IMPERIAL

Bayesian Neural Network (BNN) Machine Learning (ML) Surrogate Modelling Methods for Simulating Turbulent Flow within Nuclear Power Plant (NPP) T-Junction Pipes

Primary Academic Supervisor:

Dr Michael J. Bluck

Co-Academic Supervisor:

Dr Matthew D. Eaton

Associate Academic Supervisors:

Dr Yu Duan

Dr Lisa Lampunio

Author:

Wong Yew Hoe

Department of Mechanical Engineering

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Mechanical Engineering of Imperial College London and the Diploma of Imperial College London

May 2025

Abstract

The integration of nuclear power plants (NPPs) into modern smart grids poses new challenges, particularly in ensuring component reliability under increasingly dynamic operating conditions. One area of concern is thermal fatigue in T-junctions within the primary circuit, which is a critical issue for nuclear safety that remains poorly addressed by current modelling practices. Existing approaches often focus on isolated components or employ low-fidelity models, limiting their ability to capture the complex flow-induced thermal stresses that drive thermal fatigue. This research addresses these limitations through a three-part investigation. First, mesh discretisation error within the computational models are thoroughly studies and estimated. This provides significant insights into how model accuracy can be improved, while concurrently providing model uncertainty estimates. The second would be developing a high-fidelity simulation framework to model turbulent flow in various T-junction configurations. Using Power Spectral Density (PSD), Proper Orthogonal Decomposition (POD), and Spectral POD (SPOD), the study identifies dominant flow structures responsible for thermal fatigue. These analyses provide detailed insights into unsteady flow behaviours that are often overlooked in traditional models. Results reveal underlying flow structures within turbulent flow that potentially contributes to thermal fatigue phenomena. Third, machine learning techniques are explored to recover turbulent flow features in these geometries. Several neural network architectures such as deep neural networks (DNNs), physics-informed neural networks (PINNs), Bayesian neural networks (BNNs), and Bayesian PINNs (BPINNs), are evaluated as surrogate models for flow field reconstruction. Results show that BNNs achieve the best performance, providing both accurate predictions and quantified uncertainty. This positions it as a valuable component in the broader effort to enhance predictive capabilities in nuclear thermal fatigue analyses.

Declaration of Originality

I hereby declare that the work contained in this thesis is my own unless stated otherwise. In such cases the work will be appropriately referenced.

Copyright Declaration

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-Non-Commercial 4.0 International Licence (CC BY-NC). Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that: you credit the author and do not use it, or any derivative works, for a commercial purpose. When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes. Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

Acknowledgements

I would like to thank my principal academic supervisor Dr Michael J. Bluck, my co-academic supervisor Dr Matthew Eaton, associate academic supervisors Dr Lisa Lampunio and Dr Yu Duan for their utmost support and advice during my PhD candidature. I would like to express my sincere gratitude to Professor Chung Keng Yeow and Dr Garvin Mak Kar Heng for recognizing my potential and supporting my academic journey, the Singapore Nuclear Research and Safety Institute (SNRSI) and the National Research Foundation (NRF) for funding and supporting my PhD project. Lastly, I am profoundly grateful to my parents and younger brother for their unwavering love, encouragement, and belief in me. Their constant support has been my greatest source of strength throughout this journey.

Contents

Abstract **2**

1 Introduction **27**

1.1 The Nuclear Landscape of Singapore and Southeast Asia 27

1.2 Thermal Stripping and High-cycle Thermal Fatigue Damage within T-junctions of NPPs 30

1.3 Review of Experimental and Computational Studies on Turbulent Thermal Mixing within T-junctions 34

1.4 Review of current work done for Surrogate Modelling 43

1.5 Overview of Literature Survey 49

1.6 Proposed Methodology for Modelling & Simulations and Surrogate Modelling . . 50

1.7 Project Scope 51

1.8 Thesis Structure 52

2 The Mathematical and Computational Modelling of Turbulent Flows **55**

2.1 A Primer on Turbulent Flows 55

2.2 Turbulence Modelling 57

2.2.1 Introduction to Turbulence 57

2.2.2 Turbulent Boundary Layer 58

2.2.3 Types of Turbulence Models 59

2.3 Finite Volume Method (FVM) for Spatial Discretization 65

2.3.1 Diffusion Term 66

2.3.2 Convection Term (MUSCL 3rd Order Scheme) 67

2.3.3 Time Discretization 67

2.4 Mesh Discretization . . . 68
2.4.1 Wall Treatment Methods . . . 69
2.5 Concluding Remarks . . . 70

3 Description of Computational Fluid Dynamics Model 73
3.1 Model Settings . . . 74
3.1.1 Geometrical and numerical settings . . . 74
3.1.2 Boundary Conditions . . . 77
3.2 Mesh Sensitivity Analysis (Conventional) . . . 77
3.2.1 Results and Discussion . . . 78
3.3 Mesh Sensitivity Analysis (Adaptive Mesh Refinement) . . . 82
3.3.1 Mathematics of Conventional and Goal-based error estimates for AMR . 83
3.3.2 Results for Gradient-based Refinement (Velocity) . . . 88
3.3.3 Results for Dual-weighted error-based Refinement (Velocity) . . . 91
3.3.4 Results for Gradient-based Refinement (Temperature) . . . 95
3.3.5 Results for Dual-weighted error-based Refinement (Temperature) . . . 100
3.4 Conclusion . . . 104

4 Description of Machine Learning Model 107
4.1 Deep Neural Network (DNN) . . . 107
4.2 Physics Informed Neural Network (PINN) . . . 109
4.3 Bayesian Neural Network (BNN) . . . 113
4.4 Bayesian Physics Informed Neural Network (BPINN) . . . 115
4.5 Concluding Remarks . . . 118

5 Effects of Momentum Ratio and Reynolds Number on Turbulent Flows 119
5.1 Description of CFD Models and BCs . . . 119
5.2 Results and Discussion . . . 120
5.2.1 Mean Flow Field . . . 122
5.2.2 Turbulent QoIs . . . 133
5.2.3 Fast Fourier Transform (FFT) Frequency Analysis . . . 139
5.3 Conclusions . . . 142

6 A POD and SPOD Study on the Effects of Different Momentum Ratios and

Reynolds Number in a T-junction with an Upstream Elbow **145**

6.1 A Primer on Proper Orthogonal Decomposition (POD) . . . 146

6.2 Description of CFD Models and BCs . . . 148

6.3 Results and Analysis . . . 149

6.3.1 Mean and Instantaneous Flow Structures . . . 149

6.3.2 POD analysis . . . 154

6.3.2.1 POD Y-Z in-plane analysis . . . 154

6.3.2.2 POD X-Z in-plane analysis . . . 160

6.3.3 SPOD analysis . . . 164

6.3.3.1 SPOD Y-Z in-plane analysis . . . 164

6.3.3.2 SPOD X-Z in-plane analysis . . . 169

6.4 Summary and Conclusions . . . 171

7 Assessment of Neural Network Parameter Selection **175**

7.1 Description of Lid Driven Cavity Problem . . . 176

7.2 Activation Functions . . . 178

7.3 Network Performance Analysis . . . 181

7.3.1 Number of Hidden Layers . . . 181

7.3.2 Number of Neurons per Hidden Layers . . . 182

7.3.3 Choice of Activation Functions . . . 182

7.3.4 Number of Sample Points Required for Recovery . . . 183

7.3.5 Training Time and Quantitative Error Metric . . . 183

7.4 Conclusion . . . 188

8 Deep Neural Networks as Surrogate Models for Flow Field Reconstruction **191**

8.1 Description of Computational Model . . . 192

8.2 Neural Network Settings . . . 193

8.3 Network Sensitivity Analysis . . . 196

8.3.1 Quantitative Loss . . . 196

8.3.2 Training Time . . . 197

8.4 Flow Field Recovery Comparison for Different Neural Networks . . . 199

8.4.1 Lid Driven Cavity Results . . . 200

8.4.2 Multi Cavitated Flow . . . 203

8.5 Conclusion 204

9 Future Work **207**

9.1 Coupling Surrogate Models with Whole-Plant Simulators 207

9.2 Uncertainty Quantification (UQ) and Experimental Validation 208

9.3 Further Development in Surrogate Modelling 208

10 Conclusion **211**

10.1 Summary of Research Efforts and Concluding Remarks 211

List of Figures

1 Crack growth locations along the upstream elbows of (a) RHRS mixing area of CIVAUX-1, adapted from (Laribou and Elbasset (2021)) and (b) Tsuruga-2, adapted from (ANS (1999)) 32

2 (a) Sodium expansion tank used in PHENIX reactor and (b) Crack found within the expansion tank; adapted from Gelineau et al. (2001) 32

3 (a) T-junction configuration used in PHENIX and (b) Location of crack detected along the upstream elbow; respectively adapted from (Gelineau et al. (2001)) . . 33

4 Types of jets found in within the mixing region of a T-junction, adapted from Kimura et al. (2010) 36

5 T-junction test rig used in the WATLON facility, adapted from Kimura et al. (2014) 37

6 (a) Schematic of FATHER, (b) Facility view of FATHER, (c) Endurance T-junction test rig and (d) Close-up of thermal hydraulic T-junction test rig, adapted from (Brillard et al. (2007)) 38

7 Schematic diagram of FATHRINO facility with emphasis on the T-junctions, adapted from (Braillard et al. (2018)) 39

8 Vattenfall T-junction used for generating test data adapted from (Odemark et al. (2009)) 40

9 Close-up of the sharp and round corner T-junction geometries used for the MOTHER project, adapted from (Braillard et al. (2018)) 40

10 Control volume in unstructured meshes adapted from Dincer and Ezan (2018) . 65

11 Three commonly used mesh types in CFD softwares: (a) Hexahedral, (b) Tetrahedral, (c) Polyhedral adapted from Michalcová and Kotrasová (2020) 69

12 Side elevation view of the computational domain for the T-junction test case with upstream bend. 76
13 Mesh domains in the main pipe with a base size of (a) 5 mm, (b) 3 mm, (c) 2 mm. 78
14 Fully developed flow profile based on mean velocity at (a) branch inlet and (b) main inlet. 79
15 Law of the wall plot for different mesh size (universal velocity profile) 80
16 Cross sectional locations of interest for analysis, using parameters in Table 2 as the base test case. 80
17 Normalised mean streamwise velocity for the different mesh resolutions. 80
18 The contour of IDDES-SST blending function (utilised to estimate of RANS and LES regions) for mesh sensitivity simulation with base mesh size of (a) 2 mm, (b) 3 mm & (c) 5mm 81
19 (a) Contour of magnitude of Grad of Velocity on the YZ symmetry plane, (b) Histogram of r$r_{k,grad}$ against number of cells 90
20 Refinement Level for Gradient Based Refinement 91
21 Streamwise velocity profiles obtained with adaptive mesh refinement ($r_{k,grad}$) high fidelity model (2mm mesh) and LES model (LES mesh) 91
22 Contour of adjoint error estimate of volume uniformity of velocity on symmetry plane 92
23 Histograms of different refinement parameters against number of cells (a) $r_{k,mean}$, (b) $r_{k,max}$, (c) $r_{k,total}$, (d) $r_{k,rms}$ 92
24 Refinement Levels based on various adaptive mesh refinement metric for (a) $r_{k,mean}$, (b) $r_{k,max}$, (c) $r_{k,total}$, (d) $r_{k,rms}$ 93
25 Streamwise velocity profiles obtained with various adaptive mesh refinement models, high fidelity model (2mm mesh) and LES model (LES mesh). 95
26 Locations of Line Probes Within Flow Domain 96
27 (a) Contour of magnitude of Grad of Temperature on the YZ symmetry plane, (b) Histogram of $r_{k,grad}$ against number of cells 97
28 (a) Contour of magnitude of Grad of Temperature on the YZ symmetry Refinement Level for Gradient Based Refinement 98
29 Mean Temperature plots along the X-Z Plane for probe at locations (a) 0.007m, (b) 0.009 and (c) 0.014m away from the upper wall 99

30 Contour of adjoint error estimate of volume uniformity of temperature on symmetry plane . . . 100
31 Histograms of different refinement parameters against number of cells (a)$r_{k,mean}$, (b)$r_{k,max}$ and (c) $r_{k,total}$, (d)$r_{k,rms}$ . . . 101
32 Refinement Levels based on various adaptive mesh refinement metric for (a)$r_{k,mean}$, (b)$r_{k,max}$ and (c) $r_{k,total}$, (d)$r_{k,rms}$ . . . 103
33 Mean Temperature plots along the X-Z Plane for probe at locations (a) 0.007m, (b) 0.009 and (c) 0.014m away from the upper wall . . . 104

34 A Schematic Diagram of a PINN with the Navier Stokes Equation as the PDE of Choice. . . . 112

35 Selected locations of cross sections with (a) point probes used for the analysis (b) a string of linear probes spanning from the top to bottom of the flow is used for sampling the turbulent QoI. . . . 122
36 Contours of normalised mean streamwise velocity U (m/s) for Cases (a) 1, (b) 2, (c) 3, (d) 4, and (e) 5. . . . 124
37 (a) Normalised mean streamwise velocity along the X-Z plane to show where the recirculation zone starts, (b) Probe line (in red) within the prism layer used sampling. Note that this plot is zoomed-in and does not show the full range of velocity values. . . . 126
38 Streamwise wall shear stress in the upper wall. Note that $x/D_m = 0$ corresponds to the exit of the T-junction. . . . 128
39 Profiles of mean streamwise velocity magnitude $\langle V_x \rangle$through the main pipe centre..128
40 (a) Cross-sectional velocity streamlines at selected locations . . . 130
41 Reynolds Stress along the components (a) $\langle uu \rangle^*$ (b) $\langle uw \rangle^*$ at various locations. . 135
42 Normalised turbulent kinetic energy (TKE^*) at prescribed locations. . . . 135
43 Turbulent heat flux in the streamwise and vertical spanwise directions at selected locations. . . . 137
44 Circumpolar plots of temperature variance at selected locations. . . . 138
45 Circumpolar plots of turbulent heat flux in the (a) streamwise and (b) vertical spanwise directions at selected locations. . . . 139

46 Power Spectral Density of the frequency based on fluctuating velocity magnitude downstream of the T-junction. 141

47 Mean flow structures at (a) a horseshoe-like recirculation at the exit of the elbow, (b) the recirculation region after the T-junction, (c) the recirculation region before the T-junction. 149

48 Instantaneous (left) and mean (right) flow field on the X-Z symmetric Plane (Side View) and Y-Z plane (cross-sectional view). 152

49 Evolution of Dean vortices based on velocity streamlines and mean velocity . . . 153

50 (a) Contours of $Q-2$ Criterion Value of $150s^{-2}$ Coloured with Streamwise Component of Velocity (b) Zoomed-in view at the T-junction exit for each case . . . 153

51 POD energy spectra based on V_z' at various cross-sectional locations; Vertical axis represents the energetic content in terms of percentage, horizontal axis represents the mode number. 156

52 Welch's estimate of the power spectral density of the first 5 fluctuating modes of V_z at each location. 156

53 POD Mode 1 of Y-Z in-plane velocity components V_z' for $x/D_m = -1.36, 0, 0.292$; the contours represent the amplitude of mode 1 of V_z. 159

54 POD Mode 2 of Y-Z in-plane velocity components V_z for $x/D_m = -1.36, 0, 0.292$; the contours represent the amplitude of mode 1 of V_z 159

55 Three instantaneous views at different phase of the cross-sectional velocity after the bend from the recomposed velocity field. 160

56 POD energy spectra along the X-Z plane based on V_x'; Vertical axis represents the energetic content in terms of percentage, horizontal axis represents the mode number. 160

57 POD Mode 1 of X-Z in-plane velocity components of V_x and V_z; the contours represent the amplitude of mode 1 of V_z while the streamlines represent vectors consisting of V_x and V_z 162

58 POD Mode 2 of X-Z in-plane velocity components of V_x and V_z; the contours represent the amplitude of mode 1 of V_z while the streamlines represent vectors consisting of V_x and V_z 163

59 SPOD energy spectra for the first four modes based on V_z', at various cross-sectional locations 165

60 SPOD Mode 1 of Y-Z in-plane velocity components V_z for $x/D_m = -1.36, 0, 0.292$ at dominant frequency of 5.36 Hz . . . 168
61 SPOD Mode 1 of Y-Z in-plane velocity components V_z for $x/D_m = -1.36, 0, 0.292$ at dominant frequency of 10.71 Hz . . . 169
62 SPOD Mode 1 of X-Z in-plane velocity components of V_x and V_z at 5.36 Hz; the contours represent the amplitude of mode 1 of V_z while the streamlines represent vectors consisting of V_x and V_z . . . 171

63 Geometry of the Lid-Driven Cavity and its Solution Comprised of Streamlines (Red-dashed Line Represents the Location of Interest for Analysis in Section 3) 177
64 Graphical Representation of Activation Functions . . . 180
65 L^2 Error (Loss) Plot During Training for Different Number of Hidden Layers; (note that this is similar for other assessment metrics where L-BFGS greatly reduces the loss after 5000 iterations). . . . 185
66 Scatter Plot Using (a) 100, (b) 500, (c) 1000, (d) 3000 Sample Points for Testing. 185
67 Velocity Profile Plots Based on Predictions for (a) Different Number of Hidden Layers, (b) Number of Neurons per Hidden Layer, (c) Activation Functions, (d) Number of Sample Points . . . 186
68 Training Time (s) for Different Network Settings . . . 187
69 Error Metric Plots for Different Network Settings (Quantitative) . . . 188

70 Computational Domain of Multi-cavitated Flow. . . . 193
71 Loss Values for Various Network Settings. . . . 197
72 Training Time for Various Network Settings. . . . 199
73 Recovered Flow Field a Lid Driven Cavity Based on Predictions by (b) DNN, (c) PINN, (d) BNN & (e) BPINN, Compared Against (a) CFD Solution . . . 202
74 Recovered Flow Field for Flow Over Multiple Cavities Based on Predictions by (b) DNN, (c) PINN, (d) BNN & (e) BPINN, Compared Against (a) CFD Solution 204

List of Tables

1 Characteristics for various types of thermal fatigue 30

2 Classification of branch jet in T-junction with an upstream elbow (Hosseini et al. (2008)). 74

3 Characteristics of the T-junction with various M_R and Reynolds number. 76

4 Cell count for velocity gradient-based refinement parameters 89

5 Cell counts for different refinement parameters based on dual weighted velocity error 93

6 Cell count for temperature gradient-based refinement parameters 96

7 Cell counts for different refinement parameters based on dual-weighted error of temperature 101

8 Likelihood and Priors Used for a Bayesian Neural Network 114

9 Likelihood and Priors Used for a Bayesian Physics Informed Neural Network. . . 116

10 Activation Functions and their Mathematical Expressions 179

11 Hyperparameters for each model and case 200

List of Algorithms

1 DNN Algorithm 109
2 PINN Algorithm 113
3 BNN Algorithm 115
4 BPINN Algorithm 117
5 Adaptive HMC 194

Nomenclature

Δs	Near-wall grid spacing
$\dot{M}_{\text{branch}}$	Mass flow rate of branch pipe flow
μ	Dynamic viscosity
ϕ	Spatial basis functions
ρ_b	Branch pipe fluid density
ρ_m	Main pipe fluid density
σ	Activation function
τ	Wall shear stress
A	Time coefficients
b	Neural network bias
C	Covariance matrix
D	Main pipe diameter
D_b	Branch pipe diameter
e	Step size
f	Sampling frequency
f_n	Nyquist frequency

k, TKE	Turbulent kinetic energy
M	Mass matrix
M	Momentum ratio
M_{branch}	Branch pipe momentum
M_{main}	Main pipe momentum
R	Radius of curvature
Re	Reynolds number of the branch inlet
S	Snapshot matrix
T_{rms}	Root Mean Square of the temperature
T_{var}	Variance of the temperature
U	Velocity component in the x direction
u	Fluctuating component of velocity in the x direction
U_{bulk}	Bulk velocity
V	Velocity component in the y direction
v	Fluctuating component of velocity in the y direction
v_b	Branch pipe velocity
V_m	Main pipe inlet bulk velocity
W	Velocity component in the z direction
w	Fluctuating component of velocity in the z direction
w	Neural network weight
ADAM	Adaptive Moment Estimation

AI	Artificial Intelligence
AIPTEC	Artificially Induced Periodical Temperature Changes
AMR	Adaptive Mesh Refinement
BC	Boundary Conditions
BERT	Bidirectional Encoder Representations from Transformers
BNN	Bayesian Neural Network
BPINN	Bayesian Physics Informed Neural Network
CFD	Computational Fluid Dynamics
CHT	Conjugate Heat Transfer
CNN	Convolutional Neural Network
DDES	Delayed Detached Eddy Simulation
DES	Detached Eddy Simulation
DNN	Deep Neural Network
DNS	Direct Numerical Simulation
FNA	Fukushima Nuclear Accident
FNN	Feed-forward Neural Network
FSI	Fluid Structure Interaction
GIS	Grid Induced Separation
GNN	Graph Neural Networks
GP	Gaussian Process
GPT	Generative Pretrained Transformer

HCTF	High Cycle Thermal Fatigue
HMC	Hamiltonian Monte Carlo
IAEA	International Atomic Energy Agency
IC	Initial Conditions
IDDES	Improved Delayed Detached Eddy Simulation
LDV	Laser Doppler Velocimetry
LES	Large Eddy Simulation
LRA	Low Rank Approximation
LSTM	Long Term Short Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MCMC	Markov Chain Monte Carlo
ML	Machine Learning
MLP	Multi Layer Perceptron
MSD	Modelled Stress Depletion
MSE	Mean Square Error
MSLL	Mean Square Log Loss
NLP	Natural Language Processing
NPP	Nuclear Power Plant
PCA	Principal Component Analysis
PCE	Polynomial Chaos Expansion

PDE	Partial Differential Equation
PINN	Physics Informed Neural Network
PIV	Particle Image Velocimetry
POD	Proper Orthogonal Decomposition
PSD	Power Spectral Density
PWR	Pressurised Water Reactor
RANS	Reynolds Averaged Navier-Stokes
ReLU	Rectified Linear Unit
RHRS	Residual Heat Removal System
RMPSE	Root Mean Percentage Square Error
RMSE	Root Mean Square Error
RMSprop	Root Mean Square Propagation
RNN	Recurrent Neural Network
ROM	Reduced Order Modelling
SELU	Scaled Exponential Linear Unit
SFR	Sodium Fast Reactor
SGD	Stochastic Gradient Descent
SGS	Sub Grid Scale
SILU	Sigmoid Linear Unit
SMR	Small Modular Reactor
SPOD	Spectral Proper Orthogonal Decomposition

SST	Shear Stress Transport
SVM	Support Vector Machine
TL	Transfer Learning
UQ	Uncertainty Quantification
WMLES	Wall Modelled Large Eddy Simulation

Chapter 1

Introduction

1.1 The Nuclear Landscape of Singapore and Southeast Asia

Throughout history, technological advancement has consistently transformed societal structures, influencing both human needs and priorities. As civilizations develop, their energy demands increase correspondingly, resulting in progressively higher levels of consumption. This escalating dependence on energy, however, has significant environmental consequences, particularly due to the continued reliance on fossil fuels. In light of these challenges, the global community has increasingly turned to cleaner and more sustainable energy alternatives to mitigate environmental degradation and promote long-term ecological stability.

