-fit value of h as a function of Re.

Modifications to this procedure were proposed by several researchers, with the most commonly used being the method of Briggs and Young [17]. In this method, a general form for the Nusselt number (Nu) is given as:

$$Nu_i = \frac{h_i D_{h_i}}{k_i} = C_i Re_i^m Pr_i^n. \tag{2.7}$$

This form assumes that the heat transfer coefficient and Nu are dependent on Re, raised to some exponent, m, and the Prandtl number, Pr, raised to some exponent, n. By modifying the flow rates, i.e. Re, and fluid properties, i.e. Pr, of each of the two fluid streams, linear-regression analysis may be performed to yield best-fit values for the exponents m and n.

The Wilson coefficient, C_i, remains, to this point, uncalculated. When testing is performed on a heat exchanger that is geometrically identical for both fluid streams, i.e. the same heat transfer enhancement surface is used on both sides, the values of C_1 and C_2 are equivalent. Under the assumption that R_w is small (as is true in the majority of heat exchangers), substitution into Eq. 2.6 yields:

$$\frac{1}{UA} = \frac{2}{C Re^m Pr^n}. \tag{2.8}$$

Thus, for a known UA, the Wilson coefficient can be calculated. Subsequently, the value of C can be used for the calculation of Nu via Eq. 2.7 for some arbitrary geometry. In this fashion, it is possible to build up a library of coefficients for a variety of surfaces, provided enough samples and sets of test data are available.

The Wilson plot method is not, however, without its drawbacks. It is assumed that the correlations follow a strict power law for Re and Nu, shown in Eq. 2.7, when this may not, in fact, be true. If the flow transitions from laminar to turbulent within the test data, a modified, or blended power law function may be required for accurate data reduction. Should additional physics phenomena present themselves during testing, the required correlations may become even more complex, following any of the forms put forth by Churchill and Usagi [18]. If the wall resistance, R_w, becomes large, this may lead to inaccuracies in the basic assumptions made when performing the analysis technique outlined above. Additionally, the correlations are typically only valid for a relatively small range of Pr. In some instances, such as applications involving viscous oils that undergo a large temperature change, the viscosity may change by several orders of magnitude, making the assumption of constant fluid properties invalid and leading to increased uncertainty in the experimental results.

2.2.2 Colburn j Factor, Fanning Friction Factor and Goodness

Summary of heat transfer enhancement surface test data typically follows in the preferred method of Kays and London [1, 19] by plotting two specific values: the Colburn j factor and the Fanning friction factor, f. As both of these factors are based on non-dimensional groups, they may easily be applied under a wide variety of flow conditions.

The Fanning friction factor, f, is related to the pressure drop per unit length (Pa/m) by:

$$\frac{\Delta p}{L} = \frac{1}{2}\rho u^2 \frac{4}{D_h} f \text{ [2].} \tag{2.9}$$

This value, not to be confused with the Darcy friction factor, f_D, which is 4 times larger, is useful in estimating the pressure drop of a given enhancement surface when the flow Re is known.

Similarly, the Colburn j factor is a non-dimensional measure of the heat transfer of a given surface:

$$j = \frac{Nu}{RePr^{1/3}} = StPr^{2/3}, \tag{2.10}$$

where St is the Stanton number [2]. As the j factor incorporates Nu, Re and Pr, it is dependent on flow geometry, fluid velocity and fluid properties. Provided that the fluid properties do not change significantly within the heat exchanger, this method provides a concise way of summarising heat transfer data such that it can be easily applied to any heat transfer surface where the flow details and geometry are known.

With the availability of these two factors, one further metric for comparison may be obtained: the flow area goodness. This concept, introduced by London [20], is simply the ratio of:

$$Flow\ Area\ Goodness\ Factor = \frac{j}{f}. \tag{2.11}$$

The goodness factor is a measure of the "cost" in terms of pressure drop at which a given amount of heat transfer occurs. This is not, however, directly related to the compactness of a heat transfer, and is therefore no guarantee that a heat transfer surface with a high goodness factor will be a suitable choice for a given application.

2.2.3 ϵ-NTU

The effectiveness-number of transfer units methodology, often abbreviated as ϵ-NTU, was introduced by London as a way to predict the performance of heat exchangers with varying layouts [1]. To this point, the effect of a given surface has been discussed and the performance characterised, but the ϵ-NTU method provides a way of predicting the total amount of heat transfer occurring within an ideal heat exchanger with two fluid streams and known flow arrangement.

In order to use this method, the heat capacity rate for each of the two fluids must be determined:

$$C = \dot{m}c_p, \tag{2.12}$$

where $\dot{m}$ is the flow rate and c_p is the constant pressure specific heat capacity. The *minimum* value of C of the two streams is required for the calculations, hereafter referred to as C_{min}, as this is the limiting stream for the heat exchanger. The number of transfer units, NTU, can then be determined by the equation:

$$NTU = \frac{UA}{C_{min}}\ [1]. \tag{2.13}$$

The value of UA is the same as that introduced previously in Sec. 2.2.1.

The effectiveness of a heat exchanger is very simply measured as the fraction of the maximum theoretically available heat transfer that is achieved, where

$$\epsilon = \frac{q}{q_{max}} = \frac{C_h(t_{h,in} - t_{h,out})}{C_{min}(t_{h,in} - t_{c,in})} = \frac{C_c(t_{c,out} - t_{c,in})}{C_{min}(t_{h,in} - t_{c,in})} \text{ [1].} \tag{2.14}$$

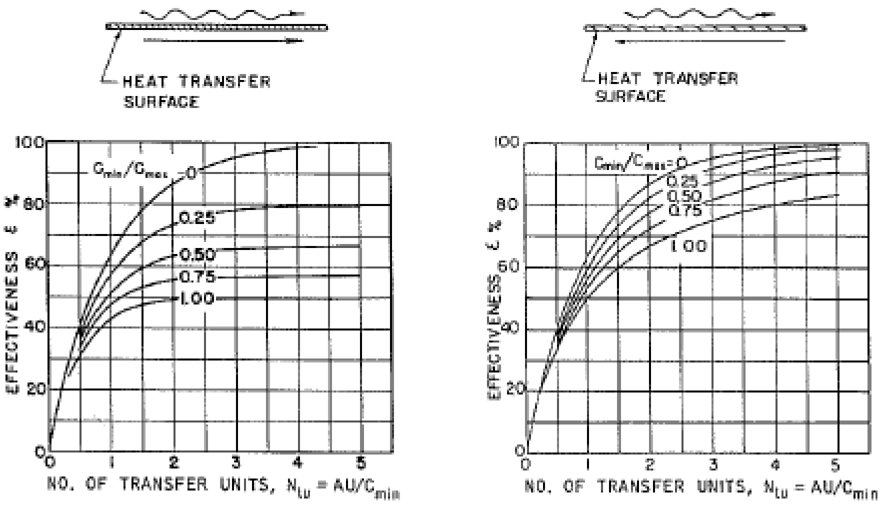
The use of C_{min} is an important part of Eq. 2.14. The denominator in this equation signifies that the maximum heat transfer cannot exceed the minimum flow capacity C_{min} multiplied by the bounding upper and lower temperatures, $t_{h,in}$ and $t_{c,in}$. Regardless of the size and layout of a heat exchange, neither fluid can ever exceed these two boundaries, putting a practical limit on the amount of heat transfer available.

The usefulness of the $\epsilon - NTU$ method becomes apparent in the correlation between the effectiveness and the flow arrangement. In general, it can be stated:

$$\epsilon = \Phi\left(NTU, \frac{C_{min}}{C_{max}}, \mathit{flow\ arrangement}\right) \text{ [1],} \tag{2.15}$$

where the expression is mathematically derived for varying flow arrangements. Use of this general equation allows for the quick and accurate sizing of a heat exchanger based on known parameters. Visual inspection of the shapes of the curves is also beneficial, as it gives the engineer a good “feel” for how the various layouts are affected by flow capacity ratios. Several examples of the $\epsilon - NTU$ curves are shown in Fig. 2.1. In each of the examples, several important trends can be observed. Firstly, each of the curves tends to plateau as the NTU value increases. This is indicative of the fact that additional heat transfer area does not significantly benefit heat transfer, i.e. the maximum practical heat transfer for the given flow conditions has been reached. Secondly, as the capacity-rate ratio approaches unity, a minimum effectiveness is realised for a given NTU value. In effect, this is due to the fact that both streams undergo a large temperature change, and available ΔT, the driving force for heat transfer, is exhausted. Conversely, when one side is saturated by extremely high flow rates and the capacity-rate ratio approaches 0, the ΔT between flow streams is maximised for a given flow arrangement, maximising heat transfer and yielding a high effectiveness. The capacity-rate is thus bounded between 0 and 1, which represent the two extremes of heat exchanger performance.

Each set of curves shown in Fig. 2.1 is derived from a mathematical relation [1]. Flow arrangements can vary from parallel flow, in which both fluids enter on the same side of the heat exchanger and travel parallel to each other in the same direction, to counter flow,



(a) Parallel Flow

(b) Counter Flow

Figure 2.1: $\epsilon - NTU$ relations for varying flow arrangements [1]

in which the flow direction of one fluid is reversed but is otherwise similar to the parallel flow case, to combinations of the two varying the number of passes and flow direction. Further information about these relations can be found in many heat transfer texts, including [1, 3].

Working within the bounds of the curves shown in Fig. 2.1 gives engineers a final design space to work within; for a given effectiveness and flow availability, the UA can be determined, and vice versa. Changes to geometry or flow arrangement can quickly be assessed for efficient initial sizing calculations. Therefore, use of the $\epsilon - NTU$ method is one of the most important tools available in sizing heat exchangers, and is still largely used today as the general basis for more in-depth sizing calculations. It is even possible to use the $\epsilon - NTU$ method in a hybrid numerical method, where 1-D calculations may be performed to locally calculate heat transfer in a heat exchanger. This method, however, is subject to a wide variety of assumptions about flow uniformity and Re conditions, and is not always applicable in complex flows.

2.2.4 Particle Image Velocimetry

Particle Image Velocimetry (PIV) is a method used to ascertain detailed flow data from a controlled experiment. These data can include velocity vector planes, time resolved analysis of flow structures, and, under certain circumstances turbulence statistics. Crucially, PIV is able to quantitatively measure a large number of points *simultaneously* [21]. A sample PIV measurement made of the flow around bubbles in a two-phase flow is shown in Fig. 2.2.

The general concept of PIV is simple: a particle seeding is introduced into a flow, a laser sheet illuminates the particles in a flow plane twice in very short succession, and the distance traversed by the seeding particles is converted into a velocity field over the imaged area. Although early efforts were plagued with issues, including ambiguity of flow direction [21] and uncertainty due to image resolution and timing, development of the technique continued and many of the early teething issues were eliminated, resulting in a robust and accurate measurement method. Recent developments have even led to the simultaneous measurement of a 3-dimensional flow volume in a technique known as tomographic PIV [23].

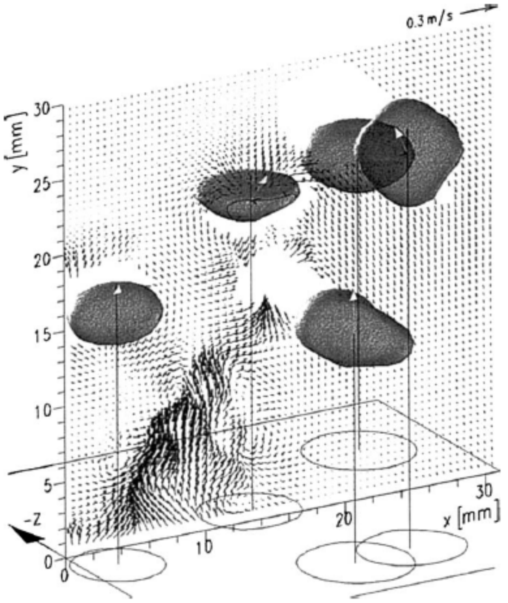


Figure 2.2: PIV measurement of the flow around bubbles [22]

In any PIV measurement, there are several requisite needs which must be met in order to reduce experimental uncertainty to acceptable levels. The flow must be seeded such that the particles faithfully follow the local flow structures, i.e. the particles accurately represent the small scale flow structures with minimal error. This is often achieved by using particles on the order of 10-30μm in water [21]. It is also important, especially in low *Re* flows, to match the density of the seeding particles to the average fluid density so that particles do not settle out of suspension and provide false velocity measurements. For obvious reasons, good optical access for both the laser sheet and the photographic equipment are crucial. Complex or convoluted geometry can create shadowed areas which the laser or photographic equipment are unable to reach, leaving areas that go unmeasured. Although this can obviously be problematic if the desired measurement region is obstructed, it does not affect accuracy of accessible measurement locations, provided the laser sheet is not scattered by boundary surfaces.

In order to work around the problems of obstructed optical access, a method known as refractive index (RI) matching has been developed [24, 25, 26]. In this method, a transparent test geometry is created, often using acrylic or other optically thin polycarbonate materials, and the fluid is treated with additives to match the flow RI with the test geometry RI. The result is that the laser sheet and photographic equipment are able to obtain unobstructed visual access to the entire flow domain. In effect, the geometry becomes optically invisible. This technique is particularly useful for complex geometry or obtaining near wall measurements on a curved surface, such as the development of a boundary layer [26].

As this section is intended only to provide a cursory overview of the PIV method, many details about the process have been omitted. Additional information regarding PIV image processing, calculation and limitations may be found in [15, 21, 27], while additional information on the processing and calculation of turbulence statistics may be found in [28, 29, 30, 31].

2.3 Heat Transfer Enhancement Surfaces

The purpose of a heat transfer enhancement surface is to increase the rate at which thermal energy is added or removed from a fluid stream and transferred to its surroundings. In most cases, the removed thermal energy is subsequently transferred into a second fluid stream. Thus, it is desirable to design a heat transfer enhancement surface which is able to effect a maximum amount of heat transfer for a minimum pressure drop penalty.

Using the case of laminar flow between two parallel plates, maximum heat transfer occurs while the flow is developing, and asymptotically approaches a minimum once the flow becomes fully developed, i.e. the cross-channel velocity profile and normalised thermal profile no longer change with distance from the entrance. Therefore, it is apparent that in the interests of improved heat transfer, the flow must be kept in some continually developing state. This is shown schematically for the case of flow over a flat plate compared to a slit fin in Fig. 2.3. Any interruptions in this laminar flow profile effectively resets the thermal boundary layer, increasing heat transfer.

While Nu of fully developed laminar flows can be shown to be independent of Re and Pr, developing and turbulent flows may, in fact, be strongly dependent on these parameters. Thus, manipulation of the channel surface, or placing any obstructions in a flow channel, can have a large impact on overall heat transfer, especially when the Pr becomes large. It is from this general trend that the concept of heat transfer enhancement surfaces arose.

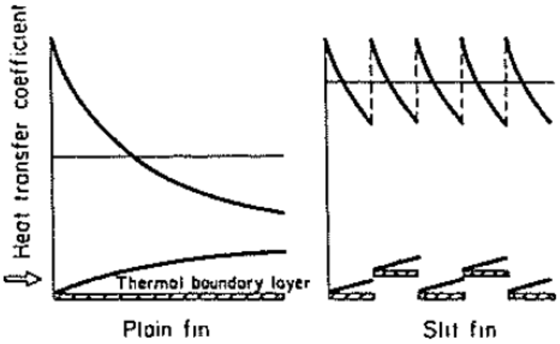


Figure 2.3: Comparison of heat transfer and thermal boundary layer growth between a smooth surface and an interrupted surface [2]

2.3.1 Shell-and-Tube Designs

One of the oldest, and perhaps simplest, heat exchanger designs is the shell-and-tube or tube-bundle type heat exchanger. In this type of heat exchanger, one fluid is passed through the interior of the tubes while the second fluid flows over the exterior of the tubes, encouraging heat transfer between the two fluid media. This concept is shown in Fig. 2.4, where the flow direction of the different fluids is shown by arrows.

In this arrangement, it is often desired to contain the fluid on the exterior of the tubes within a sealing shell. Baffles are placed within the shell, forcing the fluid back and forth over different sections of the tubes to improve overall performance, as shown schematically in Fig. 2.5. In order to increase the overall compactness of these heat exchangers a general downsizing of the diameter of the tubes occurred. The use of small tubes is beneficial to a point: increased surface area improved overall heat transfer, but additional problems such as fatigue failure due to flow induced vibrations in the tubes [32], and increased manufacturing complexity due to the number of tubes present in a given design tended to outweigh the benefits of downsizing. As a result, although these heat exchangers are

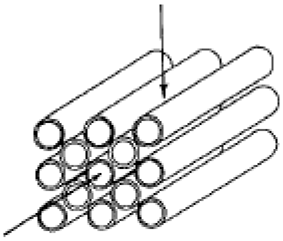


Figure 2.4: Shell-and-tube heat exchanger tube bundle [1]

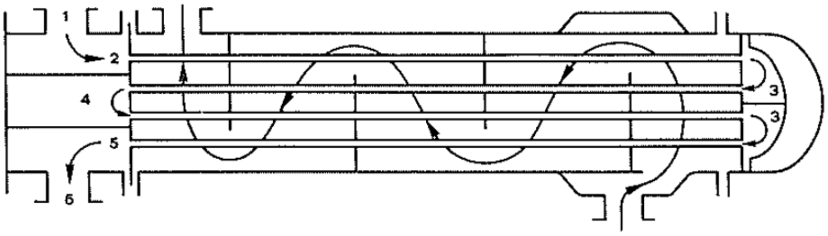


Figure 2.5: Shell-and-tube heat exchanger baffle schematic [7]

typically quite cost effective and popular in stationary applications, such as power generation plants, their relatively low level of practical compactness makes them unpopular for use in many automotive or transportation applications. Due to the irrelevance in modern automotive CHE applications, the shell-and-tube heat exchanger are not discussed further.

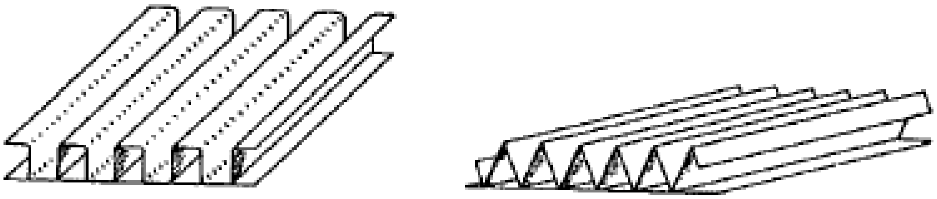
2.3.2 Fins

The purpose of fins in heat exchangers is to provide an increase in available heat transfer surface and to provide a conduction path to easily transfer heat to/from the centre of a fluid channel into the adjacent fluid channel. The use of fins in the automotive radiator is one of the most universally observed and recognised applications of heat transfer enhancement surfaces. The high surface area and relatively low pressure drop of fins provide excellent enhancement on the air side of many heat exchangers; however, the relatively long conduction path from the centre of the channel along a thin fin makes the conduction resistance, discussed previously in Sec. 2.2.1, a large component of the overall thermal resistance of the heat exchanger. Thus, it is common to relegate the fin to use primarily in applications where the flow capacity-rate is relatively low, such as air flows, where the conductive resistance does not limit the heat exchanger performance.

Plain and Louvered Fins

Plain fins are the simplest of the fin types in use on heat exchangers. The fins are made of thin strips of metal which are brazed between the upper and lower channel walls. The brazing process ensures good thermal contact between the fin material and the channel walls, thus maximising heat transfer performance. Depending on the application, the orientation of the fin can be changed such that the fin surfaces are either perpendicular to the upper and lower channel surfaces, making rectangular flow channels, or placed at some angle to the upper and lower plates, creating triangular flow channels. The specific size and shape of the channel is dependent on the intended final application. Examples of these plain fins are shown in Fig. 2.6.

The relative simplicity of the plain fin allows for the derivation of accurate analytical



(a) Parallel Flow

(b) Counter Flow

Figure 2.6: Plain fin examples [2]

correlations for heat transfer and pressure drop performance.

A very commonly seen performance adaptation of the plain fin is through the addition

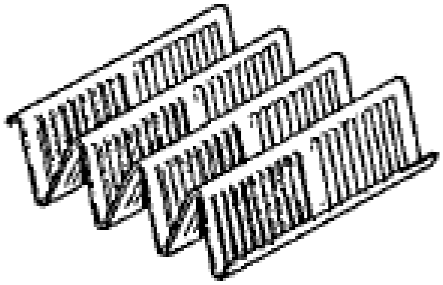


Figure 2.7: Isometric view of a typical louvered fin [1]

of louvres, Fig. 2.7. These louvres, or vertical slits, act in two capacities: the flow is redirected at some angle to the primary flow direction, elongating the total flow path, and the presence of the louvres acts to reset the thermal and viscous boundary layer as in Fig. 2.3, increasing both heat transfer and pressure drop. A cutaway view of a typical louvered fin is shown in Fig. 2.8, where the effects of the louvre redirection is shown schematically through arrows indicating flow direction. Since there are many variables that can be optimised within the fin design, for example, louvre angle, height, length spacing, etc., there are currently no general analytical solutions to predict performance which are valid for the entire practical range of each of the variables. Instead, empirical relations have been developed through extensive testing, much of which can be found in [1, 19]. The louvered fin remains the current standard for high performance density in air side CHE applications.

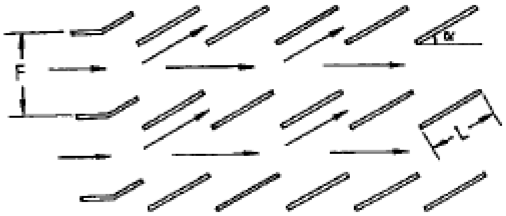


Figure 2.8: Cutaway view of a louvered fin, louvre angle and effects of flow redirection can be seen [2]

Wavy Fins

Although relatively simple in appearance, the wavy fin, so named due to the meandering sinusoidal wave shape in the primary flow direction shown in Fig. 2.9, possesses surprisingly complex fluid dynamics and heat transfer mechanisms. The direction reversal imposed by the sine wave shape sets up a secondary flow within the channel, which consists of a series of longitudinal vortices, very similar in nature to Dean vortices [33, 34] or Taylor vortices (from Taylor-Couette flow [35]). The effect of this vortical rotation is the thinning of thermal and hydraulic boundary layers on the fin surface, with a corresponding increase in heat transfer performance and pressure drop. An example of these flow features is shown in Fig. 2.10, where the pair of vortices can be seen forming in the flow within a curved pipe. Although the mechanism in which heat transfer augmentation is achieved is different than that shown in Fig. 2.3, the final effect is similar due to the localised thermal boundary layer thinning.

Although the wavy fin does not possess the manufacturing simplicity and low pressure drop of a plain fin, it does offer an improvement in heat transfer performance above the plain fin designs of equivalent fin densities. However, due to the fact that the developing thermal and viscous boundary layers are not forcibly interrupted as in the case of a louvered fin, the heat transfer performance of the wavy fin cannot generally match the louvered fin.

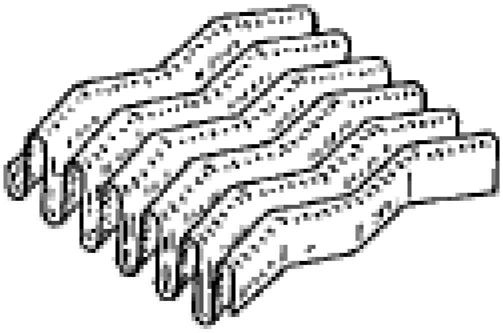


Figure 2.9: Wavy Fin [2]

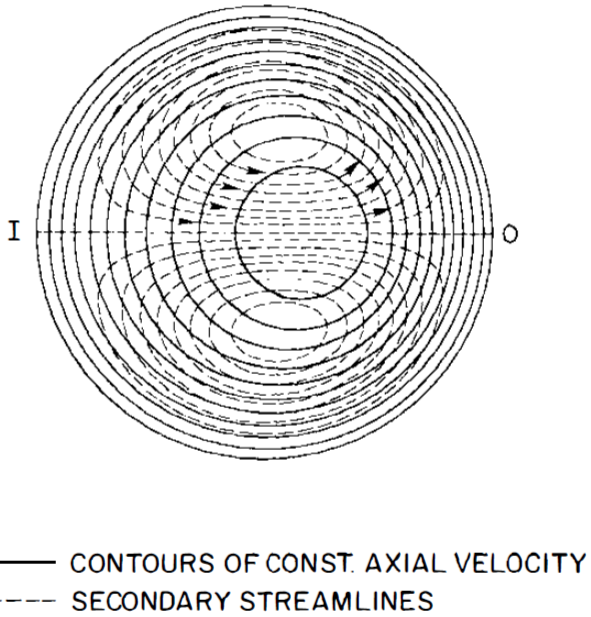

I
O
——— CONTOURS OF CONST. AXIAL VELOCITY
------ SECONDARY STREAMLINES

Figure 2.10: Dean vortex formation in a curved pipe [34]

Offset Strip Fins

The design of an offset strip fin (OSF) borrows heavily from the concept of a louvered fin, however, instead of redirecting the air with angled louvres, the offset strips serve only for thermal boundary layer thickness reduction. This effect is highlighted above schematically in Fig. 2.3. The constant "resetting" of the boundary layer leads to a very effective heat transfer enhancement surface. Additionally, the relative simplicity of the geometry allows for the derivation of analytical approximations for fin performance [36], and has such been the study of many experimental and numerical studies [9, 14, 37, 38]. A typical OSF is shown in Fig. 2.11, with the primary flow direction shown by an arrow.

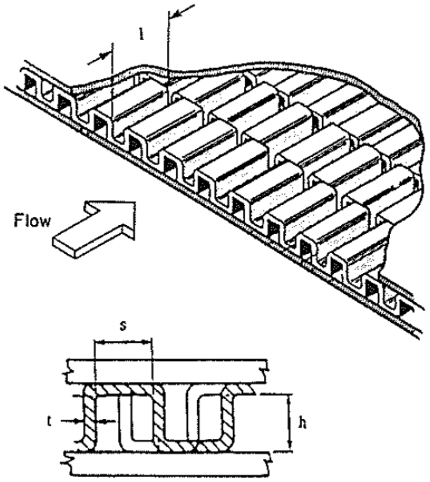


Figure 2.11: Cutaway view of OSF placed between coreplates, arrow indicates flow direction

2.3.3 Turbulizers

The concept of a turbulizer, sometimes known as a turbulator or turb, is largely based around strong mixing induced by large blockages and bluff body flow. In its simplest terms, a turbulizer can be thought of as an OSF, Fig. 2.12; however, the primary flow direction is now perpendicular to the flat fin surface. This is often referred to as the high pressure drop (HPD) orientation of the turb, due to the significant increase in pressure

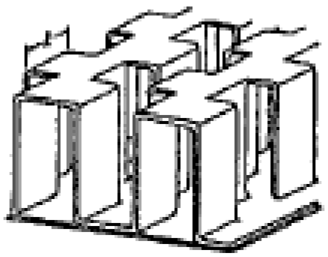


Figure 2.12: Isometric view of typical OSF

drop over the low pressure drop (LPD) flow orientation. The flow orientations are shown in Fig. 2.13. In reality, the "vertical" surfaces seen in an OSF are often oriented at some angle to vertical, as shown in Fig. 2.14.

Likely due to the competitive nature of the automotive industry, there are relatively few publications relating to the performance and characterisation of turbulizers. In one of the few available studies, Muzychka [39] provides experimental test results on macroscopic heat transfer and pressure drop data, namely j and f for a variety of different flow conditions, while performing a regression analysis on the test data in order to improve empirical relations used in the prediction of heat transfer and pressure drop. Attempts are made during the curve fitting process to align the slope of the test data with theoretical laminar or turbulent flows; however, the range of flow rates does not appear to be large enough to fully capture the change of flow regime. The Muzychka study, while comprehensive in the number of flow conditions and geometries considered, does very little to reveal the nature of the small flow structures within the turbulizer. Although the transition from laminar to turbulent flow and its impact on performance is discussed, no clear transition region is evident in the data provided.

Upon careful review of the literature, it becomes apparent that there is a significant gap relating to the understanding of the flow in and around turbulizer geometries. It is postulated that this flow becomes highly 3-dimensional and complex [39], especially when considering geometry proposed in [40]; however, no in-depth study has been performed to date which elucidates the details of this flow, including the steady/unsteady transition and the laminar/turbulent transition, both of which are highly important in designing heat exchangers that may operate within this transitional window. Additionally, the macro-

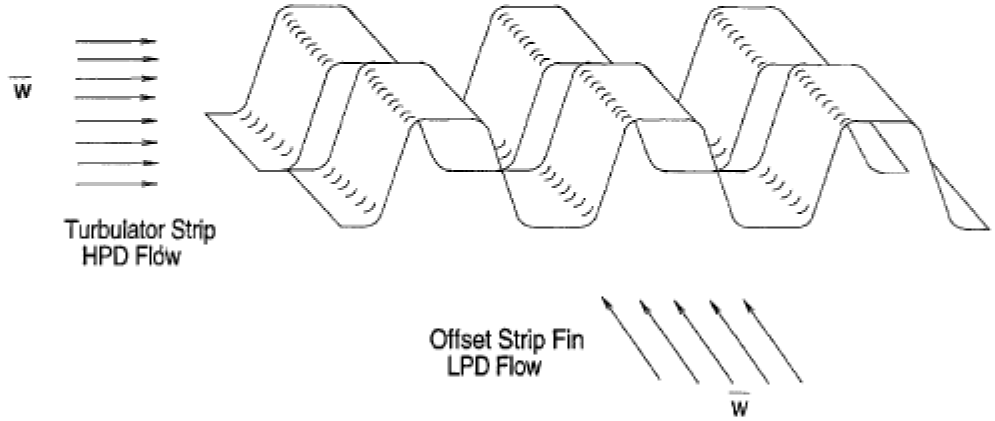


Figure 2.13: Distinction between HPD and LPD flow orientations [39]

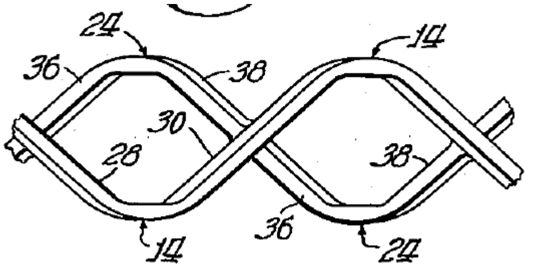


Figure 2.14: Cross section of a typical turb geometry, ca. 1961 [40]

scopic heat transfer test results are highly damped by the thermal capacitance of both the test specimen and the fluid, making it nearly impossible to understand how the changes in flow may affect heat transfer and pressure drop. Further study and understanding of the flow in these geometries will aid in the future design of more efficient turbulizer surfaces.

2.4 Numerical Modelling

Many of the early studies on heat transfer enhancement surfaces rely on constructing experimental test geometries and performing exhaustive studies with a variety of flow rates and different fluids to fully reveal the performance of a given surface [1, 19]. This method of testing, although repeatable and robust, is extremely time consuming and expensive. It becomes difficult to optimise a design due to the need for manufacturing tooling and high development cost of each prototype surface. Accordingly, the search for faster and more efficient test methods resulted in a shift towards relying on numerical modelling to ascertain these performance data. As the capabilities of numerical modelling have progressed over time, it has become a desirable enterprise to attempt to predict the heat transfer and pressure drop over a wide variety of heat transfer enhancement surfaces in a relatively short time with a high level of accuracy. This allows any investment in prototype tooling to be made with a higher level of confidence and lower financial or development time risk.

2.4.1 Computational Fluid Dynamics

The use of computational fluid dynamics (CFD) has revolutionised the study of fluid flow phenomena. Discretisation of the governing equations of fluid flow, namely the continuity equation, the Navier-Stokes equations and the conservation of energy, allows for the comprehensive simulation of flows which may be steady or unsteady, laminar or turbulent, isothermal or involve energy transfer, etc. Each particular flow condition presents its own challenges, and accordingly, various closure models have been developed to account for these challenges depending on the exact flow conditions.

Including CFD analysis as an integral part of the heat exchanger design process has allowed for new designs to be tested, optimised and prototyped for less cost and with less uncertainty than ever before. As a result, the industry has become increasingly competitive, pushing designs right to the limit of heat transfer and pressure drop performance requirements. Accordingly, there is an ongoing search for more accurate and faster methods of simulating the performance of a specific heat exchanger design, and ongoing computer and CFD code development has made this simulation goal increasingly realistic.

As there is an abundance of information available on the derivation and application of the conservation laws used in the formulation of many CFD codes, specific details will not be shown at this time. Further details can be found in many of the available texts on the subject, including [41]. The following sections will, however, discuss the roles of different turbulence modelling methodologies as they relate to the study of heat exchangers.

Reynolds-Averaged Navier-Stokes

Perhaps the most popular of the turbulence modelling approaches is the Reynolds-Averaged Navier-Stokes (RANS) method. This method relies on the decomposition of instantaneous scalar values into a mean and fluctuation component, i.e.:

$$\psi(x_i, t) = \bar{\psi}(x_i, t) + \psi'(x_i, t). \tag{2.16}$$

In this equation, the scalar ψ can represent any scalar quantity conserved in the flow, such as temperature (or enthalpy), concentration, velocity components, etc. The notation $\bar{\psi}$ represents the mean value of ψ, and ψ' is the fluctuating component. The subscript i represents Einstein notation of the position x. Using this decomposition technique, it can be shown that the continuity and momentum (Navier-Stokes) equations reduce to:

$$\frac{\partial \overline{\rho}}{\partial t} + \frac{\partial \left(\overline{\rho u_i}\right)}{\partial x_i} = 0 \tag{2.17}$$

$$\frac{\partial (\overline{\rho u_i})}{\partial t} + \frac{\partial \left(\overline{\rho u_i u_j}\right)}{\partial x_i} = -\frac{\partial \overline{p}}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} + \frac{\partial \left(\overline{\rho}\overline{u_i' u_j'}\right)}{\partial x_j} + B_i. \tag{2.18}$$

These equations are known as the Reynolds-Averaged Navier-Stokes (RANS) equations. It should be noted that even though these equations are shown with mean scalar components, it does not imply that the solution must be steady. The term created through this averaging technique in Eq. 2.18, $\frac{\partial\left(\overline{\bar{\rho} u_i' u_j'}\right)}{\partial x_j}$, represents the Reynolds stresses, while B_i represents the body forces acting on the fluid control volume, such as gravity or magnetism. Closure of this term is achieved through the use of one of the many available turbulence models. The most popular of these turbulence closure models are the $k-\epsilon$ and the $k-\omega$ models, each of which introduce two additional conservation equations, and are thus termed "2-equations models" [42, 43]. There are several variations of these models which introduce different assumptions into the formulation. Further details may be found in [28]. Both of these models have been shown to have acceptable performance in a wide range of applications, with each of the models possessing its own strengths and weakness [28, 41].

One major weakness of RANS based turbulence modelling is the assumption of a fully turbulent flow. In many cases, this assumption is valid, but in the case of low-Re flows, this assumption may force unwanted behaviour in the numerical solution. Models have been developed which do not make some of the high-Re assumptions used in the standard $k-\epsilon$ and $k-\omega$ formulation, but these models have limited applicability to general flows which may encounter a wide range of Re due to their inherent numerical stiffness and slow convergence [41].

Large-Eddy Simulation

While RANS models by their very nature model *all* eddies in the flow [41], Large-Eddy Simulation (LES) is an inherently unsteady turbulence modelling technique which resolves larger anisotropic eddies in the flow, filtering out and modelling only the smaller isotropic eddies which can be predicted much more accurately [28]. Reducing the reliance on the eddy modelling techniques provides a more rigorous and robust method, which is particularly suited for simulation of flows that involve bluff body separation, oscillating wakes and highly anisotropic eddies. Unfortunately, this increased resolution comes at the cost of numerical efficiency, as LES can incur substantially higher computational costs than steady RANS.

In the formulation of the Navier-Stokes equations, a spatial cutoff filter is applied to remove all turbulent motions below a specified threshold. This threshold, often referred to as the filter width, is expressed by Δ and can be calculated using a number of methods, including box filtering, Gaussian filtering or Spectral filtering, among others [28, 41]. Most commonly, however, the cutoff is determined by the local grid size, such that:

$$\Delta = \sqrt[3]{\Delta x \Delta y \Delta z}, \tag{2.19}$$

in the case of a standard 3 dimensional Cartesian coordinate system. Applying this filter to the continuity and Navier-Stokes equations (in the incompressible case) and substituting for the residual stress tensor, τ_{ij}^R yields:

$$\frac{\partial \overline{u_i}}{\partial x_i} = 0 \tag{2.20}$$

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial}{\partial t}\left(\overline{u_i u_j}\right) = -\frac{1}{\rho}\frac{\partial \bar{p}}{\partial x_i} + \nu \frac{\partial^2 \bar{u}_i}{\partial x_i \partial x_j} - \frac{\partial \tau_{ij}^R}{\partial x_i}. \tag{2.21}$$

The unresolved, small scale or subgrid-scale (SGS) stresses are represented in the residual stress tensor, a term which must be modelled.

Similar to RANS, numerous models exist for the SGS stresses. The most common model used is the Smagorinsky model [28, 41]; however, due to the assumptions made in the model formulation, does not accurately represent turbulent quantities near walls, in rotating flows or flows undergoing the transition from laminar to turbulent [44]. A so-called dynamic SGS eddy viscosity model has been developed for application in these circumstances and has been shown to be in good agreement with direct numerical simulations (DNS) data [44]. These modifications make the dynamic SGS model particularly well suited to modelling the wall bounded, potentially transitional flow within heat exchangers and heat transfer enhancement surfaces.

Simplified Heat Exchanger Modelling in CFD

Although a CFD code is able to provide a prediction of wall heat transfer in a general conjugate heat transfer calculation, the existing models primarily rely on the calculation and prediction of temperature gradients within a fluid to predict local fluid convective and conductive heat transfer, and a heat transfer coefficient combined with a temperature difference, or driving potential ΔT, to transfer thermal energy between a fluid and a solid boundary. These calculations are representative of the physics occurring during normal flow processes, such as bluff body or internal/external flow simulations, but when the fluid medium is replaced with a more complex heat transfer enhancement surface that is not fully simulated, the traditional wall models become inaccurate and are unrepresentative of the inherent physics occurring within a simulation [41, 45].

The current technique available within CFD codes for the modelling of a full heat exchanger replaces complex geometry with a large, relatively uniform combined fluid region that enforces relatively simple physical models. A typical representation is shown in Fig. 2.15, whereby two fluid streams are “combined” in a single, overlapping calculation zone. This zone is used to simplify the discretisation process of the otherwise complex heat transfer surfaces, which, in addition to removing the heat transfer enhancement surface itself, remove individual flow passages as well, yielding a very simplified result, as shown in Fig. 2.16. A full simulation of this geometry would instead account for the fluid flow progressing through different radiator channels in discrete locations progressing through the radiator, rather than showing a unified representation as demonstrated in Fig. 2.16.

Pressure drop is modelled using a modification of the Darcy-Forchheimer porous media model, described in further detail in the following section [45]. This simplification provides a means of handling the pressure drop in a discretised computational block, which is numer-

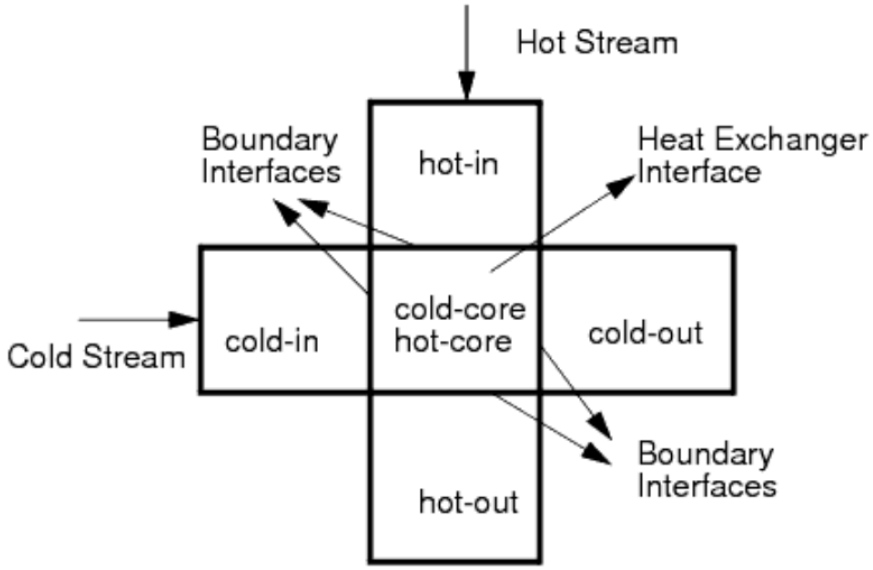


Figure 2.15: STAR-CCM+ Topological representation of the “dual stream heat exchanger” model, with a shared region allowing for the overlap of distinct hot and cold regions [45]

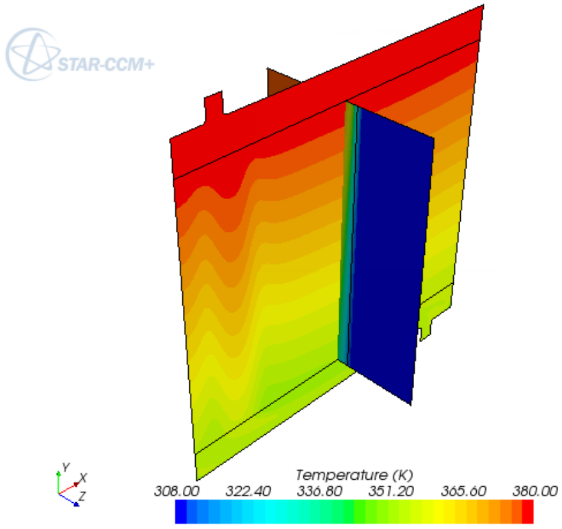


Figure 2.16: A sample representation of the STAR-CCM+ “dual stream heat exchanger” model applied to a simplified automotive radiator, as shown in the STAR-CCM+ tutorial manual [45]

ically efficient, but provides little understanding of the underlying physics being simulated. These data are calculated by a simple correlation of pressure drop as a function of velocity. This heat exchanger modelling technique thus relies primarily on frontal velocity in order to calculate pressure drop of the entire heat exchanger.

Heat transfer modelling is similarly carried out, but relies entirely on either ascertaining test data of heat transfer as a function of velocity, or predicting the UA value, as in the $\epsilon - NTU$ model. Either a single stream model, assuming a constant or parametrically driven temperature in the non-present fluid region to calculate the appropriate local ΔT, or a dual stream model, using two fluid streams to provide a more representative change in temperature, similar to the calculations shown previously for the $\epsilon - NTU$ method in Sec. 2.2.3, are used to calculate the driving potential for heat transfer. This simplification provides a limitation in extracting detailed flow data from a multi-pass heat exchanger, and is primarily useful in the case of numerically efficient systems models, encompassing multiple heat sources or sinks.

Within the framework of STAR-CCM+, a user may specify data in the form of:

1. UA_L polynomial

2. UA_L table

3. UA_L map

4. UA_G table

5. Q table

6. Q map,

where UA is the heat transfer coefficient multiplied by reference area, the subscript L corresponds to local values, G corresponds to bulk or gross values, and the value of Q refers to total heat transfer. Use of a polynomial to describe the UA_L value allows a user to specify heat transfer coefficient as a function of velocity, table allows for tabulated values from test data, or data based on a continuous or discontinuous function, and map specifies UA with respect to the geometric location within the model. Specification of Q allows a user to explicitly define the total heat transfer as a function of bulk stream flow rates or local heat flux values.

Although these models are able to provide a lossless energy balance, they do not account for the detailed physics constituted in the thermal resistance equation, Eq. 2.6. As a result of these simplifications, detailed heat exchanger models utilising this modelling technique are neither accurate nor simple to construct if the geometry is complex and the primary flow direction is not aligned with the known performance of a heat transfer enhancement surface. This is particularly problematic when dealing with fluid recirculation or direction changes, or when calculating the performance of non-orthogonally aligned geometries.

The inherent limitations in the available models have driven industry users to develop proprietary analytical or numerical models to address the aforementioned shortcomings.

2.4.2 Reduced Order Modelling

The concept of Reduced Order Modelling (ROM) revolves around taking a complex concept or system and expressing it in a simple mathematical formulation that can easily be manipulated or used in a variety of applications, without the computational cost inherent in the initial formulation. This is *extremely* desirable when considering the full CFD analysis of a heat exchanger; the small scales of turbulent motion within the heat transfer enhancement surface are separated by several orders of magnitude from the length scales associated with the full heat exchanger, making resolution of the entire heat exchanger nearly impossible with currently available computational resources. Thus, application of some type of reduced order model, or submodel, for the heat transfer enhancement surface is desired, such that an accurate prediction of pressure drop and heat transfer may be obtained for a minimal computational cost.

Darcy-Forchheimer Law

The Darcy-Forchheimer law, which is a modification of the Darcy law, is a mathematical expression to correlate the pressure drop through a loosely packed, uniformly distribution bed of particles [46]. Although initially intended for geological engineering as a means to calculate the rate of gravity driven fluid flow through a bed of sand, gravel, etc., it was discovered that the Darcy law could be applied to low speed flow through complex media, such as flow across a bank of circular cylinders. As the flow velocity increases, the validity of the Darcy law starts to decrease due to the rise in significance of the pressure drag, or Forchheimer drag [47]. In the calculation of pressure drop, this method has been successfully correlated with numerical data [47], however, in this form, no provisions exist for the correlation of heat transfer to flow.

By using dimensional analysis Patankar and Spalding proposed a method in which the steady or transient thermal performance of a shell-and-tube heat exchanger could be predicted [13]. Application within a CFD code can be accomplished multiple ways, however, as an example, CD-Adapco's implementation accomplishes this through the specification of a solid phase thermal conductivity and specific heat capacity, blended with the fluid phase via the solid volume fraction, and combined with the Darcy-Forchheimer law. Thus, the porous media region is treated as a homogeneous zone with uniform fluid properties, that allows for the easy calculation of both pressure drop and heat transfer within.

Certain limitations exist in using this modelling methodology: although extremely efficient computationally, it inherently assumes that the characteristic length of the porous media region is significantly smaller than the characteristic length of the flow channel. As the two length scales approach each other, typical of the majority of CHE's, the validity of the homogeneous porous media assumption decreases, and results are no longer necessarily representative of the physical processes occurring within the porous region.

Due to the aforementioned limitations, the Darcy-Forchheimer law must be used judiciously. As it is not necessarily representative of the flow and heat transfer processes, it cannot be used as a stand-alone predictive tool, and may only be used iteratively as a “tuning” tool for conjugate heat transfer problems, i.e. following initial simulation, the thermal conductivity or porosity of the porous region is modified until the heat transfer and temperature results match experimental measurements or the results of other predictive methods, such as the $\epsilon - NTU$ method. Despite its limitations in heat transfer predictions, it is shown to be extremely useful in determining the pressure drop within a heat exchanger under isothermal conditions or used in conjunction with other thermal models for heat transfer problems.

Lookup and Reference Libraries

One of the simplest methods for including a reduced physical model into a numerical simulation is via a lookup library. In this method, reference inputs are calculated within the numerical model, then used as reference coordinates in a lookup library which subsequently provides the output. Although compiling the library may be time consuming, the calculation is only required once, then results can be quickly referenced for an efficient numerical solution that still incorporates some form of direct physical solution into the overall simulation.

One of the most common CFD applications of the lookup library approach is seen in the Laminar Flamelet Library combustion method [41]. For a given flame chemistry, a 1-D solution of flame temperatures and species concentrations is pre-calculated into a set of lookup libraries. The calculation of these libraries depends only on two variables: the scalar dissipation rate, χ, which is a measure of the mixing intensity in the flow, and η, which is the mixture fraction, or the proportion of fuel in a given fluid parcel. With these libraries, the CFD simulation need only provide η, the variance of η and the local χ in order to obtain combustion results [41]. Although the method has been successfully used in many applications, it does contain serious limitations which affect its general applicability, leading to more complex and computationally expensive models [48, 49]. Similar limitations present themselves in various ROM formulations, and it is up to the user to understand and work within model constraints.

Another novel implementation of the use of precompiled flow libraries is seen in [50]. In this report, the development of a response system for chemical, biological or radiological terrorism attacks is outlined. This response system relies on using pre-compiled wind flow simulations over the city of Vancouver to facilitate fast evacuation, quarantine and emergency response to any airborne threat originating from an arbitrary location. By having the wind flow simulations completed in advance, scalar transport of a threat can quickly be calculated in order to most efficiently determine the threat severity and appropriate response. Use of this method reduced simulation time from ∼800000 CPU-s to ∼1500 CPU-s, or 0.2% of the initial CPU time, for calculation of the toxin dispersion. The additional time gained from reduction in calculation time can then be used to enact an emergency response.

Similar application of this modelling approach could be used in various other physical systems where the selection of boundary conditions is limited to a relatively narrow range, such as in a chemical process model or heat transfer system.

2.4.3 Reduced Order Modelling in CFD

The concept of reducing the computational cost of a calculation within the confines of CFD has long been a focus of developers. Some of the earliest work on this subject is based around the development of turbulence models created by Spalding and Launder, with the k-*epsilon* turbulence model[42], or Smagorinsky with the concept of subgrid-scale modelling in LES [51]. More recent reduced order modelling has taken on a more holistic approach of replacing large portions of the calculation by either experimental or computational representations of a small portion of the problem, using techniques such as Proper Orthogonal Decomposition (POD), Dynamic Mode Decomposition (DMD), Eigensystem Realisation Algorithm (ERA), discussed in the following subsections, or the more commonly known Artificial Neural Network (ANN) discussed in Sec. 2.4.4

Proper Orthogonal Decomposition

Proper Orthogonal Decomposition is a method used to reduce the complexity of the numerical solution of complex equations, such as the Navier-Stokes equations in the modelling of fluid flows [52]. In this approach, the complex approach to turbulence modelling is characterised by collecting either experimental or simulated data and applying it to a simplified model, in which the statistical basis of the flow provides insight into the specifics of various flows, which are fully developed and can be well predicted from a statistical point of view.

This method has been successfully applied in the context of heat transfer by Alonso et. al. for predicting heat transfer in a backward facing step [53]. In this method, “snapshots” of the flow are provided for various combinations of boundary conditions, which are then are used by the POD model to predict unsimulated conditions. In this study, it is shown that computational cost for a simulation can be reduced to only 2-5% of the time for a full simulation.

This method has also seen other applications, such as [54], among others.

Dynamic Mode Decomposition

Dynamic Mode Decomposition is a method that aims to reduce the dimensionality of a problem by the use and incorporation of different snapshots of the system in an order to understand and characterise the temporal performance of a problem. Introduced by Schmid in 2009 [55], generalised performance and stability data are gathered to understand the operation modes and underlying physical processes of a system, which can then be used to reduce the overall dimensionality of a problem from a higher order system, governed by the Navier-Stokes equations, to one that can accurately be calculated with a reduced overall computational cost and complexity. The effect is, similar to POD, a significant reduction in the computational requirements of a solution to a given problem.

Eigensystem Realisation Algorithm

The ERA method provides a ROM that focuses on macroscopic system aspects, such as stability or operational modes, rather than focusing on microscopic details, such as exact flow separation location or pressure drop. Preliminary work on this ROM is shown in Juang & Pappa [56], and focused primarily on the development and understanding of control systems for which simple inputs could be provided, such as excitation frequency.

Applications in fluid dynamics typically fall into the category of whole-system characterisation, such as aeroacoustics [57], flow induced vibration [58], or unstable control systems [59]. Due to the focus on system dynamics rather than system performance, this model is not well suited for the inclusion in heat transfer analysis.

2.4.4 Artificial Neural Networks

The concept of an artificial neural network (ANN) is a biologically inspired network of interconnected computational elements, or neurons. This network attempts to mimic the way human brains learn and develop, loosely based on complex pattern recognition [60]. This pattern recognition based learning gives a greater flexibility to the logic of a computer, and offers a great simplification to the programming required to recognise and interpret a near infinite number of potential input patterns [60]. Applications of ANN range from speech and vision recognition systems to predictions of complex networks of inputs, such as stock markets or national economies [61], to robotics control systems [62]. The flexibility of ANN's make their application ideal for many real world problems where the span of input conditions is unreasonably large or diverse to be used with traditional programming methods.

Following the biological inspiration of the ANN, each network must undergo a learning phase, much as humans, or indeed any animal, must learn basic functions for survival, such as walking, running, hunting, etc. Although there are many different algorithms that can be used to facilitate learning, in the most general sense, a set of inputs is given to the ANN which is then used to calculate a set of outputs. The generated outputs are then compared to known outputs and the error is calculated, which is then used to modify the generation of a new set of outputs, which ideally have a reduced error. This method, commonly known as backpropogation [62], is an effective way of controlling the learning process of an ANN. However, similar to a child learning multiplication tables, this process may require many iterations before it is able to acceptably reproduce the desired output.

The Neuron

The implementation of an ANN relies on several different mathematical components which can be seen as an analogue to a corresponding physiological process. The basis for this that is commonly observed is the use of the artificial neuron. The generalisation is that a neuron takes multiple inputs and provides an output, based on some computational function, g. g acts as an activation function that provides some form of processing of input data, returning a bounded output typically between $0 \leq y(x) \leq 1$ or $-1 \leq y(x) \leq 1$, depending on the choice of activation function, g. This is represented mathematically as:

$$y\left(x\right) = g\left(\sum_{i=0}^{n} w_i x_i\right), \tag{2.22}$$

where x is a neuron with n inputs, weighted by the values of w_i. The function g conditions the output based on the selected function, which are commonly either the threshold/step function model, Eq. 2.23, the sigmoid model, Eq. 2.24, or the hyperbolic tangent model, Eq. 2.25[63]:

$$g(x) = \begin{cases} 1 \text{ if } x + t > 0 \\ 0 \text{ if } x + t \leq 0, \end{cases} \tag{2.23}$$

$$g(x) = \frac{1}{1 + e^{-2s(x+t)}}, \tag{2.24}$$

$$g(x) = \tanh(s(x + t)). \tag{2.25}$$

The values of s and t are shape parameters that affect the steepness and location of the centre of the function, similar to the influence of mean and standard deviation in a normal (Gaussian) distribution. A graph of the sigmoid function is shown in Fig. 2.17 below. This function, which is arguably the most common of the demonstrated functions, is a smooth differentiable function, which is significant for implementation of the training phase. The weighting functions w_i are applied to the input before reaching these activation functions.

Feedforward Network Architecture

The actual architecture of a standard, feedforward network is generally a simple connection of neurons in various layers, with an *input* and *output* layer providing a means of

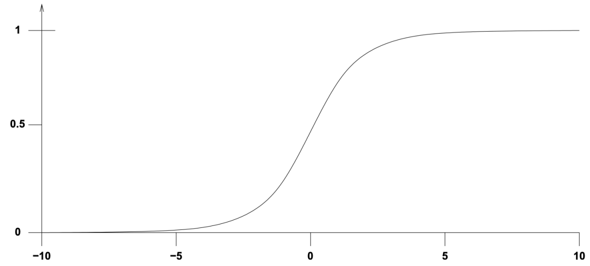


Figure 2.17: Sigmoid function detail, $s = 0.5$, $t = 0$ [63]

connecting the model to an external system. A sample of an ANN architecture is shown in Fig. 2.18.

In this architecture, there is a layer of input neurons, shown on the far left, which take

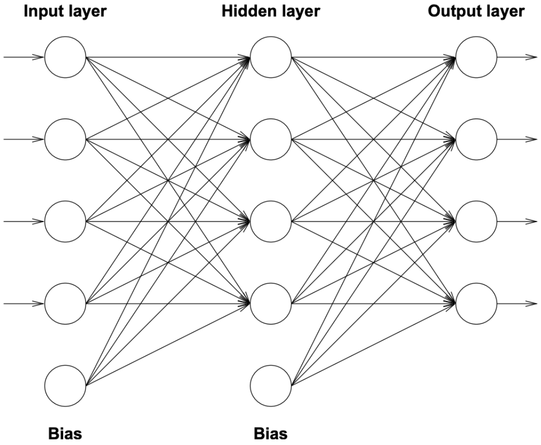


Figure 2.18: Schematic representation of an ANN, with a single hidden layer shown [63]

input values from the external system. Each of these values then has a weighting factor applied to it, and it is then fed forward to the hidden layers (hence the nomenclature of “feedforward network”). Each hidden layer may be constructed of n neurons, and m hidden layers may be used within a given network. Each neuron in a layer is connected to every other neuron in adjacent layers, where output is multiplied by a weighting factor, w_i. This is represented by the arrows shown in Fig. 2.18. Inclusion of a “Bias Node”, Fig. 2.18, serves the purpose of simplifying Eq. 2.24 and 2.25 [60, 62, 63] . Each bias node has no connection to any earlier layer (to the left, in Fig. 2.18) and simply injects a constant value into the following layer. This helps avoid numerical challenges, such as zero value inputs, and also to eliminate the need for calculating t in the sigmoid and hyperbolic activation functions [63].

The ANN functions by propagating any input values through the network and performing all neuron calculations to arrive at a series of output functions, which are the desired output from the ANN. Thus, the neural network is capable of providing relatively simple calculations of functions that can be highly non-linear [60]. Furthermore, it is hypothesised that most mathematical problems can be solved in 2 hidden layers or less, which avoids problems related to overfitting and training difficulties, which are discussed in subsequent sections [60].

ANN Training - Backpropagation

Perhaps the most crucial stage of ANN implementation is the "training" of a network to provide proper output data. Following the analogy of biological inspiration, a neural network is trained similarly to a child. A concept is "demonstrated" by providing input data, then the network attempts to provide output matching the desired output. Any errors are quantified, then the network revises itself and tries again. This is similar to a child learning to perform a task, where multiple attempts are required before a successful result is obtained. The more practice afforded to the network (child) the better the performance of the task.

In the case of ANN's, a primary difficulty may be obtaining enough appropriate training data for the network. To work with this limited dataset, the concept of backpropagation training, or supervised learning, is derived. The "backpropagation" element refers to comparing output data to known solutions for training data, and is thus heavily reliant on access to accurate training data. Other methods for training exist, but will not be further discussed in this context, as there is limited applicability to the present problem.

Backpropagation training is achieved first by assigning random values to the weighting functions, w_i between each layer. While this provides an opportunity for unreproducible results due to the inherent stochastic nature of the process, it also helps to eliminate bias and avoid the selection of local minima over global minima. This stochastic nature does provide room for error, and currently requires significant judgement when implementing the model to determine whether output is reasonable and accurate. Once the network is run for a set of input data, the error for each output neuron is calculated by:

$$e_k = d_k - y_k, \tag{2.26}$$

where y_k is the calculated output of the network and d_k is the known value from the training dataset. The error value e_k, is used in conjunction with the gradient of the the activation function, typically Eq. 2.24 or 2.25, to determine:

$$\delta_k = e_k g'(y_k). \tag{2.27}$$

The need for a smooth activation function, g is demonstrated by use of the activation function differential g', which becomes numerically problematic when step functions or other non-continuous functions are used in place of the Sigmoid or Hyperbolic activation functions [60]. The error values from the output layer are used to determine the error of the previous layer via the use of a learning rate, η, such that:

$$\delta_j = \eta g'(y_j) \sum_{k=0}^{K} \delta_k w_{jk}, \tag{2.28}$$

where K is the total number of layers present, and j corresponds to the j^{th} neuron in the layer. This error calculation is propagated throughout the entire network in order to characterise the slope, or learning rate.

Weighting values, w_i, are adjusted by a factor of Δw by the equation:

$$w_{jk,\text{new}} = w_{jk,\text{old}} + \Delta w_{jk}, \tag{2.29}$$

where the $_{jk}$ subscripts denote the node number and layer number, respectively. Δw_{jk} is determined using the equation:

$$\Delta w_{jk} = \delta_j y_k. \tag{2.30}$$

This process is iteratively repeated using the available training data until predefined stopping criteria are reached. The conclusion of training can be based solely on iterations, or can use more advanced metrics, convergence of mean square error to a predefined limit, or other more advanced methods.

ANN Overfitting

The concept of overfitting often refers to providing too few numerical constraints for the available training data. For example, if one data point is available and the equation of a line, $y = mx + b$, is desired, there exist an infinite number of solutions that pass through this nexus. Likewise, providing too many neurons or layers for the available training data may yield a situation where training residuals are extremely low, but the overfitted model fails to accurately predict real data that falls outside the exact values specified during the training process. Thus, a set of training data may be broken into two sets: one for initial training, and one (previously unused) tasked for validation of the trained model. Efficacy of the trained ANN is based on its ability to predict data *not* used for training, specifically the validation data.

ANN Summary

Although the ANN offers the advantage of computational efficiency once it has been trained, the training process may prove prohibitively expensive. Additionally, the network may only be trained using existing data sets, which may lead to unpredictable performance with new, or uncharacterised data that fall outside the range of learned data, analogous to an extrapolation calculation. Despite this, the potential for ANN's to learn patterns without preconceived restrictions, such as restrictions encountered when trying to fit test data to an assumed functional form, allows for remarkable flexibility when analysing and reducing data sets for use in a predictive model.

For the present research, the simplest ANN architecture is introduced, as the present problem does not require additional complexity afforded by other models

2.5 Summary

The methods and techniques presented in this section do not provide an exhaustive list of those available for use in the design and analysis of heat exchangers, but rather, provide a sound basis for understanding the remainder of the work presented herein. Further techniques not discussed in this report are not directly relevant to the proposed work at this stage, however, may be introduced and discussed briefly, if required.

Chapter 3

Turbulizer Numerical Modelling and Experimental Investigation

Based on the knowledge assembled within Ch. 2, this chapter provides an overview of the challenges associated with the understanding of small scale flow characteristics and heat transfer within the aforementioned heat transfer enhancement surfaces, particularly turbulizers. This portion of the study highlights manufacturing variability and numerical modelling difficulties, focusing on the specific challenges within each of these areas. As validation and expansion on the numerical modelling results, experimental investigation of the transition from steady laminar flow to turbulent flow is investigated using a combination of flow visualisation and laser Doppler velocimetry (LDV) techniques. The goal of this chapter is to provide a comprehensive investigation into the flow, modelling and overall performance of a typical turbulizer. A full understanding and use of a validated numerical model allows the present research to be applied for ROM purposes.

Specifically, this chapter provides an investigation into several key areas relating to the study of the Dana Canada turbulizer geometry:

1. **Manufacturing Variabilities and Sources of Uncertainty**

2. **Experimental Measurement Uncertainty**

3. **Numerical Modelling and Sources of Uncertainty**

4. **Current Numerical Modelling Results**

5. **Experimental Validation of Numerical Modelling Results**.

3.1 Manufacturing Variabilities and Sources of Uncertainty

As with any manufacturing process, the manufacturing of heat exchangers suffers from normal inconsistencies arising from variations in the manufacturing and forming of both the channel coreplates, the plate material isolating the individual fluid channels, and the heat transfer enhancement surface, or turbulizer. The brazing process used to join these components, however, introduces another process uncertainty that is extremely difficult to measure using non-destructive methods: the flow of the cladding, or braze filler material, during the brazing process. The cladding material flow is generally driven by capillary forces during brazing, and may cause cladding material to wick into small openings or gaps, as well as providing a "softening" radius to many otherwise sharp junctions. Furthermore, for the purposes of this study, the effect of coreplate forming tolerance is ignored, as it is relatively minor and has very little influence the overall performance of a given heat exchanger. Thus, the discussion of manufacturing uncertainties will focus on two major areas:

1. **Clad Flow and Blockage**
2. **Turbulizer Deviation From Nominal Dimensions**

Although fully eliminating these factors is largely beyond the reasonable control of any manufacturer, it is important to understand how these manufacturing variabilities can impact the heat transfer and pressure drop performance of a heat exchanger, especially during the preliminary design stages, where numerical tools are often relied upon to make engineering decisions for the specification of the specific heat transfer enhancement surface to be used. Thus, a generic treatment of geometry prior to analysis would significantly benefit the design process.

3.1.1 Clad Flow and Blockage

The process of brazing involves using a plate composed of a base alloy, typically aluminum based, with a thin layer, or cladding, of a slightly different alloy composing the outer 5-10% of the plate thickness and having a slightly lower melting temperature. Once the desired components have been assembled together and placed under a moderate clamping load, they are placed into a furnace which brings the assembly up to a specific temperature. The high temperatures within the furnace lead to melting of the clad material, while the base alloy remains solid. A schematic representation of a typical brazing material is shown in Fig. 3.1. Capillary forces draw the molten clad material into the joints between parts, and upon cooling, a single, solid piece is produced [64]. Accurately predicting and controlling the flow of molten clad can be problematic, where capillary forces are balanced, and sometimes overcome, by gravity. Poor clad flow may lead to localized pooling or uneven joint formation, leading to a partial or full solid blockage of flow channels. Any flow channel blockages are detrimental to both pressure drop and heat transfer performance and are extremely undesirable.

In an effort to understand the general effects of the brazing process, various studies have been undertaken to observe and characterise the shape and quality of the braze joints within a completed heat exchanger. Even with a well established brazing process, small variations in furnace temperature, environmental conditions or alloy composition, among other variables, can lead to a significant change in the shape and size of braze joints. Al-

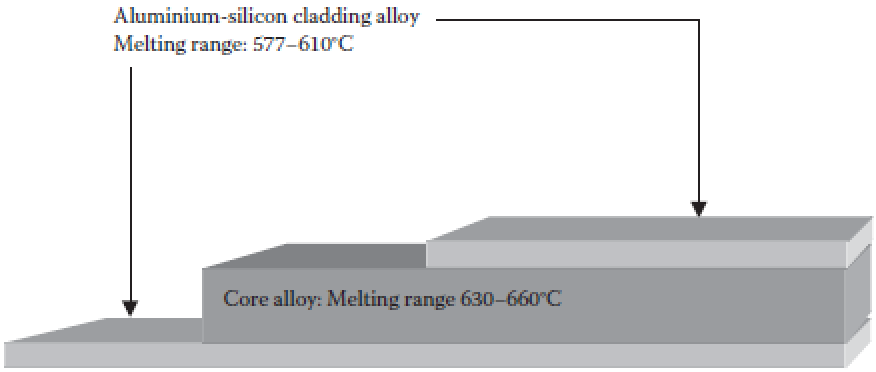
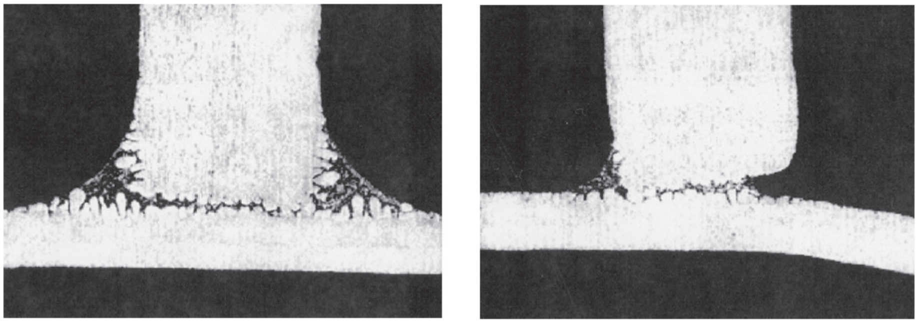

Aluminium-silicon cladding alloy
Melting range: 577–610°C
Core alloy: Melting range 630–660°C

Figure 3.1: Example of a typical brazing material, illustrating both core material and cladding [64]

though many of the past studies have focused on the structural performance of the braze joints, the geometric characteristics and information gleaned from these investigations has also been useful in determining possible origins for the variations observed in heat transfer and pressure drop performance.

In order to illustrate the potential difference between two braze joints in similar parts, the images in Fig. 3.2 a) and b) are shown. In Fig. 3.2 a), a typical braze joint can be seen. This braze yields a consistent fillet radius, with clad being pulled upward to fill the joint between the turbulizer and coreplate. This yields good strength as well as minimal thermal resistance in the joint due to larger continuous cross-sectional area, both of which are desirable in the manufacture of heat exchangers. Conversely, Fig. 3.2 b) demonstrates a poor quality braze. In this example, there is a poor thermal and mechanical connection between the coreplate and the turbulizer. This joint provides a structural weakness, which may lead to a mechanical failure, but additionally provides an increase in thermal resistance, leading to deviations from the nominal heat exchanger performance. Additionally, variations in the braze fillet radius may impact the flow structures resulting from the turbulizer, potentially leading to increased pressure drop or otherwise unexpected performance.

In some instances, particularly when dealing with low channel heights, clad flow can completely block a fluid channel, leading to a significant increase in pressure drop and decrease in heat transfer performance. In less extreme circumstances, the clad may only partially block a channel, but a deviation from nominal performance is observed nonetheless. Thus, the surface tension in molten clad material puts a practical limitation on the minimum channel height that can be reliably brazed without risking blockage, typically on



(a) High quality braze joint

(b) Low quality braze joint

Figure 3.2: Comparison of two different braze joints in similar products, in this case caused by a very small variation in alloy composition [64]

the order of 1 mm.

3.1.2 Turbulizer Deviation from Nominal

The process of forming turbulizers or other heat transfer enhancement surfaces produces some deviation from the nominal CAD dimensions specified by engineers. Understanding the limitations of this process are crucial to proper development of new or more efficient heat transfer enhancement surfaces.

Manufacturing of a turbulizer strip is performed by either stamping or rolling a flat sheet of material, typically aluminum. The die forms a set of alternating sinusoidal patterns by shearing and forming the flat sheet around the geometry in the die. There are several characteristic geometric features which are referred to in the turbulizer: the blade width, the pitch and height. The blade width, L, is defined as the width of each of the continuous sinusoidal strips formed in the die. The height is taken as the nominal distance between the highest and lowest point of the formed turbulizer strip, H. The pitch is nominally the distance for one full period of the sinusoidal pattern, s. The material thickness (or gauge) is represented by t. These measurements are shown representatively in Fig. 3.3.

One further important metric in the turbulizer, although not always directly referred to, is the blade offset. The turbulizer offset is defined as the shift between adjacent strips, or blades. An offset of 0% would be seen as corrugated metal strip since all convolutions line up perfectly with each other, resulting in no possible fluid passage, while 100% offset is defined as the blades crossing along the centre plane, as shown in Fig. 3.4. A 50% offset falls halfway in between the two, which is shown in Fig. 3.5. Despite tooling being designed for a nominal offset, various factors such as tooling wear, material quality, material thickness, etc., can lead to deviations from this nominal value. As a result, the gaps between adjacent turbulizer blades may vary dramatically, leading to partial or full blockage of the fluid flow path. Unsurprisingly, this variation may lead to heat exchanger performance drifting from the desired or tested values.

In addition to offset, the age and condition of forming dies can greatly influence performance of the turbulizer. As a turbulizer is formed from a single sheet of material, all material separations are created by shearing within the tooling. As the edges of the dies wear, the material cut becomes less of a cleanly sheared surface and more of a stretching or

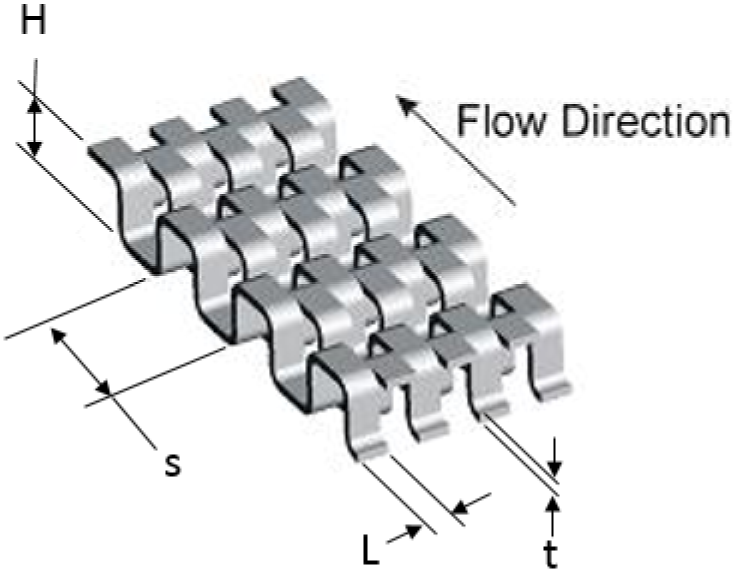


Figure 3.3: Measurement and characterisation of turbulizer geometry

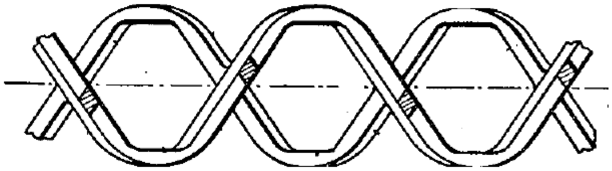


Figure 3.4: profile of 100% offset turbulizer [40]

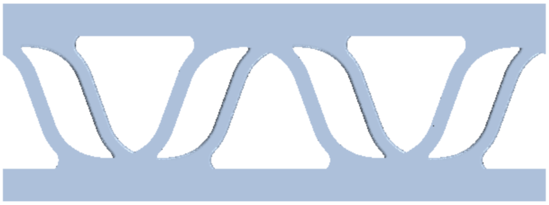
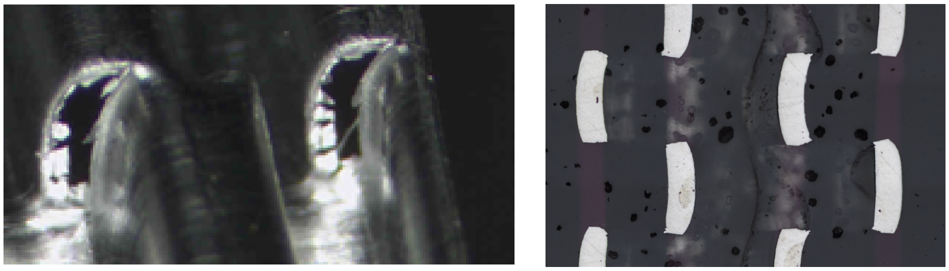


Figure 3.5: profile of 50% offset turbulizer, turbulizer blades evenly spaced along centreline

tearing of the material. The formation of burrs, rounded edges or curved turbulizer blades may result, as shown in Fig. 3.6 a) and b). As burrs are effectively slivers of material that form on shear surfaces, significant blockage is possible due to excessive burring, leading to reduced performance. In extreme cases, lack of a clean shear may lead to excessive material stretching without a physical cut being created, which can fully block fluid flow. The exact impact of rounded edges and curved turbulizer blades resulting from worn or misaligned tooling prior to the present work is not yet well understood.



(a) A typical burr blocking part of the fluid passage

(b) An example of rounded edges and curved turbulizer blades, sectioned at the turbulizer centreline

Figure 3.6: Manufacturing defects and deviation from nominal shapes

3.2 Experimental Measurement Uncertainty

Discussion on the quantification of experimental uncertainty is a broad topic, and a large amount of effort has been put forward to help standardise the analysis. The Joint Committee for Guides in Metrology (JCGM) has assembled a standard for the analysis and presentation of experimental uncertainty to reduce the discrepancies between publications [65]. The JCGM standard requires the root sum square (RSS) combination of all identifiable sources of uncertainty, however, identifying all sources of error remains up to the experimenters and can often times be difficult to appropriately or accurately quantify.

The macroscopic heat transfer and pressure drop testing of a given turbulizer geometry is typically performed by assembling a small test section of known dimension. An assortment of different fluids and flow rates are tested in order to provide some variation of fluid properties, including Re and Pr. Measurement of the total heat transfer, q, and the pressure drop, Δp, are performed via thermocouples and pressure transducers placed in fluid supply and return hoses. Mass flow rates are measured within the pumping equipment, and fluid properties are determined by an accredited external laboratory. Each of these measurements, ranging from heat exchanger geometric measurements to temperatures and fluid properties, is associated with some amount of uncertainty. Combination of these parameters using the RSS method specified within the JCGM standard yields the overall uncertainty of the experimental results. A full analysis is performed by Muzychka and reproduced with relevant modifications in Appendix A [39] for a nearly identical measurement methodology. The assumptions and calculations made by Muzychka are equally valid for the current study, and the results are summarised in Table 3.1.

Table 3.1: Experimental Uncertainty [39]

Variable	Uncertainty
$\overline{UA}$	5.19%
f	3.20%
j	7.3‘%
Nu	7.23%
Re	1.23%
Pr	0.87%

In the uncertainty values calculated by Muzychka, the thermal test results, i.e. $\overline{UA}$, Colburn j and Nu, suffer from large uncertainty due to a small temperature drop across the test section. Muzychka demonstrated that by changing the test parameters and increasing the test section temperature drop, the uncertainty could be more than halved [39]. As the uncertainty for $\overline{UA}$, j and Nu cannot be determined *a priori* without knowledge of the temperature change across the test section, Muzychka's calculations are used only for reference of typical values.

It is important to clarify that the experimental uncertainties calculated do not quantify the impact of variations in manufactured geometry, discussed previously in Sec. 3.1.

3.3 Numerical Modelling and Sources of Uncertainty

Similar to manufacturing variability, discussed previously in Sec. 3.1, numerical modelling also requires a variety of assumptions to be made which can lead to variations in obtained results. Although there are a variety of different areas within the simulation setup and solution of a numerical analysis, this section focuses on several key areas: turbulence modelling, assumed geometry, boundary conditions and further assumptions or model simplifications.

3.3.1 Turbulence Modelling

The topic of turbulence modelling within computational fluid dynamics is extensive, with a plethora of different models available following three commonly used modelling strategies: laminar, RANS and LES.

The laminar flow model is the simplest of the three strategies. It is assumed that there are no turbulent fluctuations present in the simulation, and therefore, the simulation is forced to preclude turbulent mixing. While this may be a good assumption for low Re flows, when the flow Re increases towards the transitional regime, some turbulent instabilities may develop, meaning that the model assumptions used in the simulation are no longer representative of the physics occurring within the physical domain. This is especially true of geometries that present sharp edges or discontinuities, analogous to trip wires [66], which promote the growth of flow instabilities and development of turbulent flow.

RANS type simulations present flow based on assumptions that are vastly different than laminar simulations. In RANS, it is generally assumed that the flow Re is sufficiently large that the flow is fully turbulent. As a result, the turbulence kinetic energy (TKE) is solved as a conserved scalar throughout the entire flow domain, which acts to model the diffusive effects of small scale fluctuations. Although there have been several low-Re turbulence models proposed, such as the low-Re $k-\epsilon$ and low-Re $k-\omega$ models, the numerical stiffness introduced into the model formulation may lead to issues with model convergence [41, 67]. Also, as a result of assuming a fully turbulent flow in the standard $k-\epsilon$ and $k-\omega$ formulation, these models tend to overpredict pressure drop and heat transfer due to the increase in turbulent mixing and boundary layer shape predicted numerically. Due to the increase in effective fluid viscosity, primarily contributed by the turbulent viscosity term, RANS models additionally tend to dampen any laminar unsteadiness that may otherwise be present in the modelled geometry.

LES represents a further improvement in the ability to model transitional flows over both laminar or RANS approaches. The formulation of LES necessitates an unsteady treatment, where the instabilities leading to turbulence and the evolution thereof can be much more accurately solved. LES only models the smallest scales of isotropic turbulence, leading to a reduction in the assumptions used during model formulation. Anisotropic eddy contributions are generally solved directly due to the filter size assumptions and unsteady nature of the simulation. It has been shown that in many circumstances, LES for a laminar flow will accurately converge to laminar behaviour due to the residual stress tensor approaching zero [28]. Although predicting the transition of a flat plate boundary layer has been somewhat problematic with LES [68], the transition to turbulence in more complex flows with irregular geometry tends to be predicted with more accuracy, likely due to instabilities naturally generated by the flow topography. Despite the benefits of LES in modelling complex transitional flows, the computational effort can be an order of magnitude or higher in comparison to RANS or laminar models.

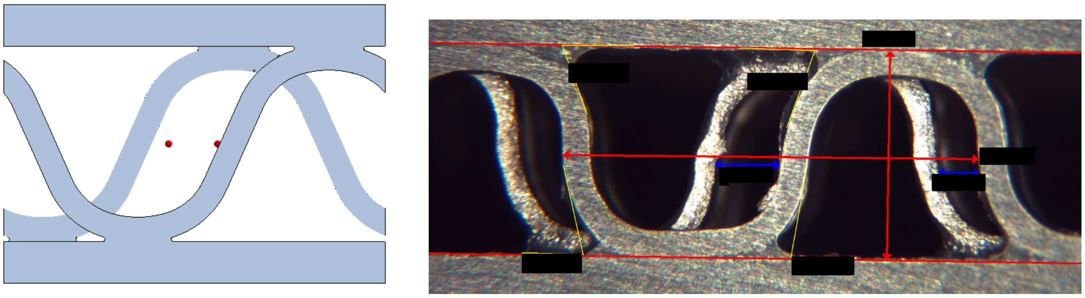
The nature of flows within turbulizers has not been studied, either experimentally or numerically, in enough detail to conclusively determine at which Re the flow begins to transition both from steady to unsteady flow and from laminar to turbulent flow. Thus, choosing the appropriate and most computationally efficient turbulence modelling methodology *a priori* becomes a challenge and impacts the overall accuracy of results and understanding of the flow behaviour. It is a goal of this research to provide insight into the transitional performance of turbulizers, as well as to provide data on the accuracy of modelling results when investigated experimentally.

3.3.2 Assumed Geometry

As previously discussed in Sec. 3.1, any manufacturing process has the potential of generating a product that possesses dissimilarities between the nominal CAD design and the as-manufactured product. Although attempts have been made to quantify the magnitude of these dissimilarities through the use of sectioning methods, very little simulation work has been performed to assess the variation in performance in heat transfer enhancement surfaces due to this manufacturing variation. Typically, CFD studies involve only the use of nominal geometry generated during the CAD process, especially when designing new turbulizer or fin designs. As these geometries require expensive tooling and are difficult to prototype, there are generally no post-braze samples available for inspection and measurement during the initial simulation phase, and thus a methodology for understanding the differences in nominal and as-manufactured geometry is required.

Although largely resembling their nominal counterparts, the as-manufactured components may exhibit slight variations in the minutiae of their shape, as exemplified in Fig. 3.7. In this figure, comparison between the nominal CAD and as-manufactured turbulizer highlights some of the particular challenges encountered when trying to perform numerical analysis on the geometry. Various feature changes, such as the curvature of the turbulizer surface, the radius of the braze fillet and jaggedness of any sheared surfaces may influence the formation of flow patterns, transition to turbulence or unsteadiness, or even the overall heat transfer and pressure drop measurements.

The overall impact of these manufacturing variations has not, prior to this study, been well documented. It is something that any manufacturing process will suffer from, to varying degrees, yet many numerical studies or scaled experimental studies performed on heat transfer surfaces largely overlook this effect [10, 69, 70].



(a) Section of nominal CAD geometry (b) Section of as-manufactured, post braze turbulizer

Figure 3.7: Examples of deviation from nominal geometry during manufacturing, of particular interest are the turbulizer curvatures and braze fillets

3.3.3 Boundary Conditions

Proper determination of appropriate boundary conditions remains a challenge for many thermal problems. In many cases, analytical solutions are typically available for *either* constant heat flux or constant temperature heat transfer boundaries [71]. Although these selections are attractive from the standpoint of providing a simple, closed form solution, it is often very difficult to replicate these boundary condition selections experimentally. Thus, numerical simulations must be created which do not follow either constant temperature or constant heat flux conditions, but rather some combination of the two.

The unfortunate consequence of this challenge is that a numerical simulation and experimental test will inevitably have a variation in the exact boundary condition applied, contributing to the accumulated discrepancy between simulations and test results.

3.3.4 Further Assumptions and Model Simplifications

When numerically modelling the performance of a full heat exchanger, the complexity of the heat transfer enhancement surface exceeds the availability of computational resources for all but the simplest of cases. In order to model most industrial heat exchangers, a simplification of the heat transfer enhancement surface is assumed, replacing it with a block of "porous media". This porous media approach, outlined briefly in Sec. 2.4.2, replaces the detailed heat transfer enhancement surface with a block of uniform media and applying the Darcy-Forchheimer law. Unfortunately, this method requires the assumption that the characteristic length scales of the heat transfer surface are much smaller than the characteristic length of the channel, which, in this case, is a poor assumption [46]. Additionally, the assumptions required to perform heat transfer analysis, as outlined by Patankar and Spalding [13], make the model unsuitable for *a priori* modelling, but rather are better suited for *a posteriori* modelling, where the coefficients can be adjusted to match the desired model performance, i.e. outlet temperature or target heat transfer, and only then can the model be used to assess the temperature distribution within the heat exchanger for conjugate heat transfer situations.

Due to these limitations, the porous media model, as it is currently formulated, is largely unsuitable for use in predicting heat transfer performance of CHE's, however, due to a lack of alternatives and the numerical efficiency of the model, it remains a popular choice for numerical modelling of CHE's. This approach is the primary method used by Dana Canada, with *a posteriori* parameter tuning performed by a combination of experimental results and analytical modelling tools. Although it cannot be explicitly known without disclosure from industry competitors, it is assumed that this is the current state-of-the-art approach.

3.4 Current Numerical Modelling Results

Following the outlining of manufacturing limitations, Sec. 3.1, measurement uncertainty, Sec. 3.2, and numerical assumptions, Sec. 3.3, the presentation of numerical modelling results follows. In this portion of the study, a detailed investigation of the numerically obtained results and correlation with experimental results, as well as a methodology and guidelines for generalised future study are presented. Topics of particular focus are: previous modelling results and how they were achieved, an investigation into the dependence of turbulence models selected during modelling, an investigation of the changes to the flowfield and its dependence on Re, characterisation of the as-manufactured geometry, dependence on any modifications made to the CAD geometry, and fluid Pr selection in an effort to improve or better understand predictive results.

These results provide the cornerstone for appropriate ROM modelling of the full heat exchanger. It is imperative that accurate results are obtained in order to represent the performance of a heat exchanger under a wide variety of flow conditions and fluid properties.

3.4.1 Previous Modelling Results

The numerical study of heat transfer enhancement surfaces within Dana Canada (formerly Long Manufacturing) began in 1990 with the evaluation of the CFD software TASCflow, a precursor to Ansys' CFX. In the preliminary study, the heat transfer of a single convolution of a turbulizer was compared with previous experimental results, which used naphthalene sublimation as a mass transfer analogy to the heat transfer process. Although results showed generally good agreement with experimental testing, the test method limited simulation Pr to a value of approximately 2. Unfortunately, this low Pr value used in preliminary simulations is not necessarily representative of the physics occurring at high Pr, and therefore cannot be extrapolated to all other fluids, such as engine oil. Also, due to the complexity and time requirements of setting up and solving a numerical simulation, a very limited number of flow rates were tested at the time [72]. Although the results were far from comprehensive, it did offer a glimpse at the potential of CFD for modelling turbulizers numerically.

Largely thanks to the improvement in CFD codes and the availability of ever-increasing computational resources, recent modelling attempts, completed using CD-Adapco/Siemens' Star-CCM+, have been able to encompass a much wider range of flow rates and fluid properties. However, despite the increase in conditions used within the simulations, there still appears to be, for unknown reasons at the outset of this project, a significant discrepancy between experimental test results and numerical simulations. Fig. 3.8 shows the discrepancy observed previously between numerical results and several different correlations based on test data. In this figure, it can be seen that the numerical results significantly overpredict heat transfer in comparison to the experimental data for the selected range of test conditions. It should be noted that in order minimise the dependence on fluid properties, all heat transfer data are presented in the form of $\frac{Nu}{Pr^{0.37}}$, with the exponent 0.37 historically found by Dana to provide the best data collapse for different Pr.

The most recent modelling attempts, corresponding to the data points shown in Fig. 3.8, were accomplished using the CFD software Star-CCM+ with a computational domain consisting of three different sections: a rectangular inlet region, specified with slip walls, 5 turbulizer convolutions in the flow direction with constant temperature walls and a solid domain to account for conduction effects, and an outlet region with slip walls to minimise gradients at the exit boundary. The purpose of the inlet and outlet regions is to prevent boundary conditions from propagating upstream or downstream and impacting the results. The flow domain is 2 blades wide, and periodic conditions are placed at the side boundaries. This configuration is shown in Fig. 3.9.

Although this domain setup has been in use for several years by Dana, recent investigation determined that significant entrance and exit effects were present while the flow was developing within the turbulizer. Variations in calculated Nu are shown on a convolution-by-convolution basis in Fig. 3.10. In this figure, it can be seen that there is

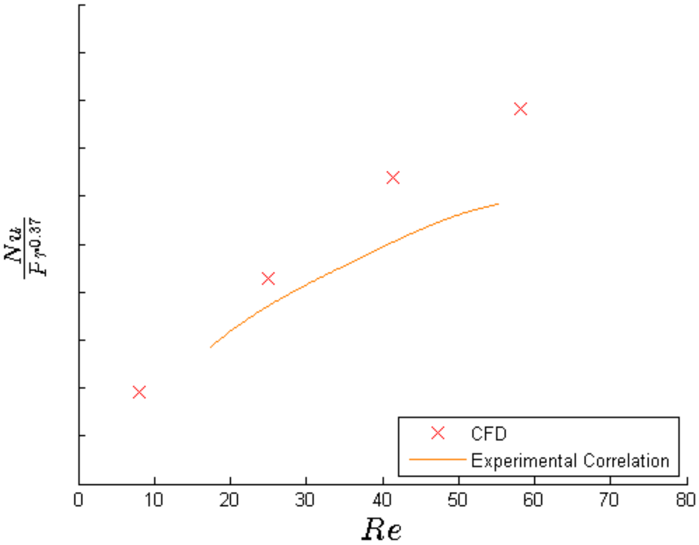


Figure 3.8: Comparison of previous modelling work to experimental data

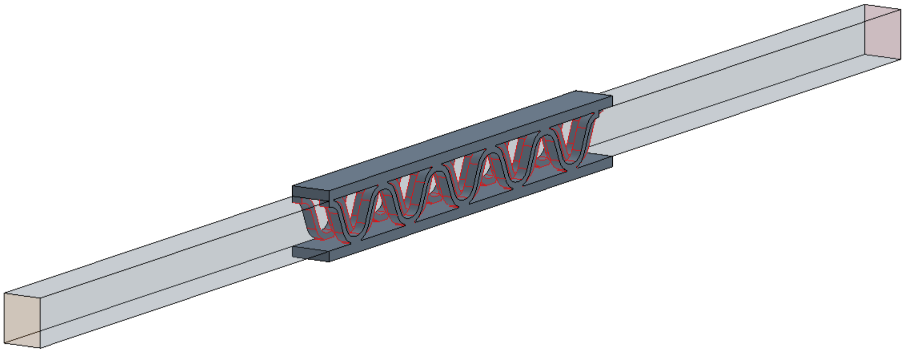


Figure 3.9: Computational domain used in previous simulation efforts consists of 5 convolutions in flow direction

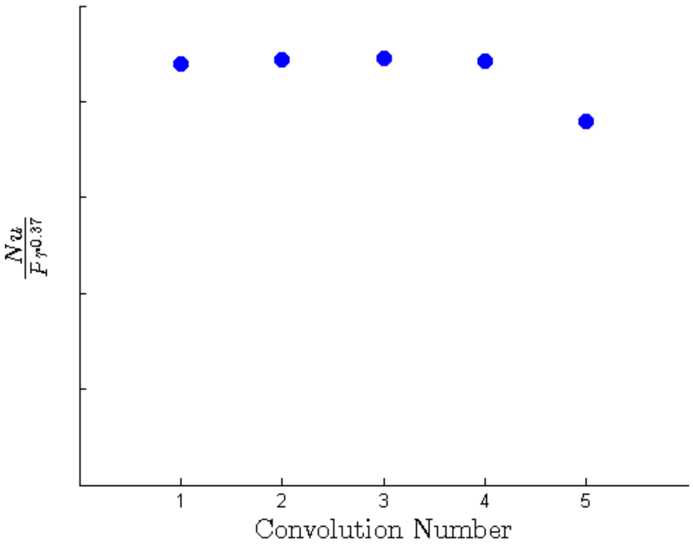


Figure 3.10: Variation of heat transfer predictions in a 5 convolution numerical model

a moderate entrance effect for the first convolution, however, the most noticeable impact actually occurs for the final convolution, where exit effects lead to a significant reduction in predicted heat transfer. As results are commonly taken as the average heat transfer across all 5 convolutions in the numerical domain, it is apparent that an improvement in modelling methodology is required in order to elucidate more accurate numerical results.

Additionally, in the past it has been assumed that relevant oil flows are typically laminar, dictating the use of a laminar flow solver, and coolant flows are either transitional or turbulent, and a steady state $k-\epsilon$ RANS turbulence model is used. To this date, no comprehensive study on the time dependent flow characteristics or the impact of turbulence modelling have been performed, and it is therefore a relative unknown as to how much the choice of turbulence models affects final results. It is expected that, while there is minimal error in modelling a fully laminar flow, the turbulence model selection may have a significant impact on results if the flow is transitional or turbulent.

Furthermore, all previous results have also exclusively used nominal CAD data: no geometeric modifications are performed to the turbulizer edges, channel heights, surface roughness or material thickness. The only modification performed to the initial geometry is the addition of a small braze fillet between the turbulizer surface and the adjacent coreplate, which can be seen in Fig. 3.7 a). Once again, the effects of this braze fillet have not been fully characterised, and when compared to section views of real geometry, such as in Fig. 3.30 b), it can be seen that this braze fillet is significantly undersized.

Post-processing of the simulation data is performed using the mass-flow averaged inlet and outlet temperatures, the average mass flow through the domain and the external wall boundary temperature, which is held constant. During post processing, fluid properties, ρ, k, μ and C_p are evaluated using the average film temperature method, where the properties are assigned at the average between the wall temperature and the average fluid temperature in the domain, i.e. $T_{film} = \frac{\frac{T_{in}+T_{out}}{2}+T_{wall}}{2}$ [73]. Simulations, however, are completed using variable fluid properties. In calculations of Nu, the log-mean temperature difference (LMTD) is corrected for the effects of conduction through the coreplate thickness by taking the average heat flux normal to the surface and calculating the temperature difference through the plate, i.e. $q = -k\frac{dT}{dx}$, with dT being the temperature difference across the plate and dx being the plate thickness. This method isolates and removes the effects of the plate thickness and allows for the data to be used in a much more general manner. All calculations in the following sections are performed using the same methodology and assumptions, unless otherwise noted.

3.4.2 Domain Dependence

As it has already been shown, previous modelling attempts have been somewhat plagued by dependence on the computational domain size selection in addition to entrance and exit effects. Choosing the proper number of convolutions for simulation, i.e. the length of the domain in the flow direction, presents a challenge: too few, and the model will suffer from overwhelming inlet and outlet effects, as shown previously in Fig. 3.10, too many, and the computational expense becomes unmanageably large. In order to simulate what can effectively be considered an infinitely long domain of identical convolutions, periodic boundary conditions are applied to the inlet, outlet and side boundaries. In comparison to previous analyses, the domain is then trimmed, eliminating the slip zones on both the inlet and outlet side so that only one convolution is encompassed within the computational domain. These modifications have the effect of not only decreasing the total number of computational cells, but also having the effect of infinitely increasing the flow length. This domain is shown in Fig. 3.11. The periodic boundary allows for both velocity and temperature to be scaled from the outlet to the inlet surface, maintaining a specified average inlet temperature and mass flow rate while capturing the velocity and temperature profiles present at the boundaries. Further details about the numerical implementation of the boundary condition may be found in the Star-CCM+ User Manual [45].

As a comparative study, the model results presented in Fig. 3.10 are directly input to the single convolution, periodic boundary model. Virtual planes were used to extract flow and temperature data from the 5 convolution model, which was then used for inlet temperatures for the single convolution model, allowing for a separate simulation at each of the 5 operating points seen by the individual convolutions. A significant reduction in variation between convolutions is observed when this methodology is applied, as demonstrated in Fig. 3.12. In fact, less than 2% variation is seen on the single convolution model, while the 5 convolution offers a spread of more than 15% on a convolution-by-convolution basis. The differences between the periodic model are seen to be minimal when compared with the heat transfer predictions of the central convolutions, producing less than 3% difference at convolution 3, indicating that the supposed periodicity is a good assumption of the actual physics occurring within the turbulizer. It is possible that this difference may decrease as

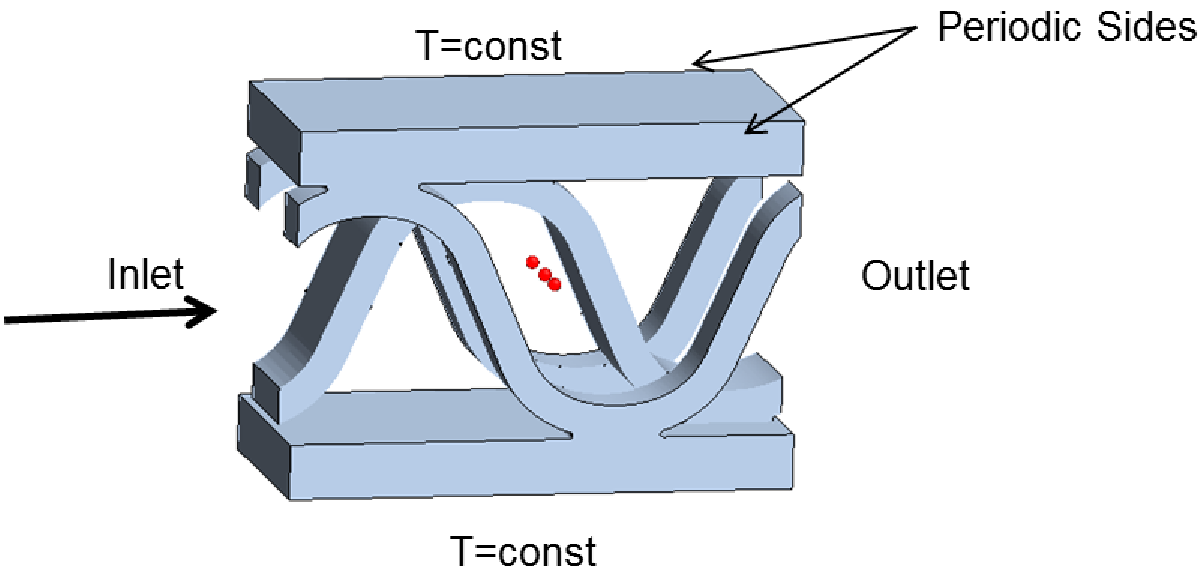


Figure 3.11: Single convolution computational domain and applied boundary condition types

further convolutions are added to the model, however, for the current purposes, the results show a sufficient improvement to warrant adoption of this methodology.

Although the previous comparison demonstrates that the single convolution is much more consistent than the previously used 5 convolution model, some uncertainty still exists as to whether large scale coherent, vortical flow structures form across multiple convolutions of the turbulizer, especially when considering turbulent or unsteady flows. In the case of an unsteady flow, the size of the single convolution domain limits the maximum size of larger coherent structures, since any flow patterns exiting the domain outlet are immediately mapped to the inlet. In order to ascertain whether or not this is occurring, or more specifically, whether this has an impact on heat transfer and pressure drop performance, the single convolution model is expanded by both a factor of 2 and a factor of 3 in both the streamwise and spanwise directions, resulting in what will be referred to as a 2x2 and 3x3 domain (referring to the multiples of the original domain that are present). When expanding the domain in this fashion, the numerical efficiency of the model quickly decreases, as the number of computational cells in the mesh increase proportional to the square of the expansion factor. An example of the 1x1 and 2x2 domains are illustrated in Fig. 3.13 for comparison.

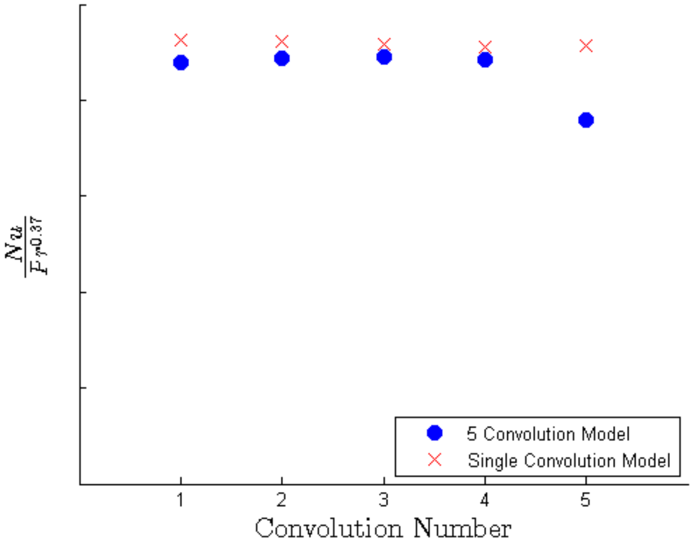
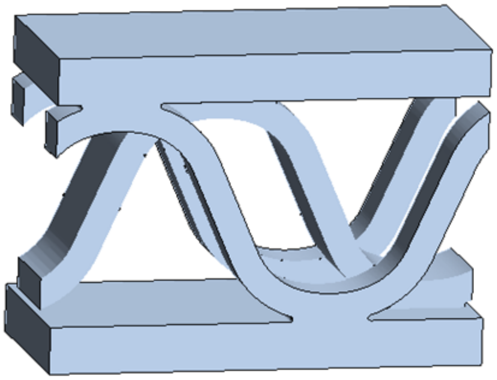
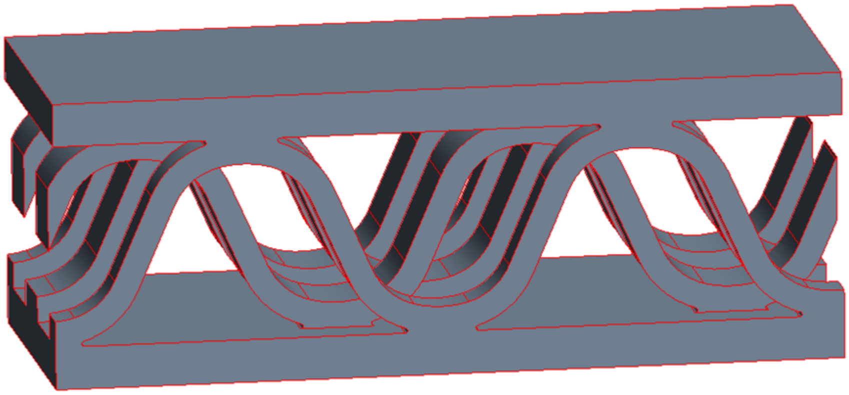


Figure 3.12: Comparison of heat transfer measurements from single convolution model and 5 convolution model

In order to test the dependence of heat transfer performance on domain size, two



(a) Example of nominal CAD 1x1 domain size



(b) Example of nominal CAD 2x2 domain size

Figure 3.13: Comparison of 1x1 and 2x2 domain sizes for simulation

different test points are selected for comparison. At each of these test points, matching flow conditions and fluid properties are selected in order to minimise sources of potential measurement uncertainty. Results are shown in Fig. 3.14. It can be seen that the $k-\omega$ model provides slightly more consistent performance at higher Re, where the $k-\epsilon$ model provides more consistent performance at the low Re test point. In either case, the maximum variation in heat transfer is approximately 3% for a given turbulence model and flow conditions. This difference is on par with the variability associated with inlet temperature conditions, as previously illustrated in 3.12. Therefore, there is no benefit seen in using a larger computational domain, and thus, the single convolution is used for the remainder of the study.

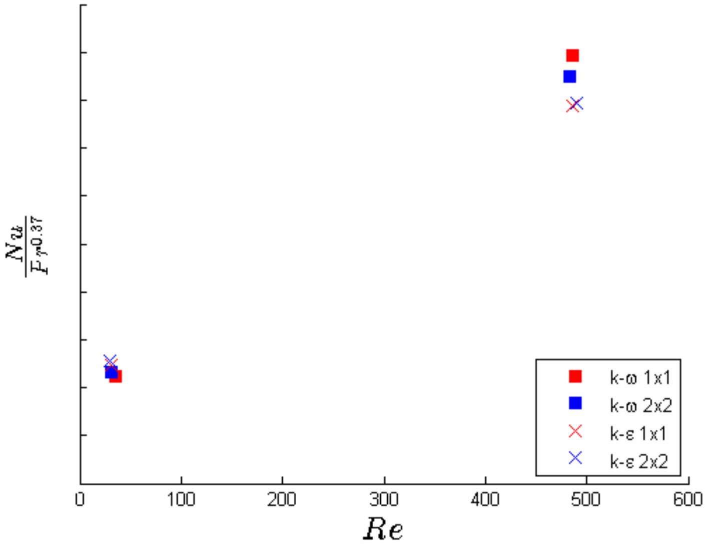


Figure 3.14: Comparison between results for 1x1 and 2x2 geometry, only minor differences exist, mainly attributed to the change in fluid properties through the domain

3.4.3 Turbulence Model Dependence

Selection of an appropriate turbulence model is perhaps one of the most difficult decisions to make when performing a simulation of a turbulizer. Although performance of the $k - \epsilon$, $k - \omega$ and LES models have been well documented for cases such as spreading of a jet or external aerodynamics [41], internal flows, such as that produced by a heat transfer enhancement surface, are not as well understood, largely due to the variety and complexity of such flows, but also relating to the proprietary nature of many heat transfer enhancement surfaces. Furthermore, when dealing with low-Re flows, particularly those that encounter the transition to turbulence, the selection of an appropriate model becomes even more critical due to the difficulties in modelling the transition to turbulence.

This section focuses on the comparison between 3 different turbulence models for the simulation of a turbulizer. The selected models include the standard $k - \epsilon$ RANS model,

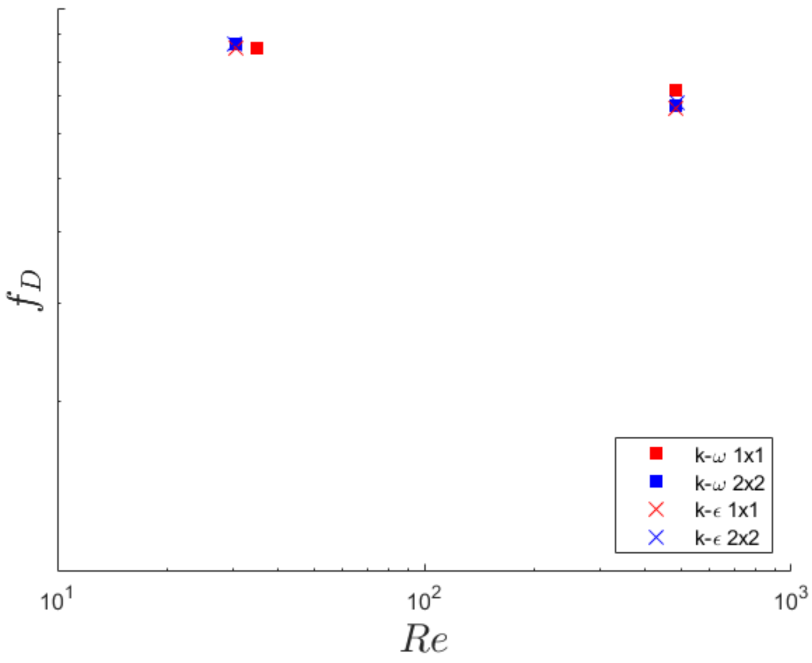


Figure 3.15: Comparison between results for 1x1 and 2x2 geometry for friction factor, very similar results are observed for all models, with the choice of turbulence model contributing more than domain selection

the SST $k-\omega$ RANS model, and the dynamic Smagorinsky subgrid scale LES model. The RANS models are primarily used in a steady state CFD solver, while LES is necessarily run in an unsteady solver. The relatively wide variety of turbulence models is selected in order to provide a good cross section of capabilities, as it is not currently known whether the flow is laminar, transitional or turbulent, and steady or unsteady. One particular area of concern is the performance of RANS models if the flow is determined to be either unsteady or transitional, whereas LES has the inherent capabilities to deal with either situation without *a priori* knowledge of the flow characteristics.

For the purposes of comparison of the aforementioned turbulence models, four specific flow conditions are selected and tested, as shown in Table 3.2. The lower *Re* conditions, nominally $Re = 8$ and $Re = 39$, are performed under cooling conditions, i.e. heat being transferred out of a hot oil, while the two higher *Re* conditions, 119 and 476, are representative of a coolant being heated, i.e. heat being transferred into the flow. It is important to note that the mass flow rates and *Re* initially calculated are based solely on inlet fluid properties, and are therefore modified based on the average film temperature assumption following collection and analysis of data.

Table 3.2: Flow conditions

	Re = 8	Re = 39	Re = 119	Re = 476
Fluid	ATF	ATF	Coolant	Coolant
Inlet Temperature	120°C	120°C	80°C	80°C
Wall Temperature	100°C	100°C	100°C	100°C

$k-\epsilon$ Model

The $k-\epsilon$ model is the most dissipative of all the turbulence models tested in this study. The increased dissipation is realised through a higher turbulent viscosity effected through the model implementation. As a result, the $k-\epsilon$ model tends to suppress any unsteadiness in comparison to other models, which can be especially detrimental when trying to predict the performance of an unsteady flow, or more specifically, the point at which transition to unsteadiness occurs. Qualitative observations of simulation results indicate that the $k-\epsilon$ model predicts a fully converged, steady flow solution under the full range of flow conditions. Thus, the converged solutions for all tested *Re* facilitate clear and easy comparison of developing flowfield results, which is further explored in Sec. 3.4.4.

In comparison to the experimental correlation derived from the most recent test data, the results of the $k-\epsilon$ model are seen to overpredict heat transfer at low Re, while underpredicting heat transfer at high Re, shown in Fig. 3.16. The suspected cause of the intersection of these two curves is related to the handling of turbulence within RANS models. As these models are documented as being poorly suited to predicting laminar or transitional behaviour, it is most likely a numerical "forcing" of unrealistically high turbulent kinetic energy, even at low Re, which subsequently leads to the overprediction of Nu. Conversely, performance at high Re, namely at $Re = 476$, is seen to slightly underpredict the extrapolation of the experimental curve. Although this falls slightly outside of the experimental test range, deviation of the results from the trend of results could be attributed to one of several key factors, such as misrepresentation of geometry (due to manufacturing uncertainties), turbulence model performance or even experimental error.

Pressure drop results have been generally acceptable in previous modelling attempts using the $k-\epsilon$ turbulence model. It generally follows the experimental correlation used by Dana Canada, as shown in Fig. 3.17, and has returned accurate overall predictions in previous modelling attempts. The general trend is a slight to moderate underprediction, approximately 10-15%, throughout the entire Re range.