Among the available renewable and low-carbon energy sources, such as solar, wind and nuclear, nuclear power remains one of the most efficient and viable options. Recognizing its potential to provide a stable, large-scale energy supply with minimal carbon emissions, many countries have increased their investments in nuclear research and development. Despite this growing interest, the integration of nuclear energy into existing power grids presents several challenges. Singapore, as a highly urbanized and land-scarce nation, faces unique obstacles in considering nuclear energy as part of its long-term energy strategy. Key barriers include geographical limitations, political safety concerns due to its proximity to neighboring countries, and the stigma surrounding nuclear energy.

Singapore is a small country located in Southeast Asia, covering just 700 square kilometres, with much of its area expanded through land reclamation. It is complemented by the fact that it is just one kilometer from Johor, Malaysia's southernmost city, and the closest Indonesian island, Batam, lies just 20 kilometers to the south. A key factor in Singapore's energy security is its political relationships with these neighbouring countries. About 80% of Singapore's electricity is generated from natural gas supplied by Malaysia and Indonesia, while its oil imports come from the Middle East. The idea of Singapore adopting nuclear power could potentially cause unease among its neighbours and affect diplomatic relations.

Despite its size, Singapore's total electricity consumption is at a whopping 50.8 TWh in 2020 compared to an annual power consumption of 47.5 TWh in 2015 ((EMA) (2023)). This provides an indication of the power consumption in Singapore as well as its increasing demand for electrical power. Since Singapore does not have any natural resources and its location does not favour the production of renewable energy, the idea of adopting nuclear energy for the nation is not new. The dependence on imported natural resources makes it dangerously vulnerable to supply hindrances and inflation. The other major issue is that nuclear energy has always been a public misunderstanding, mainly due to major nuclear disasters such as Fukushima and Chernobyl. We note that the Chernobyl disaster halted future construction of nuclear power plants (NPPs) globally for several years. In research conducted on the Chinese public's attitude towards nuclear energy after the 2011 Fukushima Nuclear Accident (FNA), it was reported that the incident indeed had dynamic effects on land markets within 40km of NPPs in China itself (Zhu et al. (2016-09)).

Nonetheless, continued advancements in nuclear technology, particularly in small modular reactors (SMRs) and next-generation safety systems, may provide viable pathways for Singapore to incorporate nuclear energy into its future energy mix. The International Atomic Energy Agency (IAEA) defines Small Modular Reactors (SMRs) as a generation of new reactors which are designed to generate electric power up to 300MW (IAEA (2016)). SMRs are designed such that it capitalises on its small size to reach certain performance characteristics (Ingersoll (2009)). At the time of writing, there are several SMRs at various stages of development (IAEA (2016); Locatelli et al. (2013); Karol et al. (2015)). SMR designs can be employed in reactors such as: water-cooled reactors, high-temperature reactors, gas-cooled reactors with fast neutron spectrum and molten salt reactors (Locatelli et al. (2013)). In the beginning of the nuclear

power age, NPPs were mostly viewed as a benchmark or base-load source of electricity.

It is because operation of the NPPs at the rated power level is usually more efficient and had less complexity. Its share in the countries' energy mix where they were introduced in was relatively low at first, but it gradually increased as governments recognized the benefits of stable base load power, low greenhouse gas emissions, and long-term energy security. SMRs are known to offer unique benefits, including lower capital costs compared to large NPPs, the ability to be manufactured in factories, and the flexibility of deploying multiple units with diverse designs for the same installed power. This is due to its small size with respect to other types of reactors (Mignacca and Locatelli (2010); Carelli and Ingersoll (2021)). One of the major attractions of SMRs is its modularity. It simplifies construction by dividing the plant into modules which can be manufactured and transported to the site for on-site assembly. This modularisation allows for functional and system testing during the manufacturing and assembly stage, which enables a higher level of parallelism and thus a shorter time (Maronati et al. (2016)). This is one of the many factors which contributes to the popularity of SMRs.

As Singapore considers nuclear energy as a potential solution to its energy security challenges, especially given the global shift towards low-carbon power sources, it must also address the technical complexities and operational demands associated with modern nuclear reactors. One of the critical factors in ensuring the safe and reliable operation of these reactors lies in understanding the mechanical and thermal stresses that their components are subjected to during regular and load-following operations. Thermal striping and high-cycle fatigue, particularly in key components like T-junctions within the primary circuit of pressurized water reactors (PWRs), represent significant engineering concerns. These phenomena can affect the reliability and safety of the plant, making them crucial areas of study for nations like Singapore, which aim to integrate nuclear power into their energy mix.

In the following chapter, these engineering challenges are discussed, focusing on how thermal fluctuations and stress in reactor components, specifically T-junctions, can lead to fatigue and potential failure. Understanding these issues is essential for optimizing the design, operation, and maintenance of NPPs, which is a key step in supporting Singapore's nuclear aspirations. By addressing these technical challenges, Singapore can better position itself to safely adopt nuclear power in its future energy landscape.

1.2 Thermal Stripping and High-cycle Thermal Fatigue Damage within T-junctions of NPPs

There are two major kinds of thermal fatigue that have been observed in T-junctions within NPPs. Thermal stripping, or also known as high-cycle thermal fatigue, is a complex thermo-fluid phenomenon and refers to the fluctuating temperature field on a structure's surface resulting from the turbulent thermal mixing of cold and hot fluids near the surface (Brunings (1982); Wakamatsu et al. (1995); Lee et al. (1999)). On the other hand, cyclic thermo-mechanical stresses, or also known as low-cycle thermal fatigue, arise from repeated cycles of thermal and mechanical loading of structures within a power plant surface (Jones (2003); Coffin Jr (2022)). They are often seen in components exposed to reactor start-up and shutdowns, where temperature fluctuations exist at various points in the system. A table (Table 1) below briefly summarizes the differences between high cycle thermal fatigue and low cycle thermal fatigue. This PhD project is primarily motivated by the challenges associated with high-cycle thermal fatigue, which arises from turbulent thermal mixing and is a key contributor to crack initiation in components such as T-junctions. As high-cycle thermal fatigue represents the central phenomenon under investigation, the following chapters will focus exclusively on this mechanism. As low-cycle thermal fatigue falls outside the scope of this study, it will not be considered in subsequent analyses.

Table 1: Characteristics for various types of thermal fatigue

Types of thermal fatigue	**Thermal Stripping (High cycle thermal fatigue)**	**Cyclic Thermo-mechanical Stress (Low cycle thermal fatigue)**
Causes	Rapid, localized mixing of hot/cold fluids	Repeated heating and cooling cycles
Location	Localized (e.g., T-junctions)	Component-wide (e.g., pipes, vessels)
Frequency for occurrence	High (1–10 Hz)	Low (e.g. < 0.01 Hz)
Amplitude	Low	High
Damage	Surface fatigue, crack initiation	Bulk fatigue, crack propagation
Mechanism	Turbulent temperature fluctuations	Thermal expansion/contraction

As mentioned above, T-junction piping is extremely prone to high cycle thermal fatigue due

to its geometric nature and the mixing of different inlet fluid temperatures and velocities. Explicit information on the amplitudes and frequencies of fluid thermal fluctuations is crucial to prevent possible fatigue failure. The usual methods employed to study and analyze thermal fatigue heavily involve the use of Computational Fluid Dynamics (CFD) simulation to derive the temperature fluctuations which plays a crucial role in structural mechanics analysis (Hannink et al. (2008)). However, at high Reynolds number, a higher order of accuracy for flow predictions is needed, which demand large computational power due to the range of flow scales to be resolved numerically. This led to further work done on the mixing that occurs within them. Most geometries considered in such papers usually consist of a 90 °junction between two near-fully developed flows, through straight pipes (Chapuliot et al. (2005); Courtin (2013); Lampunio et al. (2021, 2022)).

There are many examples of damages caused by such high cycle thermal fatigue. For example, thermal fatigue induced cracks in pipes have been observed in the French PWR CIVAUX 1 NPP in 1998 and the Japanese PWR Tsuruga-2 NPP in 1999 (IAEA (1999); Dahlberg et al. (2011)). CIVAX 1 NPP formed the basis of Cipiere et al's study, which found a through wall crack near a T-junction, within the residual heat removal system (RHRS) (Cipiere and Le Duff (2002)). The crack can be seen in Figure 1(a) where the location of the crack was found on the upstream elbow just downstream of the T-junction (at the exit of the inlet branch), where the temperature difference was 150. As for Tsuruga-2 NPP, a leakage was detected from a crack at one of the elbows of the regenerative heat exchangers, and this is further illustrated in Figure 1(b).

In the French sodium fast reactor (SFR), Phenix, cracks were observed in the sodium expansion tank around the region where coolant was discharged. This was coupled with a through wall crack found within the walls of a T-junction, seen in Figure 2 and Figure 3 respectively (Gelineau et al. (1994, 2001)). Thermal fluctuations resulting from the turbulent mixing of the flow were observed within the frequency range of 0.1 Hz to 3 Hz and this is due to the discharge of hot sodium, located within a recirculation zone inside the tank. In its secondary circuit, a sodium leak was detected within a T-junction where the branch inlet pipe was channeling hot sodium to the main pipe. The assessment confirmed that the observed cracks were not the result of other fatigue mechanisms, such as low-cycle thermal fatigue, but were instead directly linked to the effects of high-cycle thermal loading, highlighting the need for focused monitoring and

mitigation strategies to address this specific form of fatigue in nuclear components.

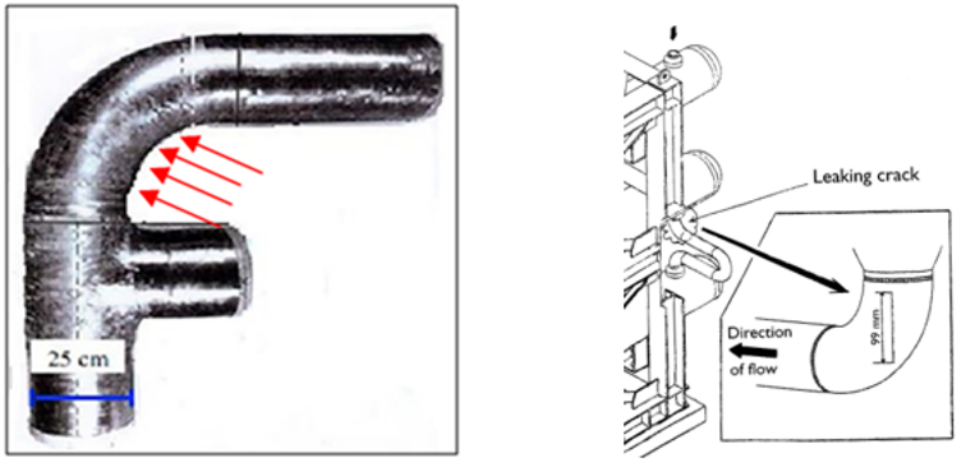


Figure 1: Crack growth locations along the upstream elbows of (a) RHRS mixing area of CIVAUX-1, adapted from (Laribou and Elbasset (2021)) and (b) Tsuruga-2, adapted from (ANS (1999))

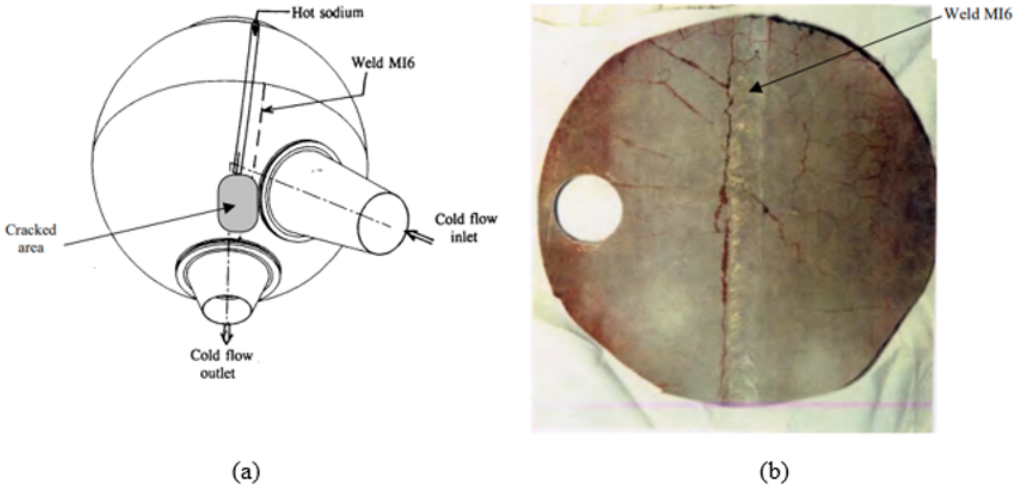


Figure 2: (a) Sodium expansion tank used in PHENIX reactor and (b) Crack found within the expansion tank; adapted from Gelineau et al. (2001)

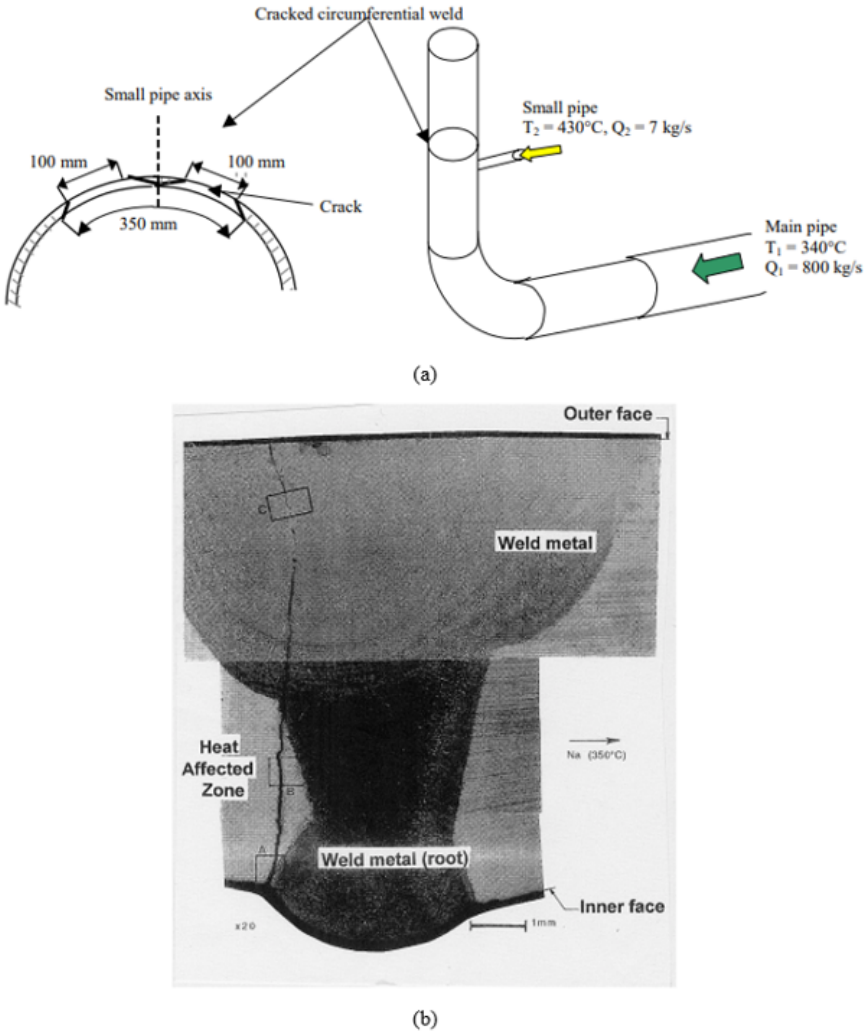


Figure 3: (a) T-junction configuration used in PHENIX and (b) Location of crack detected along the upstream elbow; respectively adapted from (Gelineau et al. (2001))

The same phenomenon was observed in the same SFR reactor, where an experiment was conducted over a four-day period in which a partially closed valve allowed hot sodium to enter an auxiliary pipe in the secondary circuit (Gelineau et al. (1994)). This led to an unexpected thermal mixing zone at a nearby T-junction with a temperature differential of approximately 200°C. This thermal disturbance led to a sodium leak, which was traced to a crack on a circumferential weld located just downstream of the T-junction. Fracture mechanics analysis indicated that under such conditions, temperature oscillations at a frequency of 1 Hz could initiate thermal stripping roughly 30 cm downstream of the junction. At this frequency, only about 500 thermal cycles, accumulated over the four days, would be sufficient to propagate the crack completely through the pipe wall.

The review of thermal stripping and high-cycle thermal fatigue in T-junctions demonstrates that these phenomena have been extensively observed in both experimental and operational contexts, with well-documented cases of material degradation in nuclear piping systems. While prior work has identified the primary mechanisms, such as cyclic thermal loading induced by turbulent mixing layers and characterised their general influence on structural fatigue life, there is limited progress in developing predictive CFD-based frameworks that can resolve the unsteady thermal gradients driving fatigue under realistic reactor operating conditions. In particular, current safety evaluations often rely on semi-empirical design margins rather than fully resolved thermo-fluid simulations capable of capturing the spatial and temporal characteristics of temperature fluctuations at fatigue-critical wall regions. This gap motivates the present research to advance a simulation and modelling framework that integrates high-fidelity CFD with targeted mesh refinement and surrogate modelling, enabling accurate, efficient, and physically grounded prediction of thermal fatigue risk in nuclear-grade T-junction geometries.

1.3 Review of Experimental and Computational Studies on Turbulent Thermal Mixing within T-junctions

This section provides a systematic and in-depth review of the current work done in both experimental and computational aspects as growing research aim to address the issue of thermal fatigue and NPPs. Some work has been done to quantify stresses and strains that are seen in the near wall regions of T-junction systems. In Kamaya et al's work, a procedure for calculating ranges of stress and strain fluctuations a given wall temperature transient was derived, and this can be used for assessment of fatigue life at T junctions where there are different fluids with different inlet temperatures (Kamaya and Nakamura (2011)). Within the same work. the elastic and elastic-plastic structural analysis were carried out. Results obtained by the elastic and elastic-plastic analyses were compared to investigate the influence of the plastic deformation on stress and strain fluctuations. It was shown that the stress and strain fluctuations were relatively large near the boundary of the hot spot, where relatively large compressive stress was caused by hot water that came from the branch pipe. A coupled approach was explored in Suryavanshi et al's work, where thermal fatigue in junction that involved turbulent mixing was studied. This involved coupling both fluid-flow and mechanical calculations, which is also known as a Fluid-Structure Interaction (FSI) analysis (Suryavanshi and Tawade (2015)). Since

there is a significant amount of energy exchanged between a moving fluid and a solid structure, thermal hydraulic analysis was employed, to find temperature distributions at different zones. This result was then applied to the structural model of the piping system to find the thermal stresses.

Although geometries and configurations of coolant channels have not yet been extensively studied, there is a handful of studies that have been done in attempts to investigate how certain configurations play a part in contributing to thermo-stresses. In Huseyin et al's work, they looked into how branch pipe diameter ratios can potentially affect the thermal stresses at junctions. In both their experimental and computational set-up, it was found that the main reason for such a phenomenon is due to incomplete mixing of hot and cold fluids that encounters the wall. This experiment confirmed that the most likely distance of thermal fatigue is two main pipe diameters at downstream post-mixing (Huseyin and Sokmen (2013)).

A separate study was done on thermal mixing processes along a horizontally oriented T-junction. It aimed to correlate mixing characteristics with thermal fatigue (Zhou (2018)). FSI analysis between circumferential pipe weld connections and the nearby mixing flow was investigated with various measurement methods, where fatigue/crack was initiated by using Artificially Induced Periodical Temperature Changes (AIPTEC). The outcomes seem to support that rimmed weld root can increase the temperature fluctuation in the nearby mixing flow, which possibly increases the possibility of thermal fatigue occurrence. The mixing envelope from the results suggests that the thermal mixing characteristics is the property of the specific mixing tee and can provide an efficient way of reducing/avoiding mixing flow induced thermal fatigue via adjusting the inlet flow boundary condition in the range of deflecting jet, which is due to the lower momentum jet that stems from the branch getting overpowered by a higher-momentum jet from the main pipe.

The influence of upstream elbow in the main pipe was investigate in a separate study by Kimura et al. The main motivation is the vicinity between the elbow and the mixing tee as it affects velocity distribution and the secondary flow, which decays unsteadily. It was found that for the straight pipe configuration, the momentum ratio effectively characterizes the mixing flow patterns between the main and branch pipes. Further information regarding the momentum ratio is discussed in chapter 2. This is further illustrated in Figure 4 with the flow regimes of the branch pipe jet in the elbow configuration classified based on the momentum ratio, just as in

the straight case (Kimura et al. (2010)). A similar test was carried out in the WATLON facility where the temperature distribution in the mixing tee was measured by a movable thermo-couple tree and velocity field were measured by high speed Particle Image Velocimetry (PIV) (Igarashi et al. (2009)). Figure 5 presents the piping configuration found in WATLON. The temperatures measured showed that fluctuation intensity near the wall was larger with the elbow present, compared to a straight wall under a wall jet condition where the jet exiting from the branch pipe was bent down to the pipe wall by high flow velocity in the main pipe. The result was that biased flow velocity distribution and fluctuation due to the elbow induced/influenced bending of branch pipe jet and the temperature fluctuation intensity around the jet.

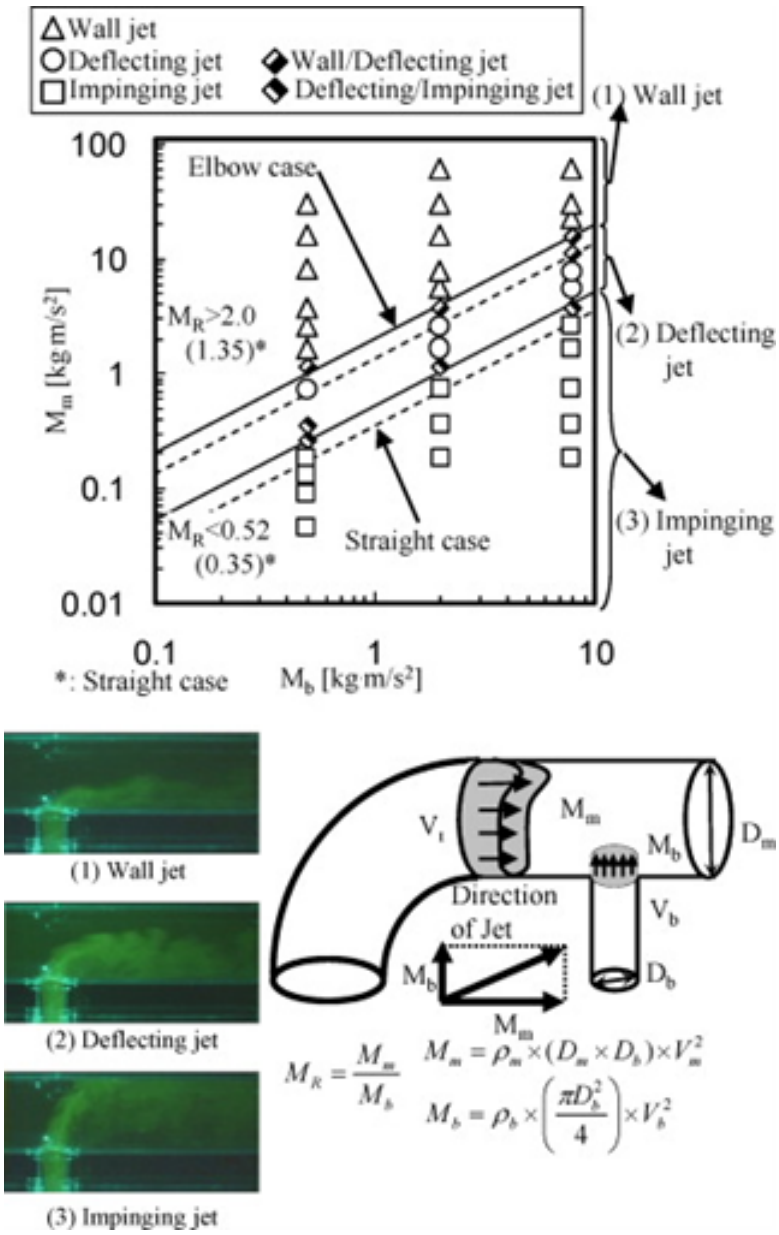


Figure 4: Types of jets found in within the mixing region of a T-junction, adapted from Kimura et al. (2010)

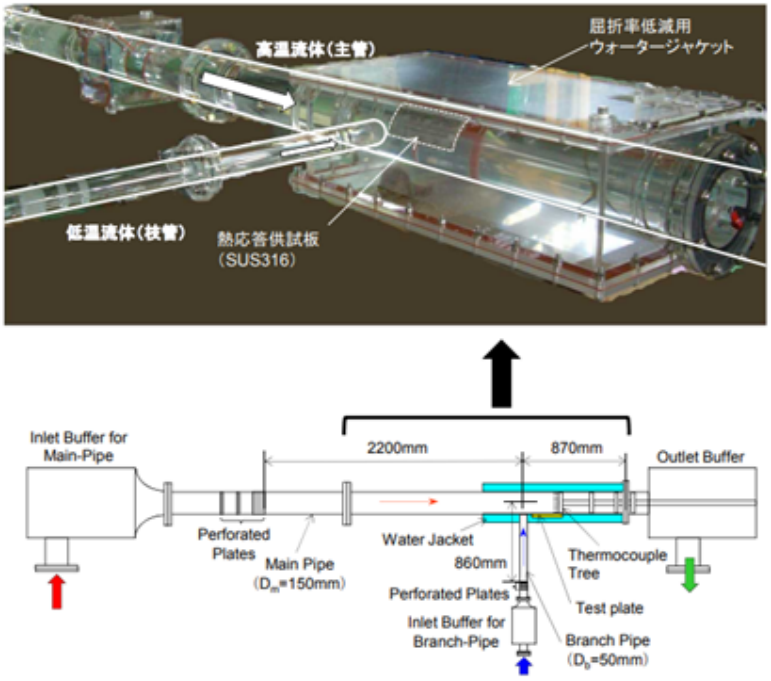


Figure 5: T-junction test rig used in the WATLON facility, adapted from Kimura et al. (2014)

As thermohydraulic fluctuations arising from the turbulent mixing of the flows may affect the inner wall of the pipes and lead to fatigue damage, the FATHER experiment was carried out in order to better understand this phenomenon of High Cycle Thermal Fatigue (HCTF). This was a mock up that solely focuses on T-junction experiments was commissioned under the EDF/CEA/AREVA agreement (Courtin (2013)). Crack growth was investigated as cracks were seen to have propagated along all three weld seams. Further investigation on the fatigue damage in the FATHER experiments were done and consolidated in this work (Taheri et al. (2017)). The influence of thermal stress changes and weld residual stress with respect to the crack depth in the weld were analysed and it was confirmed that thermal fatigue damages are most likely to be initiated close to the weld connections. The catalyst of fatigue was due to the impact of the turbulent mixing flow on the pipe walls.

Another T-Junction test facility, named the FATHERINO facility was also built at the CEA for doing more research on thermal mixing pipe flow. It is a specialized experimental setup developed to study thermal fatigue phenomena in nuclear reactor components, particularly in areas where turbulent thermal mixing occurs, such as T-junctions. It is capable of simulating significant temperature differences between mixing streams (up to 160°C), operates at pressures up to 50 bars, and handles flow rates as high as 1000 m^3/h. Key findings from FATHERINO ex-

periments include the observation that low-frequency thermal fluctuations can penetrate deeper into materials, increasing the likelihood of fatigue damage. In Khun's paper, computational study on the conjugate heat transfer in FATHERINO's T-Junction was carried out, where data from FATHERINO's experiments and CFD simulations were used (Kuhn et al. (2010)). The result strongly suggested that Large Eddy Simulation (LES) could be used to locate the maximum temperature fluctuation along the walls in the pipe, to estimate the stress distribution for accurate prediction of thermal fatigue damage within the pipe's materials. Further information on LES can be found in the next chapter. Both FATHER and FATHERINO's test rigs can be found in the figures below, where Figure 6 contains a schematic of FATHER facility and the various T-junctions of interest, while Figure 7 contains a schematic of FATHERINO facility.

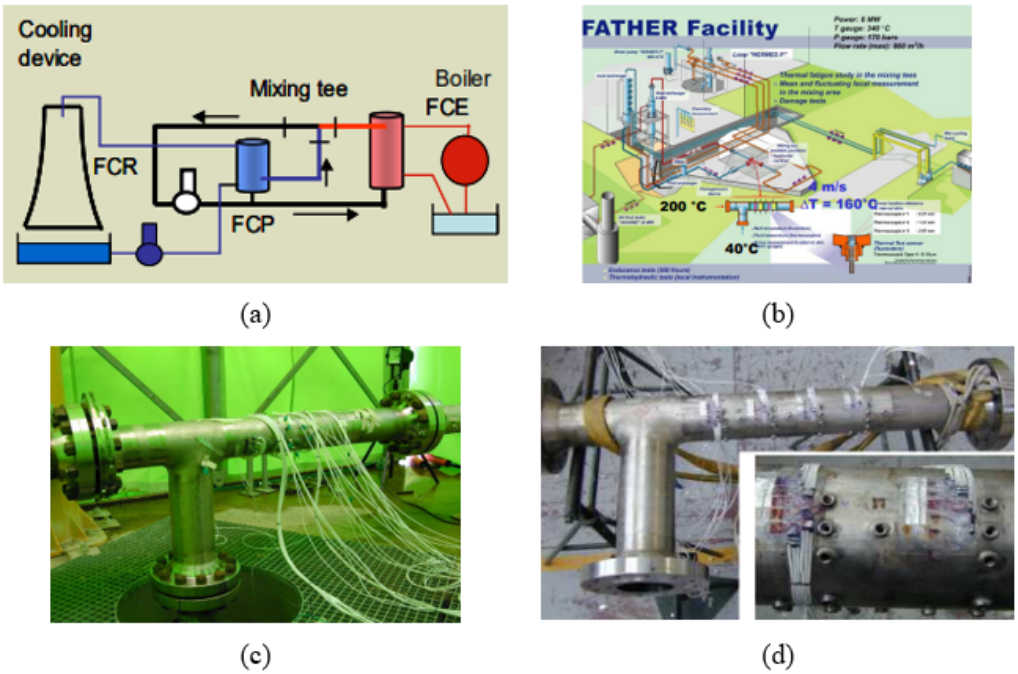


Figure 6: (a) Schematic of FATHER, (b) Facility view of FATHER, (c) Endurance T-junction test rig and (d) Close-up of thermal hydraulic T-junction test rig, adapted from (Brillard et al. (2007))

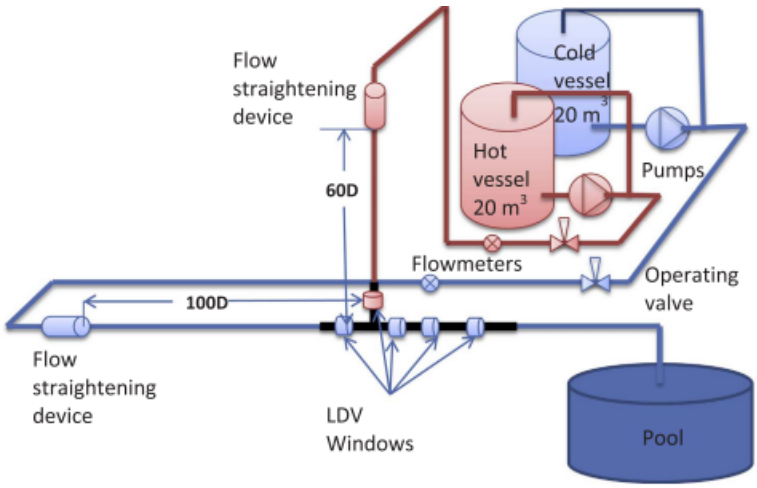


Figure 7: Schematic diagram of FATHRINO facility with emphasis on the T-junctions, adapted from (Braillard et al. (2018))

The Alvkarleby Laboratory of Vattenfall Research and Development in Sweden is widely recognized for their work on investigating thermal fatigue in T-junctions, particularly through its leadership in the OECD/NEA–Vattenfall T-Junction Benchmark project. This initiative was launched in response to the 1998 Civaux-1 incident in France, where unexpected high-cycle thermal fatigue cracks were discovered in a T-junction of the residual heat removal system. To address this issue, a series of detailed thermal mixing experiments simulated the mixing of hot and cold-water streams in a transparent Plexiglas T-junction (Andersson et al. (2006)). A snapshot of Vattenfall's test rig can be found in Figure 8 below, where it details how velocity and temperature fluctuations are generated downstream of the junction due to the turbulent mixing of two inlet fluid streams. The OECD's CFD benchmark for Vattenfall was created with the primary aim of assessing the capability of advanced CFD codes to predict thermal mixing phenomena and the associated high-cycle thermal fatigue in T-junctions

However, this effort provided insufficient insight into how the fluid-solid structures interacts in such flow regimes and highlighted the need for more insights into the heat transfer phenomenon from fluid to solid walls. The MOTHER project was created to validate various CFD approaches such as LES, Reynolds-averaged Navier-Stokes (RANS), and RANS/LES solvers for transient heat transfer in T-Junctions with experimental data. Two different T-junctions (one with round corner, the other with sharp corner) have been constructed and tested in the MOTHER series of experiments. These can be found in Figure 9 below, where close-ups of the T-junction geometries are presented.

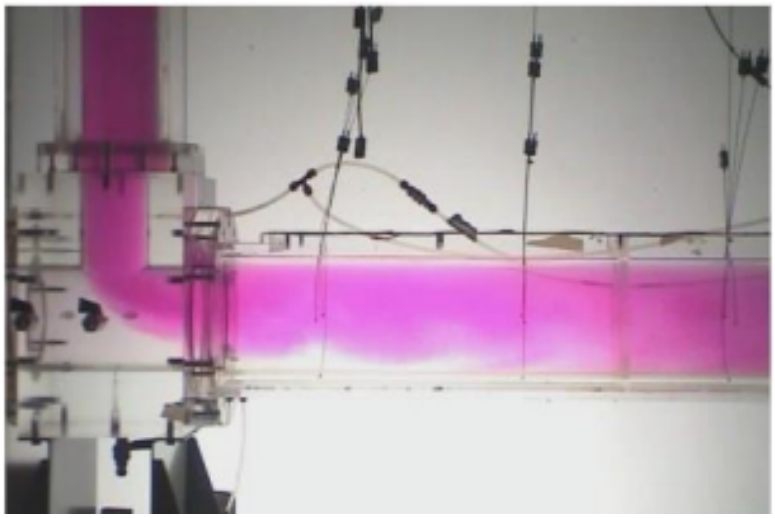


Figure 8: Vattenfall T-junction used for generating test data adapted from (Odemark et al. (2009))

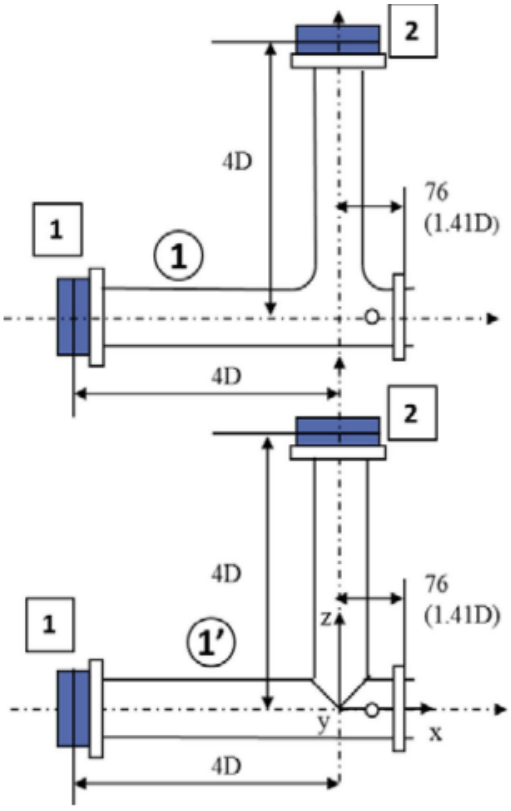


Figure 9: Close-up of the sharp and round corner T-junction geometries used for the MOTHER project, adapted from (Braillard et al. (2018))

More experimental data was created as a result to further investigate and validate the LES/RANS models used in MOTHER projects (Braillard et al. (2018)). In their paper, the FATHERINO facility at CEA in Cadarache was used as a test facility, to obtain experimental data. It is

worth noting that the facility was specifically designed to study the thermal loads for mixing in T-Junction geometries. The instrumentation includes Laser Doppler Velocimetry (LDV) and thermal measurements, from temperature fluctuations in the solid wall and in the fluid. Similar T-Junction mock-ups using 304L stainless steel material were tested with two different internal geometries: a round and a sharp corner. The fluid temperatures used were 15 °C and 30 °C for the cold and hot streams, respectively. While keeping the flow rates in both pipes to be the same, two flow-rates have been tested, with the Reynolds number 40,000 and 60,000. The results from the experiments seem to suggest that the Reynolds number does not have an enormous impact on the flow. However, the rounded and sharp corners show some differences in some areas. The round corner did not show a separation zone, which occurs when a fluid flow detaches from a surface, creating a region of low-pressure and low-velocity fluid behind an obstacle or a change in flow direction. This contrasts to that of a sharp corner where the separation zone was very pronounced. The relaxation towards a uniform velocity profile is faster too, for the case with a round corner. For the case with a sharp corner, its temperature fluctuations within the fluid and the wall have a specific Strouhal number, unlike the case with the round corner. The Reynolds number has a large effect on the amplitude of temperature fluctuations as it causes an increase in heat energy transferred.

The abovementioned works mainly involve experimental configurations and set-ups, which can be quite demanding in terms of resources and time. Hence, there is a need to supplement experimental studies with computational simulations. Numerous numerical simulation codes have been developed within the past two decades, which aim to computationally simulate certain phenomena within these T-junctions. For instance, MUGTHES was originally developed to investigate thermal stripping phenomena, caused by the turbulent mixing of fluids across a range of inlet temperatures. This, coupled with an LES model allows for the accurate prediction of unsteady thermal mixing phenomena (Tanaka et al. (2010)). It is worth noting that MUGTHES has been validated by the WATLON experimental date, which estimates high-cycle thermal fatigue in sodium cooled fast reactors (SFRs) (Tanaka and Myake (2015)).

A study that directly compared experimental data for turbulent mixing in a T-Junction simulation with LES was carried out in Kuczaj et al's work (Kuczaj et al. (2010)). This direct comparison assesses the accuracy of LES predictions for Smagorinsky and Vreman models. The results from the Vreman model are the closest to the experimental results, the opposite is true

for the Smagorinsky model where it provided the least accurate results. This paper also went further into showing that computational meshes must first resolve crucial turbulence length scales to obtain accurate results. LES has been known to capture the mixing phenomena and accompanied turbulent flow fluctuations in a T-Junction. However, its accuracy was not tested and compared with models in the above-mentioned papers. In Kuczaj & Komen's paper, an assessment of the accuracy of LES predictions were carried out utilizing the Vreman model. This was done via a direct comparison with available experimental results, detailed in the paper. It was found that the mesh resolution with the average cell-sizes thrice as small as the Taylor microscale length is enough to provide comparable results to those from finer meshes. What this signifies is that it could potentially be the initial engineering guidelines for the construction of computational meshes for accurate turbulent mixing predictions. Aside from modelling and simulation, much research has been done to directly investigate what and how the mechanisms which leads to cracking (thermal fatigue) within pipes due to thermal loading in flow mixing zones.

In another work, a direct numerical simulation (DNS) was designed based on a T-Junction configuration from the MOTHER project, as mentioned above (Santis and Shams (2018)). A sharp T-Junction configuration was calibrated in this study, where several RANS computations was used to achieve an optimized T-Junction for DNS feasibility. The MOTHER project was unable to accurately predict the thermal field, due to the insufficient statistical convergence on the solid. The conjugate heat transfer (CHT) problem is still too computationally expensive. Different thermal boundary conditions were tested, with an emphasis on the usage of Robin boundary. The Robin boundary is an alternative to the solution of the computationally expensive CHT problem mentioned above. It was implemented in the spectral element code NEK5000 and tested for the DNS of a turbulent channel flow. The calibration study consisted of scaling down the Reynolds number, optimization of the computational domain and investigation of the Prandtl number. The results obtained seem to show a good agreement between that of the CHT problem and that from the Robin boundary condition. This boundary can be considered as the next best alternative to the CHT problem.