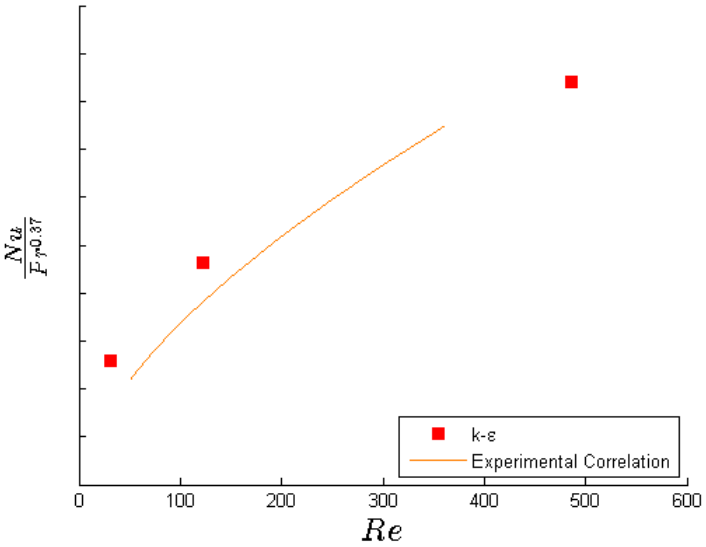


Figure 3.16: $k-\epsilon$ model results

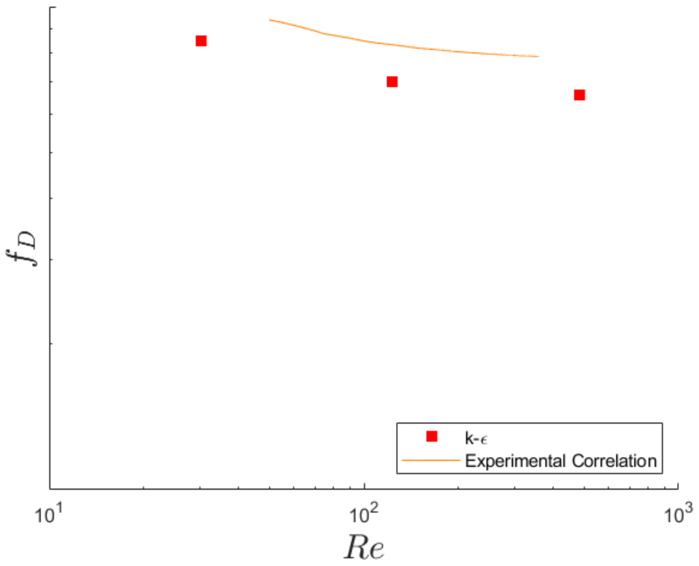


Figure 3.17: $k - \epsilon$ model results, f_D

$k - \omega$ Model

Development of the SST $k - \omega$ turbulence model was targeted at addressing some of the concerns with the $k - \epsilon$ implementation, particularly near-wall and boundary layer flows [41]. In the case of an internal flow, such as that in a compact heat exchanger, the SST $k - \omega$ model seems particularly well suited. As such, it is chosen as an alternate RANS based turbulence model for comparison to $k - \epsilon$. The SST $k - \omega$ model is generally less dissipative than $k - \epsilon$, and is therefore expected to predict earlier transition to unsteadiness.

Under identical boundary conditions to those used previously, the $k - \omega$ model is seen to predict a significantly earlier transition to unsteadiness. At $Re = 119$, the flow is observed to be fully unsteady, with any fluctuations receiving minimal damping. It is also at this Re that the $k - \omega$ model demonstrates the poorest agreement with the experimental results, with heat transfer predicted to be 60% higher than that seen experimentally, shown in Fig. 3.18. Interestingly, at both $Re = 119$ and $Re = 476$, the $k - \omega$ model is observed to predict higher heat transfer than the $k - \epsilon$ model. It is likely that the unsteadiness in the flow domain partially contributes to the higher predictions in heat transfer.

Overall, agreement of numerical and experimental results shown in Fig. 3.18 is not sufficiently strong that the use of this model can easily be justified. However, one area of significant interest that requires further investigation is the $k - \omega$ model's prediction of transition to unsteady flow.

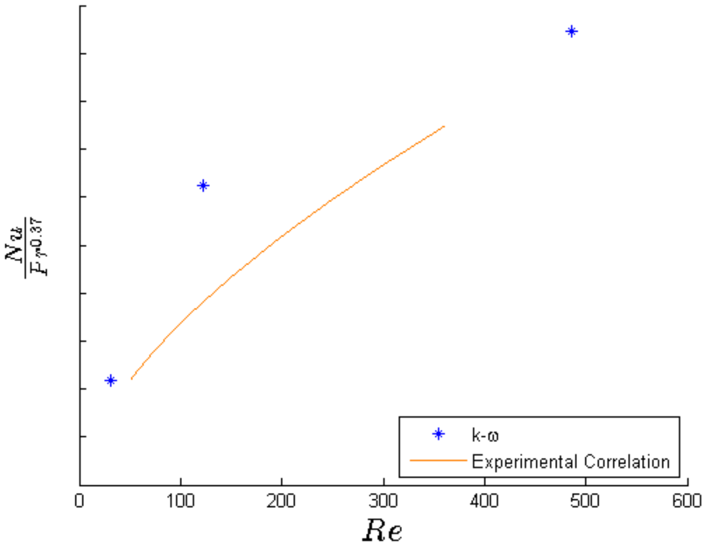


Figure 3.18: $k - \omega$ model results

Although there is a substantial difference in the physical response predicted with the $k - \omega$ turbulence model, namely the pattern of unsteady flow patterns, it once again provides a good correlation with experimental data, with only a slight increase to the f_D predictions of the $k - \epsilon$ model, as shown in Fig. 3.19, which reduces the overall error and brings it more in line with experimental test results.

Large Eddy Simulation

Largely due to the increased temporal and spatial resolution required for grid independent results, LES is often assumed to be the most accurate of generally applicable turbulence models (direct numerical simulation, or DNS, is often accepted as an "exact" solution,

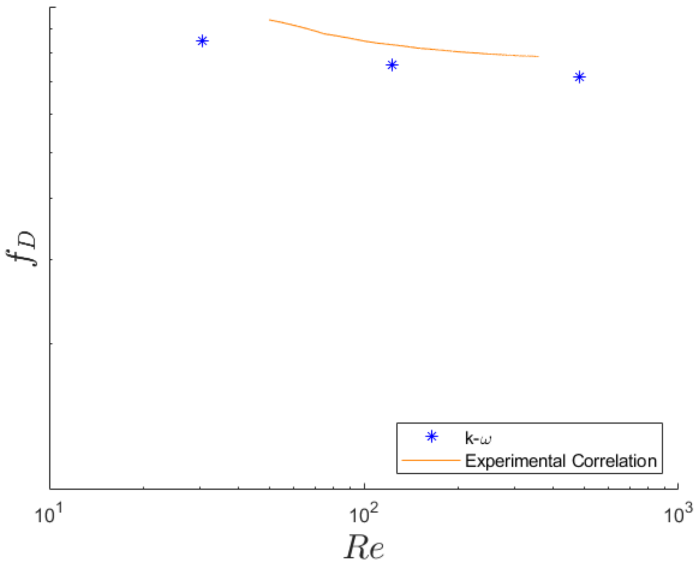


Figure 3.19: $k-\omega$ model results, f_D

but astronomical computational expense often make it impractical or impossible to use). Despite the increased resolution, the advantage of LES lies in the fact that with a sufficiently fine grid, only the smallest energy dissipating, isotropic eddies are modelled, while the larger anisotropic structures are solved. The reduction of reliance on specific assumptions during model formulation, such as isotropic turbulence, leads to a turbulence model which is able to much more accurately resolve highly anisotropic flows. Additionally, the filtering used in LES allows solutions, in many cases, to converge very closely to the laminar solution for low Re flows. Subgrid scale models specifically developed to account for near-wall behaviour also allow for excellent performance in either laminar, transitional, or fully turbulent internal flows, such as those encountered within a heat exchanger.

The flexibility in simulating a wide range of Re afforded by LES provides the most confidence in the present flow conditions out of any of the turbulence models tested. The onset of unsteadiness is predicted at some point between $Re = 39$ and $Re = 119$, with further simulations required to determine the exact point of transition. Upon reaching $Re = 476$, the flow is observed to be fully turbulent, with the turbulent energy spectrum following Kolmogorov's -5/3 power law [74, 75], shown in Fig. 3.20. Agreement with experimental results is generally good overall, with the maximum deviation of approximately

20% found at $Re = 119$, while agreement above and below this point is generally improved, as illustrated in Fig. 3.21.

As the transition from steady to unsteady flow is of particular interest in this study, additional simulation results are required to determine the approximate Re at which the transition occurs, and whether it is sensitive to variations in geometry or flow conditions.

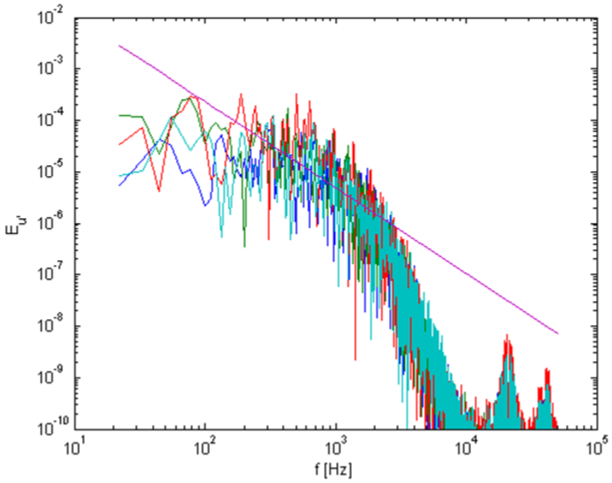


Figure 3.20: The turbulent kinetic energy cascade measured from LES, with a reference line of $\frac{-5}{3}$ slope shown for reference, $Re = 476$

Due to the rigorous modelling methodology of the LES model in comparison to the simplifications assumed for the $k - \epsilon$ and $k - \omega$ RANS models, it is generally understood that LES is better able to predict both transition to unsteady flow as well as f_D than corresponding RANS methodologies. The results shown in Fig. 3.22 closely follow the predictions made by the $k - \omega$ model, and provide no degradation in the ability to predict pressure drop of the geometry. The benefit of the LES model is therefore in its ability to accurately predict heat transfer performance, rather than significant improvements in pressure drop prediction.

Turbulence Modelling Summary

As a final summary of all turbulence models investigated, results are combined into a single plot in Fig. 3.23. In this figure, the areas of strength and weakness of each model

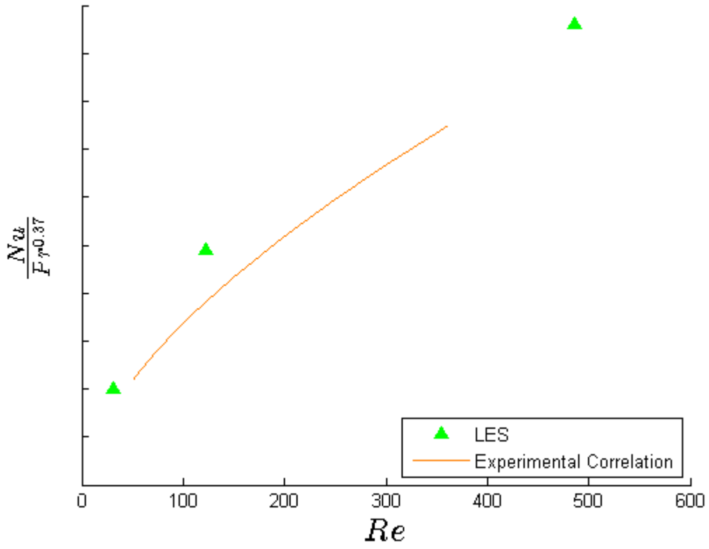


Figure 3.21: LES Results

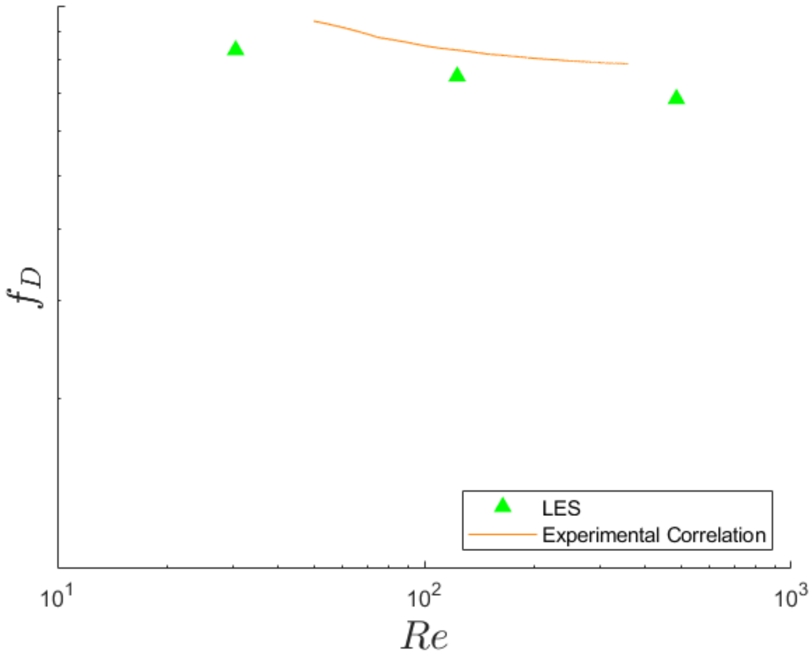


Figure 3.22: LES Results, f_D

are seen. The overall performance of the LES model appears to be in the best agreement with experimental results, however, an area of generally perceived weakness of all models appears to be in the moderate Re range, particularly $Re = 119$. It is possible that this Re range is approximately where some type of flow transition occurs, although whether it is a transition from steady laminar to unsteady laminar flow or laminar to turbulent flow must be determined experimentally. As both $k-\omega$ and LES predict a transition to unsteadiness below $Re = 119$, there is, however, strong evidence to support the transition also occurring experimentally. This transition cannot be fully confirmed, however, until detailed experimental measurements are available.

Despite providing the least overall agreement with experimental data, the $k-\epsilon$ results

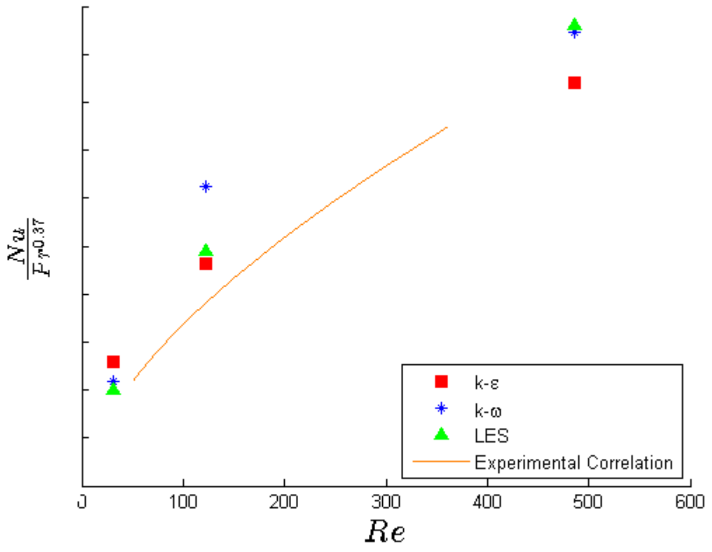


Figure 3.23: Comparison of all numerical results to the experimental correlation, significant overpredictions observed for moderate Re

Figure 3.24: Comparison of all numerical results to the experimental correlation

are particularly useful in that a steady flowfield is predicted. The availability of steady results permits a useful comparison of the evolution of flow structures. This is addressed in Sec. 3.4.4.

As shown in Figs. 3.17, 3.19 and 3.22, all turbulence models predict a generally accurate representation of f_D. At low Re, the steady nature of the flow is well represented and low levels of error are seen, however, as Re increases, the discrepancies in the models begin to reveal themselves. The difference between the steady results of the $k-\epsilon$ model predict a lower friction factor in comparison to experimental data, but it is not until the highest Re tested that the results of the $k-\omega$ model and LES diverge, with the LES model delivering intermediate results for f_D. The following sections will delve further into some of the reasons for these changes, as well as a further investigation into the effects of modifications to the geometry to represent the manufacturing process.

3.4.4 Reynolds Number Dependence

In order to better understand exactly how a heat transfer enhancement surface achieves its level of performance, it is useful to evaluate the specific changes in flow which occur at different Re. Although the flow within a turbulizer is specifically intended to be complex and tortuous, and by its very nature is extremely difficult to understand, detailed examination of the evolving flowfield provides insight into why a turbulizer is an effective heat transfer enhancement surface, as well as how it may be possible to extract higher levels of performance. This section evaluates the evolving flowfield formed by the turbulizer and documents how it changes with Re. Comparisons are primarily completed using the $k-\epsilon$ model, as the steady predictions lend themselves well to comparison. Although this study pertains primarily to heat transfer, temperature plots do not lend themselves to easy interpretation, and thus are presented separately in Appendix B for the interested reader.

The first set of images, taken through the domain centreline of the single convolution model, as demonstrated in Fig. 3.25, shows the evolution of the velocity field as Re is increased from 8 to 476. The first image, $Re = 8$ in Fig. 3.26 a), shows a flow structure reminiscent of potential flow. The flowfield, which is completely symmetric from side to side at this location, quickly fills in the wake area of the turbulizer blades. This lack of separation is characteristic of low Re flows, and as a result, pressure drop is almost entirely dictated by viscous resistance instead of inertial contributions, such as pressure drag. As Re increases to 39, Fig. 3.26 b), the flow separation clearly occurs at the leading corner of the turbulizer blade. The separation region grows in size significantly, and some recir-

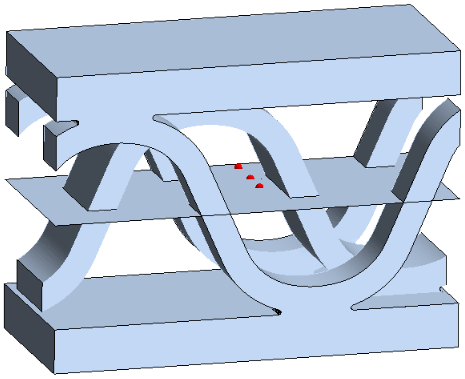
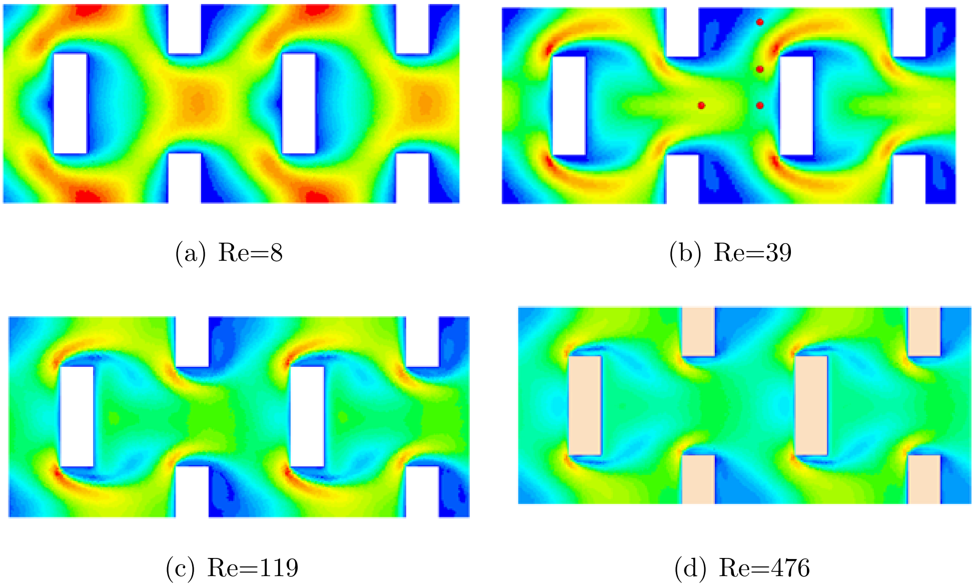


Figure 3.25: Illustration of centreline plane used for velocity and temperature measurements

culation can be seen forming behind the blades at the centre of the image. It is also clear that, despite the centreline image showing an equal spacing between all blades, there is, in fact, a distinction between a leading and trailing blade, as can be seen in Fig. 3.25. Accordingly, the flowfield forming in the wake region of the blades can be distinguished from one another, with the leading blade inducing more recirculation and higher local velocity magnitudes, while the trailing turbulizer blade forms a much lower velocity wake area. The increased recirculation is expected to contribute to additional heat transfer, as both sides of the blade provide effective heat transfer area, whereas the low recirculation area zone is essentially stagnant, providing little, if any additional heat transfer. For $Re = 119$, the flow patterns forming around the turbulizer blade become much more distinct. Increased influence of the inertial effects create a smaller accelerated flow zone around the edge of the turbulizer blade while forming a slightly larger and more distinct wake structure. The dissimilarities observed between the leading and trailing blade wakes are further accentuated, with the high-recirculation wake velocities nearing those of the freestream. The case of $Re = 476$ further accentuates the trends observed at $Re = 119$, with strong mixing present throughout the entire section plane. For the purposes of privacy and confidentiality, velocity scales are removed from all images, however, the colours span the range from 0 to the maximum velocity within the section.

Further exploration of the flow structures is performed by observing an approximation of the wake. The images in Fig. 3.27 and Fig. 3.28 are created by using an isosurface where the velocity component in the flow direction is equal to zero, i.e. $v_i = 0$. Thus,


(a) Re=8

(b) Re=39

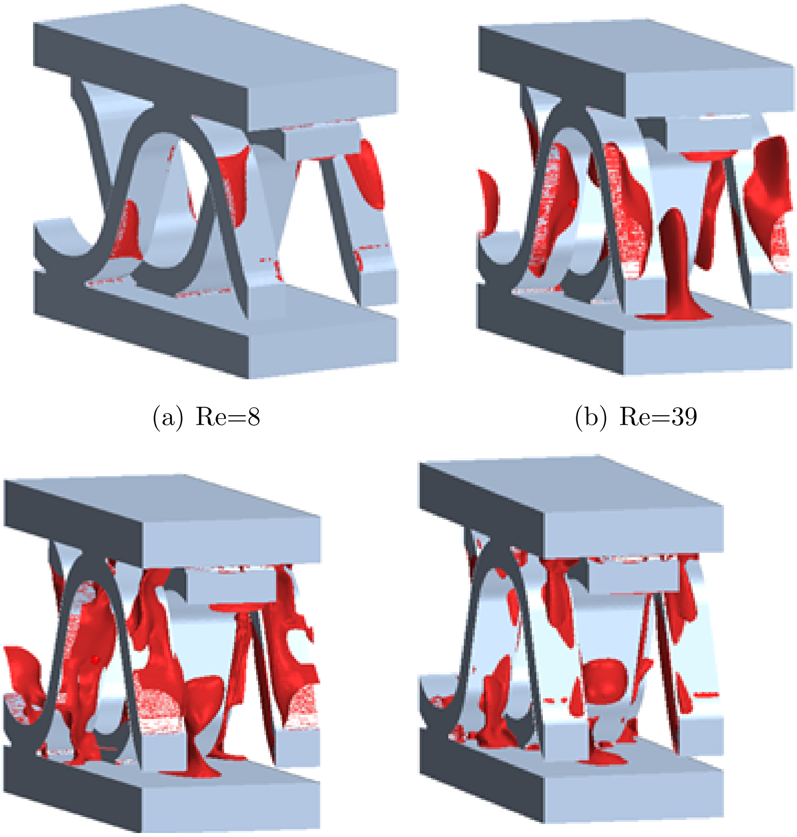
(c) Re=119

(d) Re=476

Figure 3.26: Comparison of the centreline velocity for a range of Re, $k-\epsilon$ model used to provide steady state flowfield

the surface is placed wherever the the streamwise component changes direction, as would be expected to occur at the edge of a recirculation zone. Although multiple methods for visualising recirculation and flow separation are commonly used, including isosurfaces of vorticity and total pressure, the current method is found to be the most consistent for the current geometry, mainly due to the large pressure gradient and strength of mixing within the domain. Figs. 3.27 a) and 3.28 a) demonstrate the predicted recirculation zones at $Re = 8$. Similar to what was shown along the domain centreline, the flow at low Re is very similar to what could be expected from a potential flow or creeping flow scenario. Recirculation and separation zones are extremely small, and only exist in the wake of the leading turbulizer blade. An increase to $Re = 39$, Figs. 3.27 and 3.28 b), demonstrates a wake that is becoming larger and increasingly complex. Separation is mainly limited to the area directly behind the turbulizer, with only a small separation occurring at the leading corners of the turbulizer blade. Further increase to $Re = 119$, Figs. 3.27 and 3.28 c), show an increasingly large separation occurring at the corner of the turbulizer blade. This separation forces fluid away from the turbulizer blade and contributes to the shift away from viscosity dominated drag to inertially dominated drag. Increasing to $Re = 476$, Figs. 3.27 and 3.28 d), yields extremely intense mixing within the domain, and as a result, the wake size decreases due to the diffusion of momentum. Small separation zones are observed in these regions. As a caveat to the observations presented by the isosurface method, the choice of visualisation method provides some influence on the predicted separation zone shape, and some discrepancies may be observed, depending on how post-processing of the results is completed. Due to the complex nature of the flow, the chosen method is deemed to provide the most consistent results.

The onset of unsteadiness is a major area of focus within this study. Although the results presented thus far have only been steady, it is important to glean an understanding of the complexity of the flow induced by the turbulizer when it does become unsteady. Use of LES results are ideal for demonstrating the complexity and structure of the flow at a specific instant in time. The same isosurfacing technique and criterion shown previously is

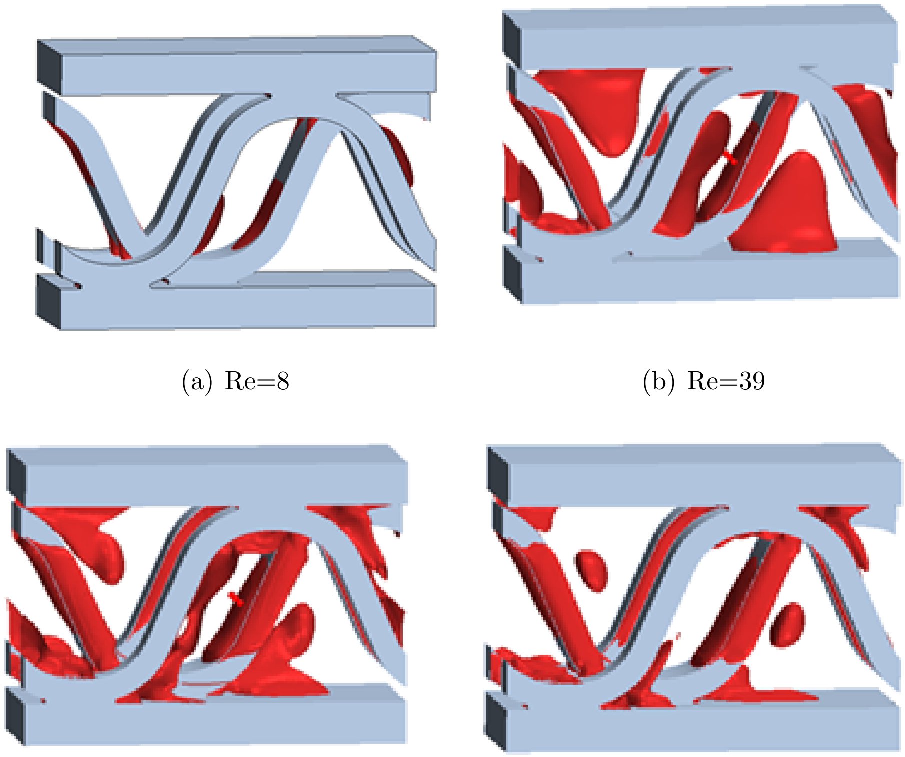

(a) Re=8

(b) Re=39

(c) Re=119

(d) Re=476

Figure 3.27: Isometric view of recirculation regions for a range of Re, $k - \epsilon$ model



(a) Re=8

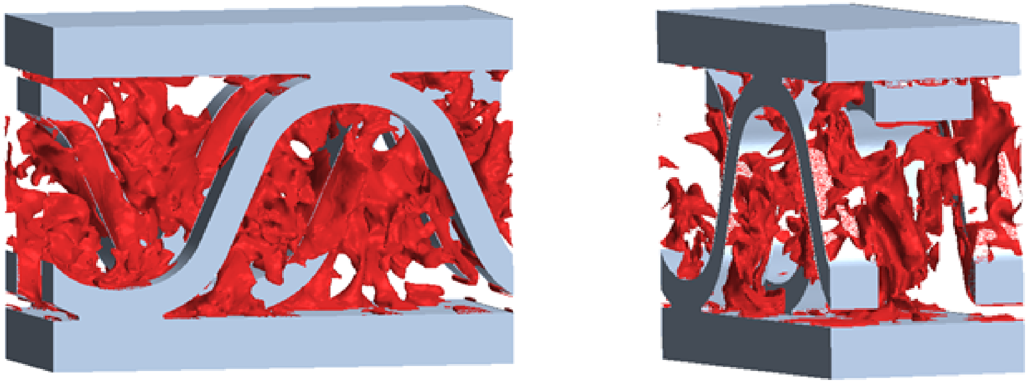
(b) Re=39

(c) Re=119

(d) Re=476

Figure 3.28: Side view of recirculation regions for a range of Re, $k - \epsilon$ model used

again used to generate the images in Fig. 3.29. In comparison to the results obtained from the $k - \epsilon$ model, the additional complexity introduced by turbulent mixing is instantly apparent. The complex flow structures become intertwined and convoluted, and are constant evolving and advecting through the computational domain. Despite the apparent disorder, some flow structures are observed to be very similar to the steady results. For example, a separation point at the leading corner of the turbulizer surface can be seen. Thus, it is expected that although the unsteady results may be very important in predictions of heat transfer, some of the geometric characteristics of the turbulizer may have a large impact on performance predictions.



(a) Side view of recirculation regions

(b) Isometric view of recirculation regions

Figure 3.29: Recirculation regions observed during LES, note the significant increase in flow complexity in comparison to $k - \epsilon$ model

3.4.5 Geometry Characterisation and Dependence

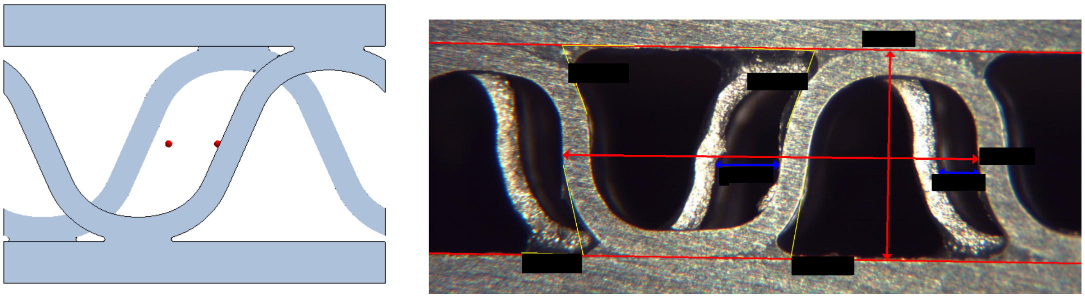
Geometry Characterisation

As discussed previously in Sec. 3.1, a significant amount of geometric variability may be present in the final product compared to the nominal CAD design. Previous investigations have involved mounting a heat exchanger channel in an acrylic base, and sectioning along various planes. Various results of these sectioning studies are shown in Fig. 3.6 b) and

Fig. 3.7 b). Although this offers some insight into the general shapes that could result from the manufacturing process, since only a 2D slice is captured, it does not offer a full understanding of the variations that may exist within a full channel. Also, because it is a destructive method, the process does not always allow one to capture all salient features of the geometry. In order to gain a better understanding of the 3 dimensional variability, a full 3D reconstruction method was sought out. The process offered by Robo-Met.3D is an ideal match for the 3D reconstruction requirements of this project [76].

The Robo-Met.3D process involves sectioning and imaging a heat exchanger sample at predefined depth intervals. Each of these images is then reconstructed to form a full 3D image of the part. In the context of this project, sections were carried out at depth intervals of 15 μm to a total depth of approximately 4.2 mm, enough to capture several blade widths. For comparison, a section image of the prepared sample is shown next to its CAD counterpart in Fig. 3.30. This figure clearly shows some of the main differences that can be observed from only using a 2D slice. Clearly, the real braze fillet radius is quite large, on the order of 0.5 mm, while previous simulation attempts have assumed a braze fillet radius an order of magnitude smaller, as seen in the CAD representation. Additionally, the turbulizer blades do not show the same curvature as the CAD model in the section between coreplates. When this image is combined with the 300 other section images, a fuller picture of the turbulizer geometry begins to emerge, and one can much better appreciate the differences that may be present between the two representations.

Once the reconstruction process has been completed, the reconstructed geometry is

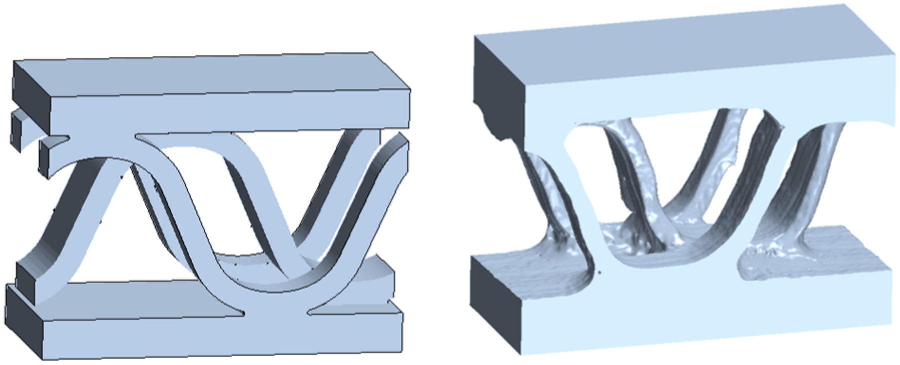


(a) Section of nominal CAD geometry (b) Section of as-manufactured, post braze turbulizer, obtained during the Robo-Met.3D process

Figure 3.30: Nominal CAD shown in comparison to section obtained during the 3D reconstruction process

suitable for use in either simulations or for measurement. The nominal CAD representation of a turbulizer is shown isometrically in comparison to the 3D reconstruction of the real geometry in Fig. 3.31. Features which may be otherwise difficult to observe, such as surface roughness, material tearing or thinning on shear surfaces, rounding on edges or burrs now become somewhat obvious to the observer.

Extracting specific measurements from the 3D reconstruction allows for easy application to the CAD model. Variations of edge radius, braze fillet radius, turbulizer blade curvature, channel height and material thickness can be directly applied to the CAD model in order to determine the heat transfer performance sensitivity to these features. Additionally, a CFD analysis of the exact geometry seen in a turbulizer may be conducted for comparison and validation of the geometric assumptions made during model preparation. The impact of these modifications on heat transfer and pressure drop performance are discussed below, in Sec. 3.4.5.



(a) Nominal CAD geometry

(b) 3D reconstructed geometry, note the rounded edges, uneven shear surface, braze fillet radius and material roughness

Figure 3.31: Comparison between nominal CAD geometry and 3D reconstructed turbulizer geometry

At the request of Dana Canada, only generalities of the features will be discussed, and any actual measurements are withheld from the present document for confidentiality reasons. If specifics on the geometry are required, please contact Dana Canada for permission and further details.

Geometry Dependence

The problem of relating the numerical performance of the real turbulizer to a CAD derived model presents an open-ended problem: how much modification of the initial CAD data is required to obtain good agreement of results, and is this modification practical enough to be performed on a regular basis? Although minute details are taken from the 3D reconstructed model and applied to the CAD geometry, this requires prototyping of the turbulizer shape to understand manufacturing limitations as well as performing the expensive and time consuming 3D reconstruction process. It is desired to capture the salient features that are expected to commonly occur within a turbulizer and apply them to CAD models, without *a priori* knowledge of the as-manufactured shape. Therefore, this section attempts to characterise the most important features of the turbulizer and how their variation affects heat transfer performance.

This section is separated into two distinct phases. The modifications covered in Phase 1 relate to changes which are expected to occur during the brazing process. This includes the impact of braze fillet radius, and channel height, both of which can be expected to vary due to slight changes in brazing conditions. Phase 2 relates to the variations in turbulizer geometry arising from the manufacturing process. This specifically includes material thickness and corner radius, both of which are affected by the stretching and tearing of material during forming of the turbulizer shape. In addition, the role of turbulizer offset is included, as previous studies have indicated that the offset may vary by several percent from nominal as tooling wears. Specific issues, such as burrs, are not covered in this section.

Phase 1: Braze Related Modifications

The feature which is perhaps most commonly associated with brazing is the formation of a "braze fillet". During the brazing process, the surface tension of the molten alloy draws the clad along material joints, as shown previously in Fig. 3.2. There are many factors which may affect the braze fillet radius, such as material finish, cleanliness, surface preparation, oxide formation, etc. Due to the large number of variables present in the process, it is common for the braze radius to vary somewhat from part to part. Isolating the effects of the braze fillet radius on heat transfer performance is considered an important step in the geometry preparation of the turbulizer. In addition to fillet radius, it was also observed that the channel height actually decreased slightly from the nominal height of the turbulizer. This reduction in height is attributed to the turbulizer "sinking" through the clad layer and contacting the base alloy. Due to the clad thickness being approximately 5-10% of the total coreplate thickness, this reduction ranges from 0.1 *mm* to 0.2 *mm*. The effects on heat transfer and pressure drop performance of this channel height reduction are mainly due to an effective increase in *Re*. It is the purpose of this phase of investigation to quantify the importance of these variations on numerical results.

As can be clearly seen in Fig. 3.30 a) and b), there currently exists a significant discrepancy in the fillet radius applied when preparing the CAD data for simulation. In order to rectify this, a larger fillet radius, on the order of 0.5 *mm* is applied to the CAD where the braze fillet typically occurs. The results of this modification are depicted in Fig. 3.32. Using the large fillet radius as a the starting point, the channel height is then reduced in steps by 0.1 *mm* and 0.2 *mm*, creating an area reduction (and corresponding increase in velocity) of approximately 4% and 8%, respectively. This height reduction is achieved by extruding both the top and bottom coreplate surface towards the centre of the channel, which for 0.1 *mm* reduction is shown in Fig. 3.33.

Numerical simulation of these geometry modifications highlights several noticeable trends. Firstly, as can be seen in Fig. 3.34, the addition of large fillets yield only a moderate difference in heat transfer, despite the presence of the fillets accounting for approximately a 30% reduction in wetted heat transfer area. Investigation of the flowfields indicate that, although there is a significant reduction in wetted area due to the fillets, this area was relatively ineffective in contributing to heat transfer. The area around the fillets is generally somewhat stagnant on the upstream side of the turbulizer, and is almost entirely situated within the downstream wake region, with both areas being characterised by large amounts of recirculation and low flow velocities. The second major trend that can be seen is the downward shift in heat transfer results as channel height is reduced. As the flow

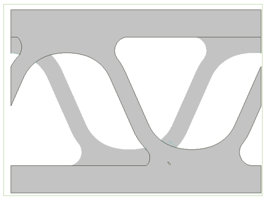


Figure 3.32: 0.5 *mm* braze fillet radius placed between turbulizer surface and coreplate

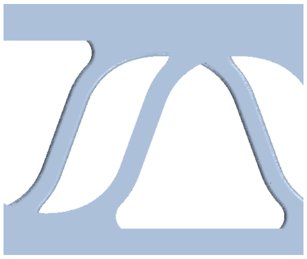


Figure 3.33: 0.1 *mm* channel height reduction in conjunction with 0.5 *mm* braze radius. The truncated radius due to coreplate extrusion can be seen in comparison to the previous figure

velocity is increased, the fluid spends less time in contact with the heat transfer surface over a given area. As the channel height decreases further, with a 0.2 *mm* reduction from nominal, the trend continues, although to a lesser extent. Any further reductions beyond 0.2 *mm* have not been tested, but are assumed to follow a similar trend. Please note that the results shown in Fig. 3.34 have been corrected to yield *Re* based on the *nominal* channel height, i.e. the channel height reduction is *not* included in the evaluation of *Re*.

From this phase of the analysis, it is shown that although braze fillet radius does

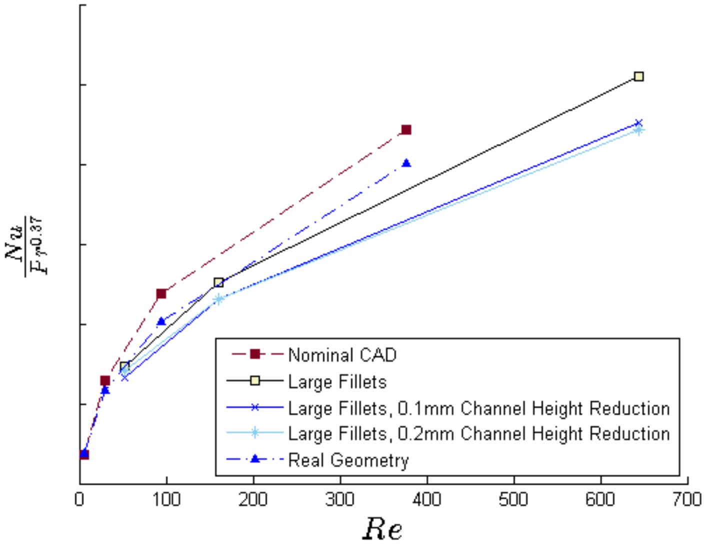


Figure 3.34: Results from the first phase of geometry modifications related to the brazing process

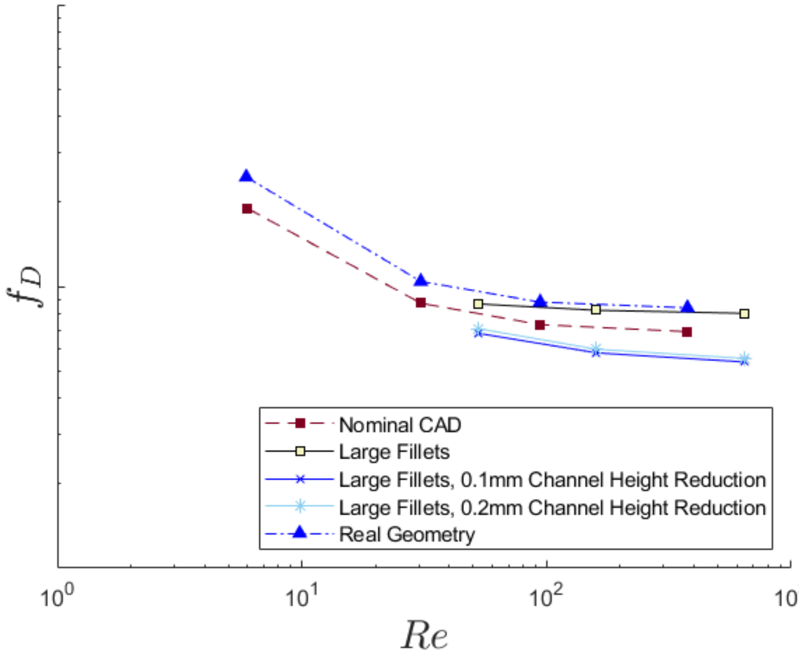


Figure 3.35: Results from the first phase of geometry modifications related to the brazing process, f_D

not play a large role in the prediction of heat transfer, the channel height, indeed, has a significant impact. In order to properly assess what could be considered an average channel height, further analysis of samples is required. Based on current measurements, it is believed that the channel height reduction is largely dependent on the coreplate clad thickness, and that knowledge of the material makeup will allow for accurate prediction of height reduction. With these results, it will be possible to quantify what reduction should be applied to future CAD data in order to create a more realistic geometric starting point.

In comparison to the predictions of heat transfer, f_D representations shown in Fig. 3.35 are also significantly affected by the presence of a smaller channel height. The reduction tends to reduce the friction factor more than any other modification. Conversely, large braze fillets tend to increase f_D in comparison to the nominal geometry. While the direction of the change from the nominal results differs for pressure drop for the aforementioned changes, interestingly, the heat transfer is shown to decrease in both cases. This could be an indication that the performance of the turbulizer may actually be beneficially affected if these changes can be acted upon in a meaningful way during the manufacturing process. The real geometry change in f_D is most closely represented by the addition of the large fillets on nominal geometry, which is similar to the findings related to heat transfer magnitude.

Phase 2: Turbulizer Related Modifications

Perhaps more difficult to quantify than the variables in Phase 1, the forming and final shape of the turbulizer is highly variable. The age and condition of the tooling can have a large impact on the quality of shear surfaces. Additionally, as the forming process elongates the turbulizer form somewhat, a small amount of material thinning is expected. Finally tooling wear often results in a shift of turbulizer phasing, which is observed as a change in the offset between adjacent blades. Although testing every combination of geometric modifications is prohibitively time consuming, several key modifications are selected and applied to the nominal CAD model.

The first parameter, material thickness is varied from nominal thickness, t, and reduced in increments to $0.83t$ and $0.67t$. To account for the shearing process, varying edge radii are applied to the corners of the turbulizer surface, demonstrated in Fig. 3.36. Finally, the turbulizer offset is modified from the nominal 50% to 43%. As there is significant overlap in the testing parameters and the results are not expected to be highly dependent on different modifications, not all possible permutations are tested, but rather, 7 cases are selected for comparison, as listed in Table 3.3.

Across the range of tested Re, the variations in geometry preparation are seen to provide approximately 18-20% variation in heat transfer performance, Fig. 3.37. As this variation far exceeds the experimental uncertainty predicted from analysis of the experimental test rig [39], determining the proper CAD modifications required to match heat transfer performance is key to improving the accuracy of future numerical predictions. In general, the trends observed from these results are: a decrease in thickness decreases heat transfer, an increase in corner radius improves heat transfer, and deviation from nominal offset yields a net reduction of heat transfer. The best agreement with the real geometry is observed with Test Case 4 from Table 3.3, which corresponds to an edge radius of $0.17t$, a thickness of $0.83t$ and a 50% offset. When comparing a centreline section, shown in Fig. 3.39, it can be seen that the prepared geometry appears to be largely similar to the real

Table 3.3: Selected test cases for geometry dependence

Test Case	Corner Radius	Material Thickness	Offset
1	$0t$	t	50%
2	$0t$	t	43%
3	$0.07t$	$0.83t$	50%
4	$0.17t$	$0.83t$	50%
5	$0.27t$	$0.83t$	50%
6	$0.27t$	$0.67t$	50%
7	Real Geometry		

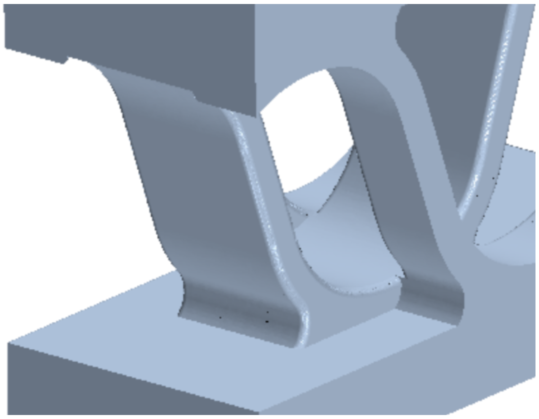


Figure 3.36: The rounded off edges can be seen between turbulizer blades to mimic the deformation and edge shape from the forming process

CAD geometry.

Although the agreement of performance of the heat transfer predictions has been improved, the pressure drop is still slightly underpredicted by approximately 15% in comparison to the real geometry across the entire range of test conditions. The only modifications able to directly improve upon the nominal CAD geometry is a combination of $0.83t$ material thickness and $0.07t$ corner radius. This highlights the challenges in providing a single geometry that is able to both predict heat transfer and pressure drop. Despite this disagreement between heat transfer and pressure drop, the magnitude of the improvement for heat transfer outweighs the loss in performance in f_D, and therefore, the chosen modifications for the remainder of the study are selected as Case 4 from Table 3.3.

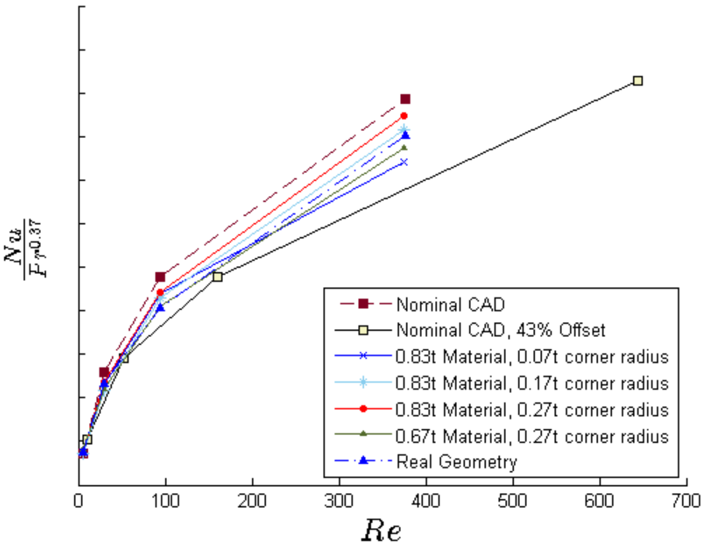


Figure 3.37: Results from the second phase of geometry modifications related to the turbulizer forming process

Combined Geometry Modifications

As the majority of analysis in the preceding sections is completed using steady results from the $k - \epsilon$ turbulence model, further investigation is required to determine the effectiveness of the selected geometry modifications when applied to LES. Accordingly, a larger number

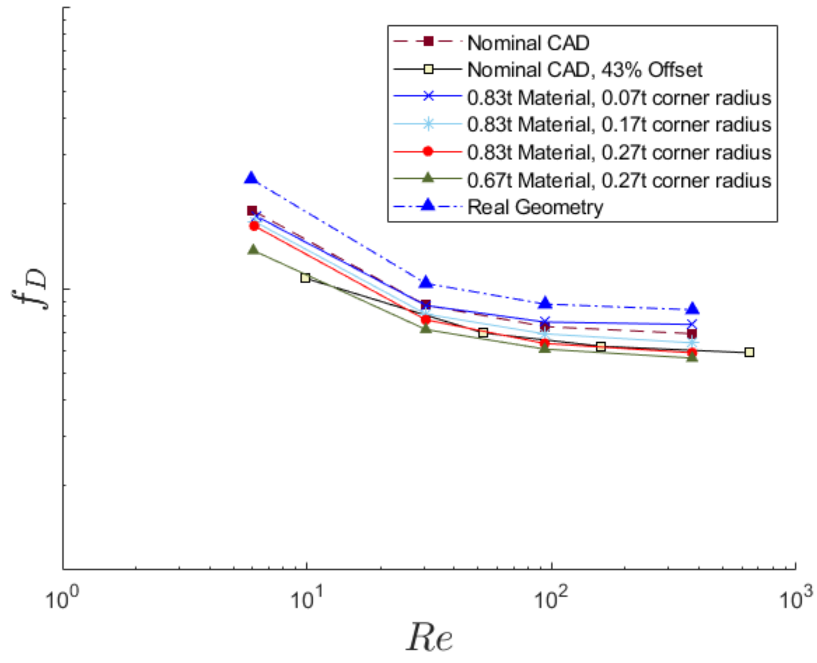
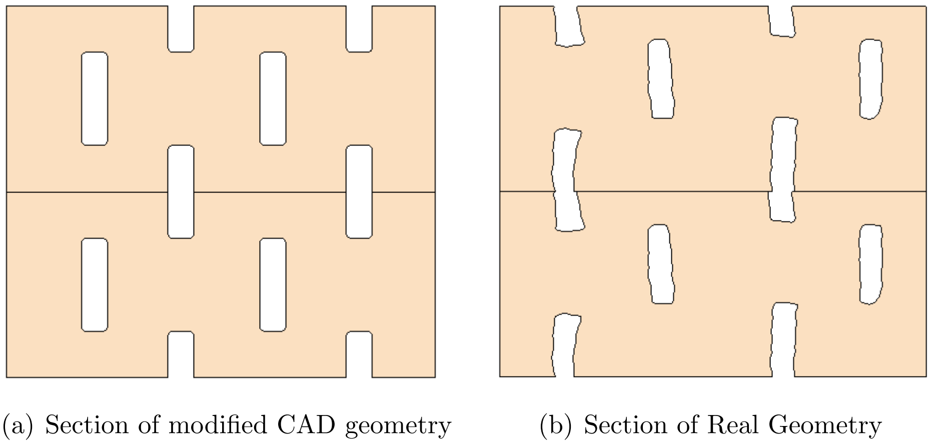


Figure 3.38: Results from the second phase of geometry modifications related to the turbulizer forming process, f_D



(a) Section of modified CAD geometry

(b) Section of Real Geometry

Figure 3.39: Section of modified geometry compared to section of real geometry, note the similarities between material thickness and edge radius

of Re values are simulated allowing for a much finer resolution and comparison of the $\frac{Nu}{Pr^n}$ curve to the experimental correlation, with the results of each unsteady simulation time averaged to determine heat transfer and pressure drop results.

In Fig. 3.40, the results from the modified geometry-LES are shown in comparison to experimental results. It can be seen that there is a significant improvement in the accuracy of the predictions, with error more than halved in the moderate Re range (centred about $Re \approx$120) compared to preliminary simulations. This result is extremely encouraging, indicating that the steps taken to mimic real geometry are effective at reproducing the desired heat transfer results. It also lends credence to the fact that, although LES incurs a higher computational expense, the accuracy afforded justifies this increased cost.

In comparison to the heat transfer results shown in Fig. 3.40, the values shown in Fig. 3.41 do not show a corresponding magnitude of change in the prediction of f_D. This is an encouraging result, which further reinforces the conclusions made previously about prioritising the modifications to match heat transfer results, rather than placing emphasis on the pressure drop predictions. This is also an indication of the sensitivity of heat transfer results to input geometry. The modifications proposed in the simulations have been able to closely follow the experimental correlation, however, there still remains a small error in

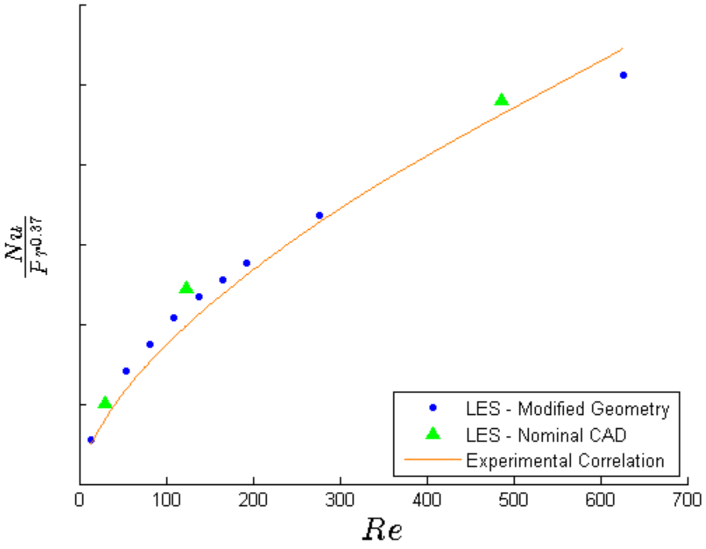


Figure 3.40: Comparison of initial LES results, LES results following geometry modification and experimental correlation

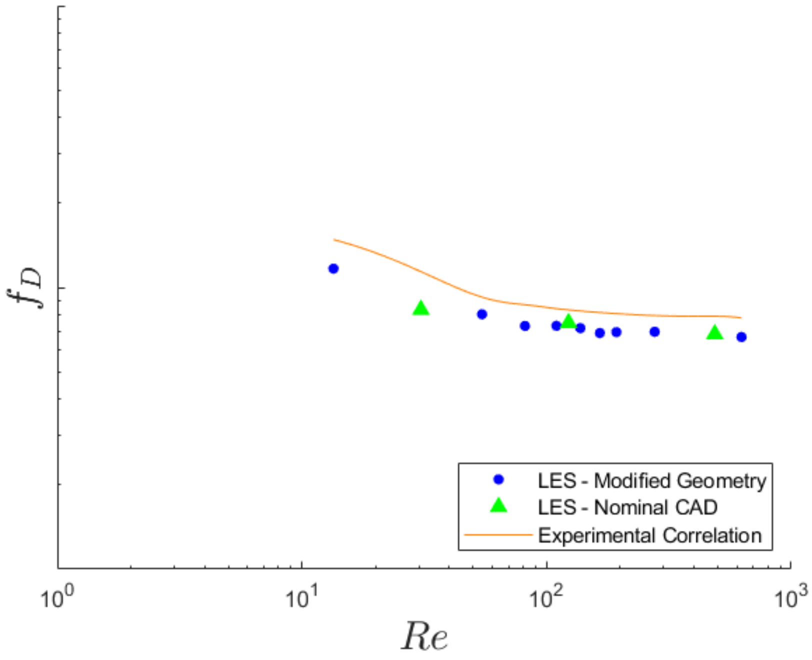


Figure 3.41: Comparison of initial LES results, LES results following geometry modification and experimental correlation

the magnitude of the predictions, which is on the order of 5%.