The experimental and computational studies surveyed demonstrate substantial progress in understanding turbulent thermal mixing in T-junctions, including the influence of branch orientation, momentum ratio, and Reynolds number on flow structures and thermal fields. Nev-

ertheless, most existing investigations adopt simplified geometries or steady-state conditions that fail to replicate the complexity of operational flow environments, and there is insufficient integration of high-fidelity CFD with fatigue-relevant evaluation metrics in geometries featuring additional complexity, such as upstream elbows. Moreover, mesh refinement strategies are often based on generic error indicators rather than goal-oriented criteria designed to resolve wall temperature fluctuations and gradients most relevant to fatigue initiation. The integration of such physics-driven refinement with reduced-order or surrogate modelling remains underexplored, despite its potential to reduce computational cost without sacrificing predictive accuracy. These limitations directly motivate the present research to (i) apply advanced CFD methods with goal-oriented mesh refinement, (ii) develop deterministic and probabilistic neural network surrogates to reproduce key flow and thermal features, and (iii) assess prediction reliability for thermal fatigue in realistic reactor-grade T-junctions.

1.4 Review of current work done for Surrogate Modelling

Surrogate modelling is the construction of a computationally inexpensive approximation of a high-fidelity numerical model or experiment. The surrogate is trained on a representative set of input–output data pairs from the original model, enabling it to predict outputs for new inputs with significantly reduced computational effort while preserving acceptable accuracy.

There are various surrogate model techniques which includes polynomial chaos expansion which considers a polynomial approximation made of multivariate orthogonal polynomials, low-rank tensor approximation which considers the surrogate model as a summation of rank-one functions, Gaussian process modelling aka Kriging, neural networks, and support vector machines (Santner et al. (2023); Rasmussen and Williams (2005); Smola and Scholkopf (2004)). Multivariate orthogonal polynomials are polynomial functions of several variables that are mutually orthogonal under a specified inner product, typically defined with respect to a weighting function over the domain of interest. In surrogate modelling, they serve as basis functions that enable the decomposition of complex, multivariate relationships into orthogonal components for efficient approximation. On the other hand, low-rank tensor approximation refers to representing high-dimensional data arrays (tensors) in a compressed form by retaining only the dominant components in their decomposition. This approach significantly reduces computa-

tional cost and memory requirements while preserving the primary structural features of the data.The following few paragraphs below will lightly touch on the essence of the various surrogate modelling techniques that are currently employed.

Rank-one functions are separable functions that can be expressed as the product of univariate functions, each depending solely on a single input variable. Such functions are fundamental elements in tensor decomposition methods for simplifying the representation of multivariate functions. Another commonly used technique is Polynomial Chaos Expansion (PCE), is a method for representing a random variable as a form of polynomial function of another random variable. PCE is potentially more computational efficient than Monte Carlo for a sparse number of unknown parameters. However, it still suffers from the curse of dimensionality which is the significant growth in the number of unknowns as the numbers of stochastic variables increases. The PCE method shifts the burden from computing stochastic solutions to computing deterministic parameters or expansion coefficients of the PCE functional expansion. It also allows for the handling of distributions without convenient forms. However, applying PCE might prove cumbersome for inputs with high dimensionality, mainly due to the sharp increase in the number of unknown coefficients. To counter this, another form of surrogate model has been proposed in recent years. Low-rank tensor approximations (LRA) are constituted by sums of products of univariate functions (Konakli and Sudret (2016)). They are usually represented by high-dimensional data arrays (tensors) in a compressed form by retaining only the dominant components in their decomposition. This approach significantly reduces computational cost and memory requirements while preserving the primary structural features of the data. As the user will not have predetermined knowledge of the univariate functions, they are expanded onto basis functions. A common method is to use univariate orthogonal polynomials seen in PCE expansions. The total number of unknowns only grows linearly with the dimensionality. LRA will be less biased in the tails of its output distribution, making this method suitable for the computation of probability of failure.

A technique which has been gaining traction is Gaussian process (GP), a generalization of the Gaussian probability distribution. It is based on supervised learning where a model is trained using labelled datasets, where each input is associated with a known output. The algorithm learns the mapping from inputs to outputs, enabling it to predict outcomes for unseen data. As supervised learning requires assumptions to be made about the characteristics of the underlying

function (that makes predictions from all the possible input values), some solutions have been proposed to deal with this problem. One such method involves restricting the class of functions to be considered. The other is to assign a prior probability to every possible function, where higher probabilities are given to functions which are considered more likely to happen. However, the issue with the former approach is that if the target function is unlike the class of functions within the selection, the target function will not be well modelled, and the outcomes/predictions will be quite poor. The latter approach, although it does not face the issue, it involves an infinite set of possible functions, which cannot possibly be computed within finite time. GPs aim to overcome both problems as inference in GPs will produce a result that is the same if one were to take every point within the infinite set of functions and if one were to ignore the infinitely many other points, except for the properties of the function at a finite number of points, within the infinite set of functions. One of the major advantages of a GP framework is its ability to unite a convoluted and coherent view with computational controllability. However, GPs have their own set of limitations too. For example, GPs utilizes entire samples to execute predictions. As the number of features increase, GPs lose efficiency in such spaces of higher dimensions.

A class of surrogate modelling technique that existed in the early days of machine learning is Support Vector Machine (SVM), which is a computer algorithm that learns from a dataset to assign labels to objects (Boser et al. (1992)). The objective of support vector machine algorithm is to find a hyperplane in a space with a certain amount of dimensions. Hyperplanes are flat surfaces that separate data points in a dimensional feature space, commonly used in classification tasks. Data points that lie on either side of the hyperplane can be assigned to different classes. The dimension of the hyperplane is dependent on the number of features. Support vectors are data points that are nearer to the hyperplane and affects the position and orientation of the hyperplane. By utilizing the support vectors, one can maximize the margin of the classifier. The main idea behind this algorithm is to maximize the margin between the data points and hyperplane.

More recently, neural networks have gained heightened interest, especially in the field of fluid mechanics. Various types of neural networks exist, with different types of architectures designed for specific tasks and data structures. The simplest form, the feedforward neural network (FNN), consists of input, hidden, and output layers, where data flows in one direction. In neural networks, layers are the fundamental building blocks of a neural network where com-

putations are performed. Each layer transforms input data to a higher-level representation. The multi-layer perceptron (MLP) extends this concept by introducing multiple hidden layers, allowing the model to learn hierarchical representations of data. Variants of MLPs have emerged to address domain-specific challenges. Physics-informed neural networks (PINNs) integrate governing physical laws, typically represented by partial differential equations (PDEs), into the loss function, enabling the learning of solutions to complex physical systems without requiring large, labelled datasets. Loss functions simply measure how far the network's predictions are from the actual target values, guiding how the model should improve during training. It was popularised by Raissi et al within the last decade (Raissi et al. (2019)). Bayesian neural networks (BNNs) incorporate uncertainty quantification by treating network weights as probability distributions, making them useful in safety-critical applications where predictive confidence is essential (Polson and Sokolov (2017)). Network weights are the parameters that the model learns and adjusts during training to minimize the loss and improve prediction accuracy. Building on this, Bayesian physics-informed neural networks (BPINNs) combine the principles of PINNs and BNNs to provide not only physically consistent solutions but also uncertainty estimates, which are crucial for tasks like turbulence modelling and heat transfer analysis.

Convolutional neural networks (CNNs), first introduced by LeCun et al, are particularly effective for image processing tasks due to their ability to capture spatial hierarchies through convolutional and pooling layers (LeCun et al. (1989)). These layers are building blocks of CNNs, where convolutional layers apply filters to detect spatial features and pooling layers reduce the spatial dimensions to retain essential information while lowering computational cost. The transformer architecture, introduced by Vaswani et al. has further revolutionized neural network research by replacing recurrence with self-attention mechanisms, significantly enhancing the efficiency of processing long sequences (Vaswani et al. (2017)). It involves combining RNNs with self-attention, which allows the model to weigh the importance of different positions in the input sequence and capture long-range dependencies more effectively. This innovation has led to breakthroughs in natural language processing, with models such as BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pretrained Transformer) achieving state-of-the-art performance across multiple benchmarks. Meanwhile, graph neural networks (GNNs) have emerged for learning on graph-structured data, proving useful in applications like drug discovery, social network analysis, and molecular modelling. These archi-

tectures, each designed for specific types of data and tasks, highlight the adaptability of neural networks across various domains.

The success of neural networks depends on their ability to generalize from training data, which is achieved through optimization techniques. Training data usually consists of labeled or structured input-output examples used to teach a machine learning model how to map inputs to desired outputs. It forms the foundation on which the model learns patterns and generalizes to new data. The backpropagation algorithm, formulated by Rumelhart et al, enables efficient weight updates by computing gradients using the chain rule. Gradient descent methods, particularly stochastic gradient descent (SGD), form the backbone of neural network training (Rumelhart et al. (1986)). Using the chain rule to propagate error signals backward through the network layers allows those weights to be updated during training. However, due to the challenges of finding optimal solutions in high-dimensional spaces, several advanced optimization techniques have been developed. These include ADAM (Adaptive Moment Estimation), where momentum, adaptive learning rates and RMSprop (Root Mean Square Propagation) are combined which helps stabilize training in non-stationary environments. Note that the learning rate is a hyperparameter that controls how much a neural network's weights are updated during training in response to the estimated error. Regularization techniques are essential for preventing overfitting in neural networks. Overfitting occurs when a model learns not only the underlying patterns in the training data but also the noise, resulting in poor generalization to new or unseen data. Dropout randomly deactivates neurons during training to enhance model generalization, while batch normalization standardizes activations to improve training stability. L1 and L2 regularization are techniques used to prevent overfitting by adding penalty terms to the loss function based on model weights (Srivastava et al. (2014); Ioffe and Szegedy (2015)). L1 encourages sparsity by penalizing the absolute values of weights, while L2 penalizes the squared values, promoting smaller but non-zero weights. Additionally, data augmentation techniques artificially expand training datasets to improve model robustness, particularly in computer vision tasks.

Neural networks have demonstrated unparalleled success across various domains. In computer vision, CNNs power applications such as image classification (e.g., ResNet, AlexNet, VGGNet), object detection (YOLO, Faster R-CNN), and medical image analysis (e.g., tumor detection, anomaly identification). In natural language processing (NLP), transformer-based models like

BERT, and T5 have enabled advancements in machine translation, sentiment analysis, and text generation. In reinforcement learning, deep neural networks guide decision-making in applications such as autonomous driving, robotics, and game playing (e.g., AlphaGo and AlphaZero). Neural networks have also been instrumental in scientific computing and engineering applications. They are widely used in fluid dynamics, where PINNs solve partial differential equations for turbulence modelling and heat transfer problems. In financial modelling, deep learning aids in stock price prediction, risk assessment, and fraud detection. Additionally, neural networks play a crucial role in healthcare, where deep learning models assist in medical diagnosis, personalized medicine, and genomics research.

Given the wide variety of surrogate modelling techniques available, the choice of method often depends on the nature of the underlying problem, the dimensionality of the input space, and the complexity of the system being studied. While methods like PCE, Low-Rank Tensor Approximations, GP, and SVM each offer unique advantages, they also come with limitations especially when handling high-dimensional, nonlinear, and noisy data. In contrast, neural networks have shown remarkable versatility and performance across a wide range of domains, including fluid mechanics, where governing equations are often nonlinear, multiscale, and computationally expensive to solve using traditional methods. Neural networks, particularly PINNs and BPINNs, are capable of capturing complex patterns in data while incorporating prior physical knowledge and uncertainty quantification (UQ). Their adaptability to various data types, scalability to high-dimensional input spaces, and capacity for generalization make them especially attractive for surrogate modelling in CFD applications.

The literature on surrogate modeling provides a rich set of techniques, but few have been applied systematically to turbulent thermal mixing in nuclear components, and even fewer combine predictive accuracy with quantified uncertainty. While methods like Gaussian processes, polynomial chaos expansion, and low-rank approximations offer value, neural networks—particularly probabilistic variants—hold promise for capturing complex, nonlinear flow-thermal interactions at reduced computational cost. Despite these advances, their application to turbulent thermal mixing in nuclear T-junctions remains limited, and predictive uncertainty is rarely quantified, even though such quantification is essential for safety-critical engineering applications. Furthermore, there is a lack of work coupling surrogate modelling with adaptive meshing strategies that explicitly target quantities of interest linked to structural fatigue, limiting the operational

relevance of current surrogate frameworks. This research addresses these deficiencies by systematically evaluating and optimising deterministic and probabilistic neural network architectures for the surrogate modelling of turbulent internal flows, integrating them with goal-oriented mesh refinement, and comparing both deterministic and Bayesian techniques to ensure reliable prediction of the overall flow fields.

1.5 Overview of Literature Survey

This literature survey has provided a comprehensive overview of both experimental and computational studies on turbulent thermal mixing in nuclear T-junctions, with emphasis on their implications for structural integrity and thermal fatigue. Past research has advanced the understanding of flow behavior under different geometries, including classical vertical-horizontal pipe configurations and more complex arrangements with upstream elbows. Significant progress has been made in using CFD to predict flow structures and temperature fluctuations, supported by experimental benchmarks that validate unsteady mixing mechanisms and their effect on near-wall conditions.

Nonetheless, several critical knowledge gaps remain. Firstly, many studies focus on steady or simplified boundary conditions and do not fully capture the range of transient and turbulent effects seen in realistic operational environments. Secondly, there is a lack of integrated approaches that couple adaptive meshing, data-driven modeling, and uncertainty quantification to target fatigue-relevant quantities of interest such as wall temperature gradients and stress cycles. Thirdly, while surrogate modeling is emerging as a tool for accelerating predictions, there remains limited insight into how neural network architectures can be optimized for complex internal flows or how such models behave under varying levels of turbulence and flow asymmetry.

This thesis builds on these findings by combining advanced CFD methods with surrogate modeling and adaptive meshing strategies, specifically aimed at capturing the physics responsible for thermal fatigue in nuclear T-junctions. In doing so, it aims to bridge the gap between high-fidelity simulation and efficient, uncertainty-aware predictive tools.

1.6 Proposed Methodology for Modelling & Simulations and Surrogate Modelling

This thesis aims to improve the predictive modeling of turbulent thermal mixing in nuclear T-junctions by integrating adaptive meshing, neural network surrogate modeling, and uncertainty quantification. The investigation is motivated by key gaps identified in the literature which are specifically, the limited use of goal-oriented mesh refinement to capture fatigue-relevant quantities, the lack of systematic benchmarking of neural network architectures for turbulent internal flows, and the absence of robust benchmark comparisons in surrogate predictions.

To address these gaps, the study focuses on three core research objectives. The first aim is to assess whether adaptive mesh refinement strategies, particularly goal-based refinement, can better resolve local flow and thermal gradients that contribute to thermal fatigue. The second aim is to extract and analyse the dominant turbulent flow structures using data-driven spectral methods which include Proper Orthogonal Decomposition (POD), Spectral POD (SPOD), and frequency-domain analysis. This is done to gain deeper insight into the unsteady flow behaviour that underlies wall temperature fluctuations, a precursor to thermal fatigue. The third aim is to develop and evaluate neural network-based surrogate models for reconstructing key flow and temperature fields. This involves both deterministic and probabilistic framework where Bayesian neural networks and physics-informed Bayesian models are further explored.

These objectives give rise to specific research questions: How effective are goal-oriented AMR techniques in capturing quantities of interest relevant to fatigue? What are the dominant turbulent structures in complex T-junction flows, and how are they affected by Reynolds number or geometry? What neural network designs offer the best trade-off between accuracy and generalizability for modeling turbulent flows in T-junctions? And how reliably can uncertainty in predictions be quantified using Bayesian approaches under realistic, high-gradient flow conditions?

To answer these questions, a staged methodology is adopted. The approach begins with high-fidelity CFD simulations using a hybrid RANS/LES turbulence model to generate training and evaluation datasets. Adaptive meshing is implemented and assessed against fixed-grid baselines to determine its effectiveness in resolving key physical features. Subsequently, flow

data are decomposed using POD and SPOD techniques to extract coherent structures and identify dominant frequency modes commonly associated with thermal mixing and wall-shear loading. In parallel, several neural network models are trained on benchmark and realistic data to evaluate their performance in recovering flow fields. The methodology concludes with a study using both deterministic and Bayesian surrogates to assess model robustness. This integrated framework ensures that the selected methods are not arbitrarily chosen, but directly motivated by the research questions and gaps identified through the literature review.

1.7 Project Scope

The aim of this PhD is to develop computationally cheap and efficient high-fidelity M&S and data-driven neural network-based surrogate model for thermal fatigue phenomena analysis. These methodologies are touted to greatly improve deterministic and probabilistic life-cycle safety assessments for NPPs. The novelty of this thesis is detailed below:

- Investigate and quantify the effect of variations in the momentum ratio on the turbulent fluid flow within T-Junctions. It is crucial to understand the factors that affect turbulent flow behaviour in certain components, as it directly relates to the overall safety of an NPP. This is achieved by either varying the branch pipe diameter while keeping the inlet velocity constant or by adjusting the branch pipe inlet velocity while maintaining a constant diameter. An in-depth analysis of turbulent quantities within the flow field at various locations of interest is carried out, where bulk of the turbulent fluctuations occur.
- Investigate the underlying turbulent flow structures using methods such as Power Spectral Density (PSD), POD and SPOD. Concurrently, differences in flow structures produced by POD and SPOD are compared. This enables us to determine the dominant flow feature which could potentially be the root factor contributing to potential thermal fatigue within the regions of interest.
- Using a standard benchmark CFD test case, such as the lid-driven cavity, allows for preliminary investigations into key factors that significantly affect neural network performance in a controlled setting. The LDC case is widely recognized for its well-defined physics such as shear-driven vortices and recirculating flow. These make it ideal for evaluating the ability of neural networks to capture essential flow features. A partic-

ular neural network architecture is selected due to its complexity, and the LDC setup ensures the training process remains interpretable and reproducible. This is because the lid-driven cavity problem shares physical similarities with a T-junction—such as sharp geometric corners, separated flow regions, and strong shear layers—making it a suitable surrogate development test bed. This controlled foundation provides meaningful insights into model behavior, robustness, and generalization, which are critical for Point 4 and ultimately transferable to more complex turbulent flows encountered in the pipe junction analysis.

- Supplementing the benchmark CFD test case with an extended geometry for added complexity. Explicit comparison between the predictive capabilities of various neural networks determines the most appropriate choice for use in the eventual surrogate model.
- Development of a data-driven surrogate model using a custom, in-house implementation of a custom framework. An extensive grid search is conducted to identify the optimal combination of network hyperparameters for accurately recovering flow physics in geometries where sharp velocity gradients are observed. The resulting surrogate model is designed to replicate key flow and thermal characteristics with high fidelity, serving as a computationally efficient alternative to full CFD simulations. This model will be instrumental in future studies involving similar geometries, particularly in turbulent flow modelling and thermal analysis within nuclear systems

1.8 Thesis Structure

This thesis begins by establishing the broader context and motivation behind the research. chapter 2 presents the theoretical foundations necessary for understanding turbulent flow simulations. It begins with an overview of turbulence, followed by a discussion of turbulence modelling approaches and the application of the finite volume method for numerical simulation. Mesh discretization strategies, including their importance in capturing the flow structures relevant to thermal fatigue, are also addressed.

In the next chapter, chapter 3 describes the CFD model used throughout the thesis. The chapter details the model settings, boundary conditions, and the mesh sensitivity analyses, including both conventional and adaptive mesh refinement techniques. These steps ensure that the CFD simulations are robust and sufficiently resolved to capture key physical phenomena.

Building upon the CFD framework, chapter 4 introduces the machine learning models used in this work. It begins with the fundamentals of deep neural networks (DNNs), and progresses to more advanced architectures, including PINNs, BNNs, and BPINNs, which are particularly valuable for surrogate modelling.

This is followed by chapter 5, where the influence of momentum ratio and Reynolds number on turbulent flow within a T-junction is investigated through a series of CFD simulations. The results are analysed to understand how changes in boundary conditions affect flow behaviour, which has direct implications for thermal fatigue.

Complementing this, chapter 6 applies POD and SPOD to further investigate flow structures and dominant modes within the T-junction. These tools provide deeper insights into flow dynamics, particularly the frequency content and spatial patterns associated with temperature oscillations that contribute to fatigue damage.

As the next phase of the project involves the use of machine learning techniques, chapter 7 shifts the focus to the optimization of neural network architectures. Using a lid-driven cavity benchmark problem, different activation functions and network configurations are assessed to understand how design choices impact model performance, guiding the development of surrogate models.

The performance of the developed neural networks as surrogate models for flow field reconstruction is further evaluated in chapter 8. It presents a comparative study of different networks, analysing their ability to recover complex flow features from limited data, which is critical for real-time monitoring and digital twin applications.

Further work, building on the scientific questions motivating this thesis, is outlined in chapter 9, where recommendations for future research are presented. These include applying the developed methods to goal-oriented uncertainty quantification for improved reliability in nuclear thermal–hydraulic predictions. The chapter also considers coupling machine learning surrogates with whole-plant system simulators to enable faster, system-wide analyses. Finally, it highlights the exploration of alternative surrogate models for time-dependent data to better capture transient flow and thermal mixing behaviour.