As a closing note on the impact of geometry modifications, it must be noted that departures from the current manufacturing methodology are likely to create significant changes in the artefacts and characteristics of the present geometry. In the present research, turbulizer strips are formed by a rolling process, which limits the approach of geometry to a vertical wall, due to tooling interactions, however, a stamping approach, or of increasing relevance, the characteristics of an additive manufacturing process would require recharacterisation of the geometry and additional investigation. Between the start and completion of the present research, X-ray computational tomography (XCT) methods have developed sufficiently that many industrial institutions either possess an XCT machine or have relatively easy access to one. It is the recommendation of the present research that additional manufacturing methods and samples are investigated using high resolution XCT imaging to determine a wider range of characteristics.

3.4.6 Fluid Property Dependence

Up to this point, all fluid property data have been plotted using a single method: $\frac{Nu}{Pr^n}$ vs. Re. Although this has historically been shown to be an excellent way of interpreting data, it does beg the question: What other dependence might exist based on fluid properties? Certainly, this relation cannot hold true for all Pr, and would be expected to fall apart under certain conditions.

As it turns out, there is some precedent to modifying this dependence based on flow conditions. It has been documented that the value of n may vary when the wall is heated ($n = 0.4$) vs. cooled ($n = 0.33$) [71, 77]. This modification likely originates within the boundary layer: as a liquid is heated (or a gas is cooled), it becomes less viscous, and the effective local Re within the boundary layer increases. As a liquid is cooled (or a gas is heated), it becomes more viscous, and the local Re within the boundary layer decreases. This slight change of fluid properties within the boundary layer is the suspected source of this variation in n. It follows, then, that as the test fluid Pr changes, so too would its response to varying levels of heat transfer.

In order to ascertain the validity of the $\frac{Nu}{Pr^n}$ assumption, a set of identical simulations are completed, with the only variable being the fluid properties. A wide range of fluid properties is selected in order test the data reduction methods. Four specific fluids are tested: Type A ATF, 50% Ethylene Glycol-water mixture (EG-water), Paratherm OR (a fluid specifically engineered for heat transfer testing) and atmospheric air. Approximate properties of each fluid are listed in Table 3.4 for comparison. Plots of fluid property curves are shown in Appendix C, however, the correlations used to determine the properties are withheld due to confidentiality reasons.

For each fluid, 8 separate simulations are conducted, with temperature boundary conditions shown in Table 3.5. This allows for both heating and cooling conditions, i.e. $T_{inlet} = 80°\text{C}$ and $T_{inlet} = 120°\text{C}$, with the wall held at a constant $T_{wall} = 100°\text{C}$, as well as all 4 Re conditions, i.e. $Re = 8,\ 39,\ 119,\ 476$. All simulations are conducted using the $k - \epsilon$ model, which is chosen due to its propensity for quick convergence and prediction of steady results. Additionally, the real geometry has been chosen for this portion of the study, as it eliminates a large amount of uncertainty associated with geometry preparation.

Table 3.4: Approximate fluid properties

Fluid	$\mu\ [Pa - s]$	$\rho\ \left[\frac{kg}{m^3}\right]$	$C_p\ \left[\frac{J}{kg-K}\right]$	Pr
Type A ATF	0.004-0.006	782-795	2414	72-110
50% EG-Water	$5x10^{-4} - 7x10^{-4}$	1008-1017	3435	3.9-5.0
Paratherm OR	$4x10^{-4} - 6x10^{-4}$	827-839	2215-2291	75-112
Air (STP)	$2.1x10^{-5} - 2.2x10^{-5}$	0.93-0.97	1009-1011	0.7

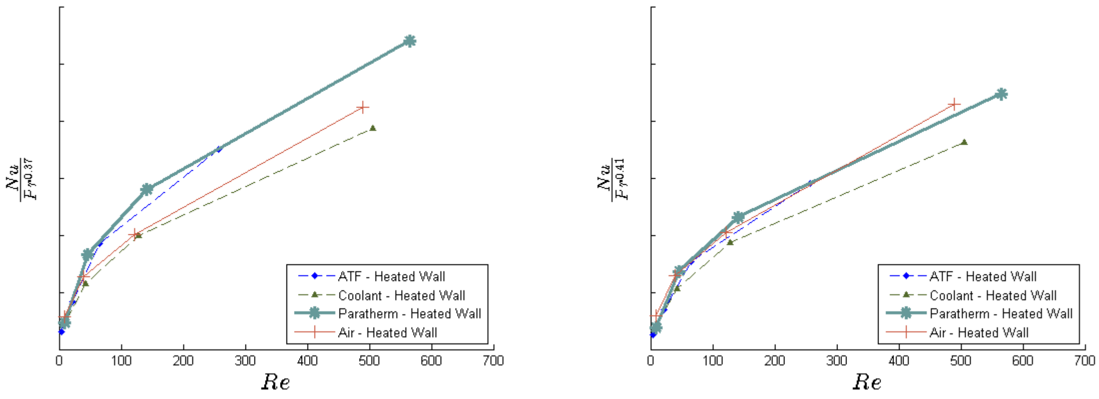
Simulation results are compiled and compared within two separate plots: the heated wall condition and the cooled wall conditions. Results for the heated wall are shown in Fig. 3.42, while results for the cooled wall are shown in Fig. 3.43. In each instance, the "standard" assumption of $n = 0.37$ is compared to the value providing the best collapse of data. It is found that the best collapse of all data is achieved at $n = 0.41$ in the case of the heated wall and $n = 0.39$ for the cooled wall. Although this spread of n does not align itself exactly with the empirically derived $n = 0.33$ and $n = 0.4$ values for an empty channel

Table 3.5: Temperature boundary conditions, fluid property dependence study

	Cooled Wall				Heated Wall			
Re	8	39	119	476	8	39	119	476
Inlet Temperature	120°C	120°C	120°C	120°C	80°C	80°C	80°C	80°C
Wall Temperature	100°C	100°C	100°C	100°C	100°C	100°C	100°C	100°C

[77], it does exhibit a spread, which is indicative that the heat transfer is dependent on fluid properties within the boundary layer.

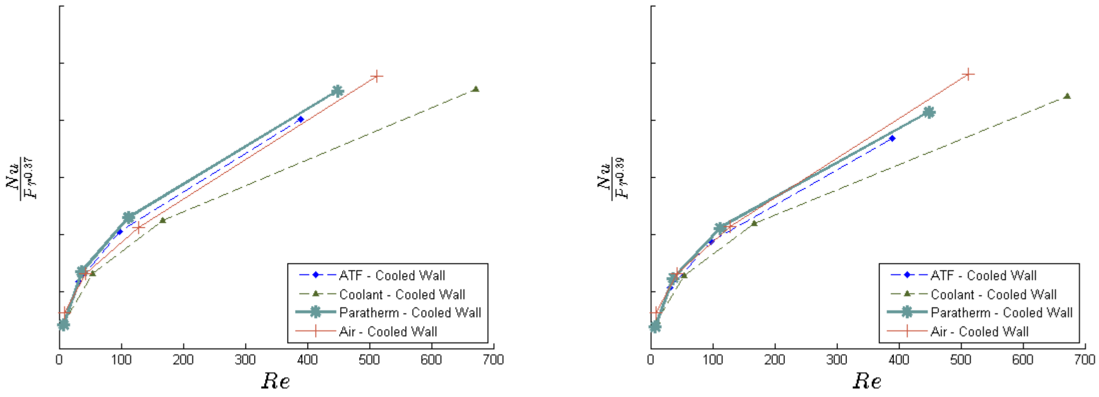
As a further test, values of n are explored outside the range of $0.3 \leq n \leq 0.4$. It is



(a) $n = 0.37$, heated wall

(b) $n = 0.41$, heated wall

Figure 3.42: Comparison of n=0.37 and n=0.41 for data collapse with a heated wall



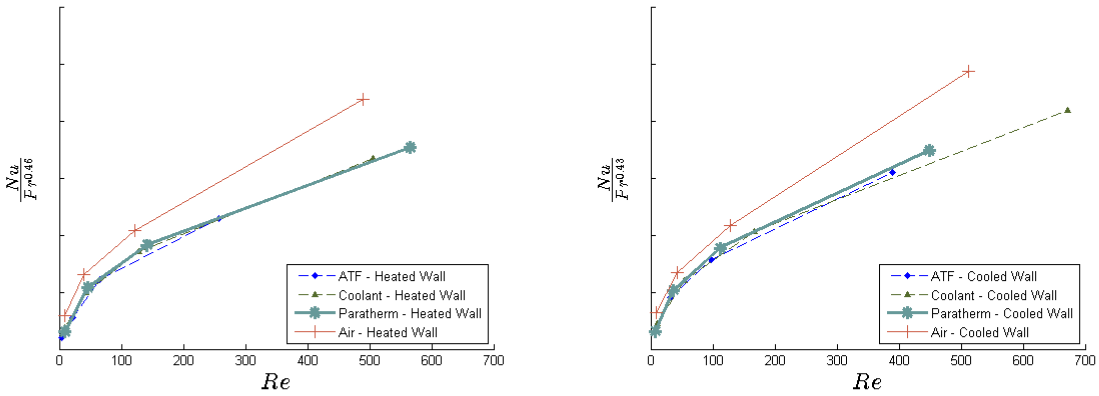
(a) $n = 0.37$, cooled wall

(b) $n = 0.39$, cooled wall

Figure 3.43: Comparison of n=0.37 and n=0.39 for data collapse with a cooled wall

found that when these values are increased to $n = 0.46$ for a heated wall and $n = 0.43$ for a cooled wall, a significant improvement in the collapse of numerical data occurs, for $Pr \geq 5$, illustrated in Fig. 3.44. This result suggests that, at least in the case of turbulizers, the standard range of validity of $0.7 \leq Pr \leq 120$ [77], does not actually apply, and that between $0.6 \leq Pr \leq 5$ (between the fluid properties of air and 50% EG-water) some change occurs which invalidates the assumptions that $n = const$. For the present type of geometry, working fluids typically fall within the range of $5 \leq PR \leq 120$. It is recommended that in order to expand this modelling methodology to lower Pr fluids, such as gases, additional work is performed to verify the exponential relationship of Pr to Nu scaling.

The results presented in this section are only a cursory investigation of how fluid properties affect heat transfer performance, but the current results point to the possibility that there may be some fundamental flaw in the assumption that heat transfer performance *should* collapse onto a curve of $\frac{Nu}{Pr^n}$.



(a) $n = 0.46$, heated wall

(b) $n = 0.43$, cooled wall

Figure 3.44: Comparison of data collapse with higher n, ignoring air results

3.4.7 Numerical Modelling Summary

The numerical work presented herein comprises the initial steps taken to investigate the heat transfer performance of turbulizers on a microscopic level. A large effort is put forth into determining what sensitivities exist in the numerical model setup, namely, selection of domain size, selection of turbulence model and solver method (steady vs. unsteady), representation of turbulizer geometry, fluid property selection and data processing techniques.

Domain independence investigation indicates that the optimal simulation domain for a turbulizer consists of a single convolution surrounded by periodic boundary conditions. The periodicity allows for representation of fully developed flow without the associated cost of providing a domain size sufficient to allow the flow to fully develop. It is also determined that the influence of large eddies of other coherent flow structures is minimal, such that a single convolution accurately captures the physics of the flow occurring within the domain. Turbulence model investigation indicates that there is likely some transition occurring between $Re = 39$ and $Re = 119$. Both LES and $k-\omega$ models predict a transition to unsteadiness, with LES further predicting fully turbulent flow at $Re = 476$. Experimental validation of the transition in flow is required, however, until these data are available, LES is selected as the most representative model due it providing the best throughout the entire range of Re in this investigation. Testing of various geometric configurations to the turbulizer model indicated that the best agreement is achieved by reducing the turbulizer thickness by approximately 15%, reducing channel height by 0.1 mm and providing a turbulizer blade corner radius of at least 0.15t provides the optimal geometry preparation to match real geometry simulations and experimental correlations. These findings are to be used in future portions of the study, which focus on ROM construction and training.

3.5 Experimental Validation of Numerical Modelling Results

As very little experimental work has been performed in identifying small scale flow structures and transitions to unsteady, or unsteady turbulent flow is available in the current literature, careful validation of numerical models and results is crucial to the complete understanding of the flow. As such, several experimental validations are undertaken in order to demonstrate the efficacy of the numerical modelling techniques. Particular focus is provided to experimental flow visualisation and laser doppler velocimetry, LDV.

Flow visualisation is particularly useful when comparing the qualitative aspects of the flow, specifically streamline or streakline comparison, as well as the transition to unsteady flow.

LDV techniques, while somewhat restrictive in understanding large scale flow structures, can be extremely useful in determining the quantitative aspects of flow transition from steady to unsteady laminar and eventually to unsteady turbulent. The high sampling rate of the LDV system facilitate capture of small scale turbulence, provided the particle seeding density is sufficient.

3.5.1 Flow Visualisation

Water Tunnel Design

For the purposes of the present research, no readily available water tunnel system exists that is sufficient for flow visualisation. A comprehensive design and development project relating to the construction of an appropriate facility is undertaken in order to test scale models of turbulizer geometry [78].

For the purposes of testing, a model scale of 10:1 is selected to provide adequate manufacturability, without requiring unreasonably low velocities for adequate *Re* range. Despite the larger scale, difficulties were experienced with 3D printing the as-manufactured geometry model, and subsequently, testing of the nominal CAD model with geometry modifications outlined in Sec. 3.4.5 are used for both experimental and numerical testing. The printed geometry is shown in Fig. 3.45. It is critical to note that this scaling is only valid for flow testing, i.e. *Re* based scaling, and is not appropriate for usage in a capacity to evaluate heat transfer performance. Heat transfer data are generated only via in-situ experimental testing of a brazed aluminum heat exchanger. These data are typically provided by dedicated designs and specific thermal testing by Dana Canada. An example of this testing is provided in Ch. 5, and used for validation of overall ROM performance. The water tunnel is constructing using a gravity driven, or static head design, outlined in Fig. 3.46. An overflow return section maintains a consistent water level and therefore constant static head for the experiment, ensuring that flow rates are subjected to consistent pressure drop. A small aquarium style pump is used to return water from the reservoir to the settling chamber. Mass flow through a gate valve at the base of the apparatus is used to determine flow *Re*, with a calculated *Re* uncertainty of 1.3%, as shown in Appendix D.

The water tunnel, Fig. 3.46, is comprised of a settling chamber/flow alignment section. A honeycomb and 3 separate layers of screens are used to minimise turbulence within the settling chamber. The aluminum honeycomb layer provides a dual purpose of dampening turbulent fluctuations while aligning the flow and preventing swirl. Upon testing, it is

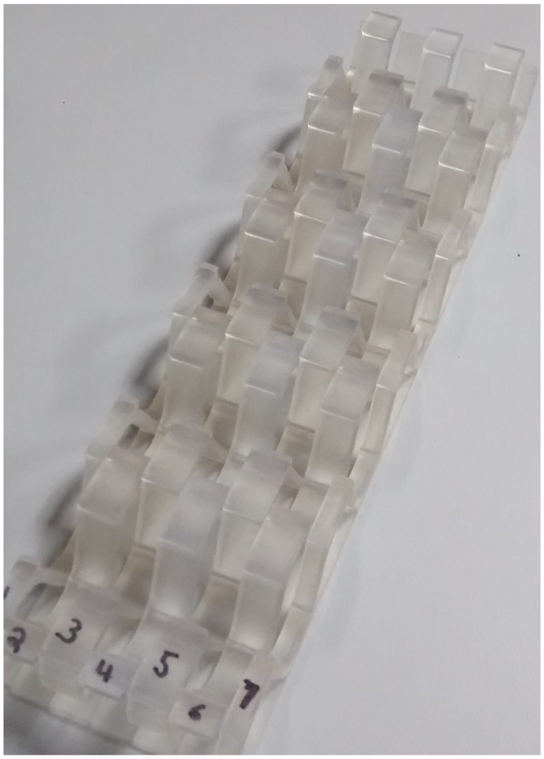


Figure 3.45: 3D printed 10:1 scale model of turbulizer

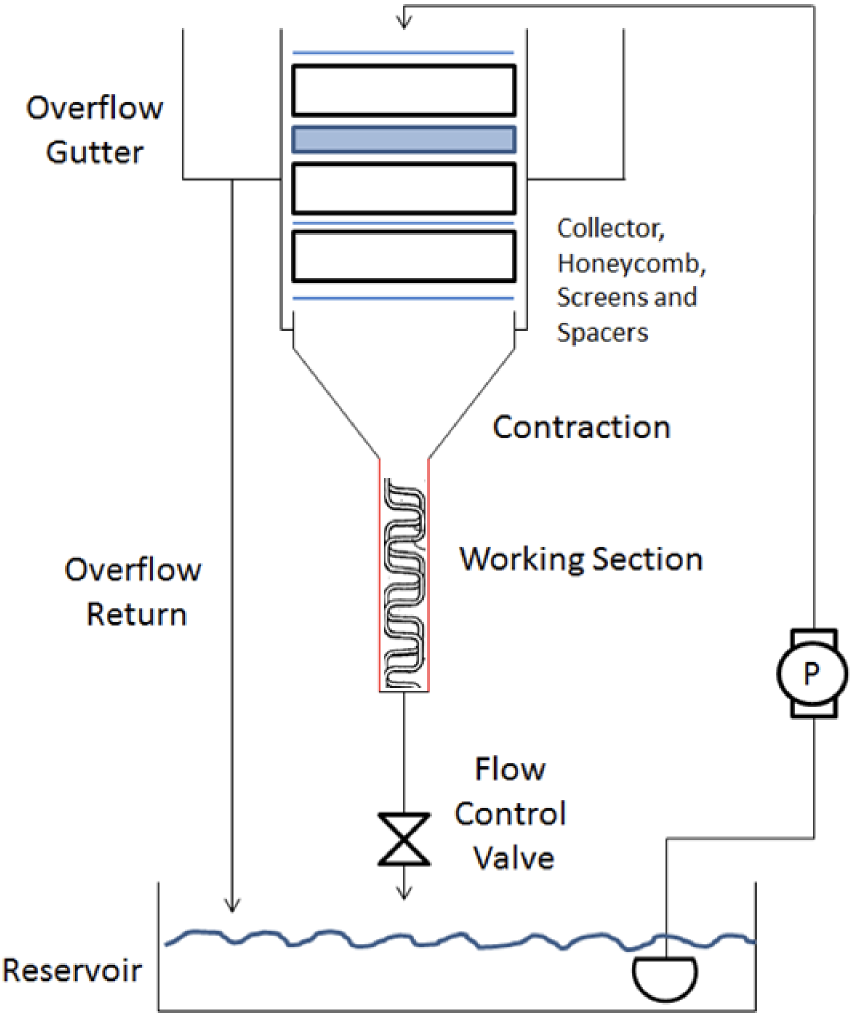


Figure 3.46: Water Tunnel Schematic

found that the discharge of the pump return provides enough momentum that a slight swirl is imparted into the flow. To correct this, a micro filter is fitted to the outlet to provide a seeping flow over a large area and effectively eliminate the additional momentum imparted by the pump flow. A CAD model of the exploded settling chamber and contraction is shown in Fig. 3.47.

The contraction shown in Fig. 3.47 is designed based on the method of Morel [79].

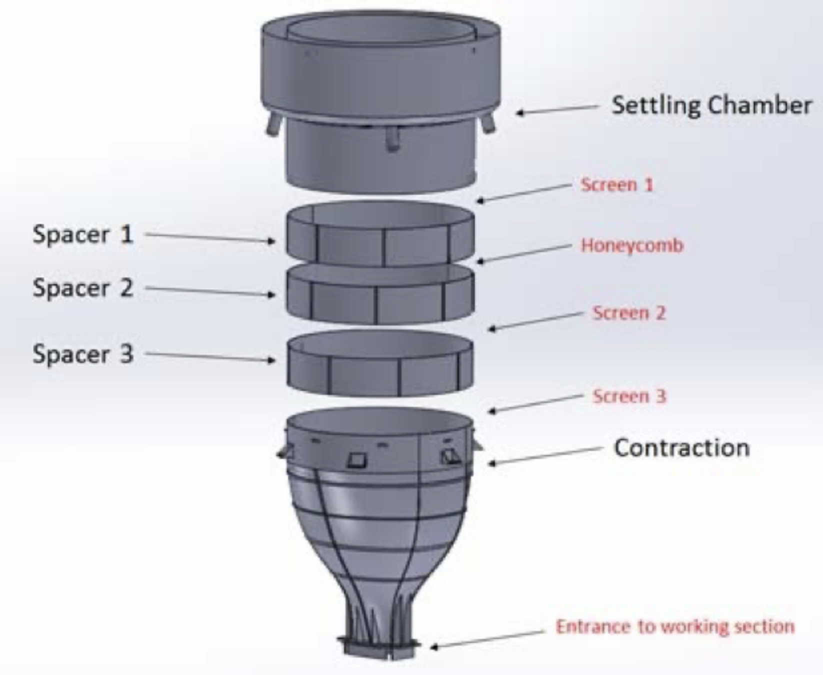


Figure 3.47: Exploded CAD view of the water tunnel

The high contraction ratio of 25:1 is based on a desire to provide a nearly uniform flow profile across the test section. CFD simulations are used to verify the design, with experimental measurements confirming the CFD results [78]. The test section is rectangular, and consists of bonded acrylic walls for maximum flow visibility. The completed water tunnel is shown in Fig. 3.48.

To further facilitate flow visualisation, a hole is built within the turbulizer model that facilitates dye injection. An external injection placed on the leading edge (stagnation point) of the turbulizer blade, while another corresponding point is placed at the trailing edge, within the wake of the blade. Access to these points is achieved through an external valving system, using hydrostatic pressure to control the flow rate appropriately. Although

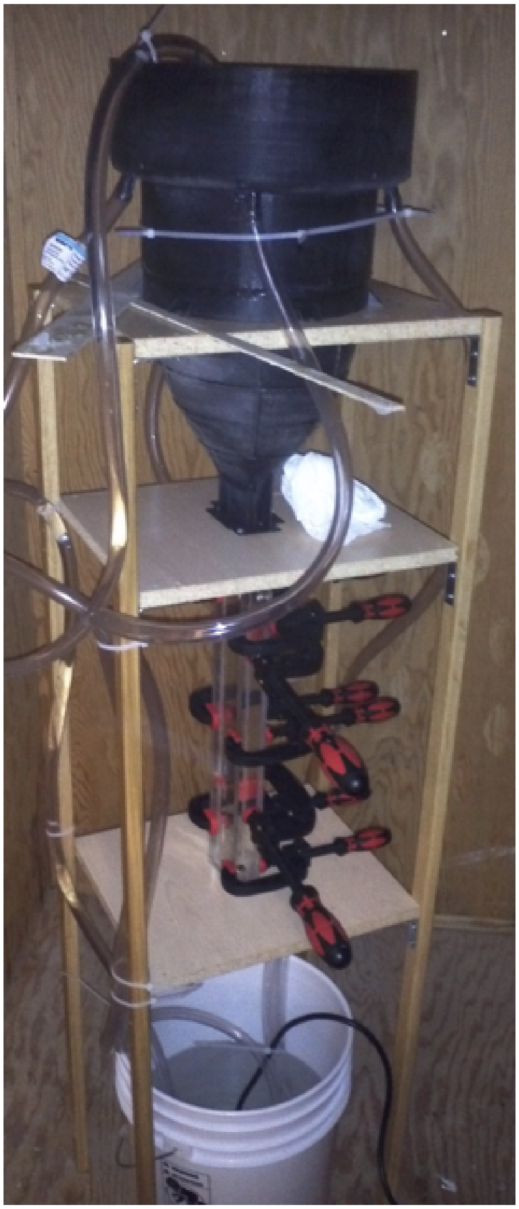


Figure 3.48: Completed water tunnel test section

the flow rate of the injection dye is not measured directly, it is diluted with water to approximate the density of the test section flow. Flow rates are kept low enough that the dye does not impact the local flow over the turbulizer. This is verified by experimentation, as flow rates that are deemed to be too large would inject a small plume of dye beyond the boundary layer, imparting additional momentum and disrupting the local flow. Flow rates that were too low would not provide sufficient dye contrast to visualise the flow.

Direct optical access to the test section also facilitates access for the LDV measurement equipment, used in Sec. 3.5.2.

A detailed review of the design, testing and validation of the water tunnel design may be found in [78].

Flow Visualisation

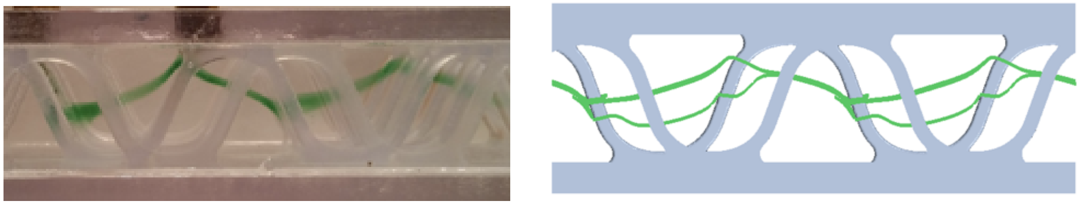
With the facilities available for detailed testing of the turbulizer, flow visualisation of the turbulizer is conducted. The two primary objectives of this section are:

1. **Confirm flow structures at low** Re
2. **Validate transition to unsteadiness**

Strictly determining the difference between unsteady flow and turbulent flow is difficult using only flow visualisation, however, some general trends are notable. If the dye filament is unsteady through the test section, but generally remains intact as a filament, it can be assumed that the flow is simply unsteady laminar flow. As the fluctuations increase in intensity and frequency, with small eddies dominating the flow, it is assumed that the transition to turbulence has occurred, and the dye filament will quickly diffuse within the test section. Exact measurements are confirmed with LDV, but flow visualisation provides the basis for later tests.

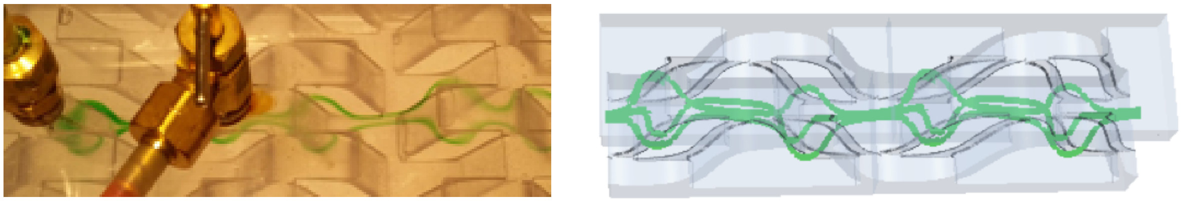
As a primary step, the aforementioned dye injection ports are used to determine local flow structures. Using the injection ports, with a flow $Re \approx 8.5$. This is shown in Fig. 3.49 for the side view of the turbulizer, and Fig. 3.50 for the top view of the turbulizer.

Fig. 3.49 and Fig. 3.50 demonstrate excellent agreement in the replication of experimental and numerical results at low Re. Validation in the fully laminar region is crucial, as



(a) Experimental, $Re \approx 8.5$ (b) Numerical, $Re \approx 8.5$

Figure 3.49: Comparison of side view of flow structures in experimental and numerical results, $Re \approx 8.5$



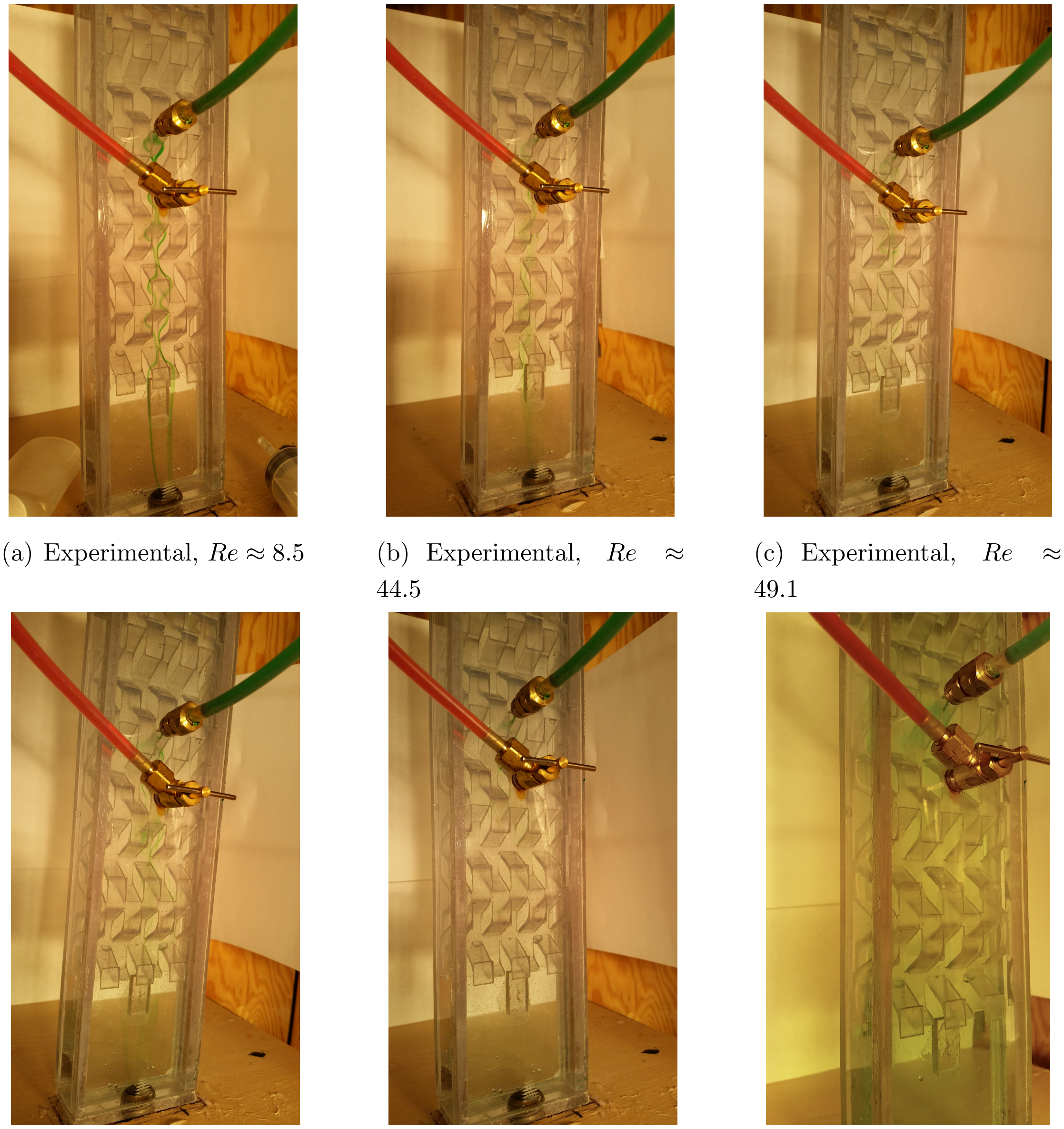
(a) Experimental, $Re \approx 8.5$ (b) Numerical, $Re \approx 8.5$

Figure 3.50: Comparison of side view of flow structures in experimental and numerical results, $Re \approx 8.5$

it demonstrates the ability of numerical analysis to accurately replicate the flow conditions prior to introducing additional complexity with time-dependent turbulence models. Heat transfer performance is not evaluated at this stage.

As the flow *Re* increases, the vortex shedding present in bluff body flows [80] manifests itself, and a transition to unsteady flow patterns occurs. With flow visualisation, a gradual "wandering" of injected dye filaments presents itself in the test section. Following the general instability theory of turbulence and generation of fully turbulent flow [28], a transition occurs where the fluctuations are no longer dampened, but rather provide positive feedback and make a transition to highly random, turbulent flow. Examples of this transition for the turbulizer geometry are seen in Fig. 3.51. In Fig. 3.51 a) and b), the flow is steady: there is no temporal evolution of the dye filament. As *Re* increases beyond this, namely $Re \approx 49.1$, the flow becomes unsteady and small movements of the dye filament are observed. Continuing to increase *Re*, shown in Fig. 3.51 d)-f), increases the temporal variation, and at the highest *Re*, significant diffusion is observed in the dye filament, signifying fully turbulent flow. At this stage, the utility of flow visualisation is limited beyond demonstration of the diffusive nature of the flow, and cannot be used to adequately show any coherent flow structures.

When discussing and analysing the transition of flow from steady to unsteady, and laminar to turbulent, the latter transition is extremely difficult to quantify in the context of turbulizers. In "traditional" transition, such as that with a flat plate or the flow over a circular cylinder [28] retains distinguishable characteristics related to vortex shedding, boundary layer growth or coherent flow structures, but in the turbulizer geometry, once the flow initially becomes unsteady, which is shown to be a clear transitional point, the only method of evaluating the onset of turbulence is by defining a fixed threshold of turbulent kinetic energy or fluctuation magnitude. This is explored in greater detail in Sec. 3.5.3, where numerical representation of the turbulent kinetic energy cascade are investigated. As the energy cascade becomes more defined, the intensity of the fluid mixing increases, which serves to provide a more homogeneous temperature distribution in the bulk flow, rather than highly stratified areas of high and low temperature. This increase in mixing aids the level of heat transfer possible under certain flow conditions, and improves heat transfer, while providing a corresponding increase in pressure drop.



(a) Experimental, $Re \approx 8.5$

(b) Experimental, $Re \approx 44.5$

(c) Experimental, $Re \approx 49.1$

(d) Experimental, $Re \approx 52.3$

(e) Experimental, $Re \approx 62.3$

(f) Experimental, $Re \approx 250$

Figure 3.51: Evolution of flow from $8.5 \leq Re \leq 250$

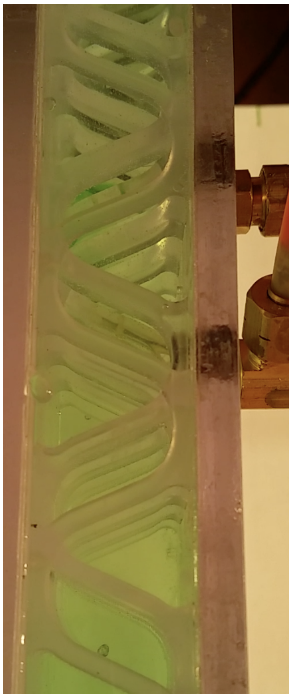


Figure 3.52: $Re \approx 250$, high diffusive nature of the flow

3.5.2 LDV Validation

As flow visualisation of the flow has adequately demonstrated that a transition to unsteady flow occurs at somewhere between $44.5 \leq Re \leq 49.3$, LDV is used to characterise the differences with respect to velocity fluctuations, particularly RMS fluctuations, indicative of turbulence. Using the same experimental water tunnel and test geometry, the LDV measurements are taken after 2.5 turbulizer convolutions along the centreline of the domain. This corresponds to a point used to probe velocity components in the numerical solution, and aids easy and direct comparison of flow unsteadiness.

Using a full range of Re, with a focus placed around the transition area, LDV measurements are obtained over a period of approximately 200 s, and the flow statistics are calculated based on collected data. An extremely sharp transition point is observed to occur approximately at $Re \approx 47$, where normalised RMS velocity fluctuations increase from 10%, signifying a slightly perturbed freestream flow, to more than 50%, where turbulent fluctuations, or at the very least, large scale fluctuations, are present in the flow.

The sharp transition observed to occur around $Re \approx 47$ is strongly indicative of reli-

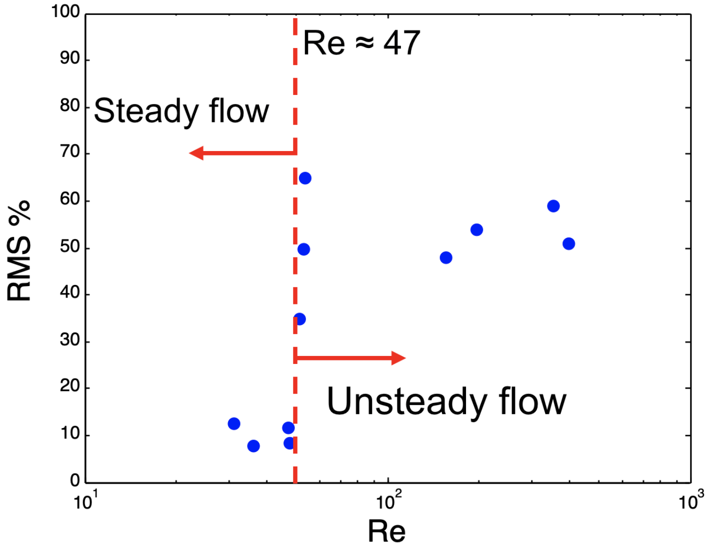


Figure 3.53: LDV measurement of normalised RMS velocity fluctuations

.

able and repeatable measurements of flow transition occurring within a relatively narrow window of flow Re.

3.5.3 Comparison with Numerical Results

In order to compare the numerical simulation results to the experimentally observed trends, a range of Re values are simulated with LES, focusing on $40 \leq Re \leq 50$. Velocity samples are taken at approximately the same location as that used in LDV, shown by the central red dot in Fig. 3.54. A Fast Fourier Transform (FFT) is performed on the data, and it is plotted as fluctuation magnitude vs. frequency. Although the -5/3 slope hypothesised by Kolmogorov [74, 75] is not observed at $Re = 43$, the FFT results indicate unsteadiness in the flow, Fig. 3.55 a). Fully turbulent flow is observed at $Re = 476$, in Fig. 3.55 b).

The transition

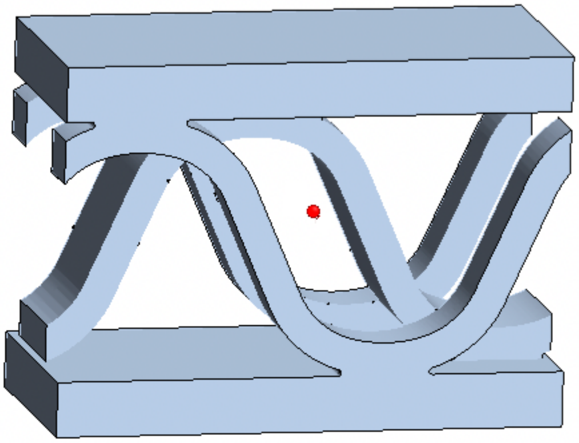


Figure 3.54: LDV and CFD measurement location

.

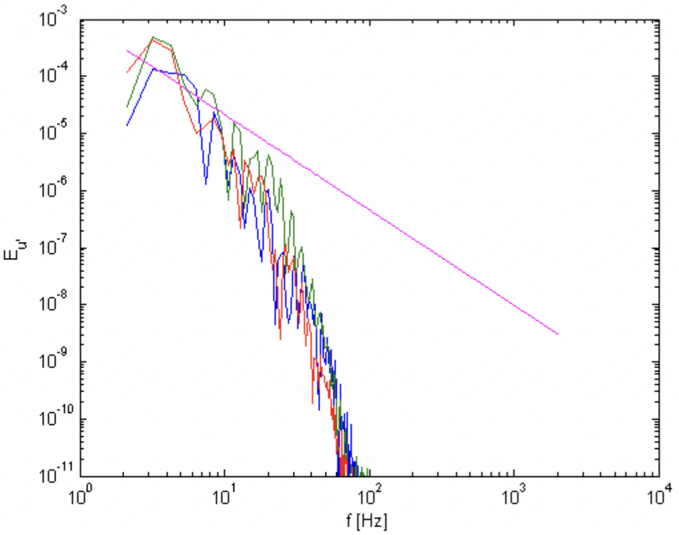
3.5.4 Experimental Validation Summary

Through a combination of qualitative flow visualisation and quantitative LDV measurements, a very well defined transition between steady and unsteady flow in the turbulizer is shown. The onset of the transition occurs consistently at $Re \approx 47$ in the experimental measurements, while numerical investigations reveal a transition to unsteadiness at $Re = 43$. Transitional flows are very commonly reported to be extremely difficult to accurately predict, so a result presenting less than 9% difference in calculated and experimental results is extremely encouraging. This is crucial to the accurate prediction of flow in the turbulizer, as many relevant flows occur at or around this Re value, particularly with oils or higher viscosity fluids.

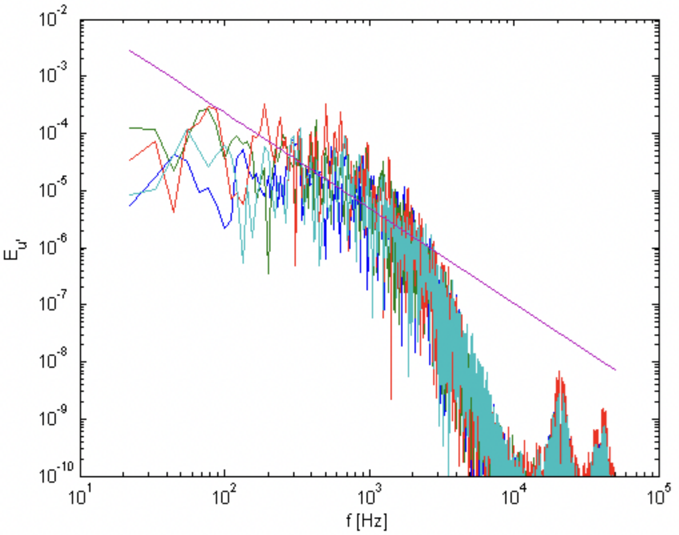
3.6 Summary

This chapter provides a comprehensive investigation of the construction, simulation and validation of turbulizer performance.

An array of different modifications are applied to nominal CAD turbulizer geometry to determine the most accurate representation of a "real" turbulizer. Furthermore, numerical



(a) Numerical, $Re \approx 43$



(b) Numerical, $Re \approx 476$

Figure 3.55: Comparison of the turbulent kinetic energy cascade at $Re \approx 43$ and $Re \approx 476$, -5/3 slope for reference

simulations of geometry of an as-manufactured turbulizer are compared to demonstrate efficacy of the model. Excellent agreement, far surpassing previous modelling attempts, is shown for both f_D and Nu with the modified geometry. The Nu error has been more than halved, while a slight reduction in nf_D is found due to the modifications. Optimal results are achieved using $0.83t$ as the nominal material thickness, with $0.17t$ as the corner radius of the turbulizer blades and 0.1 mm channel height reduction. These modifications are recommended for any future modelling prior to investing in prototype material for new turbulizer design.

Using the modified geometry, sensitivity of results is compared for several different turbulence models, including $k - \epsilon$, SST $k - \omega$ and LES. Laminar models are omitted as it is assumed that the flow becomes turbulent at a relatively low Re. On the basis of heat transfer predictions, LES most closely matches experimental results. There is a slight overprediction of heat transfer across the entire tested Re range, but the discrepancy tends to decrease as Re increases. Furthermore, it is observed that good results are achieved by modelling a single convolution of the heat exchanger, which indicates that the model with the lowest computational requirements is adequate for future use.

Experimental validation of the LES model are extremely successful. The transitional Re (from steady to unsteady flow) calculated numerically is within 9% of experimentally obtained values. This, combined with accurate heat transfer and pressure drop results, provides an excellent level of confidence in the overall accuracy of the numerical results obtained.

Taking into account the findings presented in this chapter, the aforementioned numerical model is determined to be adequate for development of a full flow library for the present turbulizer geometry. In the following chapter, Ch. 4, the geometry is characterised for a range of Re, and ϕ (flow angle) values.

Chapter 4

Turbulizer Flow Library: Generation and Calculation

In Ch. 3, it is shown that with some modifications to existing turbulizer modelling techniques, significant improvements to the accuracy of results may be obtained. Using the improved predictions, it is possible to generate a library of flows, which may be used as the basis of a model able to predict a large scale heat transfer enhancement surface without the complexity required to fully simulate the geometry.

The flow conditions within a turbulizer almost always incorporate a wide variety of flow velocities and angles, as heat exchangers are often designed to fit available packaging space, rather than optimizing for performance. As such, compromises are often made with regards to the shape of the intended flow path. This leads to significant problems when using orthogonally oriented flows to represent performance over a wide range of flow angles. As such, it is desired to discover the performance of a turbulizer in non-orthogonal flow, i.e. flow which is not primarily oriented in the “high pressure drop” (HPD) or “low pressure drop” (LPD) directions.

Expanding on the elucidation of the aforementioned variable flow angle, it is also desired to combine the results into a model that provides a range of flow rates, angles and fluid properties. Proper representation of this performance into a single model that is dependent only upon three variables, flow rate, fluid properties and flow angle, i.e.:

$$(f_D, Nu) = f(Re, Pr, \phi), \tag{4.1}$$

is desired for easy access and representation. Pre-calculation of this model front loads the computational expense, and is thus much more desirable for practical usage in industry-relevant applications. This model calculation may be performed once for each variation of heat transfer enhancement surface, and the results subsequently used in any numerical model using that particular geometry in the future without requiring any additional computational expense.

In this chapter, it is therefore undertaken to:

1. **Study turbulizer performance under a wide variety of flow conditions**
2. **Combine the performance results into a computationally efficient ROM or submodel capable of doing a simple lookup of turbulizer performance**
3. **Compare the impact of different predictive/interpolation techniques for the ROM/submodel for maximum performance and efficiency.**

Turbulizer performance is studied under industry relevant flow conditions, which allow for a full understanding of the various transitions and characteristic changes that occur in the flow. This includes the transition from creeping, steady flow, to laminar unsteady, to fully turbulent flow. The angles included also describe the flow of the turbulizer across the entire range of possible directions, split into 19 discrete increments.

The combination of the results into the Reduced Order Model, ROM, is studied using two distinct approaches. An artificial neural network is trained using simulation data, and the results are used to provide a reference calculation that is able to predict performance with any combination of input Re, Pr, or ϕ. This is contrasted with using a simpler bilinear interpolation that uses existing test or numerical data to act as reference flow data. This calculation is then used to link the global heat exchanger model with the smaller, high fidelity model of the turbulizer.

These two approaches are then compared to understand the relative strengths and weaknesses provided by each of the two methodologies. A suitable model is then selected and discussed, and results are selected to subsequently be incorporated into the completed ROM.

With the completion of these objectives, sufficient data are collected to allow for the proposed submodel to be incorporated into a full model for prediction in a global heat exchanger model, thus bridging the performance of small scale flow with the results desired in a global heat exchanger scale.

4.1 Turbulizer Modelling

As the modelling of a turbulizer has not been previously investigated over such a wide variety of flow angles and *Re* conditions, the following sections provide an analysis of the detailed results obtained in this study. The range of *Re* is manipulated to provide a full understanding all the way from low-*Re*, creeping flows, to highly turbulent flows, with $Re \geq 150$, where the pressure drop becomes largely driven by inertial contributions. Following the description of the *Re* and ϕ ranges used in the study, both the heat transfer and pressure drop bulk values are evaluated and trends are discussed. Finally, general observations are presented and compared with prior knowledge of flow characteristics, and newly observed trends are discussed.

4.1.1 Flow Boundary Conditions

To model the turbulizer across a wide variety of flows, the model developed for predicting as-manufactured turbulizer geometry in Ch. 3 is extended. The periodic flow conditions applied at both sets of boundaries are selectively manipulated to provide both a mean effective mass flow rate and a bulk flow angle, by adjusting the proportion of the mass flow in the HPD (front and rear of the turb geometry) and LPD (side to side) directions. This results in a known mass flow, thus controlling *Re*), and a known flow angle, ϕ. These boundary conditions are selected based on both proportionality to one another, as well as consideration of the open-channel velocity, which is a component of the *Re* calculation. By maintaining steady boundary conditions, the overall flow of the turb geometry is isolated from transients and instabilities that may form in larger geometric representations.

In the pursuit of providing alternative means of specifying flow direction and appropriate boundary conditions, alternative methods were explored, but no suitable alternative was found. Investigated techniques include, generation of a circular domain, following [50], however, subdivision of the domain and appropriate specification of appropriate velocity profiles proved infeasible. Additonal, irregular patterns of geometry subdivision were also investigated, but these approaches required generation of a new geometry for each individual flow angle, and were significantly less flexible than the present approach.

Within the confines of the turbulizer CFD model, the boundary conditions are manipulated to generate a range of flow conditions. For preliminary testing, the entire dataset is modelled, then subgroups are combined and tested in order to provide an analysis of the most efficient means of generating an accurate representation of the entire dataset.

In particular, the flow conditions are changed to achieve the range of Re and ϕ listed in Table 4.1. The resultant 19 x 19 matrix yields a total number of 361 different flow conditions desired for simulation. Each of these conditions represents a flow condition that can be used for lookup or validation at a later date. Generation of this table represents the most computationally intensive portion of the entire study.

On average, a single flow condition requires approximately 3h of wall time using Dana Canada's 196 CPU computational server. The server consists of a total of 14 nodes with 14 CPU's each. Each computational node possesses a total of 128 GB of memory, with intern-

Table 4.1: Turbulizer flow Re and ϕ conditions ($0^\circ = HPD$ and $90^\circ = LPD$)

Re	Flow Angle																		
-	0°	5°	10°	15°	20°	25°	30°	35°	40°	45°	50°	55°	60°	65°	75°	75°	80°	85°	90°
0.1																			
1																			
3																			
10																			
20																			
30																			
40																			
50																			
75																			
100																			
125																			
150																			
200																			
250																			
300																			
350																			
400																			
450																			
500																			

ode communication handled by optical connections for the highest memory communication bandwidth. Total simulation time is therefore 588 CPU-h per simulation, or $\approx$ 24.2 CPU-y for generation of the entire simulation matrix. As such, it is desired to reduce this computational loading, if possible, to facilitate easier implementation. Despite this seemingly large computational cost, it can be accomplished using Dana Canada's computational resources in a total of $\approx$ 45 days of wall time. As computational servers continue to grow and improve in speed, this wall time requirement can be expected to rapidly decrease, generally following Moore's Law [81]. Further improvement can be expected by foregoing the LES model in favour of a more computationally efficient RANS model.

In consideration of the computational expense required for each $19x19$ matrix of flow conditions, it is elected to restrict the simulations to only $Pr = 7.0$. This moderate value is, in the experience of Dana Canada and as discussed in Ch. 3, capable of being scaled using the relation:

$$Nu_{new} = Nu_{reference} \left(\frac{Pr_{new}}{Pr_{reference}} \right)^n , \tag{4.2}$$

with the value of $n = 0.33$ commonly assigned. The advantage of this approach, in addition to numerical efficiency, is the ability to rescale the data should a change be desired due to updated experimental or numerical data. Care should, however, be taken not to scale Pr below a value of ≈ 5, due to the error cited in Ch. 3 for low Pr values seen in gases.

Each of the aforementioned flow conditions is post-processed to determine the non-dimensional friction factor, f_D, and Nusselt number, Nu. Post-processing computational requirements are trivial, as all salient details are time-averaged and output during the simulation into a text file.

4.1.2 Pressure Drop Results

The calculation of the pressure drop, or f_D, of the turbulizer through a variety of Re and ϕ conditions is shown in Fig. 4.1. This surface is seen to be at a minimum at high Re, and while oriented in the LPD direction. As the flow angle rotates towards the HPD direction, f_d increases. Of note is that it is not a linear increase for a fixed Re, but rather, there are subsections where f_D appears relatively insensitive to flow angle. The horizontal portion

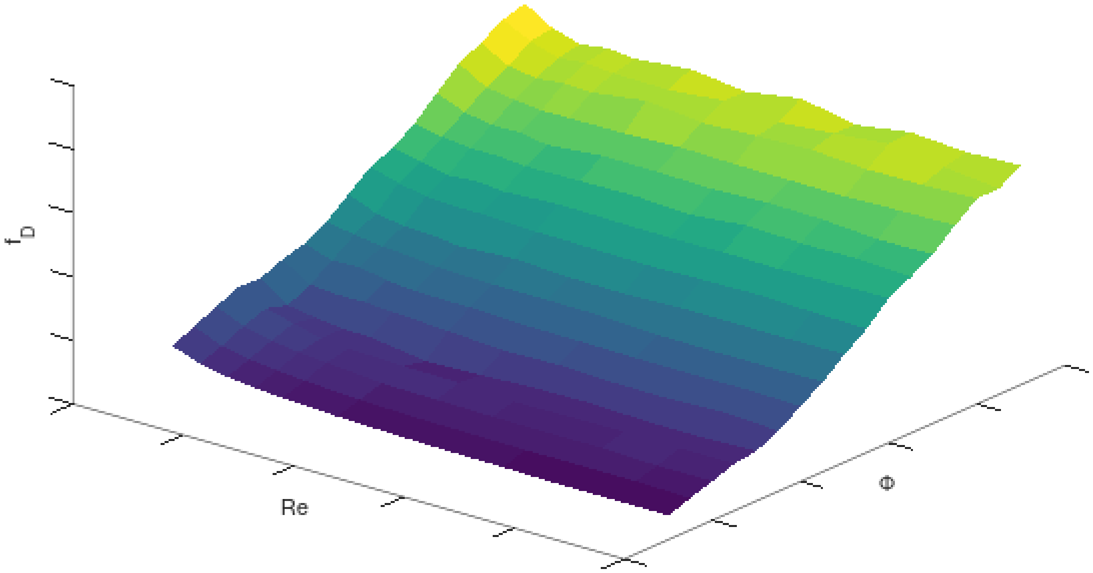
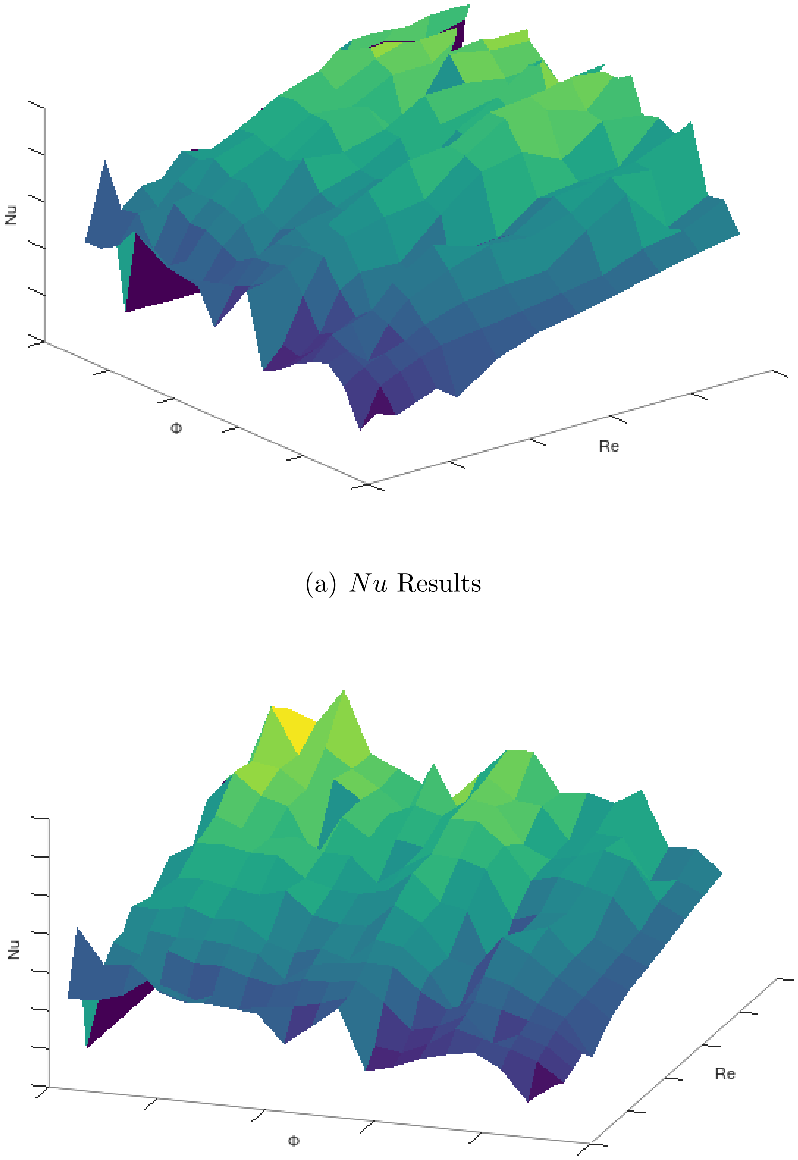


Figure 4.1: f_D results calculated from CFD

of f_D compared to ϕ at the bottom of Fig. 4.1 indicates that within a few degrees of the LPD direction, the flow generally isn't influenced. The slope increases as ϕ increases, but again levels out as the flow approaches the HPD direction. Furthermore, at a fixed angle, for example, either HPD or LPD, the change of f_D is most notable at low Re. It is common for the friction factor to remain relatively constant for highly turbulent flow, but increase quickly as Re decreases. This is typically due to the laminar nature of the flow where viscosity effects play a larger role, and offset the u^2 term in the friction factor equation. As Re continues to decrease, the friction factor will continue to increase due to this relationship. It is important to note that the actual pressure drop at low Re is typically quite low, unless dealing with a highly viscous flow.