Finally, chapter 10 summarizes the findings of the research, discusses its contributions to the

field, and outlines potential directions for future work.

Chapter 2

The Mathematical and Computational Modelling of Turbulent Flows

This chapter begins with a general description of equations which govern fluid flows and heat transfer in section 2.1. The basis of turbulence in introduced in section 2.2, which involves cascading energy to the length & timescales. The methodology used for modelling and simulation of turbulent flows used in this study is further described in detail within this section with reference to in-house turbulence models provided by Simcenter STAR-CCM+ 2023.1.1 (Siemens Digital Industries Software (Siemens 2023)). The equations for RANS k ω SST SST and IDDES-SST turbulence models are also explored. The following section, section 2.3 provides a brief description of the numerical algorithms used for the Finite Volume method employed in Simcenter STAR-CCM+. In the last section, section 2.4 discusses the discretization schemes and solution methods utilized in the creation of computational models and are further detailed within this section.

2.1 A Primer on Turbulent Flows

In fluid flow and heat transfer analysis, the behavior of fluids is described by a set of governing equations derived from the fundamental principles of physics, such as conservation of mass, momentum, and energy. These equations provide a framework for understanding how fluids move and exchange energy, allowing us to solve complex engineering problems involving heat and mass transfer. The working fluid considered in this project is assumed to have a constant

and uniform viscosity, regardless of temperature changes. For an incompressible fluid, it can be characterized by the continuity equation and Navier Stokes equation and the thermal energy equation. Respectively, these can be expressed as:

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u_j)}{\partial x_j} = 0 \tag{1}$$

$$\frac{\partial(\rho u_i)}{\partial t} + \rho u_j \frac{\partial u_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{1}{Re}\frac{\partial^2 u_i}{\partial x_j^2} + \frac{1}{Fr^2} g_i \tag{2}$$

$$\frac{\partial T}{\partial t} + u_j \frac{\partial T}{\partial x_j} = \alpha \frac{\partial^2 T}{\partial x_j^2} \tag{3}$$

Where ρ, u, p and α are density, velocity, pressure and the thermal diffusivity of the fluid. The thermal diffusivity can be further defined as:$\alpha = \frac{k_{thermal}}{\rho c_p}$, where $k_{thermal}$ is the thermal conductivity, and c_p is the specific heat capacity at constant pressure. c_p is taken to be constant and uniform in this project but it is still shown for completeness and generality. Fr refers to the Froude number. For an incompressible flow where the density is constant and the divergence of velocity is negligible, both the continuity and Navier Stokes equation can be reduced to the following equations below. The term including the Froude number and gravitaitonal acceleration is dropped under the assumption that gravity is negligible or not included as a driving force in the flow. This simplifies the momentum equation to represent pure forced convection (or inertial-dominated flow) without buoyancy effects.

$$\frac{\partial u_j}{\partial x_j} = 0 \tag{4}$$

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j^2} \tag{5}$$

Where ν refers to the kinematic viscosity

2.2 Turbulence Modelling

2.2.1 Introduction to Turbulence

Turbulence is a chaotic and complex phenomenon that occurs in fluid flows, characterized by irregular fluctuations. One of the most important concepts in understanding turbulence is the energy cascade, which was first proposed by Richardson. His description consists of how energy is transferred across different scales of motion within a turbulent flow. In turbulent systems, large-scale eddies (or swirls) are created by external forces, and these large eddies continuously break down into smaller ones, transferring their energy into a cascading process until they are dissipated at the smallest scales through viscosity. This understanding of energy distribution is fundamental in modeling and predicting the behavior of turbulent flows.

This was further formalized by Kolmogorov in 1941 (Kolmogorov (1941)). According to Kolmogorov's theory, the largest scales in turbulent flow (called the integral scales) contain the most kinetic energy, introduced by external sources such as pressure gradients, shear, or flow obstacles. These large eddies eventually break down into smaller eddies through nonlinear interactions, transferring energy without significant loss. The intermediate range, called the inertial subrange, is where the energy transfer occurs and is considered self-similar, meaning that the flow characteristics are independent of the specific scale. In this range, the energy distribution follows Kolmogorov's law which states that the energy spectrum decreases with the wave number. The Kolmogorov length and time scales of small-scale motions can be expressed as:

$$\eta = \left(\frac{\nu^3}{\varepsilon}\right)^{1/4} \tag{6}$$

$$T_\eta = \left(\frac{\nu}{\varepsilon}\right)^{1/2} \tag{7}$$

Where η and T_η are measures of the smallest eddies within the flow while $\varepsilon = \frac{u_0^3}{l_0}$, where u_0 is the characteristic velocity and l_0 is the characteristic length scale. These are only applicable and valid for the dissipation range. At a sufficiently high Reynolds number, the characteristic

length scale and the smallest length scale η will be dominated by the effects of ε and not the kinematic viscosity, such that l_0 now becomes

$$l_h = \frac{k^{3/2}}{\varepsilon}$$

while T_η now becomes

$$T_h = \frac{k}{\varepsilon}.$$

There will be a transition between Kolmogorov and integral scales, which is also known as the Taylor micro-scales. The length and time scales of the Taylor micro scale can be expressed as:

$$\lambda = \sqrt{\frac{10\nu k}{\varepsilon}} \tag{8}$$

$$T_\lambda = \sqrt{\frac{15\nu}{\varepsilon}} \tag{9}$$

2.2.2 Turbulent Boundary Layer

Turbulent boundary layers form when fluid flows over a surface at high enough velocities or Reynolds numbers such that inertial forces dominate and instabilities cause chaotic, three-dimensional motion. Unlike laminar boundary layers, turbulent layers are characterized by enhanced mixing and momentum transfer, which lead to steeper velocity gradients near the wall and higher shear stresses. In duct flows, these boundary layers grow from the walls and eventually merge at the center of the duct, marking the transition to a fully developed flow regime where the entire cross-section is influenced by turbulence.

This behavior is particularly relevant in thermal-hydraulic applications where internal flows are driven by mechanical means, such as pumps or flow controllers. In the present study, forced convection is used to replicate realistic conditions observed in operational test rigs and industrial components. This allows the simulation to reflect the dominant transport mechanisms and flow

structures associated with turbulent mixing and wall heat transfer.

2.2.3 Types of Turbulence Models

Accurately predicting turbulent flows is crucial in engineering applications, ranging from aerospace and automotive industries to environmental and nuclear engineering. Due to the chaotic and multiscale nature of turbulence, solving the full set of Navier-Stokes equations directly using Direct Numerical Simulation (DNS) is computationally expensive and often impractical for most real-world applications, especially at high Reynolds numbers (Kasagi (1998); Moin and Mahesh (1998); Sharma and Patankar (2005)). To overcome these challenges, turbulence models and closure models are used to approximate the effects of turbulence on the mean flow, providing a feasible way to solve these flows numerically.

Reynolds-Averaged Navier-Stokes (RANS) models are among the most widely used approaches for simulating turbulent flows due to their relatively low computational cost (Coroneo et al. (2011); Zhao et al. (2020)). In RANS models, the instantaneous flow variables are decomposed into mean and fluctuating components, and the governing equations are averaged over time. This can be expressed in Equations 10 and 11 below, which represent the Reynolds-averaged continuity and Reynolds-averaged momentum equations. δ_{ij} is the Kronecker delta and $\bar{S}_{ij}$ is the mean strain rate tensor while T_{ij} is the Reynolds stress tensor. The transport of thermal energy for RANS can be found in Equation 12 below, where $\tilde{T}$ refers to the mean temperature, α is the thermal diffusivity seen in Equation 3, α_t is the turbulent thermal diffusivity and Pr_t is the turbulent Prandtl number. Note that these equations assume an incompressible flow as mentioned in the beginning of this chapter.

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho \bar{u}_j)}{\partial x_j} = 0 \tag{10}$$

$$\frac{\partial (\rho \bar{u}_i)}{\partial t} + \frac{\partial (\rho \bar{u}_i \bar{u}_j)}{\partial x_j} = -\frac{\partial (\bar{p} \delta_{ij})}{\partial x_j} + 2\mu \frac{\partial \bar{S}_{ij}}{\partial x_j} - \frac{\partial \bar{u}_i \bar{u}_j}{\partial x_j} \tag{11}$$

$$\frac{\partial \widetilde{T}}{\partial t} + \widetilde{u}_j \frac{\partial \widetilde{T}}{\partial x_j} = \frac{\partial}{\partial x_j}\left[(\alpha + \alpha_t)\frac{\partial \widetilde{T}}{\partial x_j}\right] \tag{12}$$

δ_{ij} is the Kronecker delta, $\bar{S}_{ij}$ is the mean strain rate tensor while T_{ij} is the Reynolds stress tensor.

The Reynolds stress tensor is a natural additional term that is introduced to the Navier-Stokes equation even though Reynolds-averaging does not require the computation for the instantaneous flow field. It represents the effects of turbulence on the mean flow. A common method used for computing the Reynolds Stresses is via the Boussinesq approximation. It ensures a linear relationship between turbulent Reynolds stresses and the mean rate of strain tensor, which can be found in the equation below, where μ_t refers to the eddy viscosity, k is the turbulent kinetic energy per unit mass. Since $\frac{d\bar{u}_k}{dx_k}$ tends to zero for an incompressible flow, the isotropic eddy viscosity can be simplified to the expression found at the end of Equation 13.

$$-\bar{u_i}\bar{u_j} = \mu_t\left[\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} - \frac{2}{3}\frac{\partial \bar{u}_k}{\partial x_k}\delta_{ij}\right] - \frac{2}{3}\rho k\delta_{ij} = \mu_t\left[\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right] - \frac{2}{3}\rho k\delta_{ij} \tag{13}$$

The turbulent heat flux vector can be found in Equation 14, where this equation models the heat flux due to turbulence in terms of the turbulent eddy viscosity and the gradient of the mean temperature.

$$q_j^{\text{turb}} = -\frac{\mu_t}{Pr_t}\frac{\partial \tilde{T}}{\partial x_j} \tag{14}$$

As the only unknown is eddy viscosity, there are a handful of turbulence closure models, such as the k-ϵ and k-ω models which provide their own take on the eddy viscosity. The k-ϵ model is popular due to its simplicity and robustness for a wide range of flows. It uses two additional equations to model the turbulent kinetic energy (k) and its rate of dissipation (ϵ), allowing the Reynolds stresses to be approximated (Laan and Andersen (2018)). While effective for many engineering applications, the k-ϵ model often struggles with accurately predicting flows with strong adverse pressure gradients, flow separation, and near-wall behavior. Alternatively, the k-ω model uses specific dissipation rate (ω), which provides better predictions for flows in the

free stream. Variations of these models, such as the SST model, combine the strengths of both k-ϵ and k-ω models, enhancing performance in complex flow conditions.

LES offers a higher-fidelity approach compared to RANS by resolving the large-scale turbulent structures directly while modeling the smaller, sub grid-scale (SGS) turbulence using closure models (Chandy and Frankel (2009); Kotov et al. (2016)). LES provides a more accurate representation of turbulent flows since large eddies, which contain most of the energy and significantly impact the flow's behavior, are explicitly resolved. This makes LES particularly effective for capturing complex unsteady flow phenomena such as vortex shedding, flow separation, and transient mixing. Filtered Navier-Stokes equation solves for the large-scale eddies while a sub-grid scale stress (SGS) model solves for small scale eddies. Since the fluid used is considered incompressible, mass and momentum conservation equations used for such flows can be reduced to the expressions found in Equations 15 and 16. Equation 17 represents the filtered thermal energy transport equation which includes the subgrid-scale heat flux model (Equation 19). Here, $\tilde{T}$ refers to the mean temperature and α is the thermal diffusivity seen in Equation 3.

$$\frac{d\bar{u}_i}{dx_i} = 0 \tag{15}$$

$$\frac{d\bar{u}_i}{dt} + \bar{u}_j \frac{d\bar{u}_i}{dx_j} = -\frac{d\bar{p}}{dx_i} + \nu \frac{d^2\bar{u}_i}{dx_j^2} - \frac{d\tau_{ij}}{dx_j} \tag{16}$$

$$\frac{\partial \tilde{T}}{\partial t} + \tilde{u}_j \frac{\partial \tilde{T}}{\partial x_j} = \alpha \frac{\partial^2 \tilde{T}}{\partial x_j^2} - \frac{\partial q_j^{\mathrm{SGS}}}{\partial x_j} \tag{17}$$

Fluid properties are normally functions of temperature for flows that involve heat transfer. As the fluctuations may be small, it might even be the cause of the fluid motion. If the density variation is not extremely large, the density can be treated as a constant in the unsteady and convection terms and only treat it as a variable in the gravitational term. The reduced formed for incompressible flows can be further modelled with the Boussinesq approximation in Equation 18 below. The subgrid-scale heat flux for LES, which is analogous to the Boussinesq

approximation can be found in Equation 19.

$$\tau_{ij} = u_i\bar{u}_j - \bar{u}_i\bar{u}_j = \frac{1}{3}\sigma_{ij}\tau_{kk} - 2\nu_{sgs}\bar{S}_{ij} \tag{18}$$

$$q_j^{\text{SGS}} = -\frac{\nu_{sgs}}{Pr_{sgs}}\frac{\partial\tilde{T}}{\partial x_j} \tag{19}$$

where τ_{ij} is the stresses due to subgrid scale motions, σ_{ij} is the Kroneckder delta,Pr_{SGS} is the SGS turbulent Prandtl number, v_{SGS} is the subgrid viscosity and $\bar{S_{ij}} = \frac{1}{2}(\frac{d\bar{u_i}}{d\bar{x_j}} + \frac{d\bar{u_j}}{d\bar{x_i}})$ is the filtered rate of strain tensor. The isotropic term can be incorporated into the pressure for flows that are incompressible.

The sub grid scale model is essential in LES to account for the effects of unresolved small-scale turbulence. Common SGS models include the Smagorinsky model, which relies on an eddy viscosity concept to approximate the dissipation effects of small-scale turbulence, and the Dynamic Smagorinsky model, which adapts the eddy viscosity based on local flow conditions, offering improved accuracy. This model assumes that the Reynolds number is sufficiently high enough to make sure that the energy transfer occurs from large to small scales, which accounts for dissipation only. Smagorinsky defined the large eddies scale and the corresponding velocity scale as shown below:

$$l = C_s\Delta \tag{20}$$

$$\bar{S} = \sqrt{2\bar{S}_{ij}S_{ij}} \tag{21}$$

Where C_s is the Smargorinsky constant. Thus, the sub grid viscosity (eddy viscosity) used is defined by:

$$\nu_{sgs} = (C_s\Delta)^2|\bar{S}| \tag{22}$$

Where Δ is the filter length scale, computed as the cube root of the cell volume in this case. LES is widely used in applications where detailed flow information is needed, such as in the

study of turbulent combustion, aerodynamic noise prediction, and complex wake flows. However, LES remains computationally intensive, particularly in near-wall regions, necessitating significant computational resources. Wall Modelled Large Eddy Simulation (WMLES) is similar to LES models, except that they are less dependent on Reynolds number for wall-bounded fluid flows. This is possible with the use of RANS for near-wall regions of the boundary layer, while employing LES to resolve for the remaining turbulence within the boundary layer. This eradicates the need for resolving turbulent eddies located right above the viscous sublayer.

This is considered a hybrid modelling method that combines features of RANS and LES in parts of the flow. The main idea behind this model is solving for the near-wall regions using RANS model and thereafter, switching to LES model in detached regions. This greatly reduces the computational requirements and cost for turbulent simulations, while reaping the benefits of running LES in the areas of interest. An extension of DES is the DDES. A limitation observed in DES models is that it experiences limited switching from RANS to LES when the grid density is reduced. As such, flows with thick boundary layers/shallow separation will be resolved inaccurately. This is more commonly known as Modelled Stress Depletion (MSD). This leads to early flow separation due to the reduced skin friction, also known as Grid Induced Separation (GIS). The problem of GIS can be avoided by shielding the RANS model from DES within wall layers. This was made possible when Spalart proposed a more generic formulation of the shielding function, where only the eddy-viscosity and wall distance are concerned which led to DDES.

Improved Delayed Eddy Simulation (IDDES) takes DDES a step further, as it has the added ability to resolve log-layer mismatch, other than Modelled Stress Depletion (as mentioned). It involves the use of specialised shielding functions, allowing the model to accommodate the benefits from both DDES and WMLES models (Guseva et al. (2016)). The IDDES-SST involves the use various equations: Turbulent Kinetic Energy (k) equation and Specific Dissipation Rate (ω), the IDDES length scale, L_{IDDES} blending function f_d , and the DES length scale, L_{DES}. Note that P_k is the production term, ν and ν_t are the kinematic viscosity and turbulent viscosity respectively, A, B, C, $\sigma_k, \sigma_\omega, \sigma_{\omega 2}$ are model coefficients and F_1 is a blending function. L_{RANS} refers to the RANS length scale and C_{d1}, C_{d2}, r_d are constants. To 'protect' the near-wall region, $f_d \sim 1$ where RANS is mainly used while for the core fluid flow, $f_d \sim 0$. The SST model is

defined as:

$$\frac{dk}{dt} + U_j \frac{dk}{dx_j} = P_k - Ak\omega + \frac{d}{dx_j}\left[(\nu + \sigma_k \nu_t)\frac{dk}{dx_j}\right] \tag{23}$$

$$\frac{d\omega}{dt} + U_j \frac{d\omega}{dx_j} = \frac{B\omega}{k} P_k - C_{\omega^2 + \frac{d}{dx_j}\left[(\nu+\sigma_\omega \nu_t)\frac{d\omega}{dx_j}\right]}$$

$$+ 2(1 - F_1)\frac{\sigma_{\omega_2}}{\omega}\frac{dk}{dx_j}\frac{d\omega}{dx_j} \tag{24}$$

And LES sub grid scale model is:

$$\nu_{sgs} = (C_s \Delta)^2 |\bar{S}| \tag{25}$$

where C_s is the Smagorinsky constant, Δ is the grid filter width, and $|\bar{S}|$ is the magnitude of the strain rate tensor. IDDES length scale is:

$$L_{IDDES} = f_d L_{RANS} + (1 - f_d) L_{LES} \tag{26}$$

$$f_d = 1 - \tanh\left[(C_{d1} r_d)^{C_{d2}}\right] \tag{27}$$

where $L_{RANS} = \sqrt{k}/0.09\omega$ and $L_{LES} = C_{DES}\Delta$; More detailed information on the constants used can be found in this study (Gritskevich et al. (2012)). Direct Numerical Simulation (DNS) is the most accurate approach for simulating turbulent flows, as it resolves all turbulence scales directly without any modeling assumptions. DNS solves the full Navier-Stokes equations for all scales of motion, from the largest energy-containing eddies down to the smallest dissipative structures. This results in highly detailed insights into turbulence mechanics and serves as a benchmark for developing and validating turbulence models. However, DNS is prohibitively expensive for most practical flows due to the enormous computational demands, scaling roughly with the Reynolds number to the power of nine-halves ($Re^{3/2}$), making it feasible only for low

to moderate Reynolds number flows or highly simplified geometries.

2.3 Finite Volume Method (FVM) for Spatial Discretization

It is important to understand how the computational domain is discretized and the mathematical background behind various algorithms that are employed in such simulations. A common spatial discretization procedure together with the Finite Volume Method (FVM) can be found in most CFD programs. In the FVM, the computational domain is discretized into multiple control volumes within the computational grid. These are commonly called 'cells', where the general transport equation (which represents the conservation of energy, mass and momentum) are integrated over. The figure below in Figure 10 illustrates the key components used in typical finite volume method. The central node represents the location where the primary variable is calculated for a specific control volume. The adjacent nodes are in the neighbouring control volumnes whose values are used in the discretization of the flux terms across the entire control volume faces.

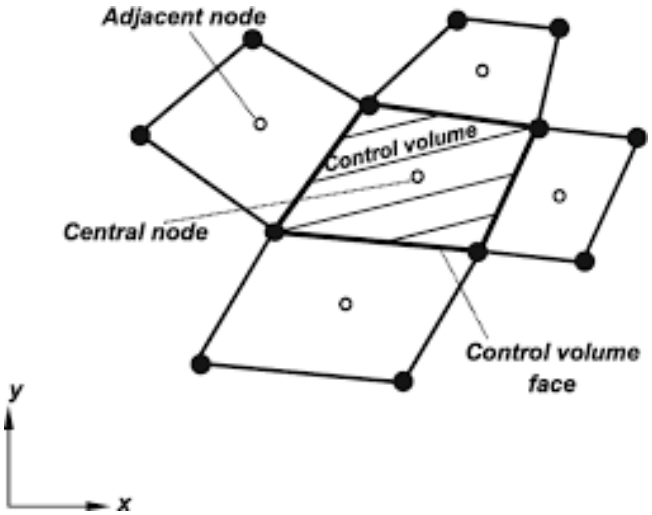


Figure 10: Control volume in unstructured meshes adapted from Dincer and Ezan (2018)

The transport equation which are integrated over each cell face can be found below.

$$\text{Rate of change} + \text{Convection} = \text{Diffusion} + \text{Source}$$

$$\frac{d}{dt}(\rho\Theta) + \nabla \cdot (\rho \mathbf{u}\Theta) = \nabla \cdot (\Gamma_\Theta \nabla\Theta) + S_\Theta \tag{28}$$

The Gauss' Divergence theorem is then applied to the transport equation to convert the volume integrals of both convective and diffusive terms to surface integrals. This transformation can be expressed below. (Note that terms used will be elaborated more in the subsequent sub-sections).

$$\int \frac{d}{dt}(\rho\Theta)dV + \int (\rho \mathbf{u}\Theta) \cdot \mathbf{n}\, ds = \int \Gamma_\Theta \nabla\Theta \cdot \mathbf{n}\, ds + \int S_\Theta\, dV \tag{29}$$

Within Star-CCM+, surface integrals are computed by evaluating variables at specific points on cell faces. These values are estimated using interpolation methods that rely on data from the centers of cells. For volume integrals, the software calculates the product of the average source term at the cell center and the cell's volume. The subsequent sections provide detailed descriptions of the discretization schemes applied to each term in Star-CCM+.

2.3.1 Diffusion Term

For a typical cell undergoing discretization, the diffusive term passed through the current face f of the cell can be expressed as:

$$D_f = (\Gamma \nabla\Theta \cdot \mathbf{n})_f \tag{30}$$

Where 'n' represents the surface area vectors of the current face 'f', Γ is the face diffusivity, $\nabla\Theta$ is the gradient of fluid property and the subscript 'f' represents the surface between cells. for an interior face can be decomposed via a second-order expression of cell values Θ_0 and Θ_1. This can be expressed as:

$$\nabla\Theta_f = (\Theta_1 - \Theta_0)\left(\frac{\mathbf{n}}{\mathbf{n} \cdot ds}\right) + \left(\frac{\nabla\Theta_0 + \nabla\Theta_1}{2}\right) - \left(\left(\frac{\nabla\Theta_0 + \nabla\Theta_1}{2}\right) \cdot \mathbf{n}\right)\left(\frac{\mathbf{n}}{\mathbf{n} \cdot ds}\right) \tag{31}$$

Where ds is the vector between the cells' centroids 0 and 1.

2.3.2 Convection Term (MUSCL 3rd Order Scheme)

The convective term passed through the current face 'f' of the cell can be expressed as:

$$(\rho \mathbf{U} \cdot \mathbf{n})_f = (\dot{m}\Theta)_f = \dot{m}_f \Theta_f \tag{32}$$

Where $\dot{m}_f$ is the mass flow rate at the call face. The IDDES-SST solver in Star-CCM+ uses the hybrid MUSCL 3rd-order central-differencing scheme for the discretized convective term. This method exhibits lower dissipation compared to both second order and bounded central-difference approaches. It is versatile and effective for both steady and unsteady flows, ranging from incompressible to high-speed compressible conditions. The scheme's boundedness is controlled by a parameter ξ. In regions with non-smooth flow, the method defaults to a first-order scheme, denoted as Θ_{1st}. Conversely, in areas with smooth flow, it employs a combination of a third-order MUSCL upwind scheme (Θ_{MUSCL3}) and a third-order central-difference reconstruction scheme (Θ_{CD3}). The convective flux can be calculated with the equation below for certain settings:

$$(\dot{m}\Theta)_f = \Theta_{1st}, \text{ if } \xi < 0, \xi > 1$$

$$(\dot{m}\Theta)_f = \dot{m}(\Theta_{MUSCL3}\sigma_{MUSCL3} + (1 - \sigma_{MUSCL3})\Theta_{CD3}), \text{ if } 0 \leq \xi \leq 1 \tag{33}$$

Where σ_{MUSCL3} is a user defined value. The Θ_{MUSCL3} relies solely on the value of ξ for incompressible flows in order to ensure the boundedness of the scheme.

2.3.3 Time Discretization

In STAR-CCM+, temporal discretization is crucial for accurately capturing transient phenomena in CFD simulations. The software offers both first order and second-order temporal discretization schemes, with the latter providing enhanced accuracy for unsteady flow analysis. The second-order temporal discretization in STAR-CCM+ employs a three-time-level scheme, which utilizes information from the current, previous, and next time steps to achieve higher accuracy. This method is particularly beneficial for simulations where precise time-dependent behavior is essential. The general form of the second-order backward differentiation formula

(BDF2) used in STAR-CCM+ can be expressed as:

$$\frac{3\psi^{n+1} - 4\psi^{n} + \psi^{n-1}}{2\Delta t} = f(\psi^{n+1}) \tag{34}$$

Where ψ represents the quantity of interest, n represents the time step, Δt represents the time step size and $f(\psi^{n+1})$ refers to the function of the discretized form evaluated at the next time step. This implicit scheme ensures numerical stability, allowing for larger time steps compared to explicit methods.

2.4 Mesh Discretization

Mesh discretization is a crucial element in CFD simulation, involving the division of a continuous fluid domain into discrete elements or cells to facilitate numerical analysis. The choice of mesh type and quality directly impacts the accuracy, convergence, and computational efficiency of simulations. These meshes are generally categorized as structured or unstructured. Structured meshes feature elements arranged in a regular, grid-like pattern, allowing for direct addressing of neighboring nodes. This regularity often results in lower computational costs concerning CPU and memory usage (Hirsch (2007)). However, generating structured meshes for complex geometries can be labor-intensive and may lead to unnecessary refinement in certain regions, causing inefficiencies.

Conversely, unstructured meshes, composed of elements like triangles in 2D or tetrahedra in 3D, offer greater flexibility in accommodating complex geometries (Baker, 1991). The figure seen in Figure 11 (adapted from Michalcová and Kotrasová (2020)) illustrates the various types of mesh which are commonly found and used in most CFD softwares. Automated meshing algorithms, such as Delaunay triangulation and advancing front methods, enable rapid mesh generation and localized refinement. Delaunay triangulation is a mesh generation method that creates triangles (or tetrahedra in 3D) such that no mesh point lies inside the circumcircle of any element, resulting in well-shaped, non-overlapping elements that improve numerical stability. On the other hand, advancing front methods generate a mesh by progressively adding new elements along the "front" of an existing boundary, allowing for local control of element size and shape, which is particularly useful for capturing complex geometries with high accuracy. Despite their adaptability, unstructured meshes often require indirect addressing and storage of

connectivity data, which can increase computational overhead. A notable type of unstructured mesh is the polyhedral mesh, which has gained attention for its advantages in CFD simulations. Polyhedral meshes typically result in a lower cell count compared to tetrahedral meshes, leading to reduced computational time and memory usage. They also provide enhanced accuracy and convergence properties, especially in complex geometries (Iqbal and Chan (2015)). Some works have demonstrated the effectiveness of polyhedral meshes in accurately simulating turbulent flows, such as thermal mixing in T-junctions. For simulations involving wall-bounded turbulent flows, setting the prism layer is crucial. The Prism Layer Mesher in STAR-CCM+ generates orthogonal prismatic cells adjacent to wall surfaces, enhancing the resolution of velocity and thermal gradients within the boundary layer. This approach improves the overall accuracy of the flow solution near walls. Achieving a balance between mesh resolution and computational cost is essential. Local mesh refinement allows for increased resolution in regions with high gradients or complex flow features, while coarser meshes can be used elsewhere to conserve resources. Conducting a mesh sensitivity study, or grid independence study, ensures that the simulation results are not duly influenced by the mesh size, thereby validating the accuracy of the solution.

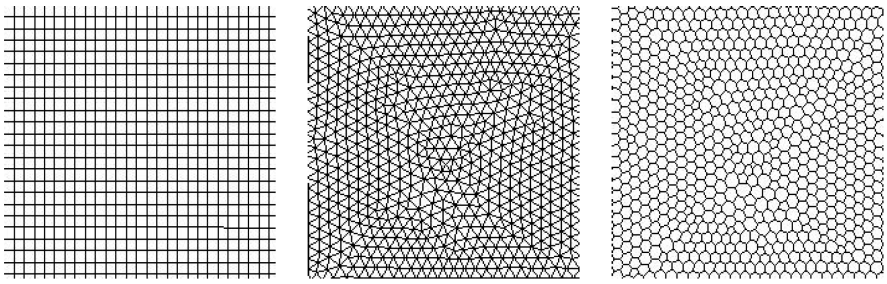


Figure 11: Three commonly used mesh types in CFD softwares: (a) Hexahedral, (b) Tetrahedral, (c) Polyhedral adapted from Michalcová and Kotrasová (2020)

2.4.1 Wall Treatment Methods

Accurately modeling the behavior of fluid flows near solid boundaries is essential due to the significant impact of boundary layers on overall flow characteristics. The near-wall layer is a thin region adjacent to the wall where velocity gradients are steep, transitioning from zero at the wall (due to the no-slip condition) to the free stream velocity over a short distance. Properly capturing physics within this layer is crucial, as it influences drag, heat transfer, and the onset of turbulence (Craft et al. (2002); Sibil (2024)). However, resolving the near-

wall region with fine mesh grids to capture these steep gradients directly is computationally expensive, especially in turbulent flows. To address this challenge, one common approach is the use of wall functions. Wall functions bridge the gap between the fully turbulent flow in the core region and the viscous-affected flow near the wall. They apply empirical formulas to approximate the velocity profile and other flow variables in the near-wall region, thus avoiding the need to resolve the viscous sublayer explicitly. This method is computationally efficient but may lack accuracy in flows with strong pressure gradients or separation.

The selection of an appropriate wall treatment is critical in CFD simulations, as inaccuracies in modeling the near-wall region can lead to significant errors in predicting flow separation, drag, heat transfer rates, and overall fluid behavior. Within Star-CCM+, there are different wall treatment options offered such as Low Y^+, High Y^+ and All Y^+ wall treatment options. For a Low Y^+ wall treatment, it resolves the boundary layer by employing a fine mesh, positioning the first cell centre within the viscous sublayer at a dimensionless wall distance, Y^+ of approximately 1. This method aligns with traditional low Reynolds number modelling, capturing detailed near-wall physics but requiring substantial computational resources due to the dense mesh. A High Y^+ wall treatment utilizes coarser meshes, placing the first near-wall cell within the logarithmic region of the boundary layer, typically where Y^+ exceeds 30. This approach corresponds to conventional wall-function methods, which apply empirical formulas to approximate near-wall behaviour, reducing computational demands but potentially sacrificing accuracy in complex flow scenarios. To accommodate varying mesh densities, the all-Y^+ wall treatment dynamically adjusts based on local mesh resolution. It functions as a low-Y^+ treatment in finely meshed regions and as a high-Y^+ treatment in coarser areas. For intermediate mesh densities, it blends the two approaches, ensuring consistent and accurate modelling across the simulation domain. This versatility makes the all-Y^+ treatment particularly suitable for simulations where mesh quality varies or in complex geometries.

2.5 Concluding Remarks

This chapter has outlined the numerical framework and modeling strategies used to simulate turbulent flow within a T-junction geometry. The selection of a hybrid RANS-LES approach using the IDDES model was justified based on its ability to resolve large turbulent structures while maintaining computational feasibility. Detailed attention was given to the mesh config-

uration, including grid independence testing and the implementation of boundary and initial conditions relevant to high-fidelity thermal mixing simulations.

The solver setup, post-processing methods, and time-averaging strategies were also documented to ensure that the results are both physically meaningful and reproducible. This numerical foundation forms the basis for the investigations presented in the following chapters, where mesh adaptation and surrogate techniques are applied to the analysis of results.

Chapter 3

Description of Computational Fluid Dynamics Model

The overarching physical phenomenon driving this study is thermal fatigue induced by turbulent mixing in nuclear T-junction pipe systems. Thermal fatigue is a structural degradation process caused by cyclic thermal loading, often arising in regions where cold and hot fluid streams interact and generate fluctuating thermal fields near pipe walls. In T-junction geometries, the unsteady mixing layer between the main and branch flows leads to temporal temperature fluctuations at the wall, which in turn impose cyclic thermal stresses on structural materials. If not resolved accurately, these stress cycles can be underestimated, leading to unanticipated fatigue failure. To simulate and assess these physics, it is essential to resolve the local flow structures and temperature gradients responsible for such unsteady thermal loading. Therefore, the objective of this chapter is to evaluate whether the computational mesh and refinement strategies are sufficient to capture the turbulent flow behavior. To quantify the fidelity of simulations in capturing thermal fatigue-relevant physics, the following Quantities of Interest (QoIs) are considered: Wall-normal temperature gradient and mean streamwise velocity profiles and velocity gradients. The base geometrical and numerical settings used in the CFD model are introduced in section 3.1. To verify that the mesh captures the physical quantities relevant to thermal fatigue, a mesh convergence study was performed using progressively refined meshes. Two methods for mesh generation are compared. The first is comparing mesh grid discrepancies by using conventional error estimates in section 3.2 and adaptive mesh refinement based on

dual-weighted error estimates in section 3.3. The mathematics behind these two approaches are presented in greater detail. This is then supplemented with results and analysis for the mesh sensitivity analysis to determine the most ideal mesh settings to be used for the rest of this project. This chapter then concludes with a conclusion found in section 3.4.

3.1 Model Settings

3.1.1 Geometrical and numerical settings

Before exploring the details of the CFD models, it is necessary to introduce the momentum ratio defined in the paper by Hosseini et al (Hosseini et al. (2008)). The momentum ratio is the ratio between the momentum in the branch pipe, M_b and momentum in the main pipe, M_m and can be expressed as:

$$M_r = \frac{M_m}{M_b} \tag{35}$$

where $M_m = D_m D_b \rho_m V_m^2$, and $M_b = \frac{\pi}{4} D_b^2 \rho_b V_b^2$. It should be noted that D_b and D_m are the diameters of the branch pipe and main pipe, respectively. Since constant fluid properties (density as $997 kg/m^3$ and viscosity as 8.89 E-4 Pa-s) are applied in all cases, the momentum ratio can be further simplified as $M_b = \frac{\pi}{4} \frac{D_m}{D_b} \frac{V_m}{V_b}^2$. Based on the values of M_b it is possible to classify the branch to main pipe flow pattern in impinging jet, turn jet, re-attached jet, and wall jet (Hosseini et al. (2008)). For context, an impinging jet occurs when a fluid jet strikes a 'surface', which can be a body of fluid, perpendicular to its flow direction, whereas a deflecting jet occurs when the fluid jet strikes the 'surface' at an angle which causes deflection. Wall jets are simpler to visualise where they occur when a fluid moves along the 'surface' with a high velocity gradient. The categorization of the branch jet in the model can be found in Table 2 below.

Table 2: Classification of branch jet in T-junction with an upstream elbow (Hosseini et al. (2008)).

Jet Type	**Momentum Ratio (M_R)**
Wall jet	$M_R > 90$
Re-attached jet	$20 < M_R < 90$
Turn jet	$2.5 < M_R < 20$
Impinging jet	$M_R < 2.5$

The T-junction in this study is formed by attaching a 10 D_b long branch pipe to a 90° curved main pipe as illustrated in Figure 12. D_b will vary in the different cases, (Table 2), whilst D_m remains 0.108 m in all cases. The curvature radius of the bend, R_c, is kept as approximately 1.4 D_m, and the main pipe's inlet is positioned 2.0 D_m upstream from the bend while the branch pipe centre line is fixed at at 2.0 D_m downstream from the centre of curvature of the bend. Moreover, the computational domain extends for 10 D_m downstream from the centre of the junction in all cases.

The inlet flow conditions of the main pipe in all cases are the same, with a bulk velocity and temperature ($T_{main,inlet}$) of 0.89 m/s and 19 °C, respectively, resulting in a Reynolds number (Re) of 107,893. The branch pipe inlet temperature, ($T_{branch,inlet}$) is set at 26.9 °C in all cases, and additional inlet flow conditions for all cases are detailed in Table 3. The choice of these temperature values is crucial as the temperature difference is very similar to what is normally observed in coolant pipes that connect the SG to the PRV and the RPV as discussed previously. It is worth noting here that M_R in this study involves manipulating either the branch pipe diameter (D_b) or the branch inlet bulk velocity (V_b). In Cases 1, 2, and 3, M_R decreases from 137.3 to 66.4 as D_b gradually enlarges from 0.015 m to 0.031 m. Meanwhile, the Reynolds number of the branch pipe inlet (Re_b) rises from 3860 to 7094 in Cases 1, 2, and 3. Case 4 and Case 5 serve as the control group, while the M_R is manipulated by elevating Vb. With Vb reaching 0.27 m/s and 0.33 m/s in Case 4 and 5, MR changes from 98 to 66.4, resembling the values in Cases 2 and 3. However, for the equivalent M_R, Re_b, and Mb in the latter two cases (Cases 4 and 5) are notably smaller than those in their counterparts (Cases 2 and 3). For instance, the Re_b is 5550 in Case 5 compared to 7094 in Case 3, and correspondingly, the mass flow rate of branch pipe flow (M_{branch}) in Case 5 is lower than that in Case 3 as well. A similar conclusion can be drawn when comparing Re_b and M_{branch} in Case 4 to those in Case 2.

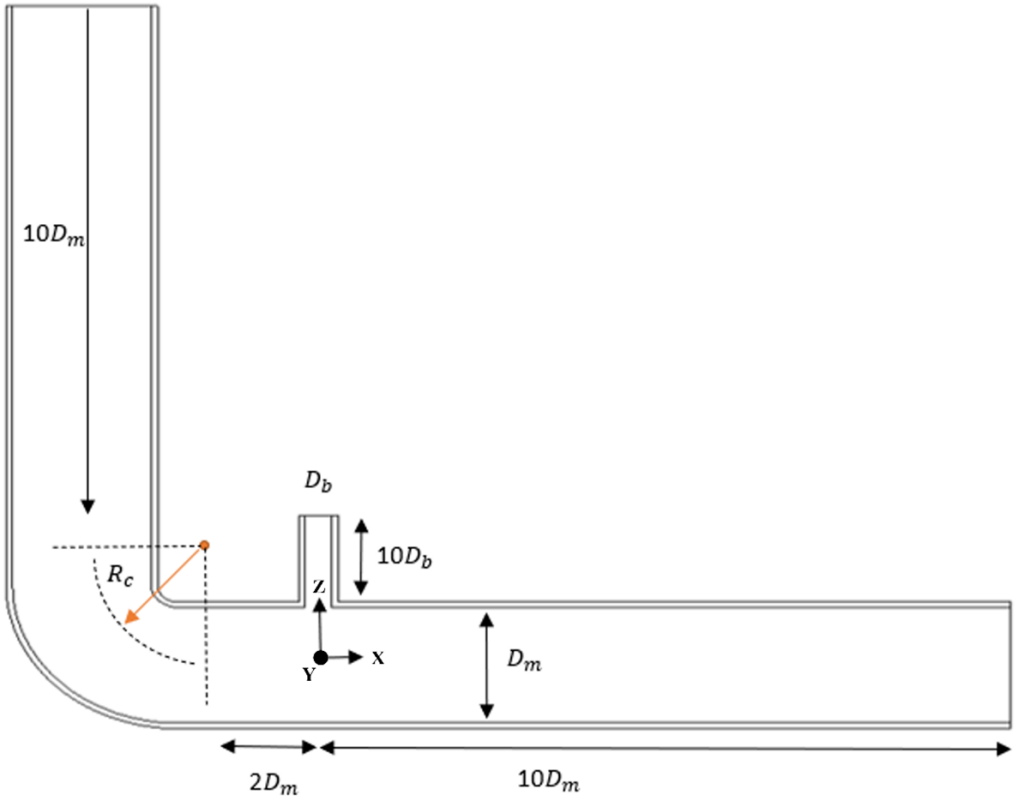


Figure 12: Side elevation view of the computational domain for the T-junction test case with upstream bend.

Table 3: Characteristics of the T-junction with various M_R and Reynolds number.

	Case 1	**Case 2**	**Case 3**	**Case 4**	**Case 5**
D_m (m)	0.108	0.108	0.108	0.108	0.108
D_b (m)	0.015	0.021	0.031	0.015	0.015
$\frac{D_b}{D_m}$	0.015	0.021	0.031	0.015	0.015
V_b (m/s)	0.23	0.23	0.23	0.27	0.33
Re_b	3860	5410	7904	4568	5550
$T_{branch,inlet}(°C)$	26.9	26.9	26.9	26.9	26.9
$T_{main,inlet}(°C)$	19	19	19	19	19
M_b	9.35E-03	1.83E-02	3.99E-02	1.29E-02	1.92E-02
M_R	137.3	98.0	66.4	98.0	66.4
$\dot{M}_{\text{mean}}$ (kg/s)	0.0406	0.0795	0.173	0.0476	0.0582
Pr	6.1	6.1	6.1	6.1	6.1
Fr	0.6	0.51	0.42	0.7	0.86

The numerical simulations are conducted using the commercial CFD software Simcenter STAR-CCM+ (Siemens Digital Industries Software (Siemens 2023)). The turbulent flow in the fluid domain is modelled with the IDDES-SST turbulence model (Gritskevich et al. (2012)). This is a hybrid turbulence modelling approach that combines elements of Reynolds-Average Navier

Stokes (RANS) and Large Eddy Simulation (LES) models. The governing equation for the fluid flow is the Navier Stokes equation and it will be solved within the turbulence model. The IDDES-SST formulation and working fluid conditions can be found in the previous chapter, in chapter 2. The computational domain is meshed using the polyhedral and prism layer mesh generation algorithms within Star-CCM+. The MUSCL 3rd-order central difference spatial discretization scheme is utilized. The time step size is prescribed as 0.001 s, with the resultant Courant-Friedrichs-Lewy (CFL) numbers ($\frac{u\Delta t}{\Delta x}$) smaller than 0.45 in all cases.