4.1.3 Heat Transfer Results

Although no published data exist, the aforementioned f_D results generally followed expectations. Despite the somewhat predictable pressure drop of the turbulizer through a wide variety of flow conditions, the heat transfer results are far less intuitive. Initial observations of Fig. 4.2 show that there is a significant amount of oscillatory behaviour in the data. When viewing the data set from alternative angles, Fig. 4.2 (b), it becomes more clear that these sections of oscillations follow very regular patterns defined by ϕ. Several local maxima exist throughout the range of ϕ. These local maxima and minima arise from the interaction of upstream and downstream turbulizer blades at different angles. Due to the arrangement of the geometry, there is either a positive or negative interaction between the wake of an upstream turbulizer surface on a downstream blade. A positive interaction could be construed as a cascading effect, where the "spillage" of the wake of a turbulizer directly impinges on a downstream surface, providing a local reduction in boundary layer thickness, therefore increasing heat transfer. The local minima observed are likely due to a negative interaction of the wake with downstream blades, for example, a low velocity region, or wake, overlapping with a downstream turbulizer blade. This is also observable in that there are certain angles through which straight line "short-circuits" of the flow bath are available to the flow. This short-circuiting minimizes the requirements for the flow to change direction, and this undisturbed flow path allows for additional boundary layer growth, reducing temperature gradients in the fluid and, therefore, reducing heat transfer



(a) Nu Results

(b) Nu Results, rotated for ease of interpretation

Figure 4.2: Nu results calculated from CFD, viewed from two angles to highlight the change in performance relative to ϕ

of the fluid. This can be imagined as a shadowing of heat transfer wetted surface area by upstream wakes.

Despite this unexpected performance, the remainder of the trends generally fall in line with expectations. The overall heat transfer, or Nu, tends to increase with Re. This would generally be expected, as a higher velocity flow would maintain a higher temperature difference to the surface, thus driving up heat transfer. Likewise, the LPD orientation generally provides a lower overall heat transfer than the HPD direction. This performance is generally understood, as HPD orientations are used in higher heat transfer scenarios, while LPD turbulizer orientations are used when less overall heat transfer is required. It is therefore expected that the highest heat transfer performance occurs near the highest Re while oriented with the HPD direction.

4.1.4 Turbulizer Performance Observations

At this time, it is appropriate to reflect upon some of the results observed while testing the turbulizer under a variety of flow conditions not previously investigated. The most notable result is the significant changes in heat transfer performance with small changes in flow orientation. When observing turbulizer models, it becomes clear that there are several orientations where the gaps in turbulizers provide a straight flow path through the turbulizer, which would be expected to provide lower heat transfer performance, but also reduce pressure drop. While it is obvious that there is a significant reduction in heat transfer at these discrete angles, Fig. 4.2 (a) and (b), the corresponding decrease in pressure drop is not as clearly observed. As f_D is a function of the square of velocity, the reductions in pressure drop are not as noticeable at the discrete angles, but some points in the f_D curve exist when compared to changing ϕ.

This phenomenon has not been observed or recorded in any previous study examining similar geometries, but is expected to exist across a wide range of periodic heat transfer geometries, including turbulizers, offset strip fins and dimple plates. This observation provides a partial explanation at the difficulty in representing heat transfer performance in both analytical or previous numerical studies. As the flow tends to follow in these lower pressure drop orientations, heat transfer performance may be compromised more often than previously believed.

4.1.5 Limitations of Methodology

While the boundary conditions selected for simulation have been chosen carefully to accurately represent the flow within periodic sections of the turbulizer, there exists an unknown impact of entry and exit to the turbulizer domain. The effects of this have been shown to some extent previously in Fig. 3.10, where the impact of prediction accuracy is shown relative to the inlet, orr developing flow region, and exit where the flow is relaxing and transitioning towards channel flow. The variation on heat transfer for at least several convolutions is notable in the case. It remains unknown if the same phenomena exist for the LPD orientation of the turb. In this configuration, it is possible that a portion of the flow migrates into a "bypass channel" where the turb is unable to be positioned due to manufacturing features, such as die radius for the stamped plate formation. Previous work by Dana Canada has investigated the impact of the flow in these regions, however, the work remains proprietary and unpublished.

In addition to the impact of the literal edge cases of entry, exit and side proximity, the impact of exceeding the tested maximum or minimum Re for both f_D and Nu prediction remains a risk or limitation for the proposed methodology. Although not implemented in the present methodology, the test library, and trained ANN may be used to extend the Re range of the test data by calculating a form of f_D using the form:

$$f_D = \frac{A_f}{Re^m} \tag{4.3}$$

or

$$Nu = A_{Nu} Re^m Pr^n. \tag{4.4}$$

In this way, a predictable extrapolation of data may be executed without blind reliance on a trained ANN, which may or may not form actual physical patterns due to limited training data.

4.2 Artificial Neural Network

The concept of an ANN, previously introduced in Sec. 2.4.4, is used as a predictive model which can be trained with experimental or numerical training data, then used to predict performance as required by the external system, or the global heat exchanger model in this case. The artificial neural network takes advantage of the fact that trends can be captured by the training data which are not immediately obvious to an external observer. The downside, for nearly the same reasons, is that improper training may present a case where the network does not accurately represent the physics involved in the present problem, or suffers from overfitting. Thus, it is the purpose of this section to elucidate the performance afforded by application of a feed forward network (FF), using a backpropagation, or supervised learning, training method. Comparisons are made to the bilinear interpolation concept introduced later in Sec. 4.3 to determine:

1. **The efficacy of each approach**
2. **The time cost of obtaining training data**

Due to the numerically intensive process of generation of the training data set, it is thus important to minimize requirements for input data where possible. Training data set reductions are compared for each of the models in order to determine what kind of input variable resolution is required for training of a new data set for functional application within the context of a FF ANN.

4.2.1 Implementation - Fast Artificial Neural Netork (FANN)

The Fast Artificial Neural Network (FANN) is a codebase developed by Steffen Nissen as library of code designed for numerical efficiency and ease of implementation within a variety of different frameworks and for general purposes [63]. The code is developed using the *C* language, which due to its pervasiveness, makes it easily compiled and accessed through other language compilers as a *.so* file, or *Shared Object*, library. This allows for direct and straightforward inclusion within the User Functions feature of STAR-CCM+, allowing for more advanced calculations than typical Field Functions allow [45]. This increase in functionality affords the ability to externally train and generate network data, which can arbitrarily be accessed when required by STAR-CCM+.

The training implementation of FANN is achieved using several functions:

```
#include "floatfann.h"

int main()
{
        const float connection_rate = 1;
        const float learning_rate = 0.7;
        const unsigned int num_layers = 3;
        const unsigned int num_input = 2;
        const unsigned int num_neurons_hidden = 4;
        const unsigned int num_output = 1;
        const float desired_error = 0.0001;
        const unsigned int max_epochs = 500000;
        const unsigned int epochs_between_reports = 1000;

        struct fann *ann = fann_create(connection_rate, learning_rate,
           ↪ num_layers, num_input, num_neurons_hidden, num_output);

        fann_train_on_file(ann, "trainingData.data", max_epochs,
           ↪ epochs_between_reports, desired_error);

        fann_save(ann, "function.net");

        fann_destroy(ann);

        return 0;

}
```

The definition and creation of the function is controlled by *fann_create*, while training data, specifically *trainingData.data* is input, along with the definitions desired for training the function. Output of a trainer neural network is provided by *function.net*, which is required to subsequently use the trained network for output calculation from an input data set [63]. When application of the trained network is required, it is simply called using:

```
#include <stdio.h>
#include "floatfann.h"

int main()
{
        fann_type *calc_out;
        fann_type input[2];
        struct fann *ann = fann_create_from_file("function.net");
        input[0] = 0;
        input[1] = 1;

        calc_out = fann_run(ann, input);

        printf(" test (%f,%f) -> %f\n",
        input[0], input[1], calc_out[0]);

        fann_destroy(ann);

        return 0;
}
```

This simplified approach to generation and usage of the library makes it an ideal candidate for cross platform development, and is thus, selected for usage in the present research. Application of the code to the present study is given in Appendix E.

Benchmarking and validation on publicly available datasets is available in greater detail in [63].

4.2.2 Training

Training of the neural network is achieved by considering the entire data set, then selectively removing training data until results are no longer suitable for use. In particular, the entire dataset consists of all entries in Table 4.2.

To aid in assessing the sensitivity of results to changes in training data, several regular and irregular permutations of the data set, including selection of borders only, Table 4.3, a cross formation, Table 4.4, and random selections, incorporating between 25% to 75% of the training data, Table 4.5 to 4.7, are chosen. The random data selections are performed 8 times while using 25% of the full data set, 6 times while using 50% and 4 times while using 75% in order to collect different distributions of randomly selected cells, and the average error of the samples is taken.

Table 4.2: Full training data set calculated from CFD, all cells have known numerical solutions

Re	Flow Angle																		
-	0°	5°	10°	15°	20°	25°	30°	35°	40°	45°	50°	55°	60°	65°	75°	75°	80°	85°	90°
0.1																			
1																			
3																			
10																			
20																			
30																			
40																			
50																			
75																			
100																			
125																			
150																			
200																			
250																			
300																			
350																			
400																			
450																			
500																			

Following the selection of the training data, several network configurations are

Table 4.3: Data set including only borders of flow conditions

Re	Flow Angle																		
-	0°	5°	10°	15°	20°	25°	30°	35°	40°	45°	50°	55°	60°	65°	75°	75°	80°	85°	90°
0.1	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
1	■																		■
3	■																		■
10	■																		■
20	■																		■
30	■																		■
40	■																		■
50	■																		■
75	■																		■
100	■																		■
125	■																		■
150	■																		■
200	■																		■
250	■																		■
300	■																		■
350	■																		■
400	■																		■
450	■																		■
500	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■

Table 4.4: Data set including borders as well as “diagonal” flow conditions

Re	Flow Angle																		
-	0°	5°	10°	15°	20°	25°	30°	35°	40°	45°	50°	55°	60°	65°	75°	75°	80°	85°	90°
0.1	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
1	■	■																■	■
3	■		■														■		■
10	■			■												■			■
20	■				■										■				■
30	■					■								■					■
40	■						■						■						■
50	■							■				■							■
75	■								■		■								■
100	■									■									■
125	■								■		■								■
150	■							■				■							■
200	■						■						■						■
250	■					■								■					■
300	■				■										■				■
350	■			■												■			■
400	■		■														■		■
450	■	■																■	■
500	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■

Table 4.5: Data set including borders plus randomly selected cells, 25% of total computational points

Re	Flow Angle																		
-	0°	5°	10°	15°	20°	25°	30°	35°	40°	45°	50°	55°	60°	65°	75°	75°	80°	85°	90°
0.1	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
1	■																		■
3	■	■														■			■
10	■									■				■					■
20	■																		■
30	■				■				■										■
40	■																		■
50	■																		■
75	■										■							■	■
100	■													■					■
125	■					■													■
150	■						■			■									■
200	■											■							■
250	■		■																■
300	■															■			■
350	■			■									■						■
400	■									■									■
450	■																	■	■
500	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■

Table 4.6: Data set including borders plus randomly selected cells, 50% of total computational points

Re	Flow Angle																		
-	0°	5°	10°	15°	20°	25°	30°	35°	40°	45°	50°	55°	60°	65°	75°	75°	80°	85°	90°
0.1	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
1	■		■			■			■				■						■
3	■	■		■	■			■	■		■		■			■			■
10	■		■	■			■			■	■			■			■	■	■
20	■											■							■
30	■	■	■		■	■	■	■	■	■				■					■
40	■	■	■			■					■	■	■		■	■	■		■
50	■			■			■	■											■
75	■			■	■			■		■	■				■	■	■	■	■
100	■	■							■				■	■	■				■
125	■		■			■		■			■	■			■	■			■
150	■				■		■			■		■					■		■
200	■	■		■		■		■	■			■	■	■					■
250	■		■		■				■	■							■	■	■
300	■				■			■			■	■		■		■	■	■	■
350	■	■		■	■	■	■				■		■		■				■
400	■	■				■		■	■	■		■							■
450	■		■			■			■				■					■	■
500	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■

Table 4.7: Data set including borders plus randomly selected cells, 75% of total computational points

Re	Flow Angle																		
-	0°	5°	10°	15°	20°	25°	30°	35°	40°	45°	50°	55°	60°	65°	75°	75°	80°	85°	90°
0.1	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
1	■	■	■	■	□	■	■	■	■	■	□	□	■	■	■	■	■	■	■
3	■	■	■	□	■	■	□	□	■	□	■	■	■	□	■	□	■	■	■
10	■	■	□	■	■	■	■	■	■	■	■	□	□	■	■	■	■	□	■
20	■	□	■	□	□	□	□	■	□	□	□	■	■	■	□	■	■	■	■
30	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	□	□	■	■
40	■	■	■	□	■	□	□	□	■	□	■	■	□	■	■	■	■	□	■
50	■	■	□	■	□	□	■	■	□	■	□	□	■	■	□	■	■	■	■
75	■	■	□	■	■	■	■	□	■	■	■	■	■	□	■	■	□	■	■
100	■	■	■	□	□	■	□	■	□	□	■	■	■	■	■	■	■	■	■
125	■	□	■	■	■	□	■	■	■	■	■	■	□	□	■	□	■	□	■
150	■	■	□	■	■	■	□	□	□	■	□	■	■	■	□	■	■	■	■
200	■	■	■	□	■	■	■	□	■	■	■	□	■	■	■	■	■	□	■
250	■	■	□	■	■	□	■	■	■	□	■	■	□	■	■	■	□	■	■
300	■	■	■	■	■	□	□	□	□	■	□	■	■	■	■	■	■	■	■
350	■	■	■	■	■	■	■	■	■	■	■	■	■	□	□	■	■	□	■
400	■	□	■	□	□	□	□	■	■	■	□	■	□	■	■	□	□	■	■
450	■	■	■	■	■	■	■	□	□	□	■	□	■	■	■	■	■	□	■
500	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■

prescribed for comparison. These configurations are shown in Table 4.8. In each test case, the number of neurons in each layer are defined. Training is performed until convergence.

The calculations take approximately 1-10 minutes each, depending on model complexity, to perform training to convergence. Simpler configurations, such as 2 neurons in a single hidden layer, Test # 1, converge extremely quickly, while the most complex case with 3 hidden layers, Test # 10 takes approximately 10 minutes for calculation. All calculations are performed in a serial-CPU manner, as no parallelization of the processor is leveraged for calculation. Despite an order of magnitude difference in computational effort observed, all data sets take a trivial amount of training time in comparison to calculation of the training data itself. Therefore, the only selection criteria is based on the overall accuracy of each network configuration.

Table 4.8: Flow conditions

Test #	Hidden Layer Neurons		
	Layer 1	Layer 2	Layer 3
1	2	-	-
2	3	-	-
3	4	-	-
4	5	-	-
5	2	2	-
6	3	3	-
7	4	2	-
8	4	4	-
9	5	5	-
10	5	5	5

4.2.3 Results

Following calculation of the neural networks, preliminary analysis investigates the overall accuracy of the Nu and f_D functions against the full available data set. It is expected that minimal error will occur on the points used for training, however, of primary interest is the performance across the entire range. Due to higher overlap of training vs. test data in the 50% and 75% utilisation scenarios, it is also expected that a lower overall error is achieved. It follows that the two primary goals of analysis are:

1. determination of the most suitable ANN architecture for both Nu and f_D
2. determination of the minimal dataset required for training that provides acceptable results

General Training Results

Overall error of the calculations is shown in Fig. 4.3 and Fig. 4.4. Preliminary investigation shows two major trends. Firstly, as the training set increases in size, the error generally decreases. Secondly, as the model increases in complexity, so too does the general accuracy. Two and three layer networks are generally seen to achieve better performance when compared to their single layer counterparts in this context. The other noticeable difference is the overall order of error observed between the two variables being predicted. Nu predictions are associated with a much higher overall level of error, typically greater than 10%. When compared to the average error for f_D, the three layer, 5 neuron per layer network (5-5-5), generally performs within 2% error of the numerical results when more than 50% of the overall dataset is used for training. This is likely due to the relative simplicity of the f_D response surface shown previously, whereas there are much larger variations in Nu with small changes in either Re or ϕ.

In consideration of these results, it appears that the 5-5-5 ANN is well suited to predictions of f_D, while the 5-5 ANN is more suited for prediction of Nu. At this stage, it is suspected that the increased complexity of the 5-5-5 network allows for overfitting of the Nu data.

Another observation is the effectiveness of the randomly sampled data sets. Even at 25%, a surprising level of accuracy is achieved. The results, however, converge more when at least 50% of the data set is used for training. Some variability in results exists due to the random nature of data set sampling. Furthermore, both the “cross” and “border” data sets are wholly unsuited for prediction, and provide significantly worse model training than the randomly selected sets. For clarity, these data are omitted from the plots.

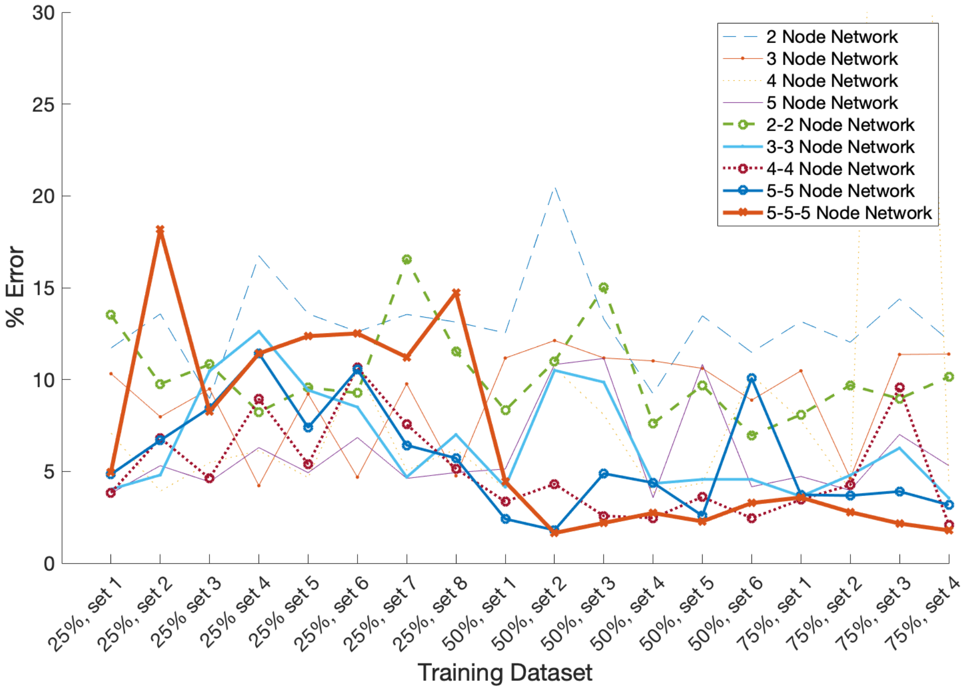


Figure 4.3: f_D average error, ANN results

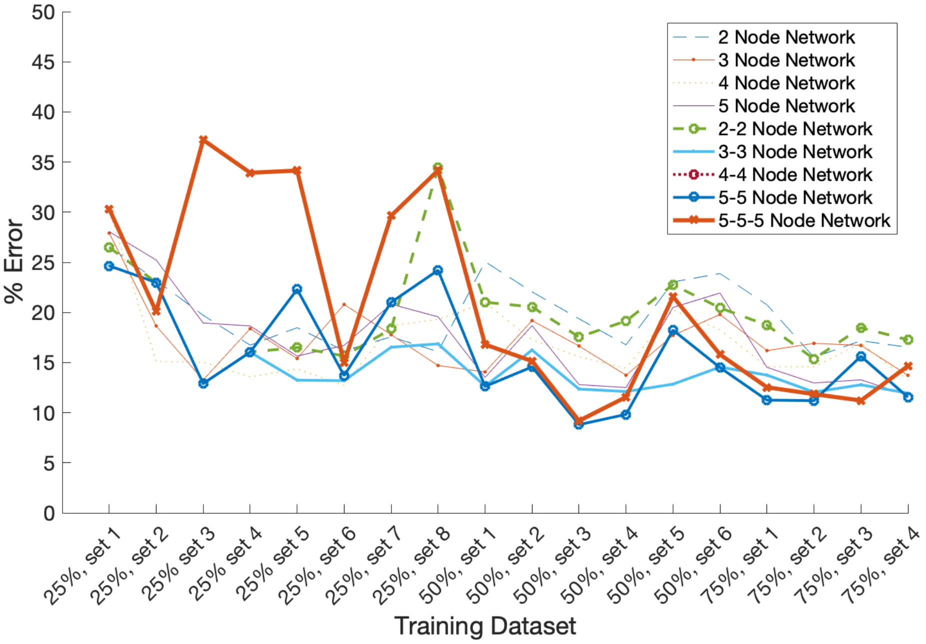


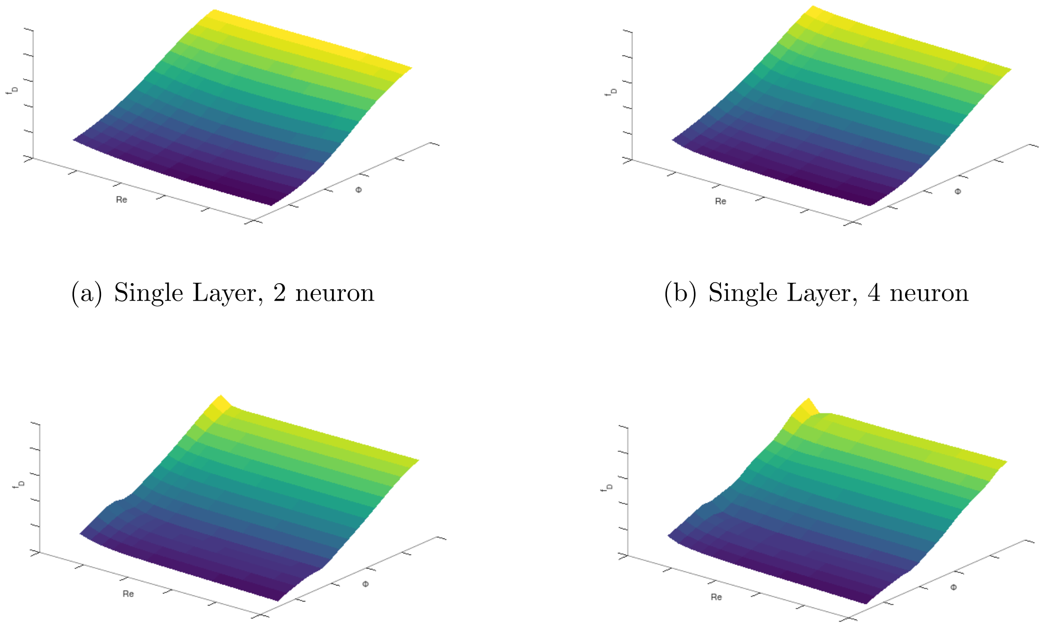
Figure 4.4: Nu average error, ANN results

Sensitivity to Network Architecture

Details of the network response for f_D are shown in Fig. 4.5 (a) - (d), and show the impact of the additional neurons and layers on the complexity of the results. As neurons and layers are added, the shape of the prediction surface begins to follow more closely with the full data set. For each of these examples, a 50% training data set is used.

In the case of f_D, the general shape of the response surface stays relatively constant, but the effects of increased complexity may be observed when comparing Fig. 4.5 (d) with the calculated values, shown previously in Fig. 4.1. This general agreement is borne out in the excellent average error observed in the 5-5-5 network, which, when using a large training data set, is less than 1%.

In the case of Nu, Fig. 4.6, there exists much less overall agreement with the shape of the results surface, Fig. 4.2. The oscillations observed in the numerical data appear to create significant issues in the more complex neural networks, where symptoms of overfitting become apparent. While Fig. 4.6 (a) and (b) generally resemble the surface in Fig. 4.2, despite giving a larger average error. As complexity of the network increases, so too



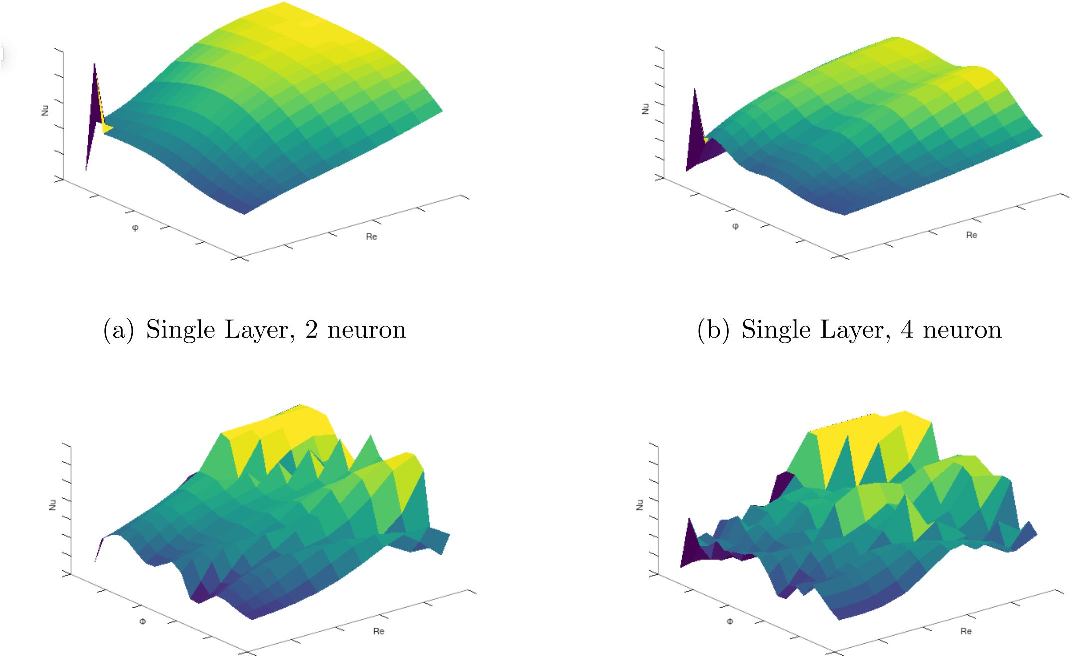
(a) Single Layer, 2 neuron

(b) Single Layer, 4 neuron

(c) Two Layer, 4-4

(d) Three Layer, 5-5-5

Figure 4.5: Comparison of ANN prediction of f_D data based on increasingly complex ANN architecture, 50% of data set used for training



(a) Single Layer, 2 neuron

(b) Single Layer, 4 neuron

(c) Two Layer, 4-4

(d) Three Layer, 5-5-5

Figure 4.6: Comparison of ANN prediction of Nu data based on increasingly complex ANN architecture, 50% of data set used for training

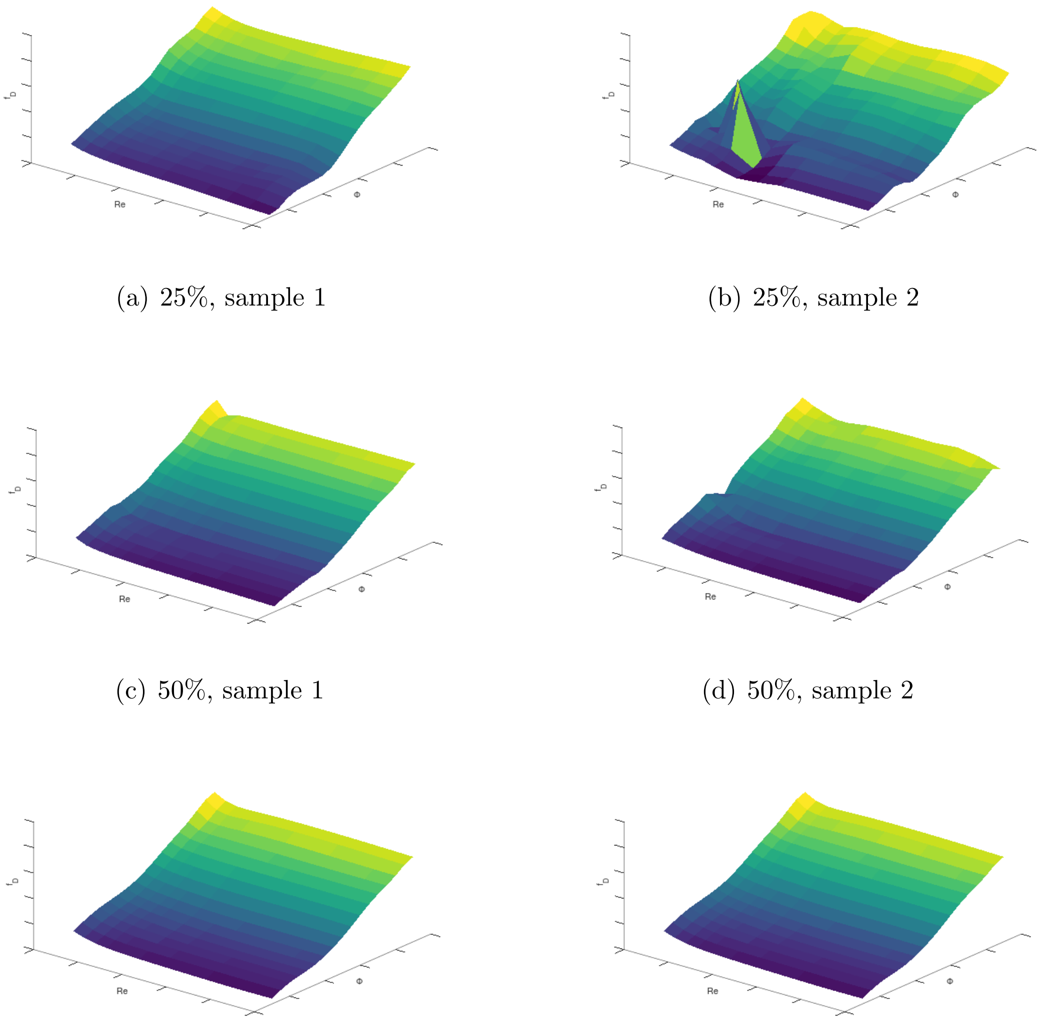
does the complexity of the results. The overfitting is seen in the "spikes" in the results, which indicate the network is trained to provide a specific result at a given coordinate. While this is obvious to an observer, it is not obvious when investigating the average error of the networks. In the case of this data set, it is fairly easy to provide an analysis of the entire range of variables, but as the problem becomes more complex and possesses more input variables, this becomes more and more difficult to identify, highlighting the overfitting challenges of ANN's. For this reason, a more simplistic network, the dual layer 4-4 network, is chosen for future use.

Sensitivity to Training Data Set

While general performance of the various trained networks has previously been examined, one overarching goal of the present analysis is to determine the minimum data set required to provide suitable network training. For each of the selected networks, a 5-5-5 network for f_D and a 4-4 network for Nu, samples of the randomly selected data sets, and their corresponding calculated results are shown in Fig. 4.7 and Fig. 4.8.

While it is obvious that the results of Fig. 4.7 show a general progression towards an improved prediction, Fig. 4.7 (b) shows a notable artefact of overfitting, where the network was most likely initialised with random weights that followed a local minima for gradient training, and was not able to reach the global minimum. As the training data set is expanded to 50%, this is no longer observed, and the predictions follow the calculated data set closely. While progressing to 75% of the available data set used for training, results are improved, but not in a manner such that it justifies the extra computational cost of calculating the training data.

In the case of Nu, it is obvious that the physical problem being represented is much more complex than for f_D. This is immediately apparent by the randomness that is presented in Fig. 4.8 (a) and (b). As the training data set grows, the surface more closely resembles the numerical results shown previously in Fig. 4.2, particularly when using up to 75% of the available training data, Fig. 4.8 (e) and (f). Only in these cases does the surface begin to represent the calculated Nu. At this stage, it is extremely important to avoid overfitting of the network, which can be introduced through increased network complexity or reduced training data availability. Thus, in order to make the fullest use of



(a) 25%, sample 1

(b) 25%, sample 2

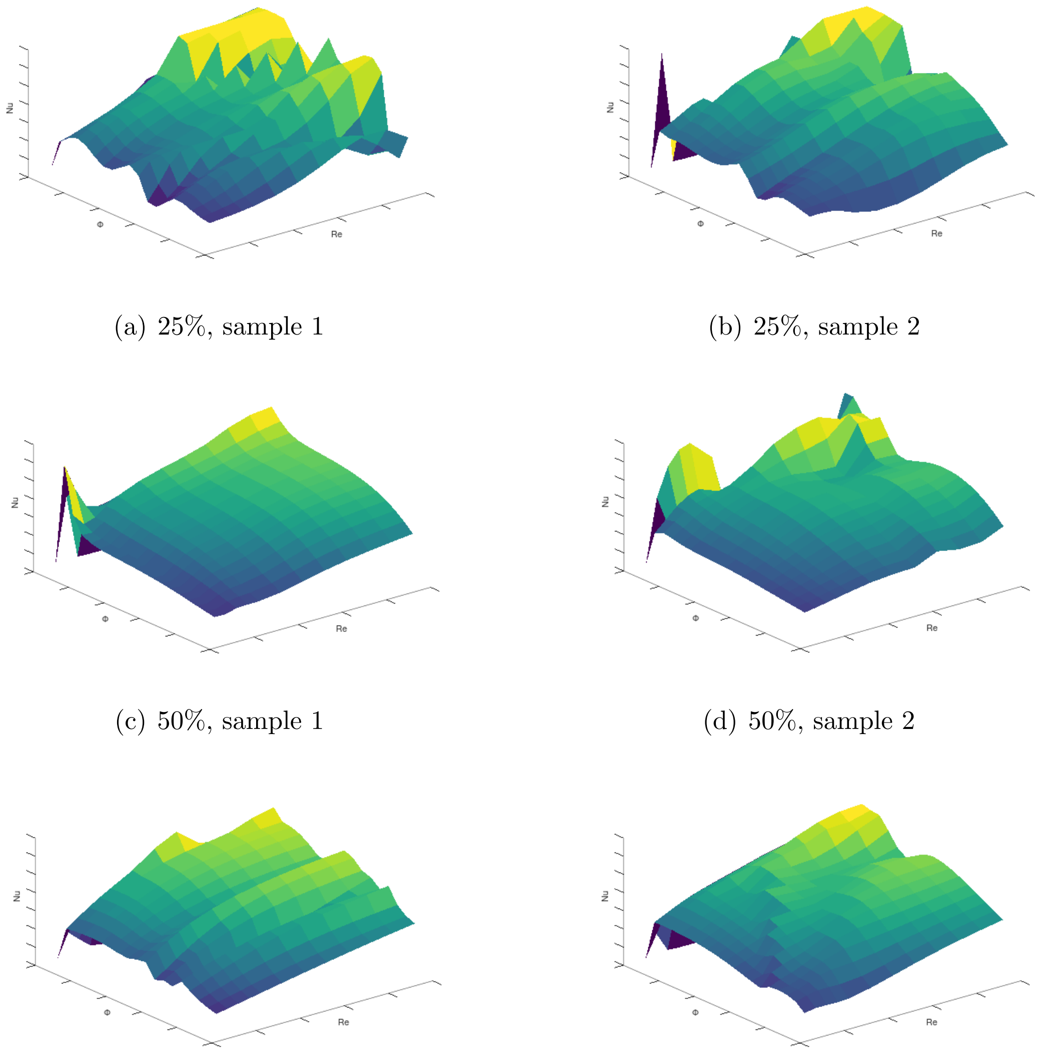
(c) 50%, sample 1

(d) 50%, sample 2

(e) 75%, sample 1

(f) 75%, sample 2

Figure 4.7: f_D sensitivity to various training data inputs, 5-5-5 network architecture



(a) 25%, sample 1

(b) 25%, sample 2

(c) 50%, sample 1

(d) 50%, sample 2

(e) 75%, sample 1

(f) 75%, sample 2

Figure 4.8: Nu sensitivity to various training data inputs, 4-4 network architecture

the training data, it is recommended that the data set be expanded to 75% of the available points in order to maximise the accuracy of the Nu prediction network.

4.3 Bi-Linear Interpolation

Perhaps the most straightforward application of the numerical results is through use of an interpolation method, particularly bilinear interpolation. Although it is acknowledged that the end form of the f_D or Nu curve may be highly non-linear, there is no *a priori* knowledge about the shape of the end function. A linear interpolation method is chosen to represent an appropriate starting interpolation methodology without knowing whether or not the response will form a higher-order function.

As shown in Table 4.1, only the values of *Re* and ϕ are available for interpolation, allowing a simplified scheme of:

$$f(x,y) = a_0 + a_1 x + a_2 y + a_3 xy, \tag{4.5}$$

where

$$\begin{aligned}
a_0 &= \frac{f(Q_{11})x_2y_2}{(x_1-x_2)(y_1-y_2)} + \frac{f(Q_{12})x_2y_1}{(x_1-x_2)(y_2-y_1)} + \frac{f(Q_{21})x_1y_2}{(x_1-x_2)(y_2-y_1)} + \frac{f(Q_{22})x_1y_1}{(x_1-x_2)(y_1-y_2)},\\
a_1 &= \frac{f(Q_{11})y_2}{(x_1-x_2)(y_2-y_1)} + \frac{f(Q_{12})y_1}{(x_1-x_2)(y_1-y_2)} + \frac{f(Q_{21})y_2}{(x_1-x_2)(y_1-y_2)} + \frac{f(Q_{22})y_1}{(x_1-x_2)(y_2-y_1)},\\
a_2 &= \frac{f(Q_{11})x_2}{(x_1-x_2)(y_2-y_1)} + \frac{f(Q_{12})x_2}{(x_1-x_2)(y_1-y_2)} + \frac{f(Q_{21})x_1}{(x_1-x_2)(y_1-y_2)} + \frac{f(Q_{22})x_1}{(x_1-x_2)(y_2-y_1)},\\
a_3 &= \frac{f(Q_{11})}{(x_1-x_2)(y_1-y_2)} + \frac{f(Q_{12})}{(x_1-x_2)(y_2-y_1)} + \frac{f(Q_{21})}{(x_1-x_2)(y_2-y_1)} + \frac{f(Q_{22})}{(x_1-x_2)(y_1-y_2)}.
\end{aligned}$$

The variables x and y are the independent variables in the interpolation scheme, where $f(Q_{xy})$ represents the function value at the independent variables enclosing the point of interest. This scheme may be extended to extrapolate outside of the region of interest, however, extreme caution must always be used with any extrapolation, as the data may become unphysical due to the nonlinear nature of the physical processes being represented.

Although slightly tedious to implement, this formulation is a simple interpolation between 4 points. The performance of this model is primarily defined by the grid density of the sample data: at any particular point of interpolation within the grid, the result will be entirely bounded by the function values at the corners of the region of interest.

The results of any point within the test data can be visualised by following the surface of Fig. 4.1 or Fig. 4.2. The plotting method used to obtain this response surface makes use of a bilinear interpolation scheme, and thus represents the model output for any point selected within the computational grid.

Although simplification of the test data set may be accomplished with *a priori* knowledge of the results, it is difficult to predict the performance of the test data set without unintentionally missing salient features, and thus simplification is performed at significant risk of degrading the quality of the results by failing to capture any features within the results.

The results of bilinear interpolation are illustrated and compared to those of the ANN in the following section, Sec. 4.4.

4.4 Interpolation-ANN Comparison

In order to calculate a baseline of error for bilinear interpolation, an implementation relying on Delaunay Triangulation [82] is used for direct comparison to the tested data sets used for ANN training. Although Delauney Triangulation is used for comparison, actual implementation of an interpolation favours the more simplified method in Eq. 4.5. As the end-user is in control of the selected training data set, this simplified interpolation method is not expected to be a limiting factor in model implementation.

When comparing the results of a more simplistic bilinear interpolation based model to more advanced ANN models, several pros and cons exist for each. The primary advantage of interpolation is that the model is numerically bounded when operating within the test data set. Thus, there are no unexpected results that may provide errant data points or misleading model results. The primary downside is that interpolation requires a relatively fine grid of test points to capture relevant data performance. ANN performance can be extremely accurate on limited training data, such as that observed when predicting f_D, or can be problematic when predicting oscillatory data sets, such as with Nu. The propensity for overfitting always exists, and is not always obvious to the user when it may have occurred, and is therefore difficult to avoid. It should also be stated that another major strength of the ANN is including additional training data sets, such as those based on experimental test data for either heat transfer or pressure drop may also be included with ANN generation, which significantly enhances their flexibility and applicability to a wider range of problems.

Each of the data sets used previously for neural network training are adapted for testing with bilinear interpolation. The results are compared against the benchmark results provided for the 5-5-5 network for f_D data and the 4-4 network for Nu data. this allows for a clear comparison of performance comparison between the two methodologies.

In Fig. 4.9, the average f_D error observed with each of the training data sets is compared against the ANN results. In nearly every circumstance, the neural network outperforms the simpler interpolation method. An interesting observation is that with the limited training data sets, particularly those that only use 25% of available training data, there is a strong correlation between error in both models, suggesting that a poor distribution of training data will inherently lead to a poor model fit, regardless of the chosen methodology.

In Fig. 4.10, the average Nu error observed between the two approaches is much more similar. Although there is no clear advantage to using neural networks, it does tend to perform better with limited data, similar to the results seen with f_D. Performance with larger data sets is of the same order between both the neural network and interpolation method.

From this comparison, the power of the trained ANN, especially with limited data availability, is observed. If the primary goal is to reduce the size of the training data set, and thus minimize computational effort in populating the model, the ANN approach is recommended. The neural network provides advantages in adaptability to various data sets, and is also capable of being trained using experimental test data, opening up an avenue not readily available with interpolation. For the remainder of this work, unless otherwise mentioned, it is assumed that the ANN model is used for calculation of turbulizer f_D and Nu values.

4.5 Summary

This chapter investigates the performance of a turbulizer across a wide range of flow conditions, as well as approaches which may be used to provide simplified models of this performance.

A grid of Re and flow angles, ϕ appropriate to conditions typically experienced by turbulizers is generated and used as a basis for comparison. Simulations are performed at all data points in order to further general understanding of performance. Several pre-

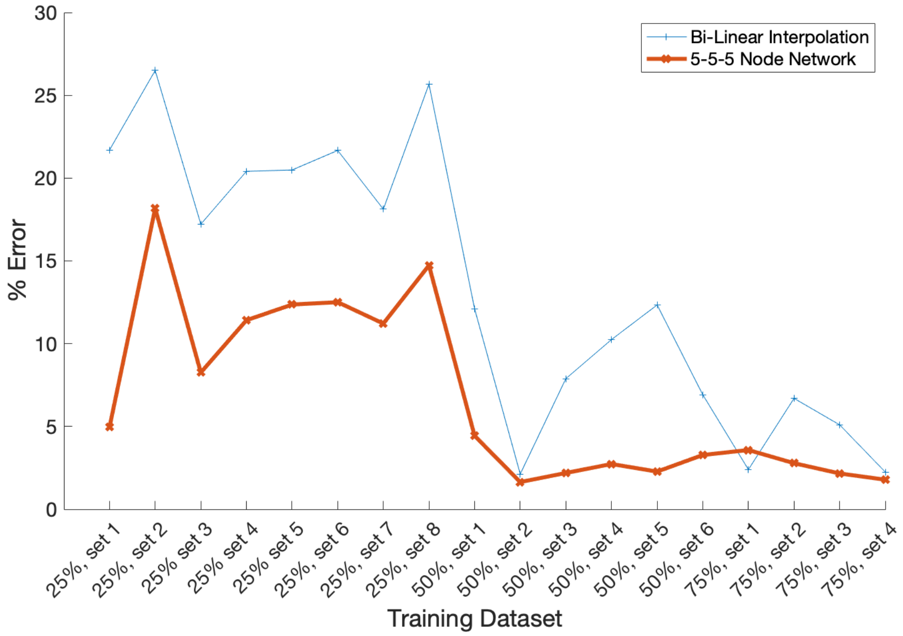


Figure 4.9: f_D average error, 5-5-5 ANN results compared with bilinear interpolation

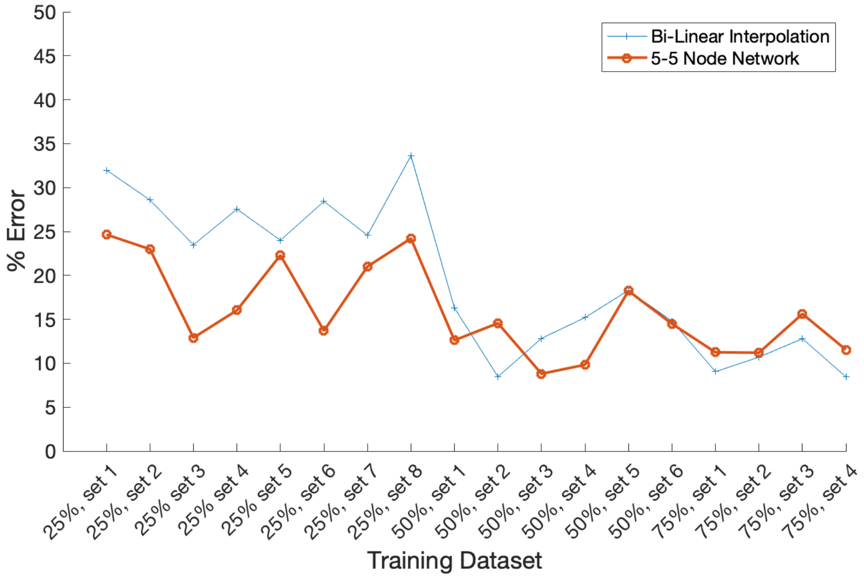


Figure 4.10: Nu average error, 5-5 ANN results compared with bilinear interpolation

viously undocumented trends are observed, particularly related to heat transfer. When certain flow angles are presented, heat transfer performance is severely reduced due to a continuous, straight fluid path available to the flow which bypasses many of the turbulizer features. This "short-circuiting" of the turbulizer degrades performance at certain flow angles, and should be avoided when designing heat exchangers. The f_D is also affected by these geometric conditions, although to a lesser extent than heat transfer. Turbulizer performance is otherwise generally confirmed to follow expected trends in terms of f_D and Nu with regards to increased, Re and changing ϕ. For reasons of computational cost, Pr sensitivity of the results is not investigated, as correlations already exist capable of scaling results to a wide range of Pr.

Two data representation/lookup techniques are introduced and evaluated.

The first, Artificial Neural Networks, are investigated as a means to reduce the required number of simulations for generation of turbulizer performance data. It is seen that f_D results respond well to application of an ANN, with relative error of less than 2% possible while using only half of the available training data, particularly with a three layer, 5 neuron per layer, ANN. Sensitivity to training data selection is also investigated and it is observed that the model generally performs very well across the range of data. Nu prediction using trained ANN's is substantially more difficult. The more oscillatory data set of the calculated Nu values presents challenges in training an ANN appropriately, and overfitting is observed with more complex networks, especially when presented with limited training data. The most appropriate results are observed using a two layer, 4 neuron per layer, ANN. This network architecture is selected due to its lack of observed overfitting on the complex data set for Nu. Average relative error is on the order of 8-10% across the full range of test values, which represents an acceptable level of error, but leaves room for improvement. An area of future development in the training of neural networks is inclusion of experimental test data. Although no such tests exist at Dana Canada, characterisation of turbulizer performance in the range of flow conditions investigated herein could provide a substantially more robust training data set for use in turbulizer predictions, particularly once a given design has entered production in one or more products, but remains available for future development.

The second evaluated method for representing turbulizer data is a bilinear interpolation technique. While this cannot be extended easily beyond two dimensions, it does provide a relatively accurate lookup for turbulizer performance. Interpolation generally relies on having a uniform, or at least regular, grid spacing. As the selection of test points may be prescribed by the user, this requirement does not provide any sort of impediment in its current usage. Simulation data set reductions as high as 50% and 75% (a factor of approximate 2 and 4 reduction of the initial data) are tested for interpolation accuracy, and compared against ANN performance. It is seen that the error of interpolation is generally related to the density of the sampling grid, but in nearly all cases, linear interpolation methods are outperformed by ANN's trained on similar quantities of data, and thus do not represent a desired solution.

Following simulation of turbulizer performance across a wide range of flow conditions and selection of ANN's as the primary method of data representation, turbulizer data may now be incorporated into the framework of a global heat exchanger CFD model, detailed in the following chapter, Ch. 5.

Chapter 5

Reduced Order Model Implementation

The reduced order model, previously mentioned in Ch. 1, has been introduced in vague terms, however, this chapter covers the detailed implementation of the proposed ROM in the commercial CFD code, STAR-CCM+. Although this particular implementation is developed specifically for use with STAR-CCM+, the formulation is presented in general terms, which will allow for an implementation of the model across a variety of different platforms, with only minor modifications required to adapt to specific software constraints.

The construction of the ROM relies heavily on the findings from Ch. 3 and Ch. 4, in which fine details about the flow are explored, and converted into a formulation that can be used as a lookup library for construction of the ROM. In general, the formulation of

$$Nu, f_D = f\left(Re, Pr, \phi\right), \tag{5.1}$$

where the local heat transfer at a location is defined by the value of Nu, and pressure drop is defined by the value of f_D. The local fluid properties and velocity define Re and Pr, while the reference flow angle is defined by ϕ. The value of ϕ is calculated based on the direction of the *local* flow relative to some reference direction, previously defined in the calculation of the flow library. Exact model implementation, including variables, submodels and assumptions required, are discussed below.

5.1 ROM Pressure Drop Formulation

Implementation of the ROM pressure drop model is relatively straightforward. The model is directly based on the implementation of the Darcy-Forchheimer model, for which the general formulation is:

$$\frac{\partial p}{\partial x_i} = -\frac{\mu}{\kappa} u_i - \frac{\rho}{\kappa_1} u_i^2, \tag{5.2}$$

where the coefficients κ and κ_1 represent the inertial and viscous coefficients, respectively. In the present formulation, local values for all variables are taken, so the equation can be reduced to a 1-dimensional formulation, where pressure drop in the *local* direction is used, and κ and κ_1 coefficients then effectively become isotropic, or rather only act in the direction of flow, which is always kept as the local reference. Due to the use of reference coefficient values that is calculated based on local Re, the use of two coefficients becomes redundant in the definition of $\frac{\partial p}{\partial x}$. Therefore, the value of $\frac{\rho}{\kappa_1}$ is arbitrarily set to zero, and κ becomes the sole reference for pressure drop. The modification to the Darcy-Forchheimer model then reduces to:

$$\frac{dp}{dx} = -\frac{\mu}{\kappa} u. \tag{5.3}$$

When this modified version is equated to the formula for Darcy friction factor, the result is:

$$\frac{dp}{dx} = -\frac{\mu}{\kappa} u = f_D \frac{\rho}{2} \frac{u^2}{D_h} \tag{5.4}$$

where

$$\kappa = \frac{-2 D_H \mu}{\rho f_D u}. \tag{5.5}$$

As the reference calculations from Ch. 4 provide data on f_D based on local flow conditions, these values directly control the pressure drop calculation in the turbulizer volume. This removes reliance on orthogonal κ and κ_1 values, and allows a representation for any fluid flow at any direction through the turbulizer. Furthermore, if variable fluid properties are used in the simulation, namely

$$\mu = f(T), \ \rho = f(T), \tag{5.6}$$

this can further be accounted for without modification to the model, providing maximum flexibility and accuracy with any calculations.

As presented, this implementation of ROM pressure drop calculation is compatible with the implementation of the porous media (Darcy-Forchheimer) model that is commonly implemented into many commercial CFD codes [45]. Modification of the coefficients to match implementation in a specific code framework may be performed on this formulation to facilitate usage.

5.2 ROM Thermal Implementation

Implementation of a thermal model requires additional modification and calculation beyond pressure drop. The primary reason for this is related to the need to calculate wall, or boundary, heat transfer values. A general formulation provides no guarantee that a given cell will be on a boundary, so manipulation of the model values to provide appropriate wall conditions is imperative. The fundamental value returned from the construction of the lookup are the wall Nu, where:

$$Nu = f\left(Re, Pr, \phi\right) = \frac{hD_h}{k}. \tag{5.7}$$

Rearranging for the local heat transfer coefficient, h, yields the form:

$$h = \frac{Nu\, k}{D_H}. \tag{5.8}$$

At a wall boundary, the heat transfer coefficient is used to determine the local heat flux from a wall-bound cell:

$$Q_{wall} = h_{bulk} A \left(T_{bulk} - T_{wall}\right). \tag{5.9}$$

The usage of bulk, or cross-stream averaged coefficients, eliminates dependence on local mesh spacing, but rather uses an average fluid temperature, which is analogous to the method used for determination of Nu in Ch. 4. Thus, any special need for the user to determine local cell spacing and dependence on internal turbulizer temperature distribution normal to the boundary is obviated.