3.1.2 Boundary Conditions

While treating the walls as adiabatic, the fully developed inlet flow profile for both the main and branch pipe are obtained using inlet flow generators. The inlet flow generators simulate flow within a straight pipe with the turbulent flow modelled by the RANS $k-\omega$ SST turbulence model. It is worth noting that while a study based on a similar geometry utilised the Synthetic Eddy Method (SEM) to generate inlet conditions, it is still largely based on RANS statistics and the turbulence model used is a wall-resolved LES (Tunstall et al. (2016)). That itself requires turbulence in the inlet flow conditions. Since the main and branch inlet pipes are straight and assumed to be adiabatic with a no-slip condition and the fact that it is computationally demanding to run SEM for the inlet flow generator, employing RANS instead is an appropriate choice. The computational domain of the inlet flow generator is obtained by volume extrusion of the surface mesh of the T-junction inlets. Values of velocity, turbulent kinetic energy and specific dissipation rate are extracted at approximately 41 and 9 diameters (D_m) away from this inlet generator and they have been used as boundary conditions at the inlets of the T-junction model.

3.2 Mesh Sensitivity Analysis (Conventional)

To validate the suitability of the mesh for the study, a mesh sensitivity study is conducted by running and comparing the outputs of CFD simulations with three distinct mesh configurations under the base case (Case 2) flow condition.

3.2.1 Results and Discussion

The meshes are generated using the STAR-CCM+ package. Figure 13 depicts the mesh grids for these configurations on the cross-section perpendicular to the streamwise direction. In particular, the base cell sizes for these meshes are set as 5 mm, 3 mm and 2 mm, respectively. To accurately capture near-wall flow and the rapid changes in velocity gradient, prism layers are utilized to refine the near-wall mesh, maintaining a fixed total thickness of 7 mm across all three meshes. This corresponds to the boundary layer where viscous forces dominate. Using prism layers allow for a more precise representation of the velocity profile within the near-wall region. Within the prism layer, a consistent configuration of 20 mesh layers is employed, with a fixed growth rate of 1.1. The total number of mesh elements for these meshes are 1.45 M, 4.01 M, and 9.23 M respectively. It should be noted that the Y^+ values are less than 1 for all generated meshes as the first grid wall-spacing, Δs for each model is has been set to 30% of the base cell size which are 0.0015, 0.0009 and 0.0006 respectively. This results in Δs values well lower than the maximum allowable wall-spacing, $\Delta s_{max} = \frac{(y^+ \mu)}{\sqrt{\frac{\tau_{wall}}{\rho}}(\rho)} = 0.011$ ($Y^+ > 1$ beyond this threshold).

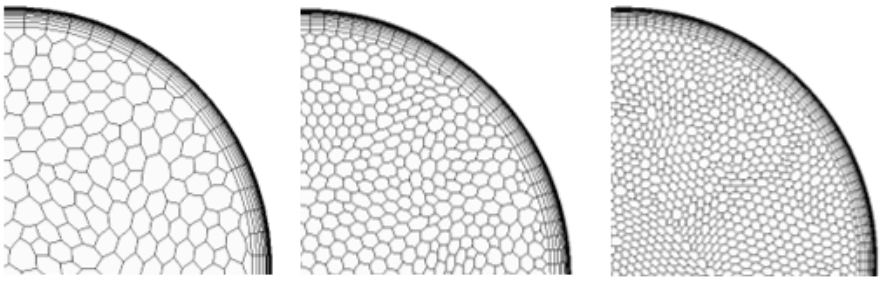
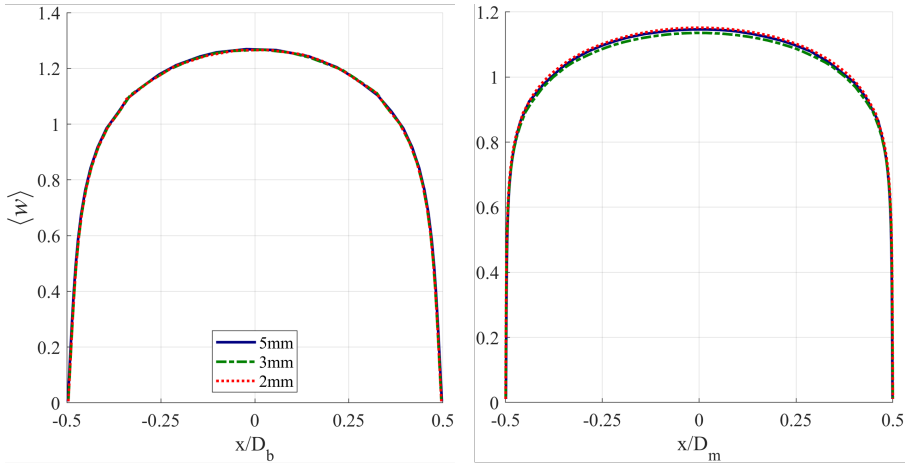


Figure 13: Mesh domains in the main pipe with a base size of (a) 5 mm, (b) 3 mm, (c) 2 mm.

The fully developed flow profiles produced by the inlet generators are imposed at both the branch and main pipe inlets using the external file table function in STAR-CCM+ (Siemens Digital Industries Software (Siemens 2023)). Figure 14 shows the flow profiles of the inlet conditions that are found at the entrance of both main and branch inlets. It can be seen that the profiles are similar for all meshes and take on a fully developed flow profile. To further confirm this, these profiles are also complemented with law of the wall plots for each respective inlet, seen in Figure 15. It is clearly shown in this figure that all the inlet velocity profiles match well with the log-law. It is again indicated that the fully developed flow profiles are achieved

at the inlet in each mesh sensitivity simulation.

To examine the suitability of the mesh, mean streamwise velocity,⟨u⟩ profiles within the same test case (Case 2) with different mesh densities are plotted at several locations of interest, as shown in Figure 16. The velocity profiles along the vertical plane (X-Z) are included in Figure 17, where w is normalised with the bulk velocity at the main pipe V_m where $V_m = 0.89m/s$. The flow profiles obtained with the 5 mm mesh in the upstream region displays some differences compared to the other two meshes. These differences are mainly in the near-upper wall region while this behaviour is not as pronounced downstream of the flow, after the T-junction exit. The results also show some slight differences compared to Tunstall et al's (Tunstall et al. (2016)) velocity data. This behaviour has been reported in Joksimovic et al which also utilised the IDDES-SST model in their work (Joksimovic et al. (2023)). This can be attributed with the behaviour of the IDDES solver that is used where RANS – LES switching occurs further away from the wall.



(a) (b)

Figure 14: Fully developed flow profile based on mean velocity at (a) branch inlet and (b) main inlet.

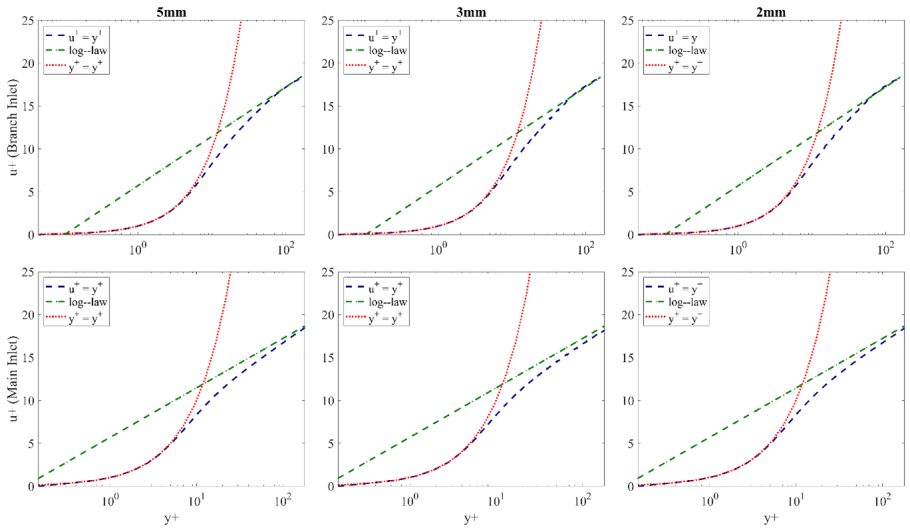


Figure 15: Law of the wall plot for different mesh size (universal velocity profile)

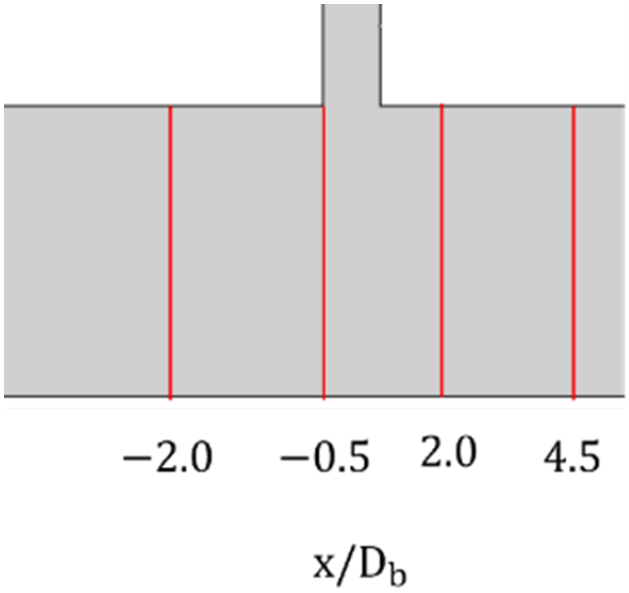


Figure 16: Cross sectional locations of interest for analysis, using parameters in Table 2 as the base test case.

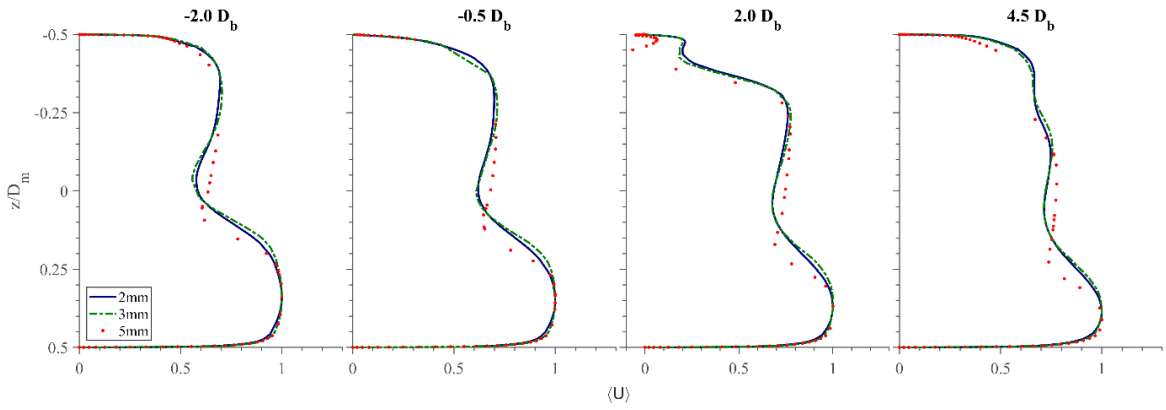


Figure 17: Normalised mean streamwise velocity for the different mesh resolutions.

The f_d of the IDDES blending function are used to show the resolved and modelled regions in the mesh sensitivity study simulations, which are illustrated in Figure 18 for the X-Y symmetry plane. It should be noted that f_d determines the extent to which the RANS or LES length scale is used, based on the local flow conditions. Therefore, the 'subgrid scale' involved in the IDDES-SST is a simply a blend of the SST model and subgrid scale stress model of the LES as dictated by the value of f_d. A value of 1 which is represented with red, refers to a 100% RANS in the respective regions. On the other hand, a value of zero, which is represented with blue, refers to a 100% LES employed in the respective regions. In the upstream region, before the T-junction, it can be seen that the switching from RANS to LES takes place in the near wall region, with the switching happening further away from the T-junction in Figure 18(c), followed by Figure 18(b) and Figure 18(a). It is reasonable to conclude that the results generated using mesh with base size as 2mm are reasonably accurate, followed by using 3 mm and 5 mm for the base mesh size. While the results produced by using 3mm shows some similarities with 2 mm, the latter mesh base size still resolves near wall flow better than the model with 3mm used as the base mesh size.

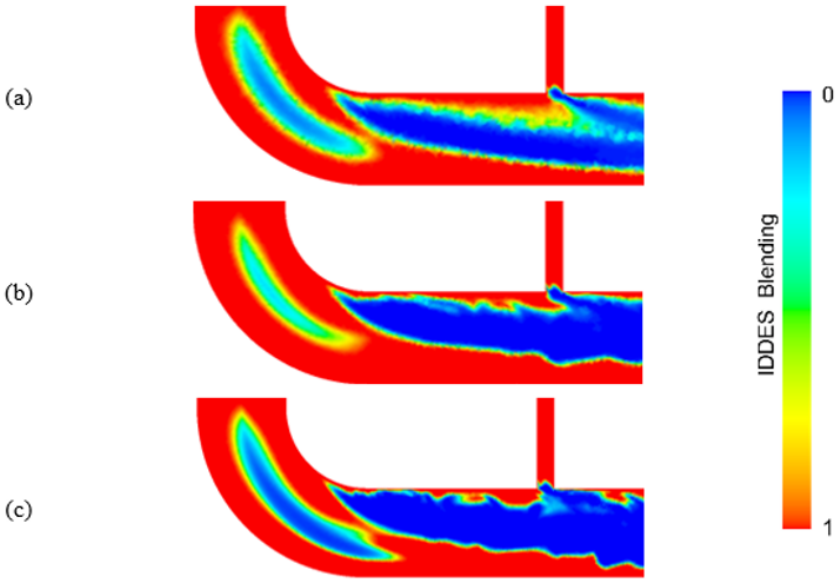


Figure 18: The contour of IDDES-SST blending function (utilised to estimate of RANS and LES regions) for mesh sensitivity simulation with base mesh size of (a) 2 mm, (b) 3 mm & (c) 5mm

3.3 Mesh Sensitivity Analysis (Adaptive Mesh Refinement)

Results from the previous section shows that a model with a higher mesh density produces results within reasonable accuracy. However, it requires more computational resources due to the high cell count. One alternative method that offers a balance between computational cost and accuracy is the use of automatic mesh refinement or adaptation method. This sub-section explores an alternative meshing approach available in STAR-CCM+, which is the adaptive mesh refinement (AMR) and compares the performance of refined models against the baseline model with a uniform 2mm base mesh used in the previous section. The goal is to assess whether AMR can provide improved resolution of key physical phenomena while maintaining computational efficiency. The comparison focuses on quantities relevant to thermal fatigue, including wall temperature fluctuations and velocity gradients near the mixing interface.

The AMR solver in STAR-CCM+ employs an algorithm that dynamically splits parent cells into smaller sub-cells in regions of interest. Conversely, it can also coarsen refined regions by merging sub-cells back into their parent cell, enabling mesh flexibility during the simulation. This adaptivity is guided by refinement criteria, which may be based on traditional gradient magnitudes or advanced adjoint-based output error estimates, where the mesh is iteratively refined in areas that most influence the error in a target quantity of interest. In this study, both velocity gradients and temperature gradients were used as refinement indicators. Velocity gradients help identify zones of high shear and turbulence such as mixing layers and recirculation zones where convective mixing contributes to breakdown of thermal stratification. Temperature gradients, meanwhile, directly pinpoint regions of sharp thermal variation and stratification, which are primary contributors to cyclic wall temperature fluctuations and thermal fatigue. By using both gradient fields, the AMR strategy targets regions where flow dynamics and heat transfer interact most intensely, especially near the wall. This dual-indicator strategy ensures that mesh refinement is physically meaningful and focused, thereby improving the model's ability to resolve thermal fatigue-relevant behaviors without unnecessary computational cost.

3.3.1 Mathematics of Conventional and Goal-based error estimates for AMR

The mathematics of both conventional and goal-based error are introduced in this subsection. The adjoint error estimate relates to approximation, truncation, and residual errors by quantifying their impact on a specific output or quantity of interest. Truncation error introduces local errors, which lead to residuals in the discretized equations. The adjoint solution acts as a sensitivity map, projecting these residuals onto the output functional to estimate how much the numerical approximation error actually affects the final result. In general, the total error consists of approximation (traditional) error, truncation error and residual error. Traditional grid error estimates measure the difference between the exact solution of a differential equation and its numerical approximation on the grid. It can be expressed as:

$$\varepsilon_h(x) = u(x) - u_h(x) \tag{36}$$

$$u_h(x) \to u(x) \quad \text{as} \quad h \to 0 \tag{37}$$

Where $u(x)$ is the exact solution and $u_h(x)$ is the numerical solution on a grid, with spacing parameter, h. Since h also indicates the quality of approximation, Equation 37 holds true when $h = 0$, which indicates a perfect, error free estimate versus a true result. This is also known as the approximation error in some literature. Truncation error, a subset of approximation error, arises specifically from approximating an infinite process by a finite one. For instance, when using numerical methods to solve differential equations, truncation error measures the discrepancy introduced by approximating derivatives or integrals. This can be expressed as:

$$T(x) = \frac{h}{2}u''(x) \tag{38}$$

Residual error on the other hand measures how well a numerical solution satisfies the original differential equation. Given a differential operator L and an approximate solution, $u_h(x)$, the

residual error can be expressed as:

$$R(x) = L(u_h(x)) - S \tag{39}$$

Where S is the source term. To obtain approximation error estimates, there are methods available such as convergence analysis, Richardson Extrapolation and a posteriori error estimation (Ainsworth and Oden (1997); Ferziger and Perić (2002); Haldar and Mahadevan (2000); Rangavajhala et al. (2011)). Richardson extrapolation estimates the exact value of a solution by combining numerical results at different mesh resolutions. It assumes the error decreases predictably and uses this to cancel leading-order error terms. This technique improves solution accuracy without refining the mesh further. It is worth noting that only Richardson Extrapolation can provide an estimation for discretization errors. a posteriori error estimates can be used on discretization errors but only with the use of a surrogate model as these methods intrinsically quantify the error arising from the use of the surrogate model (Babuška and Rheinboldt (1978); Demkowicz et al. (1984)). A method used to obtain truncation error estimates is the a priori error analysis, which can be expressed as:

$$||u - u_h|| \leq Ch^p ||u||_{H^{p+1}} \tag{40}$$

Where C is a constant which is independent on the mesh size, h and $H^{p+1}(\Omega)$ represents the Sobolev norm of the exact solution over the domain Ω. A Sobolev norm measures both the size of a function and its derivatives, capturing smoothness and regularity. It's commonly used in PDE theory and finite element methods to quantify error in function spaces. Unlike simple L^2 norms, Sobolev norms account for how "rough" or "smooth" a solution is. On the other hand, residual error estimates can be obtained via a posteriori analysis. a posteriori error analysis relies on the stability properties of A which is readily available from regularity theory, which studies how smooth solutions to PDEs are based on the properties of the equations and boundary conditions, which means that this does not pose many issues. A posteriori analysis

can be expressed as:

$$||u - u_h|| \leq C||R|| \tag{41}$$

Where the aim is to correlate the error $||u - u_h||$ to the residual $||R||$.

While a priori analysis provides theoretical error bounds based on the numerical method and the regularity of the exact solution and a posteriori analysis utilizes the computed numerical solution to assess the actual error, these methods are disadvantaged in some ways. For instance, a priori error analysis is much dependent on the assumptions on the stability of the Sobolev norm, which is a difficult task for any approximation. Additionally, it can get quite complex in computing the truncation error for most practical applications. This is an issue since true/actual outputs can be highly sensitive to either residual or discretization errors in regions which is not easily identifiable even with a priori analysis. Goal-oriented error or output-based error estimates based on adjoint analysis helps to overcome these known issues via quantification on the residual errors that affect the output while considering the effects due to propagation during the process. This method measures the discretization error within the computational model. The next section discusses on how this method works and its various applications.

The Dual Weighted Residual (DWR) method links residuals to output error for variational discretization. This allows for arbitrary quantities of interest through appropriate adaptation of the computational mesh to be obtained. The main driver behind this mesh adaptation method is the control of a physically relevant goal quantity instead of the traditional energy-norm or $L^2 - norm$. Norms are defined such that the averaged errors over the entire computational domain or part thereof can be calculated. It transforms a PDE into a system of algebraic equations by projecting the residual onto the space spanned by basis functions. The Galerkin method, which is a weighted residual approach where the trial and test functions are taken from the same function space, gives the minimum error in the energy norm, for self-adjoint forms. In the continuous adjoint approach, the adjoint equation is formulated at the differential equation level, which is then discretized (linearized then discretized).