Due to the formulation of the field functions accessible to the user in STAR-CCM+

[45], it is possible to fully define and restrict heat transfer modelling at a given boundary, while defining a known heat flux across a surface. This eliminates any contribution from sources related to the Laplacian diffusion of heat beyond the boundary and provides full control of the thermal performance of the model.

The natural risk of forcing a heat flux across a boundary is that it could artificially create unphysical performance on either side of the boundary. However, due to the formulation of Eq. 5.9, local conditions are taken into account on both sides of the boundary. Thus, the local heat flux, q_{wall} is always maintained in the proper direction, i.e. thermal energy is flowing in the direction of $T_{hot} \Rightarrow T_{cold}$.

The key to the successful application of this thermal model relies primarily on the work accomplished in Ch. 3, however, successful application within the scope of a CFD code is nuanced, and as such, the details of implementation are covered in detail in the following section, Sec. 5.3.

5.3 Implementation within STAR-CCM+

The use of STAR-CCM+ as a platform is chosen at the request of project sponsor Dana Canada. The implementation highlighted is specific to this application, however, a generalised application in most CFD codes may be based on the details outlined herein. The setup and implementation is outlined in chronological order, as required for application to a simulation.

5.3.1 Calculation Planes

Prior to embarking on any kind of calculation regarding the heat exchanger geometry, reference planes are placed at the centre of each turbulizer channel, both *hot* or *cold*, for future calculation of centreline (denoted by $_{CL}$ subscript) values of flow parameters. This reference is particularly useful when dealing with bulk, or channel average, values, which are projected onto the centreline plane for calculation.

In the case of STAR-CCM+, it is necessary to include each of the turbulizers as a separate region. This facilitates interpolation and projection of boundary values onto an arbitrary surface without interference or influence of other turbulizer regions. This is performed entirely for numerical efficiency and convenience. A more formal application of the model, for example one performed by Siemens/CD-Adapco directly, would not need such a formulation, as the accessible parameters to the user are somewhat limited in this case.

5.3.2 Fluid Property Formulation

For the purposes of heat exchanger simulations, it is crucial to include variable fluid properties. The change in viscosity of a fluid during the heat transfer surface can have a large impact on pressure drop, and heat exchanger bulk flow directions. A formulation for the different fluid properties is provided for the sake of all simulations, and follows the form:

$$\rho\left(T\right) = a_0 + a_1 T^{0.5} + a_1 T^1 + a_2 T^{1.5} + a_3 T^2 + a_4 * T^{2.5} \tag{5.10}$$

$$k\left(T\right) = b_0 + b_1 T^{0.5} + b_1 T^1 + b_2 T^{1.5} + b_3 T^2 + b_4 * T^{2.5} \tag{5.11}$$

$$\mu\left(T\right) = \left(c_0 + c_1 T^{0.5} + c_1 T^1 + c_2 T^{1.5} + c_3 T^2 + c_4 * T^{2.5}\right)^3 \tag{5.12}$$

$$C_p\left(T\right) = d_+ d_1 T^{0.5} + d_1 T^1 + d_2 T^{1.5} + d_3 T^2 + d_4 * T^{2.5}. \tag{5.13}$$

Exact coefficient values are the intellectual property of Dana Canada, and due to their proprietary nature have not been included in this text.

5.3.3 Field Functions

Much of the customisation possible in STAR-CCM+ relies on user defined *field functions*. A field function is simply a mathematical formulation that can be used to calculate other variables within the CFD code such as boundary conditions, or can be simply used for the purpose of reporting and analysis [45]. The form of a field function typically falls into either the definition of a scalar or a vector quantity.

Reference values for the calculation of flow parameters are given as (units, if applicable are shown in square braces; $_{calc}$ subscripts denote calculated based on local fluid properties):

$$h_{channel} = 0.00241\ [m] \tag{5.14}$$

$$l = 0.001\ [m] \tag{5.15}$$

$$Pr = \frac{C_{p,calc}\ \mu_{calc}}{k_{calc}} \tag{5.16}$$

$$Re = \frac{\rho_{calc}\ |u_i|\ l}{\mu_{calc}}. \tag{5.17}$$

In this context, Re and Pr are calculated from local flow parameters to obtain the most appropriate, i.e. local, values and fluid properties, where Re_{CL} and Pr_{CL} are calculated from the bulk, or channel averaged fluid parameters.

Centreline values, Ψ_{CL}, are calculated by using an averaging across the width of the channel, using n evenly spaced points. This is represented by:

$$\Psi_{CL} = \frac{\sum_{i=1}^{n} \Psi_i}{n}, \tag{5.18}$$

where Ψ_i is calculated in $\frac{h_{channel}}{n}$ evenly spaced intervals, centred around the centreline plane of each channel. For the purposes of the present simulation, $n = 5$.

Field functions are further used for the calculation of critical flow parameters used for calculation within the turbulizer library, namely flow angle, ϕ (for input reference in the turbulizer library), Nu (output from library), f_D (output from library).

$$\phi = tan^{-1}\left(\frac{u_a}{u_b}\right) \tag{5.19}$$

$$Nu = f(\phi, Re, Pr) \tag{5.20}$$

$$f_D = f(\phi, Re, Pr), \tag{5.21}$$

where the subscripts a and b denote the primary flow direction, and the in-plane direction perpendicular to the primary flow direction. The selection of "primary flow direction" is somewhat arbitrary, but must correspond to the same coordinate system used in Ch. 4. Not used is the component normal to the centreline plane of the turbulizer, as it has no relevant meaning in this context. Calculations for Nu and f_D are calculated using a STAR-CCM+ user function, which is an external applied codebase used for integration of more complex calculations and model implementations. The details of this calculation are dependent on whether a 2-D interpolation scheme or ANN method is selected for use by the end user, as discussed previously in Ch. 4. Exact details of these models and their respective implementations are shown in Appendix E.

5.3.4 Turbulizer Pressure Drop Specification

Harnessing the framework already provided within the Darcy-Forchheimer implementation, or Porous Media model, within STAR-CCM+, the calculated friction factor in Eq. 5.21 is directly used in the calculation of the porous media coefficients. Thus, the implementation in STAR-CCM+ follows:

$$\frac{\partial p}{\partial x_i} = P_v u_i + P_i |u_i| u_i, \tag{5.22}$$

where in this context, P_v is the viscous resistance component of the porous media equation, equivalent to $\frac{\mu}{\kappa}$ in Eq. 5.2, and P_i, equivalent to $\frac{\rho}{\kappa_1}$ is the inertial resistance component of the porous media equation. As previously discussed, the inertial calculation is set to $P_i = 0$ for simplicity of implementation, and the friction factor is related directly to P_v, by the equation:

$$P_v = \frac{1}{2}\frac{|u_i|}{l} * f_D. \tag{5.23}$$

An axisymmetric formulation of the porous media equation is selected, as this eliminates orthogonality or tensor formulations of P_v and P_i, and instead relies on a scalar formulation that only takes into account the local velocity magnitude. This efficiently allows the use of the aforementioned flow direction formulation, with ϕ being the only relevant directionality reference needed for prescription of pressure drop. Thus, the final implementation reduces to the equation:

$$\frac{dp}{dx} = \left(\frac{1}{2}\frac{|u|}{l} * f_D\right)|u|. \tag{5.24}$$

5.3.5 Wall Heat Flux Specification

The initial step, once a reference Nu is obtained from the lookup library, is scaling for the proper Pr. Although the variation of the exponential coefficient is discussed and analysed previously in Ch. 3, the standard value of $n = 0.33$ is selected to follow Dana Canada current convention. This is easily updated within the context of the model, but for the purposes of the present work, it is left at this value.

$$Nu = Nu_{library} \cdot \left(\frac{Pr_{CL}}{Pr_{library}} \right)^{0.33}. \tag{5.25}$$

The $_{library}$ subscript denotes the reference values obtained directly from the library. The advantage of testing using the standard $n = 0.33$ value shows that the model is capable of suitable flow prediction without intervening *a priori* knowledge on the specifics of the fluid or geometry, which is the goal of the model. Additional development work for a specific application will allow for the use of a more suitable exponent, if required. Exposing this parameter to the end user also provides a means for straightforward manipulation and exploration of alternative values.

Conversion of Nu value into a STAR-CCM+ usable variable, h, is performed by calculation of a local heat flux, using Eq. 5.8 and substituting relevant calculated values:

$$h_{bulk} = \frac{Nu\, k_{CL}}{l}, \tag{5.26}$$

and further substituting into Eq. 5.9 to achieve:

$$\frac{Q_{wall}}{A_{wall}} = \frac{Nu\, k_{CL}}{l} \left(T_{CL} - T_{wall} \right). \tag{5.27}$$

This is then substituted into a framework provided within STAR-CCM+ referred to as $UserWallHeatFluxCoefficient\{A - D\}$. The formulation of this framework follows the general form of:

$$\begin{split} UserWallHeatFlux_{net} = UserWallHeatFluxA + UserWallHeatFluxB \cdot T \\ -UserWallHeatFluxC \cdot T + UserWallHeatFluxD \cdot T^4 \end{split} \tag{5.28}$$

Similarly, the internal calculation is performed using corresponding *internal* coefficients,

$$\begin{split} InternalWallHeatFlux_{net} = InternalWallHeatFluxA + \\ InternalWallHeatFluxB \cdot T - InternalWallHeatFluxC \cdot T \\ +InternalWallHeatFluxD \cdot T^4. \end{split} \tag{5.29}$$

Full control of the wall heat flux is assumed by equating the internal coefficients with the user defined coefficient. $UserWallHeatFluxCoefficientA$ is selected for implementation of the heat flux term due to lack of temperature dependence at the wall; temperature reference for the calculation of Eq. 5.27 is taken from the centreline calculated value. Thus,

$$UserWallHeatFluxA = \frac{Nu\ k_{CL}}{l}\left(T_{CL} - T_{wall}\right) - InternalWallHeatFluxA \quad (5.30)$$

$$UserWallHeatFluxB = -InternalWallHeatFluxB \quad (5.31)$$

$$UserWallHeatFluxC = -InternalWallHeatFluxC \quad (5.32)$$

$$UserWallHeatFluxD = -InternalWallHeatFluxD. \quad (5.33)$$

This fully defines the heat flux in or out of the turbulizer region from the wall. This is only applied at solid/fluid interfaces; not at the entrance to the turbulizer region. Specification of any thermal properties within the turbulizer is left as fluid default, rather than following the traditional Darcy-Forchheimer model implementation of modifying $k_{equivalent}$ to provided a weighted average of fluid and solid phase thermal conductivity values.

5.4 Global Heat Exchanger Model and Results

Testing and validation of the proposed model is performed using an out-of-production, non proprietary oil cooler model owned by Dana Canada. This model has been selected for several reasons:

1. Experimental pressure drop and heat transfer data exist
2. Analytical model results and analysis exist, however present significant challenges
3. CAD data exist and are readily adapted for use with the proposed model
4. The design is no longer used for any active production purposes.

5.4.1 Boundary Conditions

The test data are contained in Dana Technical Report H6948 [83]. In this test, a matrix of flows using various fluids, including Havoline HELAC 50/50 EG-Water, and ATF-Type A (ATF-94-A) is used to test and validate the performance of the transmission oil cooler in question. The oil cooler is constructed using 5 turbulizer pairs, i.e. 5 turbulizer channels for the hot side and 5 turbulizer channels for the cold side. The turbulizer trim profile is 70mm x 140mm, and the same turbulizer characterised in Ch. 4 is used on both the hot and cold side. Flow rates range from 10-40L/min (ATF-94-A) on the hot side, and 10-30L/min of Havoline HELAC on the cold side. Pressure drop and heat transfer are time averaged and measured for the experimental test, with results collected only once a steady state condition is reached. Temperatures and static pressures are recorded at the inlet and outlet, as shown in Fig. 5.1.

The corresponding matrix of flow rates and temperatures is shown in Table 5.1.

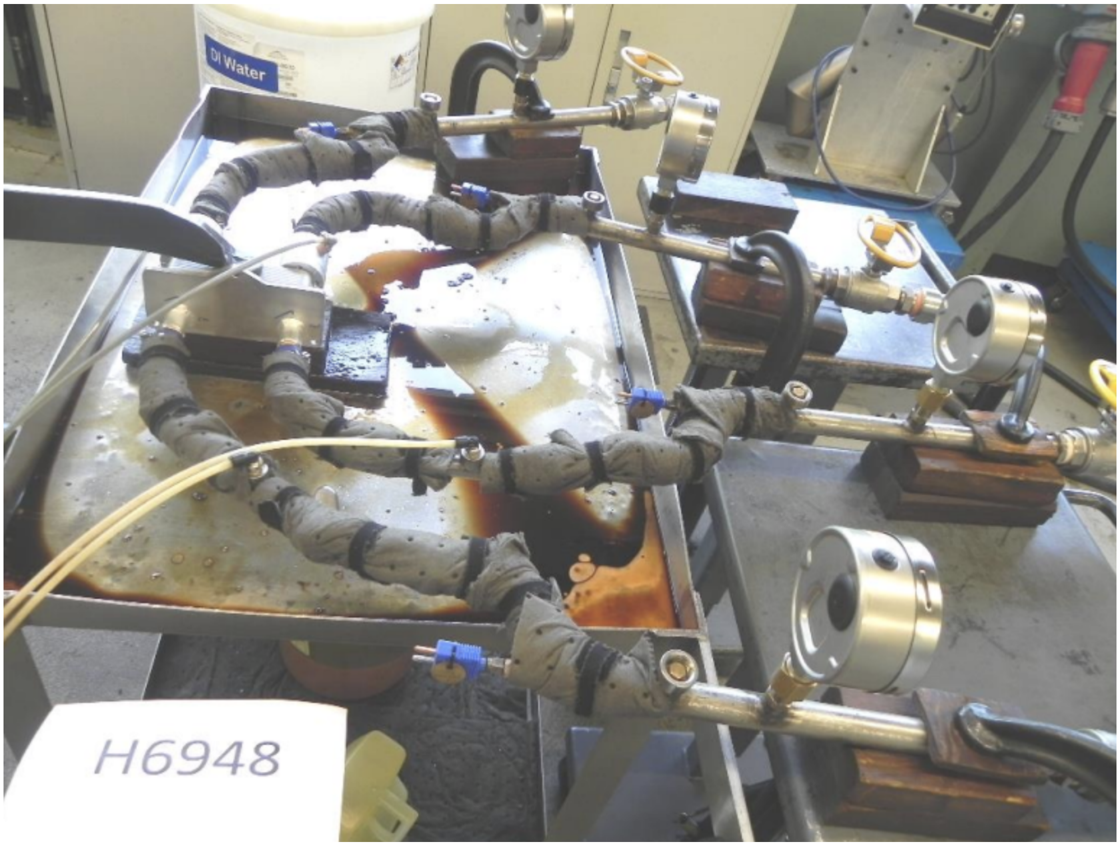


Figure 5.1: Experimental transmission oil cooler testing

The exact flow conditions (temperature and flow rate) are applied to the numerical and analytical models in order to most accurately replicate boundary conditions.

The trim profile of the cold-side turbulizer is shown in Fig. 5.2, while the trim profile

Table 5.1: Flow conditions, thermal testing

Test #	Hot Side, ATF-94-A		Cold Side, Havoline HELAC	
	Flow Rate [L/min]	Temperature [°C]	Flow Rate [L/min]	Temperature [°C]
1	10.12	119.7	10.03	78.24
2	19.96	118.69	10.04	78.9
3	29.94	120.06	9.96	80.96
4	40.05	118.97	10.06	81.47
5	10.02	119.41	20.03	78.52
6	19.96	118.44	20.05	79.31
7	29.92	120.25	20.01	81.43
8	40.06	118.6	20.03	81.72

Table 5.2: Flow conditions, pressure drop testing

Test #	Fluid Type	Flow Rate [L/min]	Temperature [°C]
1	ATF-94-A	10	119.7
2	ATF-94-A	20	118.69
3	ATF-94-A	30	120.06
4	ATF-94-A	40	118.97
5	HELAC	10.03	78.24
6	HELAC	20	78.9
7	HELAC	30	80.96

of the hot-side turbulizer is shown in Fig. 5.3. The trim profile is identical in each case, however, the turbulizer is flipped 180° to fit the alternate fitting location. The flow in the cold side is from the left of page to the right of page, while the hot side is flowing in the opposite direction, creating a counterflow arrangement. The turbulizer is arranged such that the high pressure drop direction runs from left to right on the page, thus allowing flow spreading across the height of the page, in the low pressure drop direction. Reference dimensions are taken from the furthest extent of the turbulizer profile. As can be observed, there are sections of this turbulizer profile which do not align well with the inlet and outlet fittings, and as such, is of particular challenge to model using either a standard porous media approach or the aforementioned analytical methods.

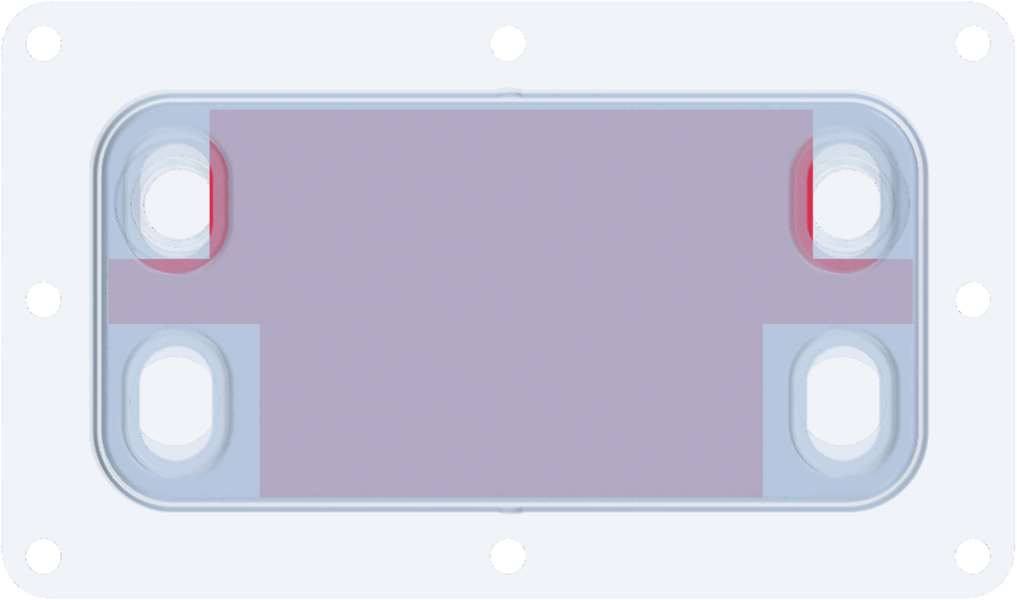


Figure 5.2: Hot-side Turbulizer trim profile, shown in red

5.4.2 Simulation Setup

The setup of the simulation follows the standard process of Dana Canada: a conjugate heat transfer simulation, consisting of 3 sets of material/fluid properties is used. ATF-A, Havoline HELAC coolant and aluminum are used to represent the hot, cold and solid

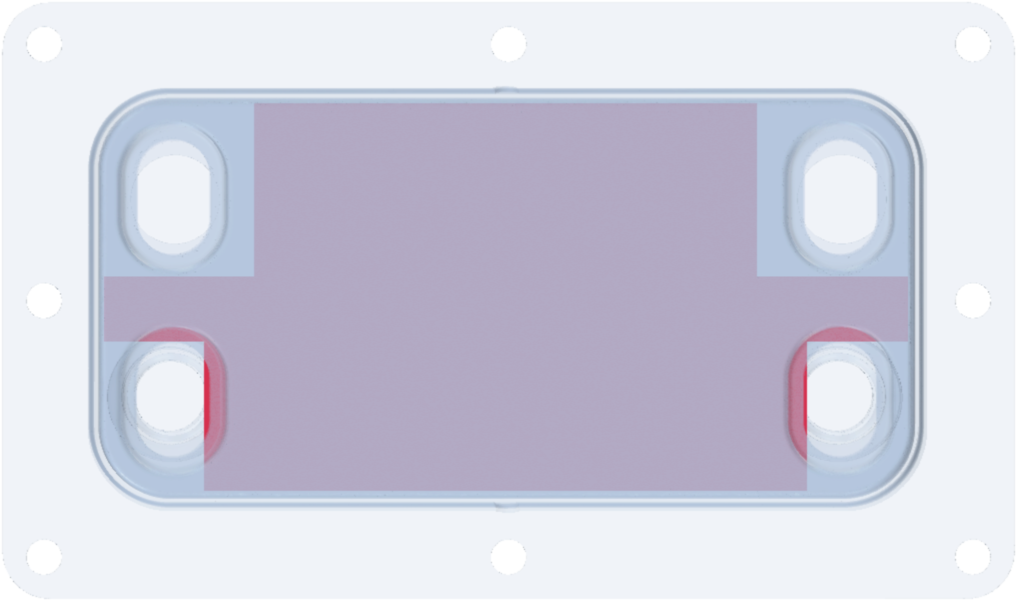


Figure 5.3: Turbulizer trim profile, shown in red

phases, respectively. The turbulizer sections are classified as separate calculation regions for application of both the present heat transfer model, as well as comparison with the standard porous media/Darcy-Forchheimer model. The $k - \omega$ SST turbulence model is used for the present set of simulations, however, it is often recommended that laminar flow models are used when low temperature, and therefore low Re, simulations are required. As demonstrated in Ch. 3, the laminar model is appropriate for use below $Re \approx 47$.

The simulation is prepared using the Java macros and scripts provided in Appendix E. No other modifcations are required of the simulation in order to enable the present model. The standard porous media model is enabled for the respective simulations, with all other simulation configuration otherwise retained.

5.4.3 Model Results and Comparison

The initial results, based on the flow rates proposed in Table 5.1, are presented below, in Table 5.3 and Table 5.4. For the sake of comparison, the porous media model results are **not** included in the thermal test conditions, as the model is tuned *a posteriori* based on output from the analytical model, or occasionally using test data when available, and thus the analytical model provides the only relevant comparison. Pressure drop comparison is performed on the basis of experimental results, the proposed model results, and porous media model results. Dana Canada standard testing protocol for pressure drop testing is to use an isothermal test condition to minimize any effect from heat transfer on the results. Furthermore, the analytical model does not include the contribution of any fluid losses outside of the turbulizer and is known to be inaccurate when predicting pressure drop of a full heat exchanger model.

Based on the results in Table 5.3 and Table 5.4, it is apparent that the performance of the proposed model exceeds the previous standards of testing: the porous media approach

Table 5.3: Thermal Test Results

Test #	Experimental Results		Proposed Model	Analytical Model
	T_{out}, hot	T_{out}, cold	% error, average	%error, average
1	96.7	89.7	-17%	18%
2	102.3	94.9	-1.7%	24%
3	107.5	99.6	5.0%	26.5%
4	109.1	101.1	8.6%	27.5%
5	94.6	84.7	-18.6%	16%
6	99.6	88.6	-7.5%	23.5%
7	105	92.8	-0.4%	25.5%
8	106.2	94.1	5.2%	27%

Table 5.4: Pressure Drop Test Results, Isothermal

Test #	Flow Rate	Experimental Results		Proposed Model	Porous Media
	[L/min]	Temperature, [°C]	dp $[kPa]$	% error	%error
1	10	119.7	15.6	0.6%	32.1%
2	20	118.7	55.9	16.1%	33.1%
3	30	120.1	119	15.4%	33.5%
4	40	119.0	207.8	17.2%	35.5%
5	10.03	78.2	18.5	18.1%	16.2%
6	20	78.9	71.7	18%	25.1%
7	30	81.0	158.6	21.3%	30.1%

for pressure drop, and the Dana Canada's proprietary analytical model.

The weakest performance of the presently proposed model is observed at low *Re* oil flow. This is not unexpected, as these conditions tend to be the most difficult to predict historically. The accuracy of the proposed model continues to increase as the oil flow *Re* increases. The coolant, or cold side, or the heat exchanger operates at a higher *Re* due to low fluid viscosity, and little impact to the results is observed when cold side *Re* is doubled in Test # 5-8. In all instances, the proposed model outperforms the analytical model. The limitations in the formulation of the analytical model preclude its accurate application for complex geometry and flow patterns expected with the present geometry, and thus, the thermal results are extremely encouraging and highlight a significant improvement in modelling capabilities. It is postulated that some of the low *Re* weakness in the model could be attributed to in-plane conduction along the aluminum walls. This is not directly addressed in the library generation in Ch. 4, and could be a contributor when the flow rate is low and thermal capacity of the fluid is small.

One departure of the heat transfer results from the general trend occurs at the lowest flow rates of oil, particularly in Test # 5, which is severely oil side limited. It is acknowledged that the performance of a heat exchanger in these conditions is entirely dominated by the performance of the high Pr/low *Re* fluid, which typically provides a lower level of overall performance. There are several potential contributors to this result:

1. Pr scaling, which has been acknowledged to be challenging for large departures from simulated test conditions, and highly dependent on the exponent n used in Eq. 5.25

2. lack of low *Re* data points during the ANN training process

3. excessive impact of experimental error due to sensor uncertainty, particularly due to the low percentage of the reading in comparison to sensor full-scale values.

Additionally, the historic performance of CFD results to overpredict both heat transfer and pressure drop in comparison to test data is not replicated for the proposed model. The present belief is that the discrepancy is due, either partially or in whole, due to the entrance and exit effects, which have been shown to degrade performance of the turbulizer when comparing computational domain models in Ch. 3. The same effects are not seen for pressure drop modelling, explaining the continued overprediction of results. This remains a point of recommendation for future exploration.

While pressure drop is typically more accurate to model both numerically and analytically in comparison to heat transfer, Table 5.4 shows that the proposed model still provides an improved ability to predict the pressure drop under nearly all flow conditions. Test 5, the only data point where the proposed model is outperformed by the standard porous media model is only a marginal difference and does not necessarily indicate a particular weakness of the model.

The general finding when comparing the proposed model to experimental, previous numerical and analytical results is that it outperforms all previous prediction methods.

The computational cost of the proposed model is identical to the previous porous media based model, for all intents and purposes. Although the exact computational cost is highly dependent on flow rates, geometry and turbulence model selected, each flow rate in Table 5.1 takes approximately 1.5h of clock time on a workspace with a minimum of 32GB memory and 16 Intel Xeon 2.4GHz processors. This equates to approximately 24 cpu-h of solution time per flow rate.

5.4.4 Detailed Results and Comparison

Although the proposed model is observed to outperform existing predictive methods in virtually all tested conditions, it is important to understand the potential impact on flow distribution. As the proposed model provides a more granular approach to specifying heat transfer and pressure drop parameters within the turbulizer, it is expected that a difference in the flow will be seen, particularly when the geometry or flow conditions do not align with the orthogonal coordinate system well.

The flow distribution of the channels is shown in Table 5.5. Channel 1 is the closest channel to the inlet and outlet fitting, while Channel 5 is the furthest from the fittings. It is typical to see the first channels starved somewhat of flow, as a fluid jet is created by the inlet fitting, and the momentum of the jet drives the majority of the fluid past the first channel. Likewise, the final channels are often starved of flow somewhat, as the pressure drop imposed by the manifold tends to restrict this same jet from penetrating to the outer edge of the heat exchanger. In this instance, the flow generally follows this same pattern on both the hot-side and cold side of the heat exchanger. The cold-side, although operating at a lower temperature, contains a lower viscosity fluid, which tends to suffer from poor flow distribution for the aforementioned reasons, while the hot side contains a higher viscosity oil which tends to offer more even flow distributions. The model selection does not have a significant impact on overall flow distribution for either of the fluid streams.

A general pattern that is observed in heat exchangers is that the higher the pressure drop of the heat transfer enhancement surface, the more evenly distributed the flow tends to be. This is a result of the manifold losses being diminished in proprotion to the entire heat exchanger pressure drop. This is occasionally used as a tool by manufacturers to help tune the performance of the heat exchanger, forcing flow more evenly through the channels, which in turn allows a higher effective "contact area" between the hot and cold fluid streams. Likewise, particularly poor flow distribution may result when a low pressure drop surface, such as a plain fin, is used for heat transfer. If this occurs, careful design must be paid to the manifold design, and when possible, larger inlet and outlet fittings should be used, which results in a lower energy jet forming into the manifold, and therefore more even flow distribution due to lower manifold losses.

Centreline fluid velocities are taken as a comparison between the proposed model and the porous media approach. These are compared directly in Fig. 5.4 to Fig. 5.9. Test data are taken from the flow conditions defined by Test # 8 in Table 5.1. For the sake of brevity, Cold Channel 1, 3 and 5 are shown, while Hot Channel 1, 3 and 5 are also compared. In each of the aforementioned figures, a slice is taken through the centreline of the respective fluid channel, with a colour scale and vector field representing the velocity of a given point. Only in-plane velocity components are considered in this presentation, as it is assumed that the out-of-plane component is approximate zero due to the small channel height relative to length and width. Furthermore, high velocity entrance and exit regions are visible in nearly all of the following figures as the flow exits the turbulizer region where

Table 5.5: Flow distribution, % of total

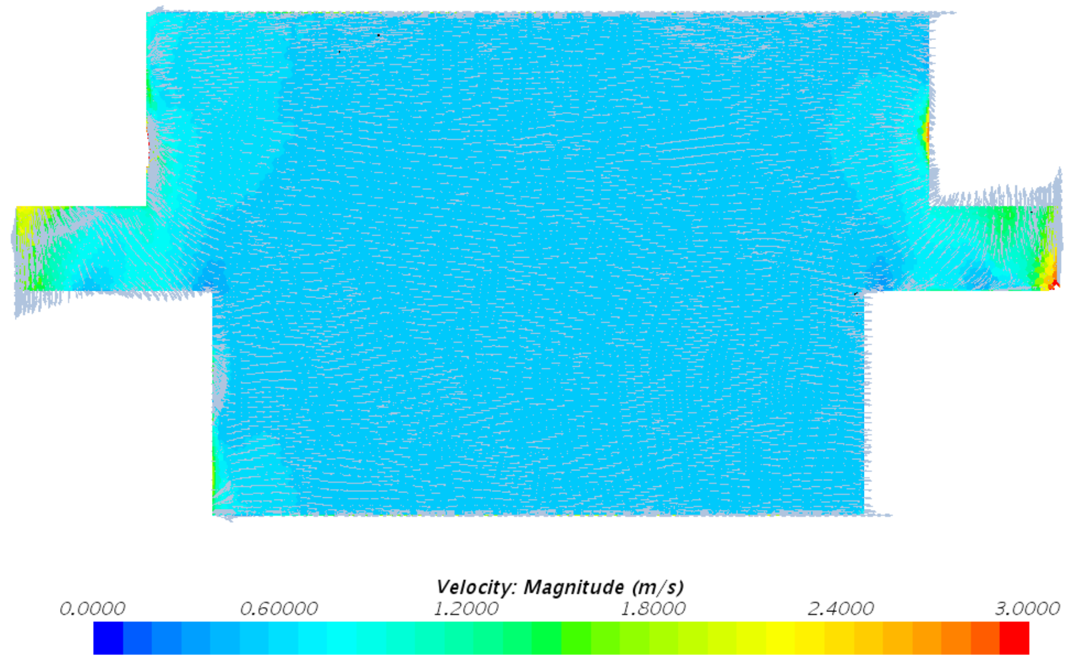
	Proposed Model		Porous Media Model	
Channel #	Cold-Side	Hot-Side	Cold-Side	Hot-Side
1	13.0%	17.5%	21.3%	21.1%
2	18.6%	20.5%	28.1%	21.7%
3	26.0%	26.9%	27.4%	26.9%
4	23.4%	19.3%	11.3%	22.0%
5	19.0%	15.8%	11.9%	8.3%

the model is active. Not shown is the remainder of the plane where the proposed model is not applied to flow calculations: that is, in the inlet and outlet manifold areas joining the plate pairs.

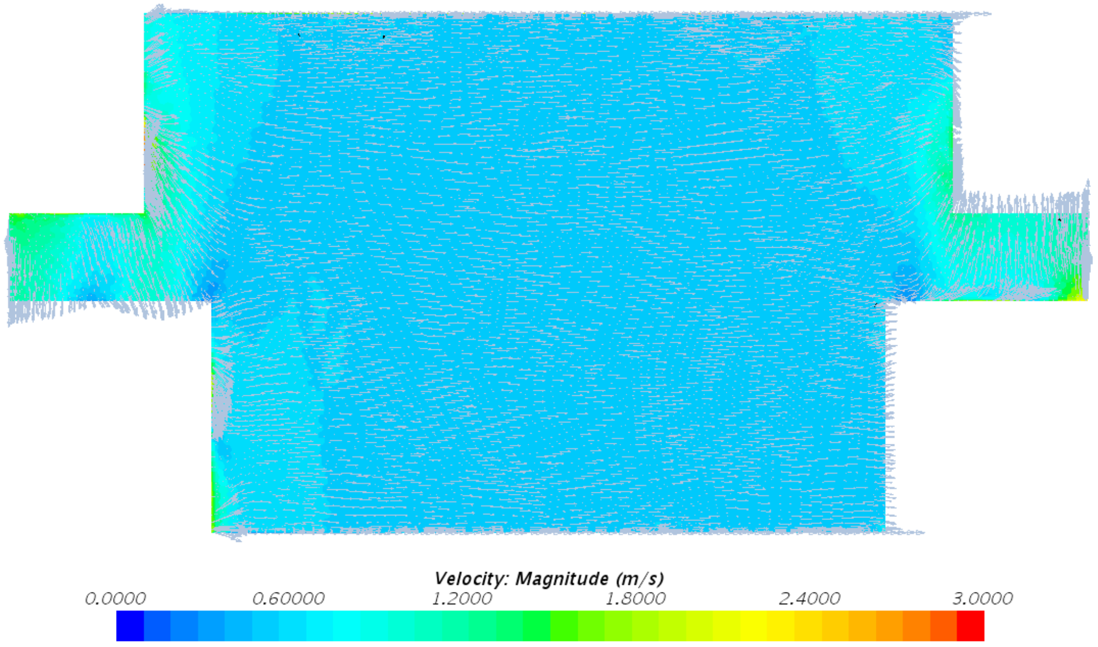
The general trend observed in all channels, shown in Fig. 5.4 to 5.9, is that the proposed model tends to have more orthogonally oriented flow, particularly near the inlet and outlet of each channel. This is exemplified by the velocity vector demarcations in orthogonal coordinates being of higher magnitude than similar locations in the Porous Media model results. This biasing of the flow to LPD direction is consistent with the assumed flow performance, i.e. the flow tends to follow the path of least resistance, and is biased towards spreading in a direction normal to the highest pressure drop, HPD, direction. This is effectively smeared due to the numerical blending achieved through the implementation of the orthogonal coordinate system of the Porous Media model, and is consistent with prior experience adapting this model for CHE usage. This result is fully expected and proves that the pressure drop results seen previously in Ch. 4 are borne out in the current results. As the pressure drop of the turbulizer is **not** a linear blend between orthogonal directions, namely the so-called "high pressure drop" (HPD) and "low pressure drop" (LPD) directions, but rather experiences a complex, non-linear transition, this smearing of the flow results is removed. Although the general flow profile on both models does form a radial expansion from the inlet of each channel to some extent, the Porous Media model accentuates this even spreading performance, where it is not expected to exist due to turbulizer geometry and orientation.

The flow distribution within a given channel is of particular interest, as it has a large impact on the available heat transfer. If the fluid were to follow a direct path between the inlet and outlet fittings, this particularly heat exchanger design would suffer from very low performance: Fig. 5.2 and 5.3 show that a horizontal line can be drawn for each geometry, with little overlap between the hot and cold fluids. The spreading performance, facilitated by orienting the LPD direction to encourage the flow to transit the entire width of the heat exchanger, is a crucial part of capturing the heat transfer performance of a particular geometry. Both models tend to predict a substantially even flow distribution within the centre of the channel flow length, where the proper overlap of fluids has the most impact on heat transfer.

Although heat transfer performance is of primary interest in this study, detailed heat transfer and temperature data are not available for comparison between the models. This

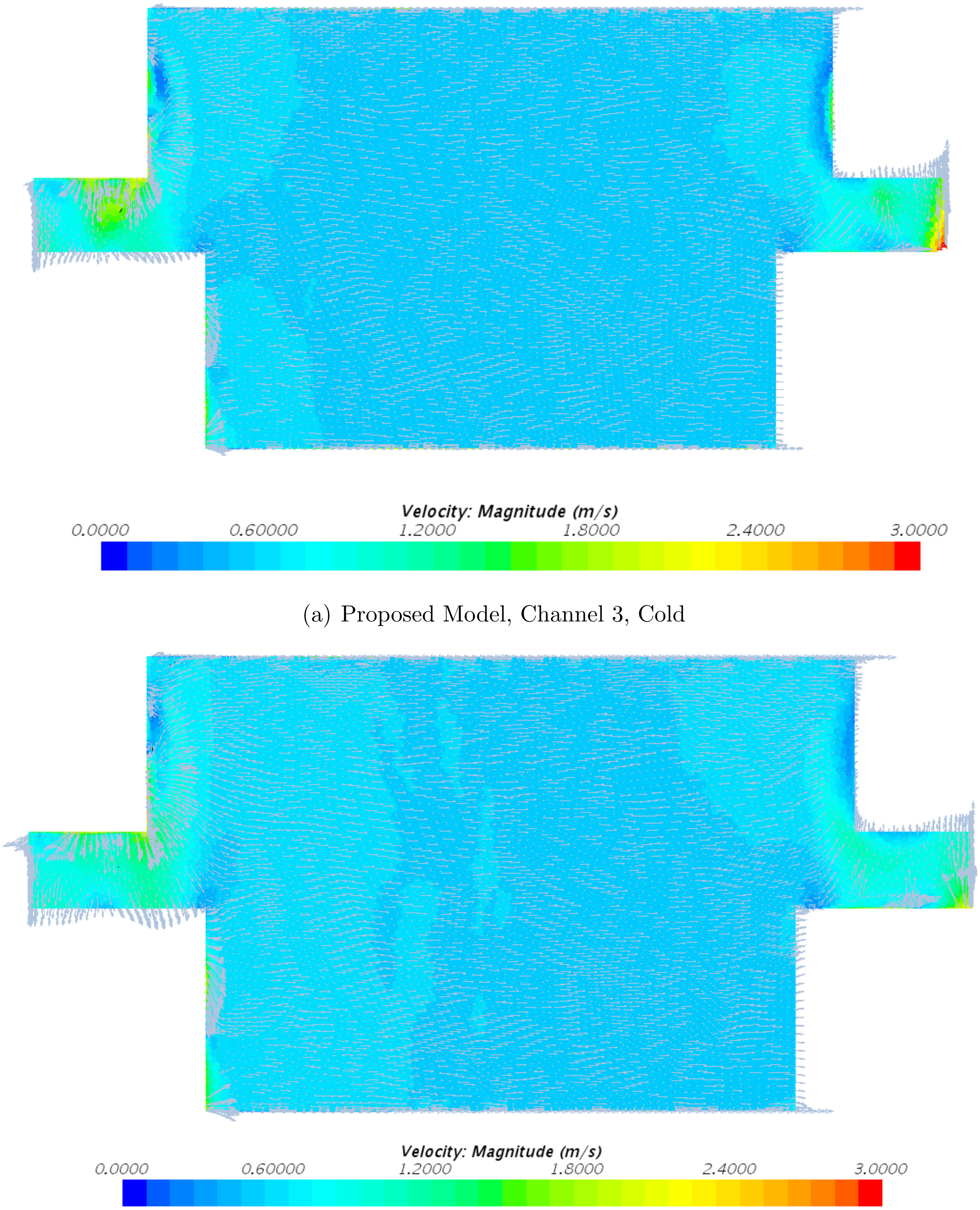


(a) Proposed Model, Channel 1, Cold



(b) Porous Media Model, Channel 1, Cold

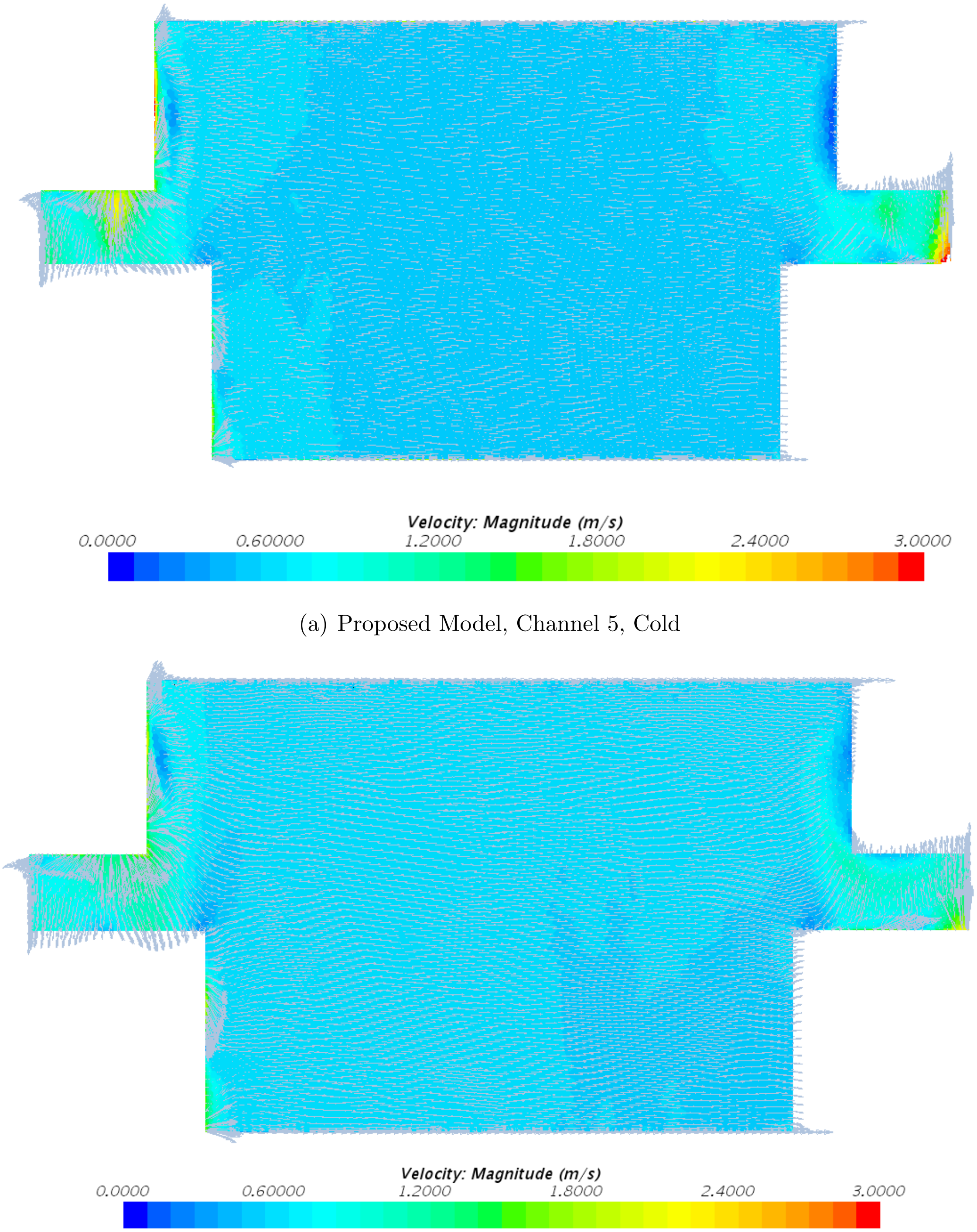
Figure 5.4: Comparison of the centreline velocity vectors for cold channel 1, with colour contours calculated by velocity magnitude



(a) Proposed Model, Channel 3, Cold

(b) Porous Media Model, Channel 3, Cold

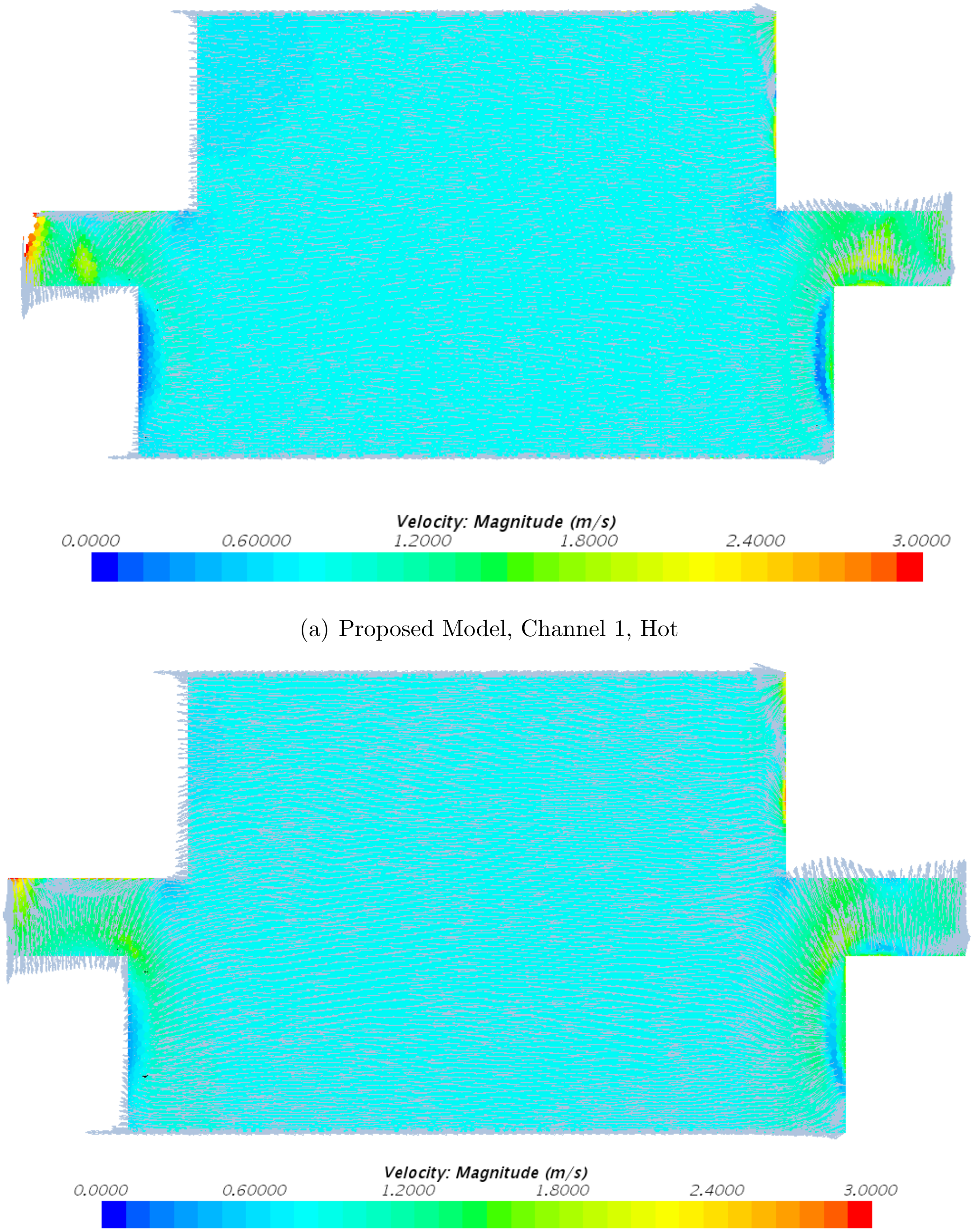
Figure 5.5: Comparison of the centreline velocity vectors for cold channel 3, with colour contours calculated by velocity magnitude



(a) Proposed Model, Channel 5, Cold

(b) Porous Media Model, Channel 5, Cold

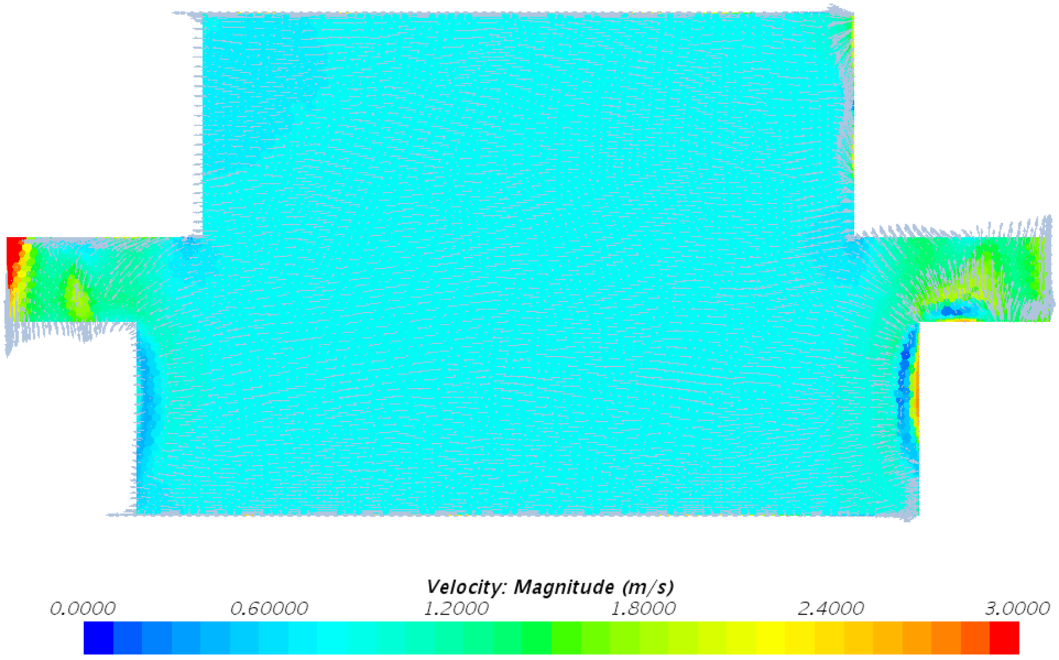
Figure 5.6: Comparison of the centreline velocity vectors for cold channel 5, with colour contours calculated by velocity magnitude



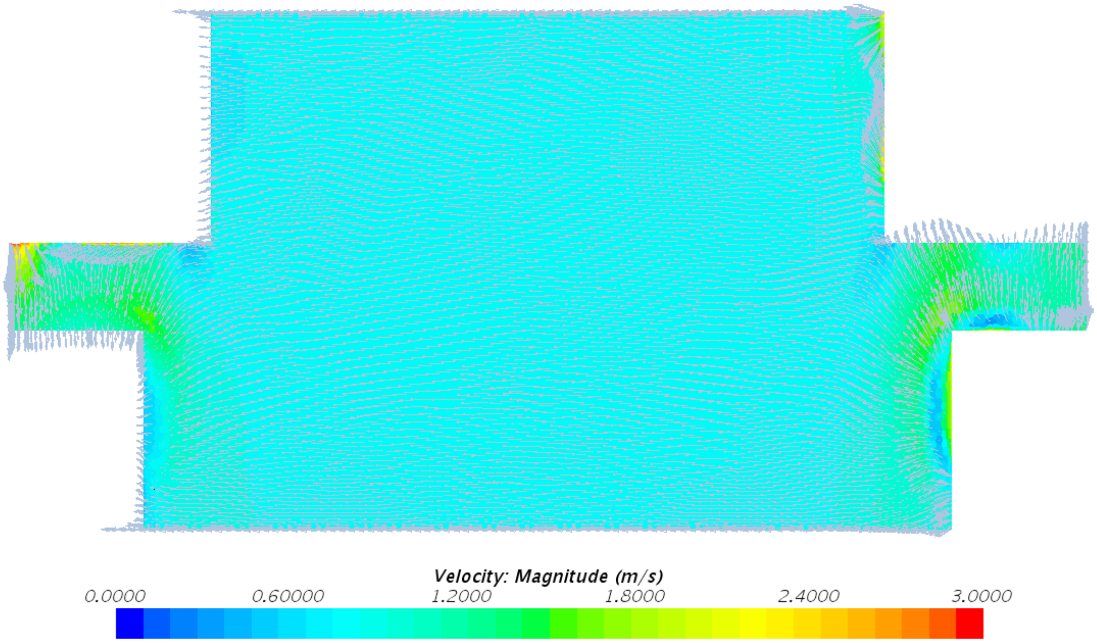
(a) Proposed Model, Channel 1, Hot

(b) Porous Media Model, Channel 1, Hot

Figure 5.7: Comparison of the centreline velocity vectors for hot channel 1, with colour contours calculated by velocity magnitude

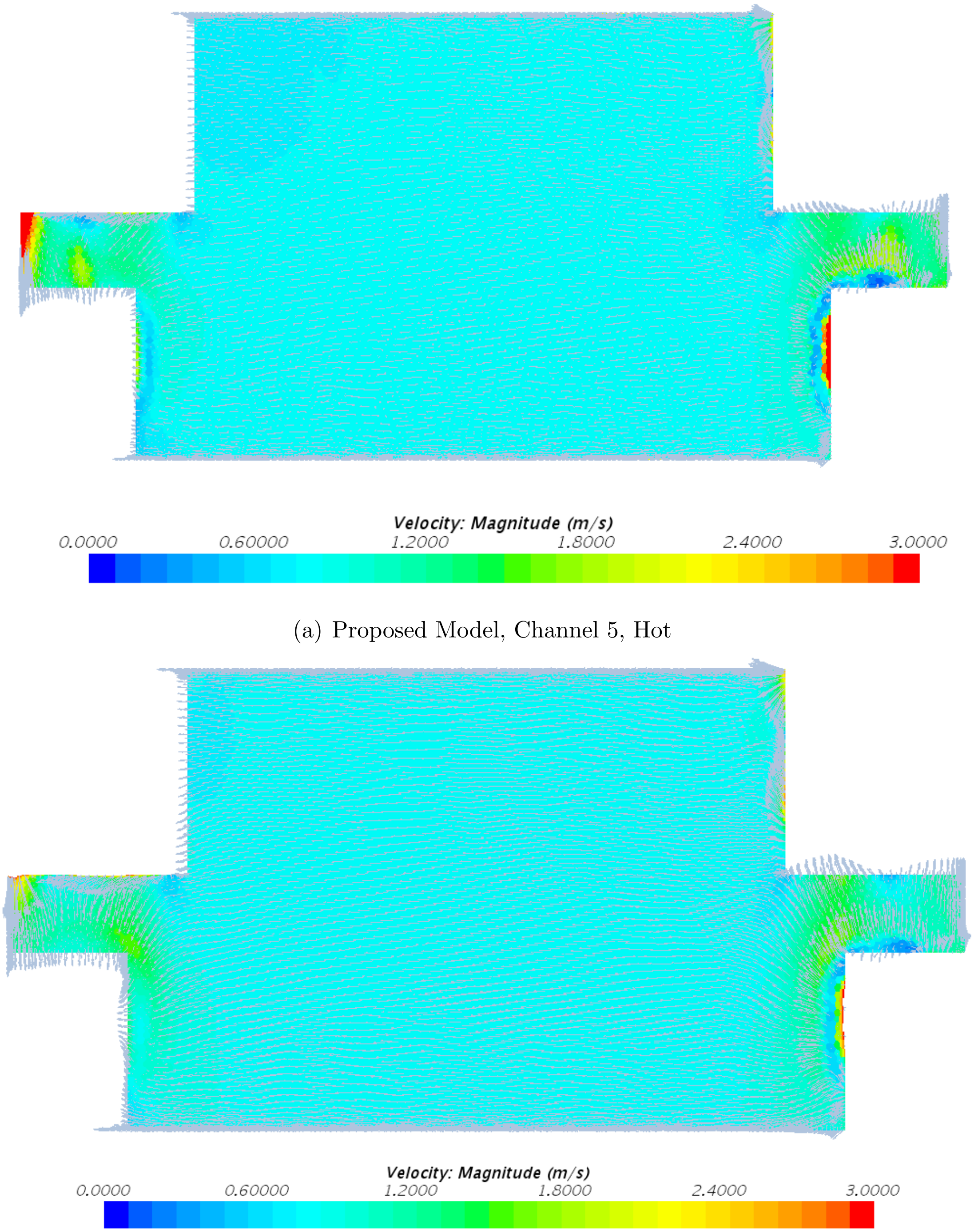


(a) Proposed Model, Channel 3, Hot



(b) Porous Media Model, Channel 3, Hot

Figure 5.8: Comparison of the centreline velocity vectors for hot channel 3, with colour contours calculated by velocity magnitude



(a) Proposed Model, Channel 5, Hot

(b) Porous Media Model, Channel 5, Hot

Figure 5.9: Comparison of the centreline velocity vectors for hot channel 5, with colour contours calculated by velocity magnitude

is primarily due to the necessity of tuning the Porous Media model to achieve the desired heat transfer *a posteriori*, and as such, is not representative of any kind of prediction, but rather a reaction on the part of the analyst performing the modelling. Likewise, no practical experimental method exists that is capable of providing detailed heat transfer and temperature data that are applicable for comparison.

5.5 Summary

In summary, the present chapter derives and demonstrates a thermal model applicable for use within multi-channel heat exchangers that is widely applicable to commercial to CFD software. Implementation of this model leverages commonly available pre-existing models, but does so in a way that can be easily generalised for direct implementation in the kernel of the CFD software in the future. The Darcy-Forchheimer equation is leveraged and combined with a newly derived formulation of a ROM to provide pressure drop performance, while the cross-channel-averaged thermal data are used to derive appropriate heat transfer coefficient data based on local flow conditions and ROM output, wall temperature and fluid temperature.

The performance of the model is observed to exceed any pre-existing predictive method available. The error observed in the test case is generally halved for both pressure drop and heat transfer. Analytical modelling, which is typically treated as the de-facto solution prior to availability of experimental data (and thus reliant on prototyping and experimentally testing a design), is not sufficiently able to handle the flow complexity induced by the present example, while existing Porous Media models are not able to work in a predictive capacity for turbulizers. Thus, within the proposed model, both pressure drop and thermal performance are improved upon in every commonly measured way, across a wide variety of flow rates. The example undertaken in this chapter provides an example of the robust performance and numerical stability of the proposed model.

Combination of the model contained within this chapter, with the learnings from Ch. 3 and 4 provide substantial evidence that the proposed methodology provides a significant improvement to existing modelling infrastructure and capabilities for use within CHE's, and conclusively addresses the primary objectives outlined at the outset of this project.

Chapter 6

Conclusions and Future Steps

At the outset of the present research, two primary objectives are outlined:

1. **Improved Understanding of flow Characteristics and Numerical Modelling of the Turbulizer**

2. **Development and Implementation of a Reduced Order Model for Turbulizers in a Global Heat Exchanger Model**

In Ch. 3, a detailed study of the numerical performance of the turbulizer is undertaken. The impact and effects of the turbulizer computational domain are investigated, and a revised model construction, composed of periodic boundary conditions is proposed to better isolate the performance of the turbulizer, while eliminating entrance and outlet effects. The impact of the turbulence model is also explored, and it is found that use of LES provides the most representative and accurate methodology, especially when compared with previous experimental data.

Following determination of the optimal numerical modelling techniques, the study of the turbulizer is continued to examine the impacts of manufacturing processes and variabilities on the overall performance. Through a detailed reconstruction of a sectioned turbulizer, a guideline for improved modelling is developed. Results are seen to be closest to the as-manufactured geometry when applying a corner radius of $0.17t$ and a material thickness of $0.83t$, combined with a channel height reduction of 0.1 mm.

Experimental validation of the the model is obtained by testing within a custom-built water channel, using both LDV and flow visualisation to determine the Re specific flow patterns, as well as the transition from steady to unsteady flow. Extremely good correlation is observed, with a transitional $Re = 43$ determined numerically, and a transitional $Re = 47$ determined experimentally.

In Ch. 4, the performance of the turbulizer, which is thoroughly tested and validated experimentally in Ch. 3, is explored under a variety of flow angles and Re. The periodic model developed in Ch. 3 is further used for this exploration. Non-linear performance between HPD and LPD directions is observed for both f_D and Nu, which has not been previously reported in the available literature. This expansion of the knowledge base further reinforces the need for more complex modelling techniques than are currently available.

The development of a training data set using a full range of ϕ and Re is used for development and comparison of both a bilinear interpolation based ROM and an ANN based ROM. In testing, it is found that f_D predictions are extremely accurate when a 5-5-5 ANN is used, but the complexity of Nu response is more difficult to characterise. Subsequently, a simpler network, 4-4, is recommended for Nu and is seen to provide the best results. The use of an ANN significantly outperforms bilinear interpolation for the case of f_D, but only provides a slight improvement over bilinear interpolation for Nu. The training data set reduction for calculation of f_D is approximately 50% from the baseline used for study, while a training data set reduction of only 25% is available for Nu. All training data sets are randomly generated from the full training data, as it is expected that novel heat transfer enhancement surfaces will have unexpected performance in some regimes that cannot always be predicted *a priori*. Calculation of the full training data set is thus approximately 18 CPU-y per heat transfer enhancement surface.

In Ch. 5 the results from the previous chapters are incorporated into a ROM suitable for reproducing turbulizer performance at a large scale. Details of the implementation are presented, which leverage control of wall heat transfer coefficients and variables presently exposed in the Darcy-Forchheimer Porous Media model, allowing for simple implementation. Results are seen to significantly exceed those observed with current state-of-the-art modelling techniques. Error relative to experimental testing is approximately halved for both heat transfer and pressure drop, leading to a significant improvement in the overall predictive capabilities of the model. Computational time is expected to be on par with existing modelling techniques.

Through the results and approaches presented herein, it is clear that both a significant improvement in the fundamental understanding of turbulizer flow performance is achieved, while a novel, simple and accurate reduced order model is presented which is capable of halving the current numerical error in comparison to existing techniques.

Future recommendations for the present research include:

1. Expansion of the ANN testing to explore areas outside of the test data, particularly in search of instabilities or non-physical representations of the flow

2. Investigation of further training data reduction through use of methods, such as orthogonal design of experiments (Taguchi Method) in order to improve computational efficiency

3. Detailed investigation of entrance and exit effects in the turb domain, which are acknowledged to provide uncharacterised behaviour in the present model

4. Investigation of the impact of Pr scaling on the final results

5. Characterise and understand the relevance of relative thermal resistance between different fluid combinations

6. Further investigation into low Pr effects in the tested geometry

As a final comment, it is noted that during the course of the present research, the automotive industry shifted significantly from a focus on traditional "cooling" type heat exchangers, i.e. oil coolers, radiators, etc. , and transitioned towards the concept of thermal management within a vehicle., especially in the case of electric vehicles. This has put an increased emphasis on the capabilities of predicting accurate heat transfer coefficients within a heat exchanger. Fixed heat rejection from power electronics or battery cooling has obviated the traditional PM approach, instead forcing the need for an improved approximation of thermal resistance at an interface in order to accurately predict surface temperatures and surface temperature distribution.

It is the recommendation of this research that Dana Canada pursues collaboration with Siemens/CD-Adapco to implement the present model into future releases of the commercial CFD code STAR-CCM+.

Bibliography

[1] W.M. Kays and A.L. London. *Compact Heat Exchangers, Third Ed.* McGraw-Hill Book Company, Inc., 1984.

[2] J.E. Hesselgreaves. *Compact Heat Exchangers: Selection Design and Operation.* Elsevier Science Ltd., 2001.

[3] R.K. Shah and D.P. Sekuliç. *Fundamentals of Heat Exchanger Design.* John Wiley & Sons, Inc., 2003.

[4] M. Prithiviraj and M.J. Andrews. Three-Dimensional Numerical Simulation of Shell-and-Tube Heat Exchangers. Part II: Heat Transfer. *Num. Heat Transfer, Part A*, 33:817–828, 1998.

[5] R.K. Shah, A.D. Kraus, and D. Metzger. *Compact Heat Exchangers.* Hemisphere Publishing Corporation, 1990.

[6] E.E. Wilson. A Basis for Rational Design of Heat Transfer Apparatus. *ASME Transactions*, 37:546–668, 1915.

[7] N.H. Afgan and E.U. Schluender. *Heat Exchangers: Design and Theory Sourcebook.* Scripta Book Company, 1974.

[8] G.F. Hewitt. *Hemisphere Handbook of Heat Exchanger Design.* Hemisphere Publishing Corporation, 1990.

[9] K. Suzuki, E. Hirai, T. Miyake, and T. Sato. Numerical and Experimental Studies on a Two-Dimensional Model of an Offset-Strip-Fin Type Compact Heat Exchanger used at Low Reynolds Number. *Int. J. Heat Mass Transfer*, 28(4):823–836, 1985.

[10] M. Ciofalo, J. Stasiek, and M.W. Collins. Investigation of Flow and Heat Transfer in Corrugated Passages - II. Numerical Simulations. *Int. J. Heat Mass Transfer*, 39(1):165–192, 1996.

[11] O. Labbé and P. Sagaut. Large-Eddy Simulation of Heat Transfer over a Backward Facing Step. *Numerical Heat Transfer, Part A*, 42:73–90, 2002.

[12] W. Wang and R.H. Pletcher. On the Large Eddy Simulation of a Turbulent Channel Flow with Significant Heat Transfer. *Phys. Fluids*, 8(12):3354–3366, 1996.

[13] S.V. Patankar and D.B. Spalding. *Heat Exchangers: Design and Theory Sourcebook; A Calculation Procedure for the Transient and Steady-State behaviour of Shell-and-Tube Heat Exchanger*, pages 156–176. Scripta Book Company, 1974.

[14] L.S. Ismail, C. Ranganayakulu, and R.K. Shah. Numerical Study of Flow Patterns of Compact Plate-Fin Heat Exchangers and Generation of Design Data for Offset and Wavy Fins. *Int. J. Heat Mass Transfer*, 52:3972–3983, 2009.

[15] S. Tavoularis. *Measurement in Fluid Mechanics.* Cambridge University Press, 2005.

[16] W. Nusselt. Die Oberflaechenkondensation des Wasserdampfes. *Z. Ver. Dt. Ing.*, 60:541–546, 1916.

[17] D.E. Briggs and E.H. Young. Modified Wilson Plot Techniques for Obtaining Heat Transfer Correlations for Shell-and-Tube Heat Exchangers. *Chem. Eng. Prog. Symp. Series*, 65:35–45, 1969.

[18] S.W. Churchill and R. Usagi. A General Expression for the Correlation of Rates of Transfer and Other Phenomena. *Amer. Inst. Chem. Eng.*, 18(6):1121–1128, 1972.

[19] W.M. Kays and A.L. London. *Compact Heat Exchangers, A Summary of Basic Heat Transfer and Flow Friction Design Data.* McGraw-Hill Book Company, Inc., 1958.

[20] A.L. London. Compact Heat Exchangers: Part 2 - Surface Geometry. *Mech. Eng.*, 86:31–35, 1964.

[21] R.J. Adrian. Twenty Years of Particle Image Velocimetry. *Exp. Fluids*, 39:159–169, 2005.

[22] R. Lindken and W. Merzkirch. A novel PIV Technique for Measurements in Multiphase Flows and its application to Two-Phase Bubbly Flows. *Exp. Fluids*, 33:814–825, 2002.

[23] G.E. Elsinga, F. Scarano, B. Wieneke, and B.W. van Oudheusden. Tomographic Particle Image Velocimetry. *Exp. Fluids*, 41:933–947, 2006.

[24] R. Budwig. Refractive Index Matching Methods for Liquid Flow Investigations. *Exp. Fluids*, 17:350–355, 1994.

[25] O. Uzol, Y.-C. Chow, J. Katz, and C. Meneveau. Unobstructed Particle Image velocimetry Measurements with an Axial Turbo-Pump using Liquid and Blades with Matched Refractive Indices. *Exp. Fluids*, 33:909–919, 2002.

[26] S. Talapatra and J. Katz. Three-Dimensional Velocity Measurements in a Roughness Sublayer using Microscopic Digital In-Line Holography and Optical Index Matching. *Meas. Sci. Technol.*, 24:1–11, 2013.

[27] R.J. Adrian and J. Westerweel. *Particle Image Velocimetry*. Cambridge University Press, 2010.

[28] S.B. Pope. *Turbulent Flows*. Cambridge University Press, 2008.

[29] *Processing of Random Signals*, Proc. Dynamic Flow Conf., 1978.

[30] L.H. Benedict and R.D. Gould. Towards Better Uncertainty Estimates for Turbulence Statistics. *Exp. Fluids*, 22:129–136, 1996.

[31] P. Lavoie, G. Avallone, F. De Gregorio, G.P. Romano, and R.A. Antonia. Spatial Resolution of PIV for the Measurement of Turbulence. *Exp. Fluids*, 43:39–51, 2007.

[32] D.S. Weaver, S. Ziada, M.K. Au-Yang, S.S. Chen, M.P. Païdoussis, and M.J. Pettigrew. Flow-Induced Vibrations in Power and Process Plant Components – Progress and Prospects. *J. Pressure Vessel Tech.*, 122:339–348, 2000.

[33] W.R. Dean. The Stream-Line Motion of Fluid in a Curved Pipe. *Phil. Magazine and J. Sci.*, 7:673–695, 1928.

[34] S.A. Berger, L. Talbot, and L.-S. Yao. Flow in Curved Pipes. *Ann. Review Fluid Mech.*, 15:461–512, 1983.

[35] G.I. Taylor. Stability of a Viscous Liquid Contained between Two Rotating Cylinders. *Phil. Trans. R. Soc. A*, 223:289–343, 1923.

[36] NATO. *Compact Heat Exchangers*, number AGARD-LS-57-72 in AGARD Lecture Series No. 57 on Heat Exchangers, Paris, 1972.

[37] H.M. Joshi and R.L. Webb. Heat Transfer and Friction in the Offset Strip Fin Heat Exchanger. *Int. J. Heat Mass Transfer*, 30(1):69–84, 1987.

[38] R.M. Manglik and A.E. Bergles. Heat Transfer and Pressure Drop Correlations for the Rectangular Offset Strip Fin Compact Heat Exchanger. *Exp. Therm. Fluid Science*, 10:171–180, 1995.

[39] Y.S. Muzychka. *Analytical and Experimental Study of Fluid Friction and Heat Transfer in Low Reynolds Number Heat Exchangers.* PhD thesis, University of Waterloo, Waterloo, Canada, 1999.

[40] E.F. Farrell. Turbulizer, June 27 1961. US Patent 2,990,163.

[41] W. Malalasekra H.K. Versteeg. *An Introduction to Computational Fluid Dynamics, The Finite Volume Method.* Pearson Education Limited, 2007.

[42] B.E. Launder and D.B. Spalding. The Numerical Computation of Turbulent Flows. *Comput. Methods Appl. Mech. Eng.*, 3:269–289, 1974.

[43] D.C. Wilcox. *Turbulence Modeling for CFD.* Griffin Printing, 1994.

[44] M. Germano, U. Piomelli, P. Moin, and W.H. Cabot. A Dynamic Subgrid-Scale Eddy Viscosity Model. *Phys. Fluids A*, 3(7):1760–1765, 1991.

[45] CD-adapco, Inc. Star-CCM+ User Guide, `http://www.cd-adapco.com/products/star-ccm/documentation`, 2014.

[46] M. Kaviany. *Principles of Heat Transfer in Porous Media.* Springer, 1995.

[47] S.L. Lee and J.H. Yang. Modelling of Darcy-Forchheimer Drag for Fluid Flow Across a Bank of Cicular Cylinders. *Int. J. Heat Mass Transf.*, 40(13):3149–3155, 1997.

[48] A.Y. Klimenko and R.W. Bilger. Conditional moment closure for turbulent combustion. *Prog. Energy Combust. Sci.*, 25(6):595–687, 1999.

[49] H. Steiner and W.K. Bushe. LES of non-premixed turbulent reacting flows with Conditional Source-term Estimation. *CTR-Annual Research Briefs*, 1998.

[50] E. Lee, F.-S. Lien, and H. Ji. A Building-Resolved Wind Field Library for Vancouver Facilitating CBRN Emergency Response for the 2010 Winter Olympic Games. Technical Report DRDC Suffield TM-2010-088, Defence Research and Development Canada, 2010.

[51] S. Manabe and J. Smagorinsky. Simulated climatology of a general circulation model with a hydrologic cycle ii. analysis of the tropical atmosphere. *Monthly Weather Review*, 95(4):155–169, 1967.

[52] G. Berkooz, P. Holmes, and J.L. Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annu. Rev. Fluid Mech.*, 25:539–575, 1993.

[53] D. Alonso, A. Velazquez, and J.M. Vega. Robust reduced order modeling of heat transfer in a back step flow. *Int. J. Heat and Mass Transfer*, 52:1149–1157, 2009.

[54] K. Willcox. Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition. *Computers & fluids*, 35(2):208–226, 2006.

[55] P.J. Schmid. Dynamic mode decomposition of numerical and experimental data. *J. Fluid Mech.*, 656:5–28, 2010.

[56] *An eigensystem realization algorithm for modal parameter identification and model reduction*, J. Guidance, Cont. and Dyn., 1985.

[57] W. Yao and R.K. Jaiman. Feedback control of unstable flow and vortex-induced vibration using the eigensystem realization algorithm. *J. Fluid Mechanics*, 827:394–414, 2017.

[58] D.C. Rebolho, E.M. Belo, and F.D. Marques. Aeroelastic parameter identification in wind tunnel testing via the extended eigensystem realization algorithm. *J. Vib. and Control*, 20(8):1607–1621, 2014.

[59] Z. Ma, S.Ahuja, and C.W. Rowley. Reduced-order models for control of fluids using the eigensystem realization algorithm. *Theor. and Comp. Fluid Dyn.*, 25(1):233–247, 2011.

[60] P. Picton. *Neural Networks.* Palgrave, 2000.

[61] I. Kaastra and M. Boyd. Designing a neural network for forecasting financial and economic time series. *Neurocomputing*, 10, 1996.

[62] J.A.K. Suykens, J.P.L. Vandewalle, and B.L.R. De Moor. *Artificial Neural Networks for Modelling and Control of Non-Linear Systems.* Kluwer Academic Publishers, 1996.

[63] S. Nissen. Implementation of a fast artificial neural network library (fann). Technical report, Department of Computer Science, University of Copenhagen, 2003.

[64] P. Roberts. *Industrial Brazing Process, Second Ed.* CRC Press, 2013.

[65] Joint Committee for Guides in Metrology. 100:2008 evaluation of measurement data - guide to the expression of uncertainty in measurement, 2008.

[66] F.R. Hama, J.D. Long, and J.C. Hegarty. On Transition from Laminar to Turbulent Flow. *J. Applied Phys.*, 28(4):388–394, 1957.

[67] V.C. Patel, W. Rodi, and G. Scheuerer. Turbulence Models for Near Wall and Low Reynolds Number Flows: A Review. *AIAA J.*, 23(9):1308–1319, 1985.

[68] F. Ducros, P. Comte, and M. Lesieur. Large-Eddy Simulation of Transition to Turbulence in a Boundary Layer Developing Spatially over a Flat Plate. *J. Fluid Mech.*, 326:1–36, 1996.

[69] J. Stasiek, M.W. Collins, M. Ciofalo, and P.E. Chew. Investigation of Flow and Heat Transfer in Corrugated Passages - I. Experimental Results. *Int. J. Heat Mass Transfer*, 39(1):149–164, 1996.

[70] C.-C. Wang, J. Lo, Y.-T. Lin, and C.-S. Wei. Flow visualization of annular and delta winglet vortex generators in fin-and-tube heat exchanger applications. *Int. J. Heat Mass Transf.*, 45:3803–3815, 2002.

[71] S. Kakac, R.K. Shah, and W. Aung. *Handbook of Single Phase Convective Heat Transfer.* John Wiley & Sons, Inc., 1987.

[72] J.G. Burgers. 5 cpi tubulizer computational fluid dynamics demonstration study. Technical Report RP00289, Advanced Engineering, Long Manufacturing Ltd., February 1990.

[73] T.L. Bergman, A.S. Lavine, F.P. Incropera, and D.P. Dewitt. *Fundamentals of Heat and Mass Transfer*. John Wiley & Sons, Inc., 2011.

[74] A.N. Kolmogorov. The local structure of turbulence in incompressible viscous fluid for very large Reynolds numbers. *Proc. USSR Academy Sci*, 30:299–303, 1941.

[75] A.N. Kolmogorov. Dissipation of Energy in the Locally Isotropic Turbulence. *Proc. USSR Academy Sci*, 32:16–18, 1941.

[76] UES, Inc. Robomet.3D, `http://www.ues.com/sites/default/files/R3D2%20Jan2013.pdf`, 2013.

[77] P.W. Dittus and L.M.K. Boelter. Heat Transfer in Automobile Radiators of the Tubular Type. *Univ. Calif. Pub. Eng.*, 2(13):443–461, 1930.

[78] A.Buckrell. ME770 Water Tunnel Design: Final Report, 2013.

[79] T. Morel. Comprehensive design of axisymmetric wind tunnel contractions. *J. Fluids Engineering*, 97:225–234, 1975.

[80] T. von Kármán. Ueber den mechanismus des widerstandes, den ein bewegter koerper in einer fluessigkeit erfaehrt. *Nachr. Ges. Wiss. Goettingen math.-phys*, pages 509–517, 1911.

[81] G.E. Moore. Cramming More Components onto Integrated Circuits. *Electronics*, 38(8), 1965.

[82] B. Delaunay. Sur la Sphère Vide. *Bulletin de l'Académie des Sciences de l'URSS, Classe des Sciences Mathématiques et Naturelles*, 6:793–800, 1934.

[83] A. Mowat. Heat transfer/pressure drop. Dana Canada Technical Report H6948, Dana Canada Corporation, 2018.

APPENDICES

Appendix A

Uncertainty Analysis

The uncertainty analysis presented below is largely based on that provided by Muzychka [39]. Certain modifications are made to the analysis in regards to specific operating conditions, however, the results are largely the same. The analysis is based on the root sum square method . For a measurement R dependent on n independent variables,

$$R = R(x_q, x_2, ..., x_n), \tag{A.1}$$

the root sum square error is expressed by:

$$w_R = \left[\left(\frac{\partial R}{\partial x_1} \right)^2 + \left(\frac{\partial R}{\partial x_2} \right)^2 + \left(\frac{\partial R}{\partial x_1} \right)^2 + ... + \left(\frac{\partial R}{\partial x_n} \right)^2 \right]^{\frac{1}{2}} [15], \tag{A.2}$$

where w_i represents the uncertainties of the independent variables x_i.

A.1 Uncertainty Due to Measurement Error

The uncertainty of experimental measurements, namely temperature, pressure and flow rate are expressed in Table A.1.

Table A.1: Uncertainty in Measurements [39]

Variable	Uncertainty
Temperature [°C]	±0.05°C
Pressure [Pa]	±1%
Volumetric Flow Rate ($\dot{V}$) [$\frac{kg}{s}$]	±1%

A.2 Uncertainty Due to Fluid Properties

The uncertainty of fluid properties, which are typically obtained by a 3^{rd} party lab, are reported to be within 0.5% or less of the reported curve fit. This is shown in Table A.2.

Table A.2: Uncertainty in Fluid Properties [39]

Variable	Uncertainty
ρ [$\frac{kg}{m^3}$]	±0.5%
μ [$Pa - s$]	±0.5%
C_p [$\frac{J}{kg-K}$]	±0.5%
k [$\frac{W}{m-K}$]	±0.5%

A.3 Uncertainty In Heat Transfer Results

The uncertainty in heat transfer measurements, Q and UA are determined from the root sum square of the constituent independent variables, i.e.:

$$w_R = \left[\left(\frac{\delta \dot{m}}{\dot{m}} \right)^2 + \left(\frac{\delta C_p}{C_p} \right)^2 + \left(\frac{\delta T_{in}}{\Delta T} \right)^2 + \left(\frac{\delta T_{out}}{\Delta T} \right)^2 \right]^{\frac{1}{2}} [15], \tag{A.3}$$

and

$$\frac{\delta \dot{m}}{\dot{m}} = \left[\left(\frac{\delta \rho}{\rho} \right)^2 + \left(\frac{\delta \dot{V}}{\dot{V}} \right)^2 \right]^{\frac{1}{2}}. \tag{A.4}$$

The calculated uncertainties are shown in Table A.3.

Table A.3: Total heat transfer uncertainty

Variable	Uncertainty
Q_{ATF}	±1.59%
$Q_{coolant}$	±4.34%
$\bar{Q}$	±4.62%

The uncertainty from the overall heat transfer coefficient is determined by the expression:

$$\frac{\delta UA}{UA} = \left[\left(\frac{\delta Q}{Q} \right)^2 + \left(\frac{\delta \Delta T_{LMTD}}{\Delta T_{LMTD}} \right)^2 \right]^{\frac{1}{2}}. \tag{A.5}$$

The resultant uncertainties are shown in Table A.4.

Table A.4: Overall heat transfer coefficient uncertainty

Variable	Uncertainty
UA_{ATF}	±2.86%
$UA_{coolant}$	±4.95%
$\overline{UA}$	±5.19%

A.4 Uncertainty in Nu and Re

The uncertainty in measured Re is calculated from:

$$\frac{\delta Re}{Re} = \left[\left(\frac{\delta \dot{m}}{\dot{m}} \right)^2 + \left(\frac{\delta D_h}{D_h} \right)^2 + \left(\frac{\delta A}{A} \right)^2 + \left(\frac{\delta \mu}{\mu} \right)^2 \right]^{\frac{1}{2}}, \tag{A.6}$$

while the experimental Nu uncertainty is determined by:

$$\frac{\delta Nu}{Nu} = \left[\left(\frac{\delta h}{h} \right)^2 + \left(\frac{\delta C_p}{C_p} \right)^2 + \left(\frac{\delta k}{k} \right)^2 \right]^{\frac{1}{2}}. \tag{A.7}$$

Uncertainty results for Re and Nu are shown in Table A.5.

Table A.5: Overall heat transfer coefficient uncertainty

Variable	Uncertainty
Re	±1.23%
Nu	±7.23%

A.5 Summary

Although the uncertainty for test data has been approximately quantified, modifications to the calculations may be required if the test experimental test method is modified during later portions of this study.

Appendix B

Additional Results

The primary focus of the current research is to ascertain the flowfield and heat transfer performance of the numerical model. For the sake of brevity, detailed flow results have been omitted from the main body of the report, primarily relating to the temperature field and pressure drop performance.

Additionally, CFD modelling methodology, mesh details and additional parameters are presented in Sec. B.1.

Although the temperature field results are critical to heat transfer performance, the method in which they are obtained can lead to some difficulty in interpreting the data. Although the domain size and geometry are identical and fully periodic in most cases, some variation in the location of the inlet and outlet planes exists due to meshing constraints within Star-CCM+. Furthermore, the temperature range does not scale as cleanly as the velocity field, leading to larger discrepancies when comparing data from differing inlet conditions. For reference, temperature fields corresponding to the majority of the velocity fields shown in Ch. 3 are shown in Sec. B.2.

Modelling pressure drop performance of the turbulizer has typically been met with good success in past studies. Performance is observed to be relatively weakly correlated with variations in geometry, and as such, is not a major focus of the study. For reference, the pressure drop performance of the turbulizer, expressed as a friction factor, is shown in Sec. B.3.

B.1 CFD Model Description and Discussion

Prior to embarking on a full study of the impacts of different modelling constraints, a brief discussion of both mesh and CFD modelling methodology is presented in order to educate the reader and permit the recreation of numerical results.

Throughout the study, several different CFD modelling methodologies are presented. While the derivation of each of these models can be found in many different research papers and texts [41], the important note is that each of them is available in contemporary commercial CFD modelling software packages without requirement of special programming or implementation. As the focus of this research is primarily of an industrial nature, focusing on application of existing foundational models rather than developing of new turbulence modelling approaches, each of the models is left to use the default settings in Star-CCM+. Although it is acknowledged that adaptation of modelling constants may result in an overall improvement of results, a finely tuned model for a particular geometry is not necessarily the goal of the present research; instead, a general model that can be widely applied is a more relevant deliverable. As such, none of the constants have been tuned in any of the primary models tested: the standard $k-\epsilon$ RANS turbulence model, the $k-\omega$ SST turbulence model and the dynamic Smagorinsky large eddy simulation turbulence model.

This approach of leaving the default settings in their beginning state emphasises the general applicability of the developed model, and provides confidence in the generalist user applying the developed approach in a way that can be beneficial among a wide range of geometries and heat exchanger problems.

Although the turbulence model has largely remained unchanged, the mesh does remain somewhat specialised. Great effort and emphasis was placed on the accurate resolution of the thermal boundary layer, and as such, highly resolved “prism layers” near the solid-fluid interface are used to capture the change of fluid properties associated with a large temperature gradient. In this study, it was found through mesh refinement analysis that 20 prism layers, with a 1.3 geometric growth factor appropriately resolved the desired physical changes. This was implemented for both a hexahedral, aligned/structured meshing technique (trim mesh in Star-CCM+), as well as the proprietary polyhedral meshing performed by Star-CCM+. The results were not seen to be impacted by the choice of meshing technique or further refinement. Due to the uniformity of the polyhedral mesh, it was found to be slightly more numerically efficient than the trim meshing methodology, with approximately 58% of the total cells for calculation. Thus, it was chosen as the primary mesh for the study. Of note is that the same mesh was used for the entirety of the training data calculation, ensuring that no artefacts of the meshing process would be responsible for the results observed. The polyhedral mesh is shown in Figs. B.1 and B.2, while the trim mesh is shown in Figs. B.3 and B.4.

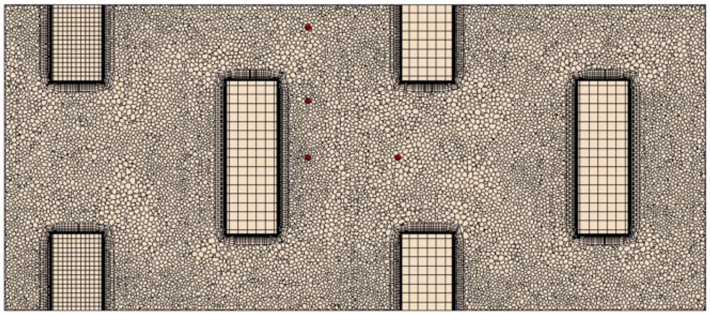


Figure B.1: Polyhedral Mesh generated by Star-CCM+ (through centre of domain)

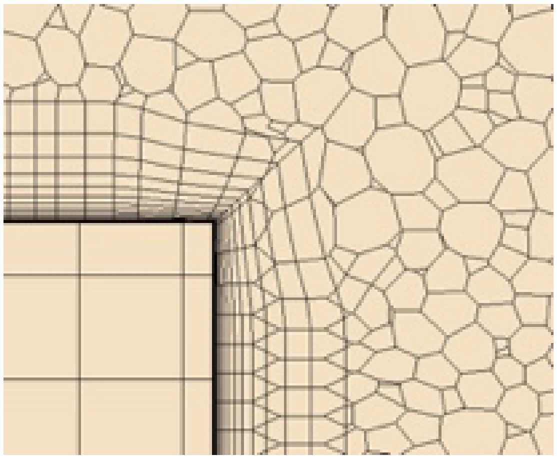


Figure B.2: Polyhedral Mesh generated by Star-CCM+, focusing on the prism layer (20 layers with a 1.3 geometric growth factor)

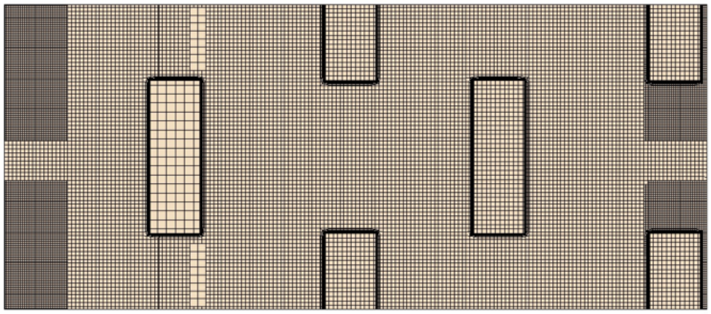


Figure B.3: Trim Mesh generated by Star-CCM+ (through centre of domain)

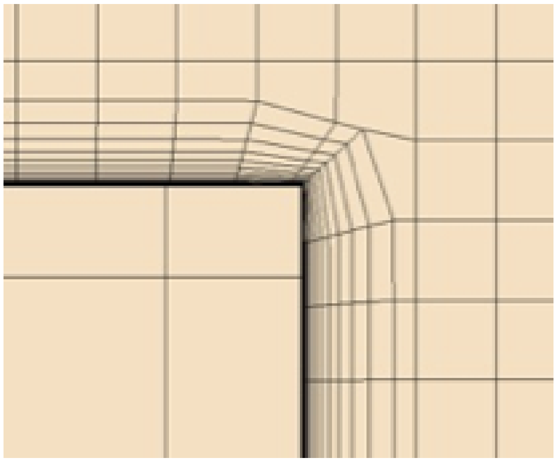


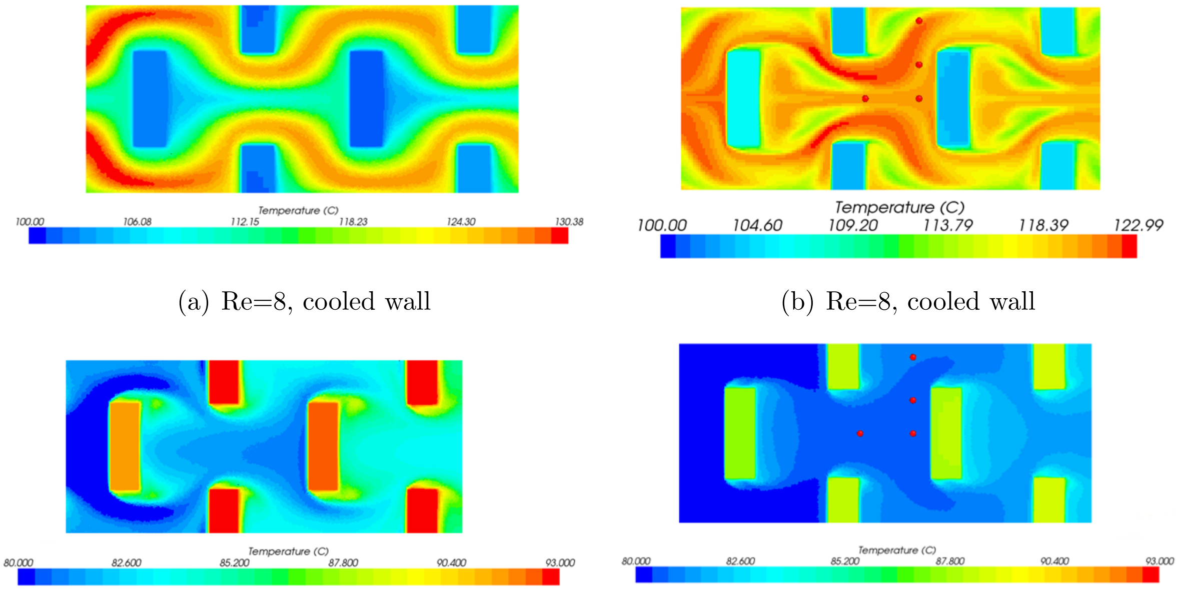
Figure B.4: Trim Mesh generated by Star-CCM+, focusing on the prism layer (20 layers with a 1.3 geometric growth factor)

B.2 Temperature Field Results

The temperature field results presented in Fig. B.5 a) through d) correspond to the flowfields shown in Fig. 3.26. In these figures, the evolution of the temperature field can be easily observed. In low Re flow, the lack of mixing can easily be observed, as high temperature “streamlines” can be observed flowing through the domain. The laminar flow prevents mixing and breakup of these high temperature regions, reducing the overall effectiveness of the heat transfer surface. As such, the performance characteristics many low Re heat transfer enhancement surfaces are limited almost entirely by wetted surface area. As Re increases, so to does the mixing within the computational domain. The difference in temperature throughout the domain also decreases, due in part to the increased mixing as well as the increase in mass flow rate. The thinning of the boundary layer can be observed as the high temperature fluid is forced closer to the turbulizer surface. At the highest Re, Re=476, Fig. B.5 d), very little temperature difference is observed throughout the fluid due to the high level of mixing and high fluid velocities. In this instance, a larger temperature change can also be observed in the turbulizer surface, indicating that the conduction resistance within the turbulizer material is now playing a larger role in overall thermal resistance.

The centreline temperature results in Fig. B.6 a) through d) and Fig. B.8 a) through d) show the variation in temperature field for Re = 476 with different variations in geometry. In this instance, the geometry tested corresponds to that tested in Phase 1 in Sec. 3.4.5. In Fig. B.6 a) through c), increasing the corner radius decreases the wake width of each turbulizer blade. As a result, the high temperature region is funneled at high velocity between turbulizer blades, reducing mixing and leading to a slight reduction in performance. Despite a relatively large corner radius in Fig. B.6 d), the real geometry case indicates that a relatively large wake is formed. The sections shown in Fig. B.8 demonstrate the temperature field taken at the vertical slice plane shown in Fig. B.7. The same general trends can be observed with a small corner radius inducing a higher temperature flow through the gap between turbulizer blades, largely due to the effective gap being reduced in size due to the separation zones.

Futher temperature field results are shown in Fig. B.9 a) through d). In this instance,



(a) Re=8, cooled wall

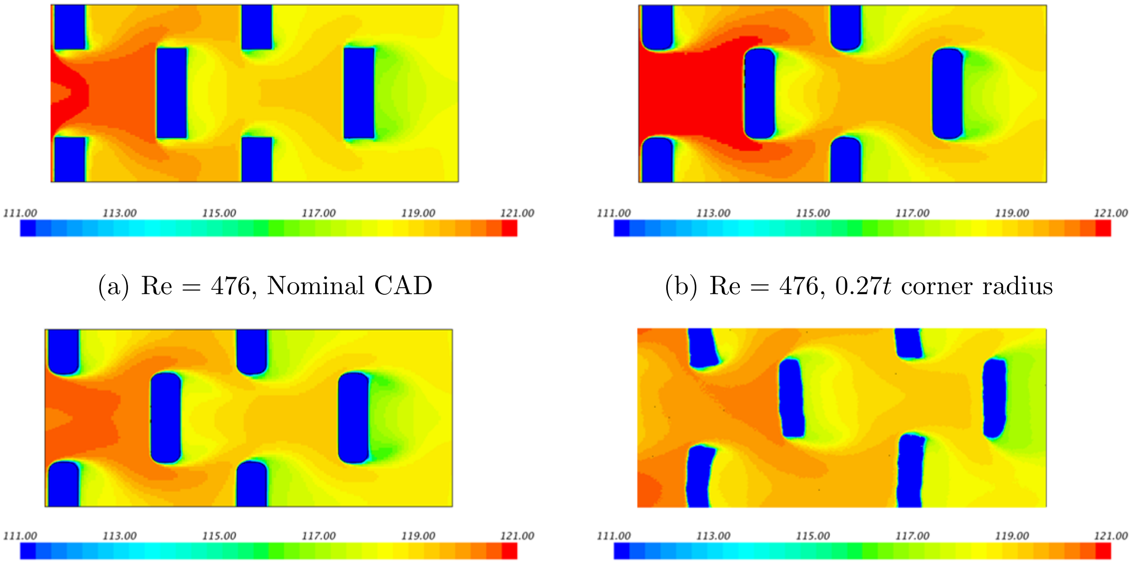
(b) Re=8, cooled wall

(c) Re=119, heated wall

(d) Re=476, heated wall

Figure B.5: Centreline temperature field results, $k - \epsilon$ turbulence model. Note: Re $=$ 8 and Re $=$ 39 for heated wall conditions, Re $=$ 119 and Re $=$ 476 for cooled wall conditions

material thickness is observed to have an influence on the wake formation. The thinner material provides a longer relaxation time for the wake, allowing the higher temperature flow to return to the centre of the blade gap. As a result, the wake of the turb blades becomes wider, as the flow is not forced through as small a gap. As the material thickness increases, the wake decreases in size.



(a) Re = 476, Nominal CAD

(b) Re = 476, 0.27t corner radius

(c) Re = 476, 0.27t corner radius, 0.1mm channel height reduction

(d) Re = 476 Real geometry

Figure B.6: Centreline temperature results for Phase 1 geometry modifications (forming related), $k - \epsilon$ turbulence model, t refers to nominal material thickness

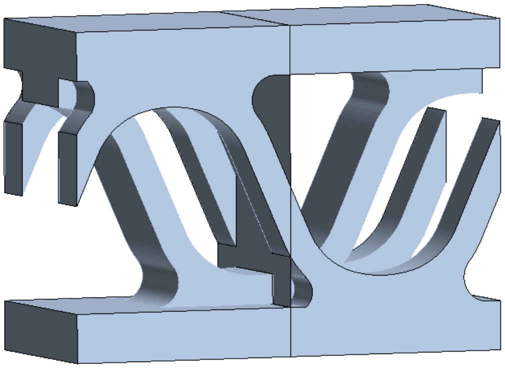
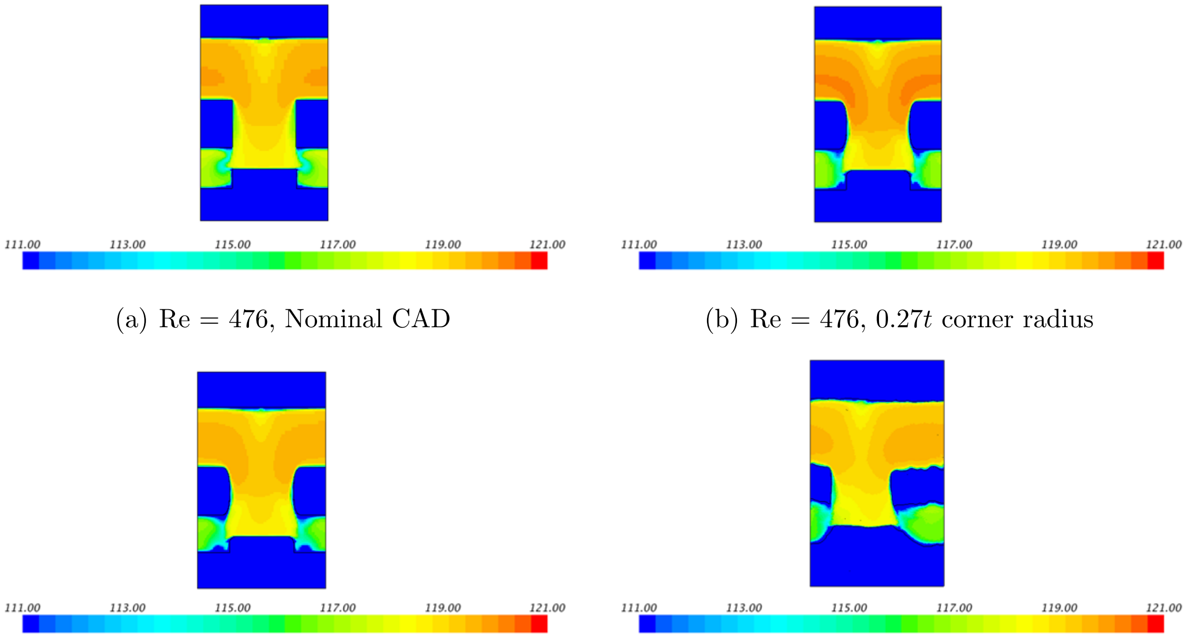


Figure B.7: Orientation and approximate location of vertical slice plane (through centre of domain)



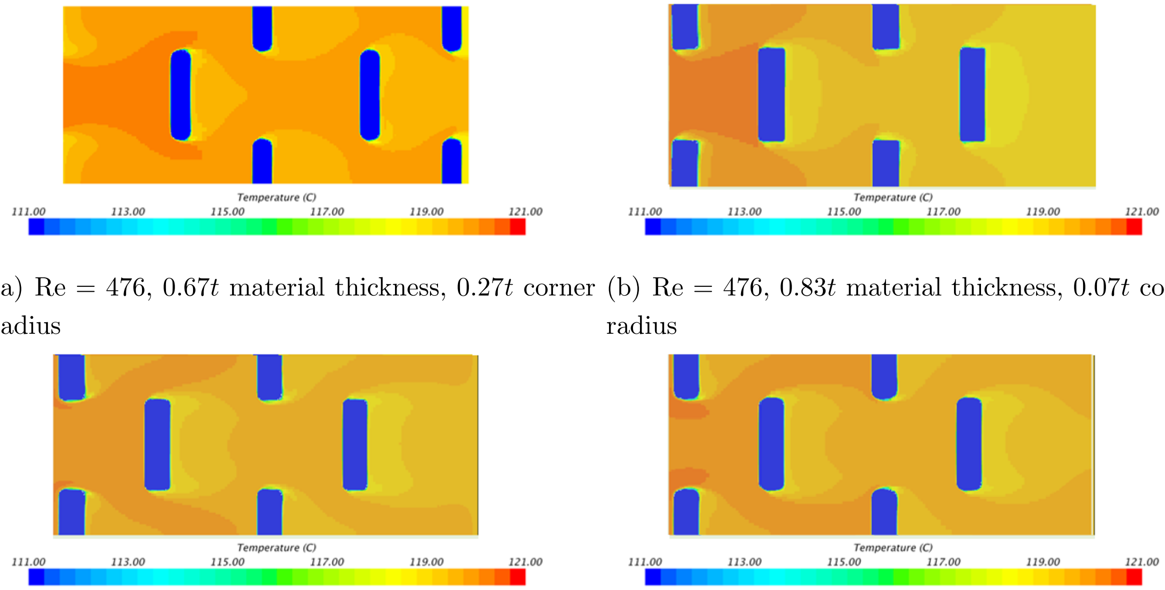
(a) Re = 476, Nominal CAD

(b) Re = 476, 0.27t corner radius

(c) Re = 476, 0.27t corner radius, 0.1mm channel height reduction

(d) Re = 476 Real geometry

Figure B.8: Vertical plane temperature results for Phase 1 geometry modifications (forming related), $k - \epsilon$ turbulence model, t refers to nominal material thickness



(a) Re = 476, 0.67t material thickness, 0.27t corner radius

(b) Re = 476, 0.83t material thickness, 0.07t corner radius

(c) Re = 476, 0.83t material thickness, 0.17t corner radius

(d) Re = 476, 0.83t material thickness, 0.27t corner radius

Figure B.9: Centreline temperature results for Phase 2 geometry modifications (brazing related), $k-\epsilon$ turbulence model, t refers to nominal material thickness

B.3 Pressure Drop Results

In the majority of previous numerical studies conducted by Dana Canada, generally good agreement with test results has been observed. Normal wear and dulling of manufacturing tooling leads to a variation in the resultant performance of the turbulizer. Although this is not ideal from a performance standpoint, it is expected within the industry, with many customers allowing a tolerance of approximately ±10% for both pressure drop and heat transfer performance.

The numerical results for pressure drop are converted to friction factor, allowing for easy comparison and application of test data to many different fluids and flow conditions. Conversion of pressure drop to friction factor is obtained using:

$$\Delta p = f \frac{L}{d_h} \frac{\rho u^2}{2} \tag{B.1}$$

In this instance, a slight change in notation is used. L represents the length of the test section over which pressure drop is measured, while d_h represents the hydraulic diameter or characteristic length scale of the flow. This equation is rearranged and solved for friction factor, with the results for a variety of geometries are presented in Fig. B.10.

From Fig. B.10, it can be seen that the pressure drop is relatively insensitive to both geometric configuration and turbulence model. Although some variation is expected due to the differing wake sizes (contributing to pressure drag), this appears to be a relatively minor contribution to overall pressure drop of the turbulizer. The largest departures from the median friction factor data appear to be the real geometry case, which results in an above median friction factor, and the $0.67t$ material thickness case, which provides a below median friction factor. While the larger flow gap in between turbulizer blades explains the low pressure drop of the $0.67t$ material case, there is more uncertainty about the reason for high pressure drop in the real geometry case. Close inspection of the 3D reconstructed geometry shows formation of small protrusions from the turbulizer blade edges, which possibly contribute to increased pressure drag. While this is expected to happen occasionally during the forming process, this may be an artifact of the particular turbulizer section tested, and cannot be verified without further samples.

Maximum variation of results spans approximately ±15%. Although this does fall slightly outside of the maximum acceptable range specified by most customers, it does

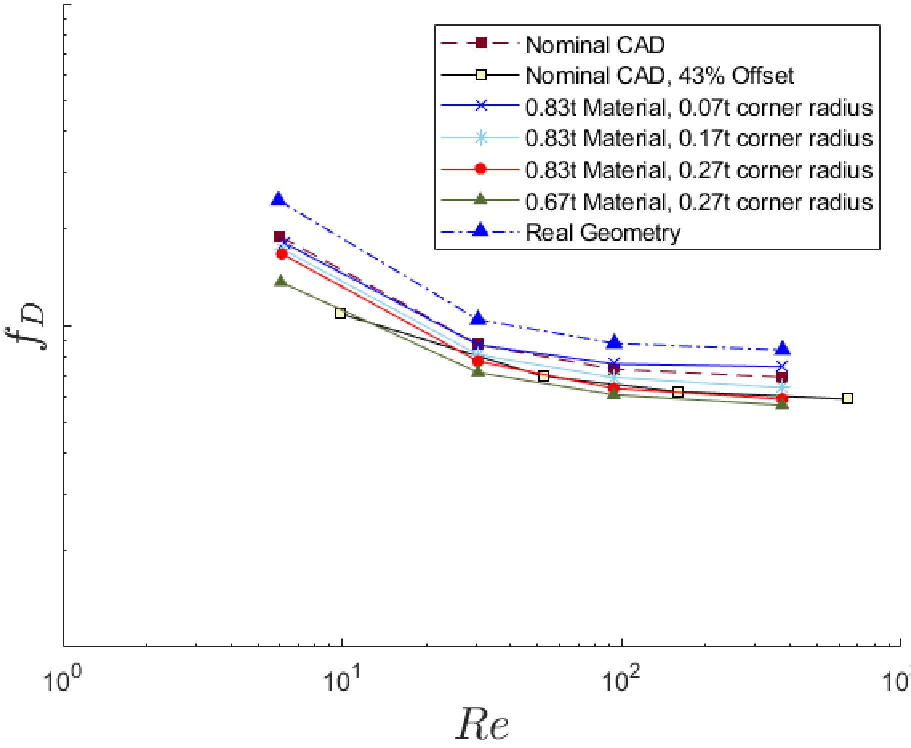


Figure B.10: Friction factor predictions for geometry modifications, all curves for $k-\epsilon$ turbulence model unless otherwise stated

represent a significantly lower source of error than the heat transfer performance. As such, it is accepted that the pressure drop performance is of relatively minor importance within the scope of the current research.

Appendix C

Fluid Properties

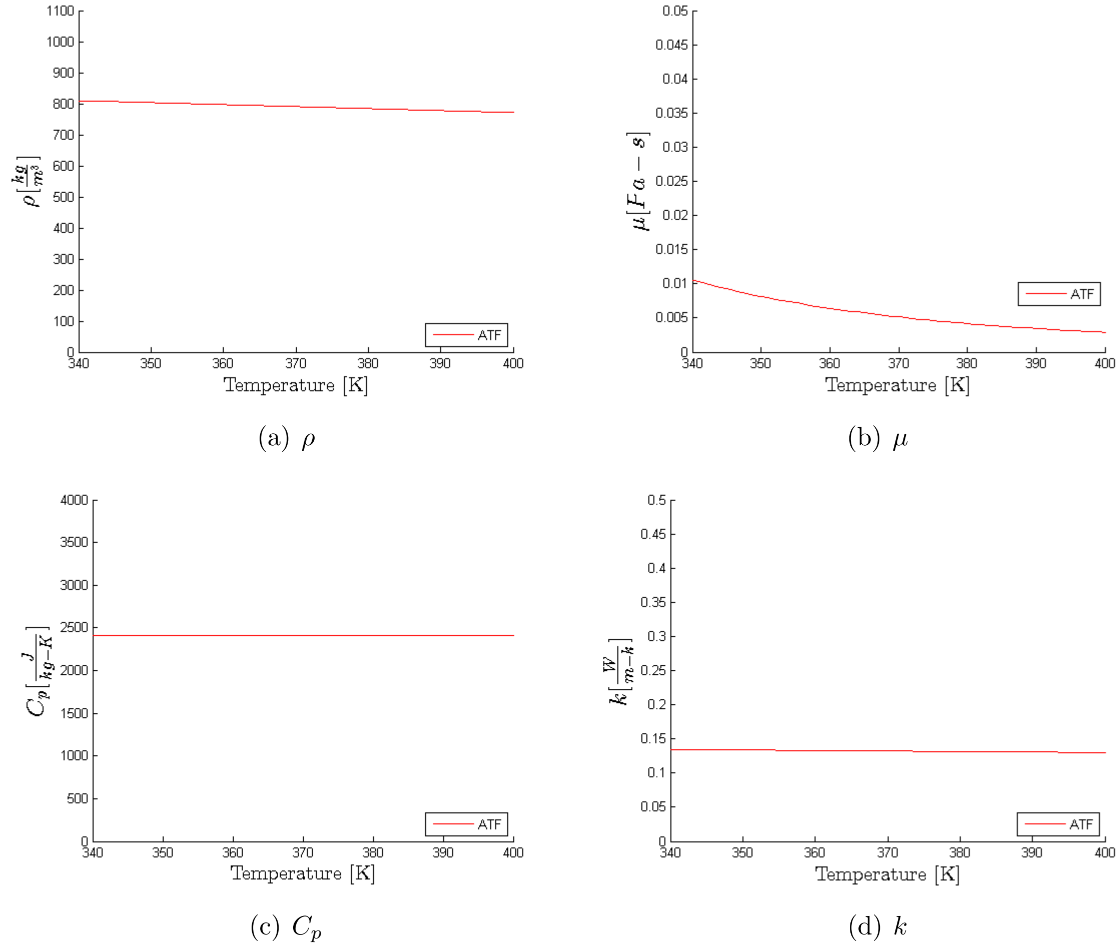
Four separate fluids are used throughout the numerical portion of this study. These correspond to:

1. **ATF-Type A**
2. **50% Ethylene Glycol - 50% water (standard automotive coolant)**
3. **Paratherm OR**
4. **Air, atmospheric pressure**

For each of these fluids, the density, ρ, viscosity, μ, specific heat capacity, C_p and thermal conductivity k are presented over a temperature range of $340K$ to $400K$, which encompasses all simulation conditions used in the present study.

C.1 ATF

Fluid properties for ATF Type A:



(a) ρ

(b) μ

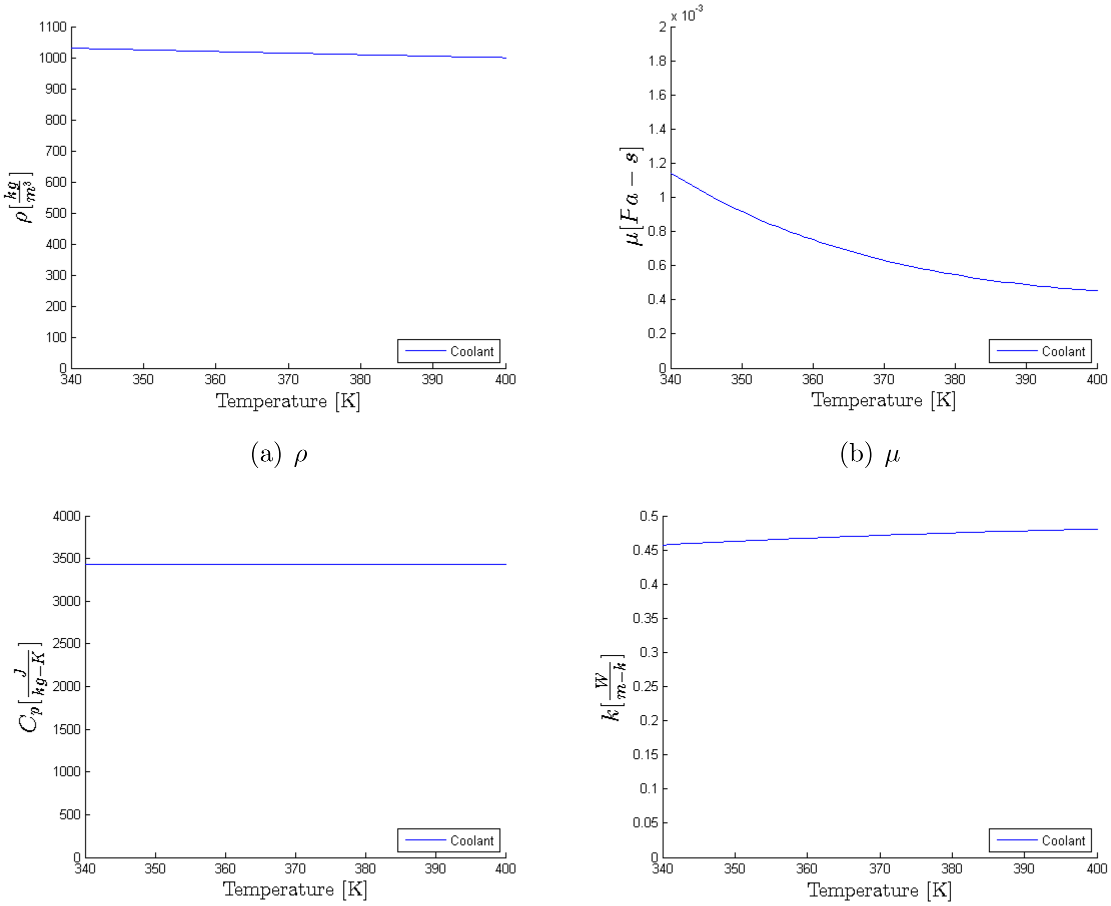
(c) C_p

(d) k

Figure C.1: ATF fluid properties from $340K$ to $400K$

C.2 Coolant

Fluid properties for 50% ethylene glycol 50% water mixture:



(a) ρ

(b) μ

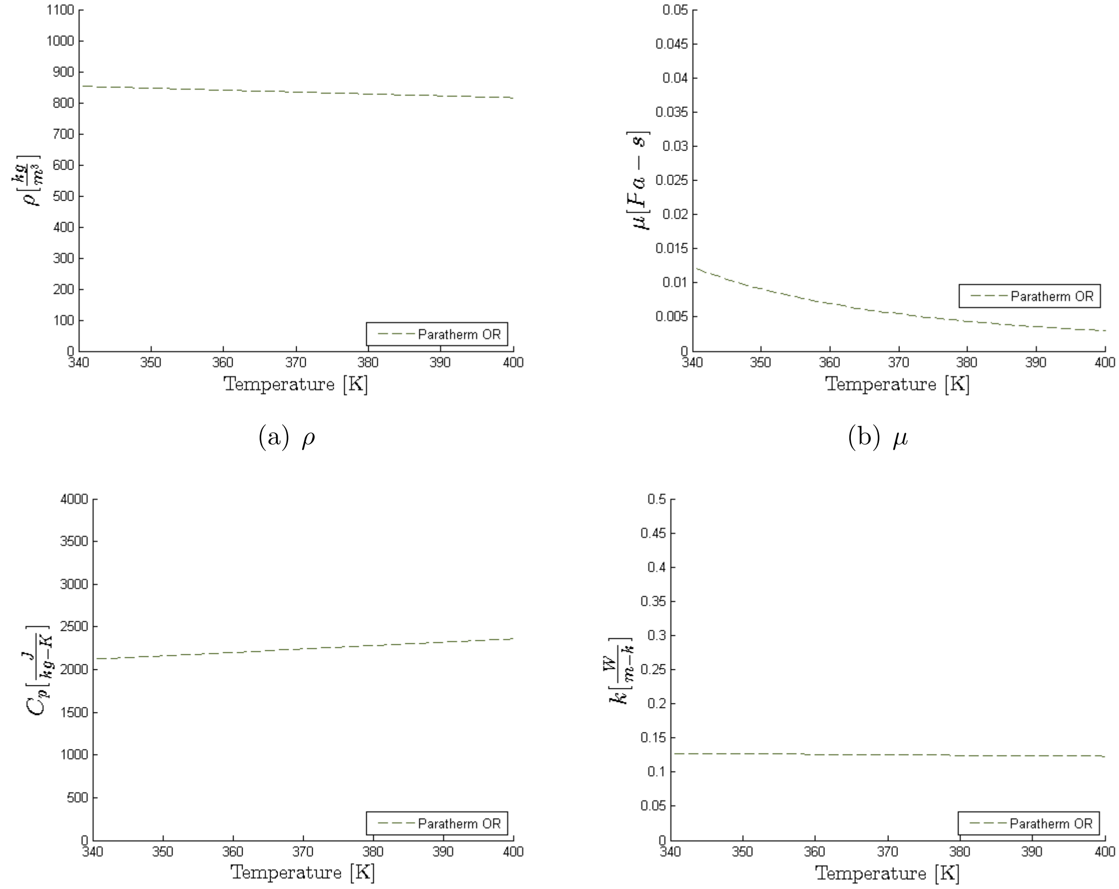
(c) C_p

(d) k

Figure C.2: Coolant fluid properties from $340K$ to $400K$

C.3 Paratherm OR

Fluid properties for Paratherm OR:



(a) ρ

(b) μ

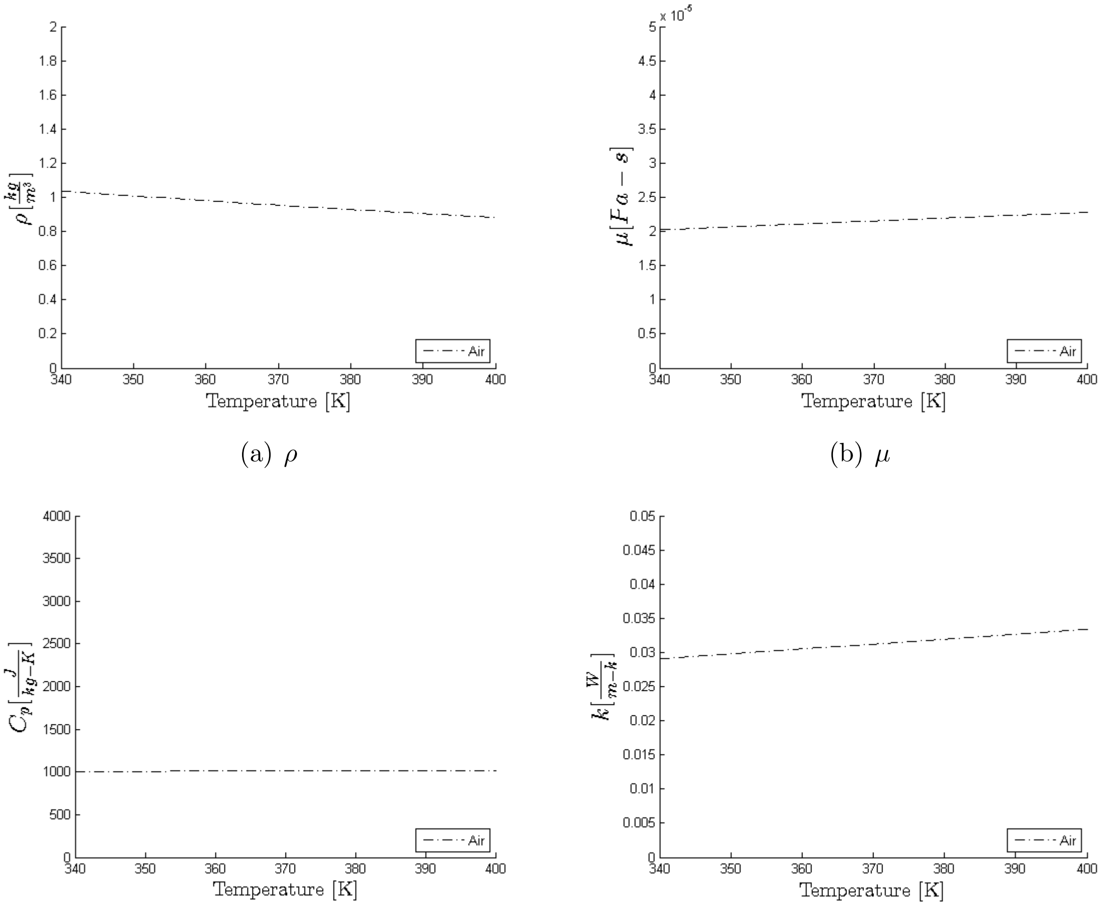
(c) C_p

(d) k

Figure C.3: Paratherm OR fluid properties from $340K$ to $400K$

C.4 Air - 1 atm

Fluid properties for air at 1atm:



(a) ρ

(b) μ

(c) C_p

(d) k

Figure C.4: Paratherm OR fluid properties from $340K$ to $400K$

Appendix D

Water Tunnel Uncertainty Analysis

Characterisation of the uncertainty in the determination of experimental *Re* of the water tunnel is explored herein. The water tunnel uses a measured mass flow over a measured time period to determine the experimental *Re*. Analysis is performed using the Root Sum Square methodology for zero order analysis, such that:

$$\frac{\Delta\phi}{\phi} = \left[\left(\frac{a\Delta u_1}{x_1}\right)^2 + \left(\frac{b\Delta u_2}{x_2}\right)^2 + \left(\frac{c\Delta u_3}{x_3}\right)^2 + \left(\frac{d\Delta u_4}{x_4}\right)^2 + \left(\frac{e\Delta u_5}{x_5}\right)^2 + \dots\right]^{\frac{1}{2}}, \quad \text{(D.1)}$$

where

$$\phi = x_1^a x_2^b x_3^c x_4^d x_5^e \cdots . \quad \text{(D.2)}$$

In this case, the value of interest is *Re*, where:

$$Re = \frac{\rho u D_h}{\mu}. \quad \text{(D.3)}$$

The calculation of u is of critical interest. It is measured by calculating the volumetric flow rate through a known cross sectional area, i.e. length l and width w, with:

$$u = \frac{\dot{V}}{lw} \quad \text{(D.4)}$$

The value of u is based on an open channel, so turbulizer blockage ratio is irrelevant for its calculation. The volumetric flow rate is taken from the mass flow rate, i.e.:

$$\dot{V} = \frac{m}{\rho\Delta t}. \quad \text{(D.5)}$$

With these simplifications substituted into the equation for Re, the expression of interest now becomes:

$$Re = \frac{mD_h}{\Delta t \mu l w}, \tag{D.6}$$

and thus, uncertainty is expressed as:

$$\frac{\Delta Re}{Re} = \left[\left(\frac{\Delta m}{m}\right)^2 + \left(\frac{\Delta D_h}{D_h}\right)^2 + \left(\frac{-\Delta t}{t}\right)^2 + \left(\frac{-\Delta \mu}{\mu}\right)^2 + \left(\frac{-\Delta l}{l}\right)^2 + \left(\frac{-\Delta w}{w}\right)^2\right]^{\frac{1}{2}} \tag{D.7}$$

The variables in this expression, and their 95% C.I. uncertainty is expressed in Table D.1. Substitution of these values in the table into Eq. D.7 yields:

Table D.1: Uncertainty in Measurements

Variable	Uncertainty
mass $[g]$	$\pm 0.5\ g$
time $[s]$	$\pm 0.5\ s$
D_h $[m]$	$\pm 0.0001\ m$
l $[m]$	$\pm 0.0001\ m$
w $[m]$	$\pm 0.0001\ m$
μ $[Pa - s]$	$\pm 0.5\%$

$$\frac{\Delta Re}{Re} = \left[\left(\frac{\Delta m}{m}\right)^2 + \left(\frac{\Delta D_h}{D_h}\right)^2 + \left(\frac{-\Delta t}{t}\right)^2 + \left(\frac{-\Delta \mu}{\mu}\right)^2 + \left(\frac{-\Delta l}{l}\right)^2 + \left(\frac{-\Delta w}{w}\right)^2\right]^{\frac{1}{2}} \tag{D.8}$$

For $40 \leq Re \leq 50$, measurement mass is approximately 1 kg, measurement time is approximately 150 s, channel width is 0.0241 m and channel length is 0.06 m. Uncertainty of fluid properties, particularly μ are on the order of 1%, while the uncertainty of the hydraulic diameter, D_h is approximately equal to the uncertainty in l or w.

Thus, the overall uncertainty is equal to:

$$\frac{\Delta Re}{Re} = 0.0125 = 1.25\%. \tag{D.9}$$

Appendix E

STAR-CCM+ Code

For the purpose of continued development, the entirety of the code used to apply the models developed herein is included.

E.1 setupAll.java

This code block serves the purpose of initialising a simulations and loading all other Java macros.

```
// STAR-CCM+ macro: nestedMacro.java
// Written by STAR-CCM+ 13.02.011
package macro;

import java.util.*;

import star.common.*;
import star.base.neo.*;
import java.io.*;

public class setupAll extends StarMacro {

  public void execute() {
    execute0();
  }

  private void execute0() {

    // loop through all macros to run the setup scripts

    new StarScript(getActiveRootObject(), new File(resolvePath("
       ↪ LoadUserFunction.java"))).play();
    new StarScript(getActiveRootObject(), new File(resolvePath("
       ↪ setupFieldFunctions.java"))).play();
    new StarScript(getActiveRootObject(), new File(resolvePath("setupFluid.
       ↪ java"))).play();
    new StarScript(getActiveRootObject(), new File(resolvePath("
       ↪ setupOfAveragingPlanesFunctions.java"))).play();
    new StarScript(getActiveRootObject(), new File(resolvePath("
       ↪ setupBoundariesAndPMCoeffs.java"))).play();

  }
}
```

E.2 LoadUserFunction.java

This code block serves the purpose of loading the user defined function used to return the turbulizer library calculations. Because the existing neural network code cannot be natively executed in java, a separate .so library is generated and linked to the simulation.

```
// STAR-CCM+ macro: LoadUserFunction.java
// Written by STAR-CCM+ 13.02.011
package macro;

import java.util.*;

import star.common.*;
import star.base.neo.*;

public class LoadUserFunction extends StarMacro {

  public void execute() {
    execute0();
  }

  private void execute0() {

    Simulation simulation_0 =
      getActiveSimulation();

    UserLibrary userLibrary_0 =
      simulation_0.getUserFunctionManager().createUserLibrary(resolvePath
         ↪ ("../2018-07-31-ValidationCasesNN-Instability/STAR-Code/
         ↪ TestDirectory/ANN-Fd-Nu.so"));
  }
}
```

E.3 setupFieldFunctions.java

This code block serves the purpose of initialising all simple field function calculations used by the simulation to calculate flow parameters and heat transfer.

```
// STAR-CCM+ macro: setupFieldFunctions.java
// Written by STAR-CCM+ 13.02.011
package macro;

import java.util.*;

import star.common.*;
import star.base.neo.*;

public class setupFieldFunctions extends StarMacro {

  public void execute() {
    execute0();
  }

  private void execute0() {

    Simulation simulation_0 =
      getActiveSimulation();

    // setup various field functions, based on what’s required. Add
       ↪ functionality for checking existence of those functions later on
       ↪ .

    UserFieldFunction userFieldFunction_0 = simulation_0.
       ↪ getFieldFunctionManager().createFieldFunction();
    userFieldFunction_0.getTypeOption().setSelected(FieldFunctionTypeOption
       ↪ .Type.SCALAR);
    userFieldFunction_0.setFunctionName("ChannelHeight");
```

```
userFieldFunction_0.setDefinition("0.00241");
userFieldFunction_0.setPresentationName("ChannelHeight");

UserFieldFunction userFieldFunction_1 = simulation_0.
   ↪ getFieldFunctionManager().createFieldFunction();
userFieldFunction_1.getTypeOption().setSelected(FieldFunctionTypeOption
   ↪ .Type.SCALAR);
userFieldFunction_1.setPresentationName("LengthScale");
userFieldFunction_1.setFunctionName("LengthScale");
userFieldFunction_1.setDefinition("0.001");

UserFieldFunction userFieldFunction_2 = simulation_0.
   ↪ getFieldFunctionManager().createFieldFunction();
userFieldFunction_2.getTypeOption().setSelected(FieldFunctionTypeOption
   ↪ .Type.SCALAR);
userFieldFunction_2.setPresentationName("f");
userFieldFunction_2.setFunctionName("f");
userFieldFunction_2.setDefinition("$UserFdInterpVal");

UserFieldFunction userFieldFunction_3 = simulation_0.
   ↪ getFieldFunctionManager().createFieldFunction();
userFieldFunction_3.getTypeOption().setSelected(FieldFunctionTypeOption
   ↪ .Type.SCALAR);
userFieldFunction_3.setPresentationName("dTSurf");
userFieldFunction_3.setFunctionName("dTSurf");
userFieldFunction_3.setDefinition("-$Temperature+$CL_Temp");

UserFieldFunction userFieldFunction_4 = simulation_0.
   ↪ getFieldFunctionManager().createFieldFunction();
userFieldFunction_4.getTypeOption().setSelected(FieldFunctionTypeOption
   ↪ .Type.SCALAR);
userFieldFunction_4.setPresentationName("CL_Temp");
userFieldFunction_4.setFunctionName("CL_Temp");
```

```
userFieldFunction_4.setDefinition("${Cold1Tavg}+${Cold2Tavg}+${
   ↪ Cold3Tavg}+${Cold4Tavg}+${Cold5Tavg}+${Hot1Tavg}+${Hot2Tavg}+${
   ↪ Hot3Tavg}+${Hot4Tavg}+${Hot5Tavg}");

UserFieldFunction userFieldFunction_5 = simulation_0.
   ↪ getFieldFunctionManager().createFieldFunction();
userFieldFunction_5.getTypeOption().setSelected(FieldFunctionTypeOption
   ↪ .Type.SCALAR);
userFieldFunction_5.setPresentationName("flowAngle");
userFieldFunction_5.setFunctionName("flowAngle");
userFieldFunction_5.setDefinition("abs(atan($$Velocity[0]/($$Velocity
   ↪ [2]+0.001)))");

UserFieldFunction userFieldFunction_6 = simulation_0.
   ↪ getFieldFunctionManager().createFieldFunction();
userFieldFunction_6.getTypeOption().setSelected(FieldFunctionTypeOption
   ↪ .Type.SCALAR);
userFieldFunction_6.setPresentationName("UserA");
userFieldFunction_6.setFunctionName("UserA");
Units units_1 = simulation_0.getUnitsManager().getPreferredUnits(new
   ↪ IntVector(new int[] {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
   ↪ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}));
Units units_2 = simulation_0.getUnitsManager().getPreferredUnits(new
   ↪ IntVector(new int[] {0, -2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
   ↪ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}));
userFieldFunction_6.setDefinition("$dTSurf*$h-${
   ↪ InternalWallHeatFluxCoefficientA}");

UserFieldFunction userFieldFunction_7 = simulation_0.
   ↪ getFieldFunctionManager().createFieldFunction();
userFieldFunction_7.getTypeOption().setSelected(FieldFunctionTypeOption
   ↪ .Type.SCALAR);
userFieldFunction_7.setPresentationName("UserB");
```

```
userFieldFunction_7.setFunctionName("UserB");
Units units_3 = simulation_0.getUnitsManager().getPreferredUnits(new
   ↪ IntVector(new int[] {0, -2, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
   ↪  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}));
userFieldFunction_7.setDefinition("0.02-${
   ↪ InternalWallHeatFluxCoefficientB}");

UserFieldFunction userFieldFunction_8 = simulation_0.
   ↪ getFieldFunctionManager().createFieldFunction();
userFieldFunction_8.getTypeOption().setSelected(FieldFunctionTypeOption
   ↪ .Type.SCALAR);
userFieldFunction_8.setPresentationName("UserC");
userFieldFunction_8.setFunctionName("UserC");
userFieldFunction_8.setDefinition("0.02-${
   ↪ InternalWallHeatFluxCoefficientC}");

UserFieldFunction userFieldFunction_9 = simulation_0.
   ↪ getFieldFunctionManager().createFieldFunction();
userFieldFunction_9.getTypeOption().setSelected(FieldFunctionTypeOption
   ↪ .Type.SCALAR);
userFieldFunction_9.setFunctionName("UserD");
userFieldFunction_9.setPresentationName("UserD");
Units units_4 = simulation_0.getUnitsManager().getPreferredUnits(new
   ↪ IntVector(new int[] {0, -2, 0, -4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
   ↪  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}));
userFieldFunction_9.setDefinition("-${InternalWallHeatFluxCoefficientD
   ↪ }");

UserFieldFunction userFieldFunction_10 = simulation_0.
   ↪ getFieldFunctionManager().createFieldFunction();
userFieldFunction_10.getTypeOption().setSelected(
   ↪ FieldFunctionTypeOption.Type.SCALAR);
userFieldFunction_10.setPresentationName("Pr");
```

```
userFieldFunction_10.setFunctionName("Pr");
Units units_5 = simulation_0.getUnitsManager().getPreferredUnits(new
   ↪ IntVector(new int[] {-1, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
   ↪  0, 0, 0, 0, 0, 0, 0, 0, 0, 0}));
Units units_6 = simulation_0.getUnitsManager().getPreferredUnits(new
   ↪ IntVector(new int[] {0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
   ↪ 1, 0, 0, 0, 0, 0, 0, 0, 0, 0}));
Units units_7 = simulation_0.getUnitsManager().getPreferredUnits(new
   ↪ IntVector(new int[] {0, -1, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
   ↪  0, 0, 0, 0, 0, 0, 0, 0, 0, 0}));
userFieldFunction_10.setDefinition("${SpecificHeat}*${DynamicViscosity
   ↪ }/${ThermalConductivity}");

UserFieldFunction userFieldFunction_11 = simulation_0.
   ↪ getFieldFunctionManager().createFieldFunction();
userFieldFunction_11.getTypeOption().setSelected(
   ↪ FieldFunctionTypeOption.Type.SCALAR);
userFieldFunction_11.setPresentationName("CL_Velocity");
userFieldFunction_11.setFunctionName("CL_Velocity");
userFieldFunction_11.setDefinition("mag($$Velocity)");

UserFieldFunction userFieldFunction_12 = simulation_0.
   ↪ getFieldFunctionManager().createFieldFunction();
userFieldFunction_12.getTypeOption().setSelected(
   ↪ FieldFunctionTypeOption.Type.SCALAR);
userFieldFunction_12.setPresentationName("PMCoeff");
userFieldFunction_12.setFunctionName("PMCoeff");
userFieldFunction_12.setDefinition("0.5*$Density*mag($$Velocity)/
   ↪ $LengthScale*$f");

UserFieldFunction userFieldFunction_20 = simulation_0.
   ↪ getFieldFunctionManager().createFieldFunction();
```

```
    userFieldFunction_20.getTypeOption().setSelected(
       ↪ FieldFunctionTypeOption.Type.SCALAR);
    userFieldFunction_20.setPresentationName("ReInternal");
    userFieldFunction_20.setFunctionName("ReInternal");
    Units units_8 = simulation_0.getUnitsManager().getPreferredUnits(new
       ↪ IntVector(new int[] {1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       ↪ 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0}));
    userFieldFunction_20.setDefinition("${Density}*${CL_Velocity}*${
       ↪ LengthScale}/${DynamicViscosity}");


//BECAUSE THE CALCULATION OF PR IS ONLY DONE LOCALLY, TAKE THE CL AVERAGED
    ↪  FLUID TEMPERATURE AND APPLY THE STANDARD CALCULATION
    UserFieldFunction userFieldFunction_30 = simulation_0.
       ↪ getFieldFunctionManager().createFieldFunction();
    userFieldFunction_30.getTypeOption().setSelected(
       ↪ FieldFunctionTypeOption.Type.SCALAR);
    userFieldFunction_30.setPresentationName("CL_Pr");
    userFieldFunction_30.setFunctionName("CL_Pr");
    userFieldFunction_30.setDefinition("${SpecificHeat}*pow(x-x*pow(
       ↪ $CL_Temp,0.5)+xe-3*$CL_Temp,3)/(x+x*pow($CL_Temp,0.5)-xe-3*
       ↪ $CL_Temp)");


    Region region_0 = simulation_0.getRegionManager().getRegion("Cold_Turb1
       ↪ ");
    Region region_1 = simulation_0.getRegionManager().getRegion("Cold_Turb2
       ↪ ");
    Region region_2 = simulation_0.getRegionManager().getRegion("Cold_Turb3
       ↪ ");
    Region region_3 = simulation_0.getRegionManager().getRegion("Cold_Turb4
       ↪ ");
```

```
Region region_7 = simulation_0.getRegionManager().getRegion("Cold_Turb5
   ↪ ");
Region region_4 = simulation_0.getRegionManager().getRegion("Hot_Turb1
   ↪ ");
Region region_5 = simulation_0.getRegionManager().getRegion("Hot_Turb2
   ↪ ");
Region region_6 = simulation_0.getRegionManager().getRegion("Hot_Turb3
   ↪ ");
Region region_8 = simulation_0.getRegionManager().getRegion("Hot_Turb4
   ↪ ");
Region region_9 = simulation_0.getRegionManager().getRegion("Hot_Turb5
   ↪ ");

UserFieldFunction userFieldFunction_21 = ((UserFieldFunction)
   ↪ simulation_0.getFieldFunctionManager().getFunction("ReInternal")
   ↪ );
XyzInternalTable xyzInternalTable_0 = simulation_0.getTableManager().
   ↪ createTable(XyzInternalTable.class);
xyzInternalTable_0.setFieldFunctions(new NeoObjectVector(new Object[] {
   ↪ userFieldFunction_21}));
xyzInternalTable_0.getParts().setQuery(null);
xyzInternalTable_0.setPresentationName("Re");
xyzInternalTable_0.getParts().setQuery(null);
xyzInternalTable_0.getParts().setObjects(region_0,region_1,region_2,
   ↪ region_3,region_7,region_4,region_5,region_6,region_8,region_9);
xyzInternalTable_0.extract();

UserFieldFunction userFieldFunction_13 = simulation_0.
   ↪ getFieldFunctionManager().createFieldFunction();
userFieldFunction_13.getTypeOption().setSelected(
   ↪ FieldFunctionTypeOption.Type.SCALAR);
userFieldFunction_13.setPresentationName("Re");
userFieldFunction_13.setFunctionName("Re");
```

```
    Units units_9 = simulation_0.getUnitsManager().getPreferredUnits(new
       ↪ IntVector(new int[] {1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       ↪ 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0}));
    userFieldFunction_13.setDefinition("interpolatePositionTable(@Table(\"
       ↪ Re\"),\"ReInternal\")");

    UserFieldFunction userFieldFunction_14 = simulation_0.
       ↪ getFieldFunctionManager().createFieldFunction();
    userFieldFunction_14.getTypeOption().setSelected(
       ↪ FieldFunctionTypeOption.Type.SCALAR);
    userFieldFunction_14.setPresentationName("h");
    userFieldFunction_14.setFunctionName("h");
    userFieldFunction_14.setDefinition("$Nu*${ThermalConductivity}/${
       ↪ ChannelHeight}");


///note SCALING PERFORMED BASED ON TEST DATA AT PR=7.00 (HAVOLINE HELAC
   ↪ COOLANT FLUID PROPERTIES)
    UserFieldFunction userFieldFunction_15 = simulation_0.
       ↪ getFieldFunctionManager().createFieldFunction();
    userFieldFunction_15.getTypeOption().setSelected(
       ↪ FieldFunctionTypeOption.Type.SCALAR);
    userFieldFunction_15.setPresentationName("Nu");
    userFieldFunction_15.setFunctionName("Nu");
    userFieldFunction_15.setDefinition("pow($CL_Pr/7,0.333)*max(
       ↪ $UserNuInterpVal,0.01)");

    UserFieldFunction userFieldFunction_16 = simulation_0.
       ↪ getFieldFunctionManager().createFieldFunction();
    userFieldFunction_16.getTypeOption().setSelected(
       ↪ FieldFunctionTypeOption.Type.SCALAR);
    userFieldFunction_16.setPresentationName("rho_HELAC");
    userFieldFunction_16.setFunctionName("rho_HELAC");
    userFieldFunction_16.setDefinition("x-x*pow($Temperature,0.5)+xe-2*
       ↪ $Temperature");

    UserFieldFunction userFieldFunction_17 = simulation_0.
       ↪ getFieldFunctionManager().createFieldFunction();
    userFieldFunction_17.getTypeOption().setSelected(
       ↪ FieldFunctionTypeOption.Type.SCALAR);
    userFieldFunction_17.setPresentationName("k_HELAC");
    userFieldFunction_17.setFunctionName("k_HELAC");
    userFieldFunction_17.setDefinition("-x+0.x*pow($Temperature,0.5)-xe-3*
       ↪ $Temperature");

    UserFieldFunction userFieldFunction_18 = simulation_0.
       ↪ getFieldFunctionManager().createFieldFunction();
    userFieldFunction_18.getTypeOption().setSelected(
       ↪ FieldFunctionTypeOption.Type.SCALAR);
    userFieldFunction_18.setPresentationName("mu_HELAC");
    userFieldFunction_18.setFunctionName("mu_HELAC");
    userFieldFunction_18.setDefinition("pow(x-x*pow($Temperature,0.5)+xe-3*
       ↪ $Temperature,3)");

    UserFieldFunction userFieldFunction_19 = simulation_0.
       ↪ getFieldFunctionManager().createFieldFunction();
    userFieldFunction_19.getTypeOption().setSelected(
       ↪ FieldFunctionTypeOption.Type.SCALAR);
    userFieldFunction_19.setPresentationName("Cp_HELAC");
    userFieldFunction_19.setFunctionName("Cp_HELAC");
    userFieldFunction_19.setDefinition("x-x*pow($Temperature,0.5)+x*
       ↪ $Temperature");
      }
}
```

E.4 setupOfAveragingPlanesFunctions.java

This code block is the least general of those used, and includes the code required to set up each of the averaging planes at the centreline of each channel. A more generalised version of the code would detect the upper and lower extents of each “porous media” allocated region, then set a plane to exist at the average of those values.

```java
// STAR-CCM+ macro: setupOfAveragingPlanesFunctions.java
// Written by STAR-CCM+ 13.02.011
package macro;

import java.util.*;

import star.common.*;
import star.base.neo.*;
import star.vis.*;

public class setupOfAveragingPlanesFunctions extends StarMacro {

  public void execute() {
    execute0();
  }

  private void execute0() {
    String[] turbNames = {"Cold1","Cold2","Cold3","Cold4","Cold5","Hot1","
       ↪ Hot2","Hot3","Hot4","Hot5"};
    String[] regionNames = {"Cold_Turb1","Cold_Turb2","Cold_Turb3","
       ↪ Cold_Turb4","Cold_Turb5","Hot_Turb1","Hot_Turb2","Hot_Turb3","
       ↪ Hot_Turb4","Hot_Turb5"};
    double[] turbMidplanes =
       ↪ {0.029025,0.023065,0.017105,0.011145,0.005185,0.026045,0.020085,
       ↪  0.014125, 0.008165, 0.002205};
    double turbDelta = 0.0009;
```

```
Simulation simulation_0 = getActiveSimulation();


for (int i=0;i<10;i++){
    Region region_0 =
      simulation_0.getRegionManager().getRegion(regionNames[i]);
    PlaneSection planeSection_0 = (PlaneSection) simulation_0.
       ↪ getPartManager().createImplicitPart(new NeoObjectVector(new
       ↪ Object[] {}), new DoubleVector(new double[] {0.0, 1.0, 0.0}),
       ↪  new DoubleVector(new double[] {0.0, turbMidplanes[i], 0.0}),
       ↪  0, 1, new DoubleVector(new double[] {0.0}));
    planeSection_0.getInputParts().setObjects(region_0);
    planeSection_0.setPresentationName(turbNames[i]);

    PlaneSection planeSection_1 = (PlaneSection) simulation_0.
       ↪ getPartManager().createImplicitPart(new NeoObjectVector(new
       ↪ Object[] {}), new DoubleVector(new double[] {0.0, 1.0, 0.0}),
       ↪  new DoubleVector(new double[] {0.0, turbMidplanes[i]-
       ↪ turbDelta, 0.0}), 0, 1, new DoubleVector(new double[] {0.0}))
       ↪ ;
      planeSection_1.getInputParts().setObjects(region_0);


    planeSection_1.setPresentationName(turbNames[i]+"-");

    PlaneSection planeSection_2 =
      (PlaneSection) simulation_0.getPartManager().createImplicitPart(
         ↪ new NeoObjectVector(new Object[] {}), new DoubleVector(new
         ↪ double[] {0.0, 1.0, 0.0}), new DoubleVector(new double[]
         ↪ {0.0,turbMidplanes[i]+turbDelta, 0.0}), 0, 1, new
         ↪ DoubleVector(new double[] {0.0}));
```

```
planeSection_2.getInputParts().setObjects(region_0);

planeSection_2.setPresentationName(turbNames[i]+"+");



//////////////////////////////////////////////
/// create tables for each of the functions ///
//////////////////////////////////////////////

PrimitiveFieldFunction primitiveFieldFunction_0 = ((
   ↪ PrimitiveFieldFunction) simulation_0.getFieldFunctionManager
   ↪ ().getFunction("Temperature"));
PrimitiveFieldFunction primitiveFieldFunction_1 = ((
   ↪ PrimitiveFieldFunction) simulation_0.getFieldFunctionManager
   ↪ ().getFunction("Velocity"));
VectorMagnitudeFieldFunction vectorMagnitudeFieldFunction_0 = ((
   ↪ VectorMagnitudeFieldFunction) primitiveFieldFunction_1.
   ↪ getMagnitudeFunction());


XyzInternalTable xyzInternalTable_0 = simulation_0.getTableManager
   ↪ ().createTable(XyzInternalTable.class);
xyzInternalTable_0.setFieldFunctions(new NeoObjectVector(new Object
   ↪ [] {primitiveFieldFunction_0, vectorMagnitudeFieldFunction_0
   ↪ }));
xyzInternalTable_0.getParts().setQuery(null);
xyzInternalTable_0.setPresentationName(turbNames[i]);
xyzInternalTable_0.getParts().setQuery(null);
xyzInternalTable_0.getParts().setObjects(planeSection_0);
xyzInternalTable_0.extract();
TableUpdate tableUpdate_0 = xyzInternalTable_0.getTableUpdate();
tableUpdate_0.setAutoExtract(true);
```

```
tableUpdate_0.getUpdateModeOption().setSelected(
   ↪ StarUpdateModeOption.Type.ITERATION);

XyzInternalTable xyzInternalTable_1 = simulation_0.getTableManager
   ↪ ().createTable(XyzInternalTable.class);
xyzInternalTable_1.setFieldFunctions(new NeoObjectVector(new Object
   ↪ [] {primitiveFieldFunction_0, vectorMagnitudeFieldFunction_0
   ↪ }));
xyzInternalTable_1.setPresentationName(turbNames[i]+"-");
xyzInternalTable_1.getParts().setQuery(null);
xyzInternalTable_1.getParts().setObjects(planeSection_1);
xyzInternalTable_1.extract();
TableUpdate tableUpdate_1 = xyzInternalTable_1.getTableUpdate();
tableUpdate_1.setAutoExtract(true);
tableUpdate_1.getUpdateModeOption().setSelected(
   ↪ StarUpdateModeOption.Type.ITERATION);


XyzInternalTable xyzInternalTable_2 = simulation_0.getTableManager
   ↪ ().createTable(XyzInternalTable.class);
xyzInternalTable_2.setFieldFunctions(new NeoObjectVector(new Object
   ↪ [] {primitiveFieldFunction_0, vectorMagnitudeFieldFunction_0
   ↪ }));
xyzInternalTable_2.setPresentationName(turbNames[i]+"+");
xyzInternalTable_2.getParts().setQuery(null);
xyzInternalTable_2.getParts().setObjects(planeSection_2);
xyzInternalTable_2.extract();
TableUpdate tableUpdate_2 = xyzInternalTable_2.getTableUpdate();
tableUpdate_2.setAutoExtract(true);
tableUpdate_2.getUpdateModeOption().setSelected(
   ↪ StarUpdateModeOption.Type.ITERATION);
```

```
    UserFieldFunction userFieldFunction_0 = simulation_0.
      ↪ getFieldFunctionManager().createFieldFunction();
    userFieldFunction_0.getTypeOption().setSelected(
      ↪ FieldFunctionTypeOption.Type.SCALAR);
    userFieldFunction_0.setPresentationName(turbNames[i]+"Tavg");
    userFieldFunction_0.setFunctionName(turbNames[i]+"Tavg");
    userFieldFunction_0.setDefinition("(${RegionIndex} == "+region_0.
      ↪ getIndex()+") ? (interpolatePositionTable(@Table(\""+
      ↪ turbNames[i]+"-\"),\"Temperature\")+interpolatePositionTable(
      ↪ @Table(\""+turbNames[i]+"\"),\"Temperature\")+
      ↪ interpolatePositionTable(@Table(\""+turbNames[i]+"+\"),\"
      ↪ Temperature\"))/3 : 0");

  }
  UserFieldFunction userFieldFunction_0 = simulation_0.
    ↪ getFieldFunctionManager().createFieldFunction();
  userFieldFunction_0.getTypeOption().setSelected(FieldFunctionTypeOption
    ↪ .Type.SCALAR);
  userFieldFunction_0.setPresentationName("ChannelTavg");
  userFieldFunction_0.setFunctionName("ChannelTavg");
  userFieldFunction_0.setDefinition("$"+turbNames[0]+"Tavg+$"+turbNames
    ↪ [1]+"Tavg+$"+turbNames[2]+"Tavg+$"+turbNames[3]+"Tavg+$"+
    ↪ turbNames[4]+"Tavg+$"+turbNames[5]+"Tavg+$"+turbNames[6]+"Tavg+$
    ↪ "+turbNames[7]+"Tavg+$"+turbNames[8]+"Tavg+$"+turbNames[9]+"Tavg
    ↪ ");
 }
}
```

E.5 setupBoundariesAndPMCoeffs.java

This code block sets up and allocates all boundaries to the proper conjugate heat transfer models. The models to leverage the local porous media coefficients are also set to accept input from the external codebase.

```
// STAR-CCM+ macro: SetBoundariesAndPMCoeffs.java
// Written by STAR-CCM+ 13.02.011
package macro;

import java.util.*;

import star.common.*;
import star.base.neo.*;
import star.vis.*;
import star.flow.*;
import star.energy.*;

public class setupBoundariesAndPMCoeffs extends StarMacro {

  public void execute() {
    execute0();
  }

  private void execute0() {
    String[] regionNames = {"Cold_Turb1","Cold_Turb2","Cold_Turb3","
       ↪ Cold_Turb4","Cold_Turb5","Hot_Turb1","Hot_Turb2","Hot_Turb3","
       ↪ Hot_Turb4","Hot_Turb5"};
    String[] boundaryNames = {"Default [Cold_Turb1/SolidVolume]","Default [
       ↪ Cold_Turb2/SolidVolume]","Default [Cold_Turb3/SolidVolume]","
       ↪ Default [Cold_Turb4/SolidVolume]","Default [Cold_Turb5/
       ↪ SolidVolume]","Default [Hot_Turb1/SolidVolume]","Default [
       ↪ Hot_Turb2/SolidVolume]","Default [Hot_Turb3/SolidVolume]","
       ↪ Default [Hot_Turb4/SolidVolume]","Default [Hot_Turb5/SolidVolume
```

```
   ↪ ]"};
Simulation simulation_0 = getActiveSimulation();

for (int i=0;i<10;i++){
  /* select the appropriate boundary name, must have consistent naming
     ↪ */
   // Region region_0 = simulation_0.getRegionManager().getRegion("
      ↪ Cold_Turb1");
     Region region_0 = simulation_0.getRegionManager().getRegion(
        ↪ regionNames[i]);



  /* set the PM method to Axisymmetric, sets normal direction to
     ↪ (0,0,1), magnitude to 0, then sets axisymmetric component to
     ↪ PMCoeff field function, derived from "f" */
     PorousViscousResistance porousViscousResistance_0 = region_0.
        ↪ getValues().get(PorousViscousResistance.class);
     porousViscousResistance_0.setMethod(
        ↪ AxisymmetricTensorProfileMethod.class);
     VectorProfile vectorProfile_0 = porousViscousResistance_0.
        ↪ getMethod(AxisymmetricTensorProfileMethod.class).getAxiAxis
        ↪ ();
     vectorProfile_0.getMethod(ConstantVectorProfileMethod.class).
        ↪ getQuantity().setComponents(0.0, 1.0, 0.0);
     ScalarProfile scalarProfile_0 = porousViscousResistance_0.
        ↪ getMethod(AxisymmetricTensorProfileMethod.class).
        ↪ getCrossStreamProfile();
     scalarProfile_0.setMethod(FunctionScalarProfileMethod.class);
     UserFieldFunction userFieldFunction_5 = ((UserFieldFunction)
        ↪ simulation_0.getFieldFunctionManager().getFunction("PMCoeff
        ↪ "));
     scalarProfile_0.getMethod(FunctionScalarProfileMethod.class).
        ↪ setFieldFunction(userFieldFunction_5);
```

```
/* set the interfaces to the A,B,C,D components for User Specified
    ↪ heat transfer*/

    //InterfaceBoundary interfaceBoundary_0 = ((InterfaceBoundary)
        ↪ region_0.getBoundaryManager().getBoundary(boundaryNames[i])
        ↪ );
    InterfaceBoundary interfaceBoundary_0 = ((InterfaceBoundary)
        ↪ region_0.getBoundaryManager().getBoundary(boundaryNames[i])
        ↪ );

    interfaceBoundary_0.getConditions().get(
        ↪ UserWallHeatFluxCoefficientOption.class).setSelected(
        ↪ UserWallHeatFluxCoefficientOption.Type.USER_DEFINED);
    UserWallHeatFluxCoefficientA userWallHeatFluxCoefficientA_0 =
        ↪ interfaceBoundary_0.getValues().get(
        ↪ UserWallHeatFluxCoefficientA.class);
    userWallHeatFluxCoefficientA_0.setMethod(
        ↪ FunctionScalarProfileMethod.class);
    UserFieldFunction userFieldFunction_1 = ((UserFieldFunction)
        ↪ simulation_0.getFieldFunctionManager().getFunction("UserA")
        ↪ );
    userWallHeatFluxCoefficientA_0.getMethod(
        ↪ FunctionScalarProfileMethod.class).setFieldFunction(
        ↪ userFieldFunction_1);

    interfaceBoundary_0.getConditions().get(
        ↪ UserWallHeatFluxCoefficientOption.class).setSelected(
        ↪ UserWallHeatFluxCoefficientOption.Type.USER_DEFINED);
    UserWallHeatFluxCoefficientB userWallHeatFluxCoefficientB_0 =
        ↪ interfaceBoundary_0.getValues().get(
        ↪ UserWallHeatFluxCoefficientB.class);
```

```
userWallHeatFluxCoefficientB_0.setMethod(
   ↪ FunctionScalarProfileMethod.class);
UserFieldFunction userFieldFunction_2 = ((UserFieldFunction)
   ↪ simulation_0.getFieldFunctionManager().getFunction("UserB")
   ↪ );
userWallHeatFluxCoefficientB_0.getMethod(
   ↪ FunctionScalarProfileMethod.class).setFieldFunction(
   ↪ userFieldFunction_2);

interfaceBoundary_0.getConditions().get(
   ↪ UserWallHeatFluxCoefficientOption.class).setSelected(
   ↪ UserWallHeatFluxCoefficientOption.Type.USER_DEFINED);
UserWallHeatFluxCoefficientC userWallHeatFluxCoefficientC_0 =
   ↪ interfaceBoundary_0.getValues().get(
   ↪ UserWallHeatFluxCoefficientC.class);
userWallHeatFluxCoefficientC_0.setMethod(
   ↪ FunctionScalarProfileMethod.class);
UserFieldFunction userFieldFunction_3 = ((UserFieldFunction)
   ↪ simulation_0.getFieldFunctionManager().getFunction("UserC")
   ↪ );
userWallHeatFluxCoefficientC_0.getMethod(
   ↪ FunctionScalarProfileMethod.class).setFieldFunction(
   ↪ userFieldFunction_3);

interfaceBoundary_0.getConditions().get(
   ↪ UserWallHeatFluxCoefficientOption.class).setSelected(
   ↪ UserWallHeatFluxCoefficientOption.Type.USER_DEFINED);
UserWallHeatFluxCoefficientD userWallHeatFluxCoefficientD_0 =
   ↪ interfaceBoundary_0.getValues().get(
   ↪ UserWallHeatFluxCoefficientD.class);
userWallHeatFluxCoefficientD_0.setMethod(
   ↪ FunctionScalarProfileMethod.class);
```

```
        UserFieldFunction userFieldFunction_4 = ((UserFieldFunction)
           ↪ simulation_0.getFieldFunctionManager().getFunction("UserD")
           ↪ );
        userWallHeatFluxCoefficientD_0.getMethod(
           ↪ FunctionScalarProfileMethod.class).setFieldFunction(
           ↪ userFieldFunction_4);
    }
  }
}
```

E.6 compileANNLink - Script

A simple BASH script that includes the appropriate compiler call.

```
#!/bin/bash


echo "compiling the shared object (.so) library for STAR-CCM+, mixed
   ↪ precision"
echo "************************************************"
echo "DO NOT USE IN THE DOUBLE PRECISION (-R8) VERSION"
echo "************************************************"
gcc -fPIC -shared -O3 -ggdb -Wall -Wformat-security -Wfloat-equal -Wshadow
   ↪  -Wpointer-arith -Wcast-qual -Wsign-compare -ansi -I../src/ -I../src
   ↪ /include ../src/floatfann.c *.c -o ANN-Fd-Nu.so -lm
```

E.7 uclib.c

The .c file required by STAR-CCM+ to initialise user functions and the associated input/output variables.

```
#include "uclib.h"

void FdInterpVal(Real*, int, CoordReal*, CoordReal*, CoordReal*);
void NuInterpVal(Real*, int, CoordReal*, CoordReal*, CoordReal*);

void
USERFUNCTION_EXPORT uclib()
{
  /* Register user functions here */
  ucfunc(FdInterpVal, "ScalarFieldFunction", "FdInterpVal");
  ucarg(FdInterpVal, "Cell", "$Re", sizeof(CoordReal));
  ucarg(FdInterpVal, "Cell", "$Pr", sizeof(CoordReal));
  ucarg(FdInterpVal, "Cell", "$flowAngle", sizeof(CoordReal));

  ucfunc(NuInterpVal, "ScalarFieldFunction", "NuInterpVal");
  ucarg(NuInterpVal, "Cell", "$Re", sizeof(CoordReal));
  ucarg(NuInterpVal, "Cell", "$Pr", sizeof(CoordReal));
  ucarg(NuInterpVal, "Cell", "$flowAngle", sizeof(CoordReal));

}
```

E.8 uclib.h

The header file required by STAR-CCM+ to define and initialize the variable types.

```
#ifndef UCLIB_H
#define UCLIB_H
#ifdef DOUBLE_PRECISION
typedef double Real;
#else
typedef float Real;
#endif
typedef double CoordReal;

#ifdef __cplusplus
extern "C" {
#endif
#if defined(WIN32) || defined(_WINDOWS) || defined(_WINNT)
# define USERFUNCTION_EXPORT __declspec(dllexport)
# define USERFUNCTION_IMPORT __declspec(dllimport)
#else
# define USERFUNCTION_EXPORT
# define USERFUNCTION_IMPORT
#endif

extern void USERFUNCTION_IMPORT ucarg(void *, char *, char *, int);
extern void USERFUNCTION_IMPORT ucfunc(void *, char *, char *);
extern void USERFUNCTION_IMPORT ucfunction(void *, char *, char *, int,
   ↪ ...);

void USERFUNCTION_EXPORT uclib();
#ifdef __cplusplus
}
#endif
#endif
```

E.9 ANNValue.c

The artificial neural network code used for the calculation of the network used for turbulizer flow library lookup and calculation.

```
#include "uclib.h"
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include "fann.h"
/* -------------------------------------------------------------- */
/* Read a scalar value from CSV file and provide a field function */
/* File format header: "X","Y","Z","value" */
/* First line needs to be header file! */
/* All values need to be separated by commas (,) only! */
/* Do not use whitespace characters within the file. */
/* -------------------------------------------------------------- */


/* -------------------------------------------------------------- */
/* There are two different types of interpolation: */
/* Interpolation is performed by tri-linear methods */
/* -------------------------------------------------------------- */


/* -------------------------------------------------------------- */
/* Debug output can be activated with the flag: */
/* debug_output == 1 */
/* -------------------------------------------------------------- */



  void USERFUNCTION_EXPORT
```

```
FdInterpVal(Real *result, int size, CoordReal *Re, CoordReal *Pr,
    ↪ CoordReal *flowAngle)
{

  /* ------------------ Set these flags as needed -------------- */

  /*int Coordinate = 2; */
  int debug_output = 0; /* 1 == display debug information */

  /* Linux: char filename[] = "/users/home/username/Value.csv"; */
  /* Windows: char filename[] = "C:\\Temp\\Value.csv"; */
  /* ------------Set the flags above as needed ----------------- */


  /* ------ Immediate exit for parallel runs -------------------*/
  /* When running in parallel, it might happen that the user */
  /* code does not need to be evaluated for any cells */
  /* of a particular partition. We can exit immediately in */
  /* this case. */
  if (size==0)
    {
      if (debug_output==1)
        {
          printf("No work to do on this partition. Exiting.... \n");
          fflush(stdout);
        }
      return;
    }


    int icell = 0;
      if(debug_output == 1) printf("starting the ANN routine\n");
      struct fann *ann;
      /*struct fann_train_data *data;*/
      /*printf("Creating network.\n");*/

      ann = fann_create_from_file("/working/andrew.buckrell/
          ↪ AndrewResearch/2018/2018-07-31-ValidationCasesNN-Instability/
          ↪ instabilityTest/STAR-Code/TestDirectory/scaling_fd.net");
      if(!ann)
      {
          printf("Error creating ann --- ABORTING.\n");
          return;
      }

      /*printf("Testing network.\n");*/
      /*data = fann_read_train_from_file("fd_test.data");*/

  float x;
  float y;
  float z;

  for (icell = 0; icell != size; ++icell)
  {
      fann_type *calc_out;
      x =7.0; /*Pr[icell]; /*Pr -- InputCoord[0][icell]*/
      y = Re[icell]; /*Re -- InputCoord[1][icell]*/
      z = flowAngle[icell]*180/3.14159; /*Angle*/
      float inputData[3] = {x,y,z};
      /*printf("input value for %f %f %f is \n", inputData[0],inputData
          ↪ [1],inputData[2]);*/

      fann_reset_MSE(ann);
      fann_scale_input( ann, inputData );
      calc_out = fann_run( ann,inputData);
```

```
       fann_descale_output( ann, calc_out );
       /*printf("Result %f original %f error %f\n", calc_out[0], data->
          ↪ output[i][0], (float) fann_abs(calc_out[0] - data->output[i
          ↪ ][0])); */
       /*fprintf(outputData, "%f %f \n", data->output[i][0],(float)
          ↪ fann_abs(calc_out[0] - data->output[i][0]));*/
       result[icell]=calc_out[0] ;
       /*printf(" %f\n",calc_out[0]);*/

       /*printf("value for cell %i is %f", icell, calc_out[0]);*/



    }


       /*fann_print_connections(ann);*/
       /*fann_print_parameters(ann);*/

       /*printf("Cleaning up.\n");*/
       /*fann_destroy_train(data);*/
       fann_destroy(ann);
       /*printf("Closing output file. \n");*/
       /*fclose(outputData);*/
       fflush(stdout);



  }

void USERFUNCTION_EXPORT
  NuInterpVal(Real *result, int size, CoordReal *Re, CoordReal *Pr,
     ↪ CoordReal *flowAngle)
```

```
{

  /* ------------------ Set these flags as needed -------------- */

  /*int Coordinate = 2; */
  int debug_output = 0; /* 1 == display debug information */

  /* Linux: char filename[] = "/users/home/username/Value.csv"; */
  /* Windows: char filename[] = "C:\\Temp\\Value.csv"; */
  /* ------------Set the flags above as needed ----------------- */


  /* ------ Immediate exit for parallel runs -------------------*/
  /* When running in parallel, it might happen that the user */
  /* code does not need to be evaluated for any cells */
  /* of a particular partition. We can exit immediately in */
  /* this case. */
  if (size==0)
    {
      if (debug_output==1)
        {
          printf("No work to do on this partition. Exiting.... \n");
          fflush(stdout);
        }
      return;
    }


    int icell = 0;
      /*printf("starting the ANN routine\n");*/

      struct fann *ann;
      /*struct fann_train_data *data;*/
      /*printf("Creating network.\n");*/

      ann = fann_create_from_file("/working/andrew.buckrell/
          ↪ AndrewResearch/2018/2018-07-31-ValidationCasesNN-Instability/
          ↪ instabilityTest/STAR-Code/TestDirectory/scaling_nu.net");
      if(!ann)
      {
          printf("Error creating ann --- ABORTING.\n");
          return;
      }

      /*printf("Testing network.\n");*/
      /*data = fann_read_train_from_file("fd_test.data");*/

  float x;
  float y;
  float z;

  for (icell = 0; icell != size; ++icell)
  {
      fann_type *calc_out;
      x = 7.0; /*Pr[icell];*/ /*Pr -- InputCoord[0][icell]*/
      y = Re[icell]; /*Re -- InputCoord[1][icell]*/
      z = flowAngle[icell]*180/3.14159; /*Angle*/
      float inputData[3] = {x,y,z};
      /*printf("output value for %f %f %f is ", inputData[0],inputData
          ↪ [1],inputData[2]);*/

      fann_reset_MSE(ann);
      fann_scale_input( ann, inputData );
      calc_out = fann_run( ann,inputData);
      fann_descale_output( ann, calc_out );
      /*printf("Result %f original %f error %f\n", calc_out[0], data->
          ↪ output[i][0], (float) fann_abs(calc_out[0] - data->output[i
          ↪ ][0])); */
      /*fprintf(outputData, "%f %f \n", data->output[i][0],(float)
          ↪ fann_abs(calc_out[0] - data->output[i][0]));*/
      result[icell]=calc_out[0] ;
      /*printf(" %f\n",calc_out[0]);*/

      /*printf("value for cell %i is %f", icell, calc_out[0]);*/



  }



      /*fann_print_connections(ann);*/
      /*fann_print_parameters(ann);*/

      /*printf("Cleaning up.\n");*/
      /*fann_destroy_train(data);*/
      fann_destroy(ann);
      /*printf("Closing output file. \n");*/
      /*fclose(outputData);*/
      fflush(stdout);



   }


/* Nusselt number interpolation scheme */
/* Code currently mirrors fd inerpolation, */
/* but just changes the output variable */
/* as well as input filename */
/* */
/* */
/* */
/* */
/* */
/* */
/* */
```