On the other hand, the discrete adjoint approach usually begins with a set of discretized equations which then formulates the respective discrete adjoint equations (discretized then linearizes). This goal-oriented analysis can be used together with the AMR seen in various

papers (Fidkowski and Darmofal (2009-01); Fabiano and Mavriplis (2016)). All these are used within the built-in adjoint solver, which allows for adjoint error estimates to be generated once the adjoint solution is obtained and the local discretization error can be measured based on the cost function used. Discrete adjoint methods are used where the flow solver discretizes equations and adjoint equations are then derived algebraically. Adjoint methods have been widely used in many applications such as design optimization (Fabiano and Mavriplis (2016); Jameson and Martinelli; Nielsen and Anderson (2002); Elliott and Peraire (1997); Lohner et al. (2003)). Since the continuous approach is more customizable at the discretization stage, less computational cost is required for such formulations, which is a huge advantage. This approach depends on a space-time variational formulation of the discrete problem, where a posteriori error estimates are found with the aid of duality techniques (Hartmann et al. (2011)). Such duality techniques are known from a priori error analysis as suggested by various works (Eriksson and Johnson (1991, 1995); Heywood and Rannacher (1982)).

The main goal behind adjoints is to optimize an objective D with respect to design parameters L. This requires the sensitivity of D and L, which can be expressed in Equation 42 below:

$$\frac{dL}{dD} = \left[\frac{\partial L}{\partial X} + \frac{\partial L}{\partial Q}\frac{\partial Q}{\partial X}\right]\frac{dX}{dD} \tag{42}$$

where the mesh is X with m number of points and number of design variables is n. The Jacobian of $\frac{dX}{dD}$ is then computed by calculating all of its first-order partial derivatives. This can be seen in Equation 43 where each column of the matrix J_x is a tangent and each row is a gradient of the Jacobian.

$$\frac{d\mathbf{X}}{d\mathbf{D}} = \begin{bmatrix} \frac{\partial X_1}{\partial D_1} & \cdots & \frac{\partial X_1}{\partial D_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial X_m}{\partial D_1} & \cdots & \frac{\partial X_m}{\partial D_n} \end{bmatrix} = J_{\mathbf{X}} \tag{43}$$

For a large number of D, computing the results for Equation 43 is computationally expensive. A method to overcome this is to compose $\frac{dL}{dD}$ purely with the tangents of J_x. For example, $\frac{dL}{dD}$ can be composed as such:

$$\frac{dL}{dD_i} = \left[\frac{\partial L}{\partial X} + \frac{\partial L}{\partial Q}\frac{\partial Q}{\partial X}\right]\frac{dX}{dD_i} \tag{44}$$

Inverting the tangent approach seen in Equation 44 allows for problems where the value of D is large, especially when most engineering problems involve a large number of design variables, D but only a small number of outputs, L. A more detailed expression of this approach can be found in Equation 45 which shows how the final sensitivity is formed.

$$\frac{dL_j^T}{dD_i} = \frac{dX^T}{dD}\left[\frac{\partial L_j^T}{\partial X} + \frac{\partial L_j^T}{\partial Q}\frac{\partial Q^T}{\partial X}\right] \tag{45}$$

The sensitivity of the cost function with respect to the solution can be found from firstly solving for $\frac{dL_j^T}{dQ}$. In the process of doing so, sensitivity of the cost function is calculated with respect to the mesh in $\frac{dL_j^T}{dX}$. The sensitivity of the cost function with respect to the mesh is then evaluated by solving the adjoint solution iteratively, which generates the sensitivity of L with respect to the residuals of the governing equations, $\frac{dL_j^T}{dR}$. Once the adjoint solution from this previous step has converged, the product of the residuals with respect to the mesh and the adjoint solution is computed to form, $\frac{dR^T}{dX}$. This is then added to $\frac{dL_j^T}{dX}$. Eventually, the overall sensitivity of the cost function with respect to the mesh can be expressed as:

$$\frac{dL_j^T}{dX} = \frac{\partial L_j^T}{\partial X} + \frac{\partial L_j^T}{\partial Q}\frac{\partial Q^T}{\partial X} \tag{46}$$

Deriving the steady adjoint solution requires the primal flow solution to be differentiated and transposed. It will result in Equation 47 where it comprises of a fully coupled linear system. Within the linear system, because the linearization process is exact, that accounted for 2nd order terms which can be seen in Equation 48.

$$\left[\frac{\partial R}{\partial Q}\right]\frac{\partial Q}{\partial X} = \frac{\partial R}{\partial X} \tag{47}$$

$$\frac{\partial Q^T}{\partial X} = -\frac{\partial R^T}{\partial X}\left[\frac{\partial R}{\partial Q}\right]^{-T} \tag{48}$$

Since calculation for the last term, $\left[\frac{\partial R}{\partial Q}\right]^{-T}$ is quite computationally expensive, a workaround for this is to multiply Equation 48 with $\frac{\partial L_j^T}{\partial Q}$which has been previously calculated. This results in the following:

$$\frac{dL_j^T}{dX} = \frac{\partial L_j^T}{\partial X} + \frac{\partial L_j^T}{\partial Q}\left(-\frac{\partial R^T}{\partial X}\left[\frac{\partial R}{\partial Q}\right]^{-T}\right) \tag{3.1}$$

$$= \frac{\partial L_j^T}{\partial X} - \frac{\partial R^T}{\partial X}\left[\frac{\partial R}{\partial Q}\right]^{-T}\frac{\partial L_j^T}{\partial Q} \tag{49}$$

Adjoint flow solution can be computed as $\wedge_Q = \left[\frac{\partial R}{\partial Q}\right]^{-T}\frac{\partial L_j^T}{\partial Q} = \frac{dL_j^T}{dR}$

The adjoint error estimate can finally be calculated by multiplying the local discretization error ψ^3 (which will be computed by a third order MUSCL correction) and the adjoint flow solution for the cost function L.

$$e_{adj} = \psi^3 \wedge_Q \tag{50}$$

3.3.2 Results for Gradient-based Refinement (Velocity)

The refinement parameter r_k, is chosen based on the gradient of velocity and defined in a similar manner as used in the previous T-junction example:

$$r_{k,grad} = |r_k|\Delta h \tag{51}$$

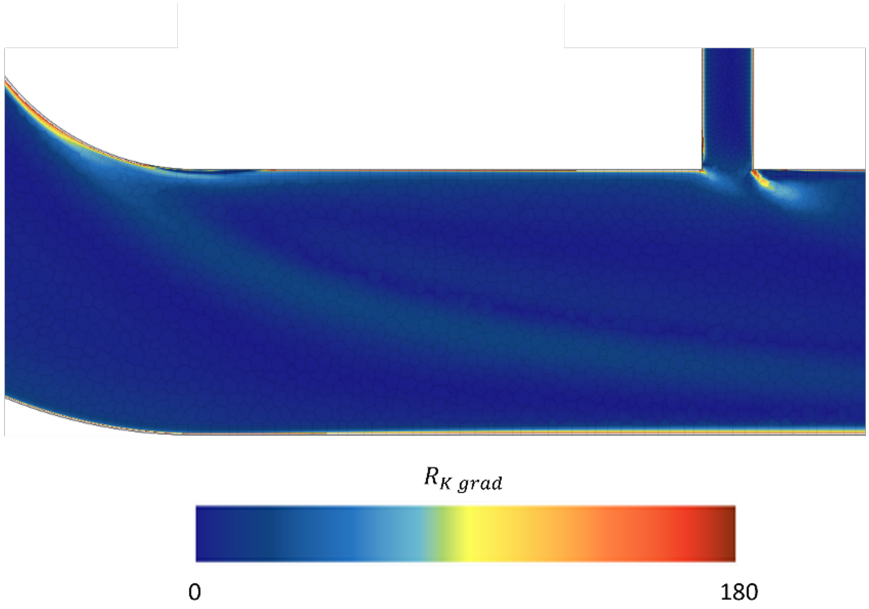
In this study, the refinement parameter r_k, is chosen based on the gradient of velocity, $grad(V)$.

Figure 19(a) below shows the contour of magnitude of the velocity gradient on the YZ symmetry plane and the histogram of$r_{k,grad}$ for the various computational regions can be found in Figure 19(b).It can be seen from Figure 19(a) there is a region of high velocity gradient that originates from the elbow itself, in addition to the high velocity gradients seen at the exit of the inlet branch. This is due to the presence of an upstream elbow. Figure 19(b) shows the histogram values of $r_{k,grad}$ at in the entire computational region. There is a large number of cells seen with higher $r_{k,grad}$. This corresponds to the refinement levels at various regions seen in Figure 20, the flow is seen to originate from the elbow and the flow downstream of the T-junction gets more refined. This is further quantified in Table 4 the number of cells are recorded within the computational domain before and after refinement, which see a 44% increase after the refinement.

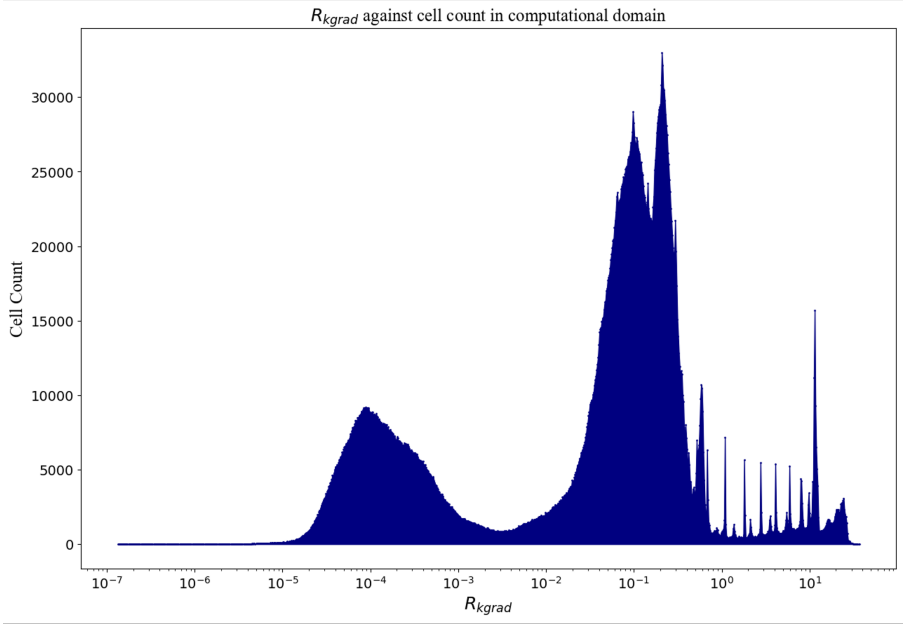
Most of the flow-field away from these regions do not get refined at all as they are away from the shear layer jet-in flow from the T-junction and from the upstream elbow. The first four refinement levels are concentrated around the thermal layer between the jet-in cross flow. They are evenly distributed throughout the computational domain, with more refined cells originating from the upstream section (where the exit of the elbow is) and downstream of the T-junction. The mean streamwise velocity profiles are plotted and represented in Figure 21, alongside velocity profiles generated with a high-fidelity mesh (2mm base size) and wall-resolved LES data from a study which this geometry was based on. It is worth noting that there are obvious discrepancies between Tunstall et al's data and the data presented here, due to the fact that the adaptive mesh refinement suite can only be executed on RANS models within STAR-CCM+. Note that there are 19,610,000 cells measured in the high-fidelity model. Within the near wall regions at all locations, using $r_{k,grad}$ as the refinement parameter leads to under predicted velocity values. The velocity in the upper half of the flow away from the wall seems to be recovered relatively well with the refined model. However, in the lower half of the flow in the upstream region, the refined model predicts velocity values.

Table 4: Cell count for velocity gradient-based refinement parameters

Refinement Metric	**Original Cell Count**	**New Cell Count**	**Percentage Change**
r_{k_grad}	805,880	1,105,217	37.1



(a)



(b)

Figure 19: (a) Contour of magnitude of Grad of Velocity on the YZ symmetry plane, (b) Histogram of $rr_{k,grad}$ against number of cells

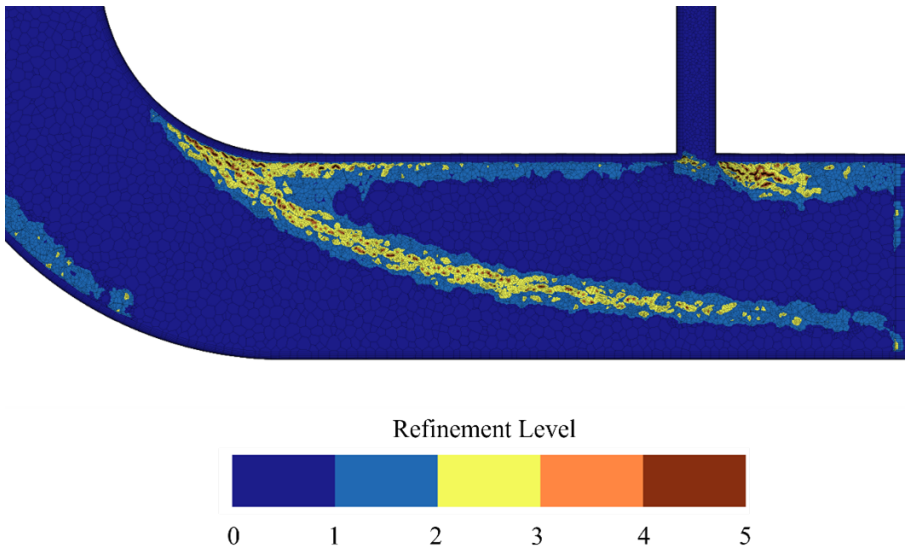


Figure 20: Refinement Level for Gradient Based Refinement

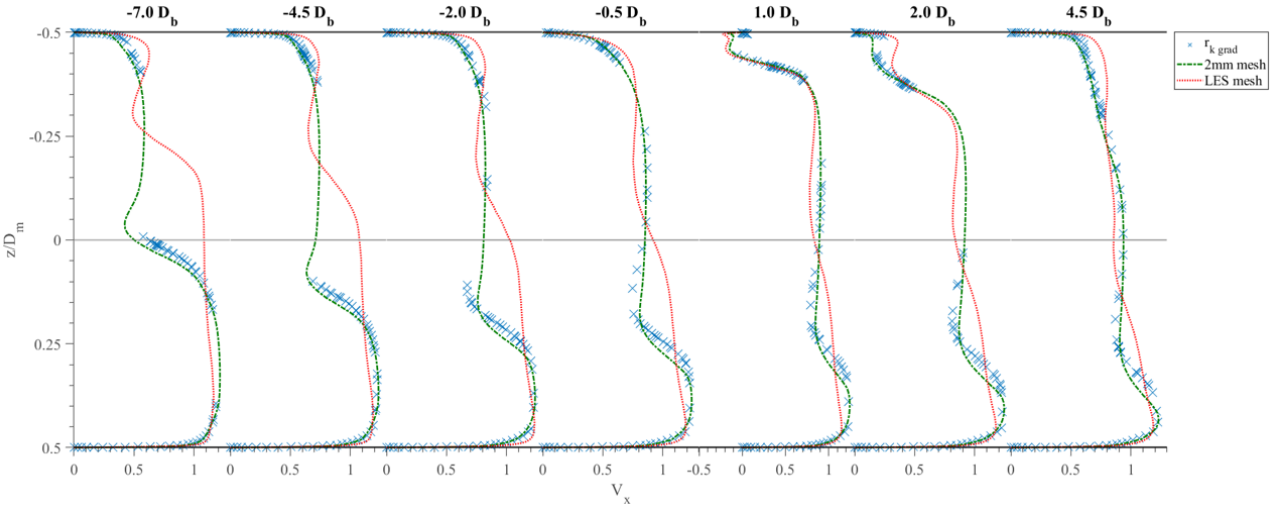


Figure 21: Streamwise velocity profiles obtained with adaptive mesh refinement ($r_{k,grad}$) high fidelity model (2mm mesh) and LES model (LES mesh)

3.3.3 Results for Dual-weighted error-based Refinement (Velocity)

The adjoint cost function of choice is defined based on the volume uniformity of velocity, where $\phi = V$. Figure 22 depicts the contour of the adjoint error estimate of volume uniformity of velocity along the XY symmetry plane while Figure 23 shows the histograms of the different refinement parameters across the whole computational domain. It can be seen from Figure 22 that the adjoint error is the highest at the shear layer of the jet-in cross flow and at the shear latter emanating from the exit of the elbow. This is represented by the red and yellow regions

in the same figure. A large number of cells with higher $r_{k,mean}$ and $r_{k,max}$ is noted, compared to the other two refinement parameters. The number of cells before and after mesh adaptation for each refinement parameter is recorded in Table 5, where new and refined models all have higher cell counts than before. Notably, the refinement parameter $r_{k,mean}$ has the largest cell count compared to the rest.

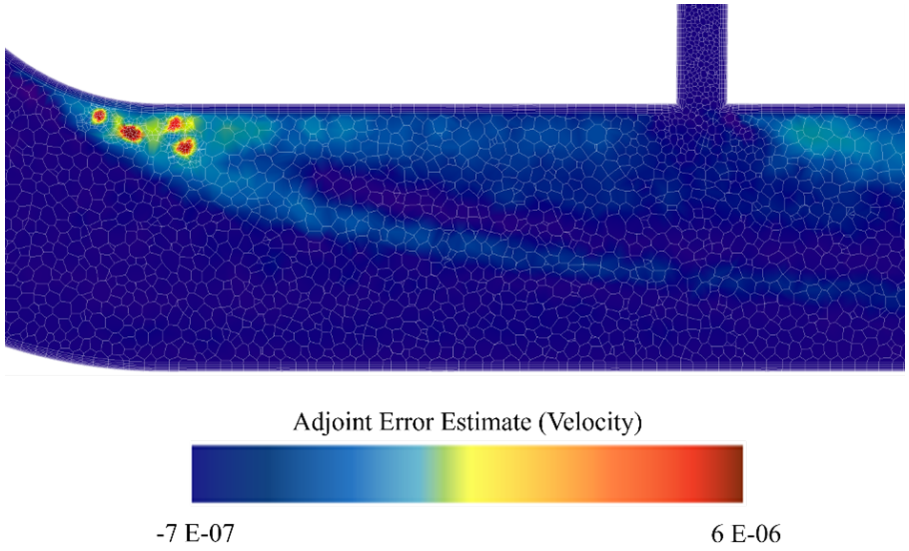


Figure 22: Contour of adjoint error estimate of volume uniformity of velocity on symmetry plane

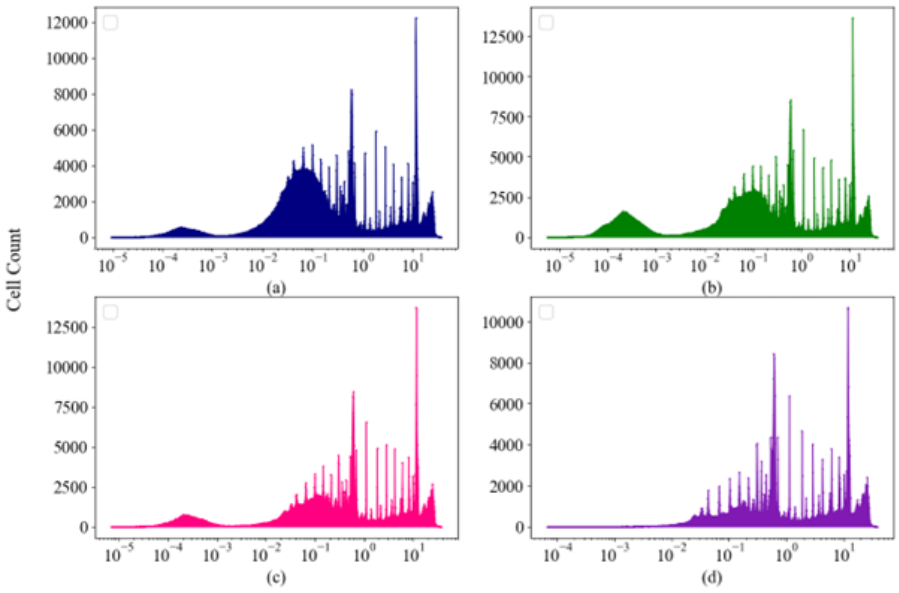


Figure 23: Histograms of different refinement parameters against number of cells (a) $r_{k,mean}$, (b) $r_{k,max}$, (c) $r_{k,total}$, (d) $r_{k,rms}$

The cells with different refinement levels found in Figure 24 show a more locally distributed

mesh refinement. It can be seen from the color bar which represents the levels of refinement taking place. These are found within the upstream elbow region and are different from that seen in Figure 20, where the mesh refinement is not locally contained. $r_{k,mean}$ shows a more spread-out region where mesh refinement is taking place. This is followed by $r_{k,max}$ where the region of refinement is more localized within the shear layer. $r_{k,total}$ has a similar behavior as $r_{k,max}$, while $r_{k,rms}$ is very localized in the upstream elbow region. The new cell count after refinement is the most for $r_{k,max}$, followed by $r_{k,total}$, $r_{k,mean}$ and $r_{k,mean}$. This is because of the high level of refinement occurring in the shear layer based on the first two refinement parameters, even though the refinement is localized.

Table 5: Cell counts for different refinement parameters based on dual weighted velocity error

Refinement Metric	**Original Cell Count**	**New Cell Count**	**Percentage Change (%)**
$r_{k\,mean}$	805,880	1,105,217	37.1
$r_{k\,max}$	805,880	1,764,102	119
$r_{k\,total}$	805,880	1,394,128	73.9
$r_{k\,rms}$	805,880	894,885	11.2

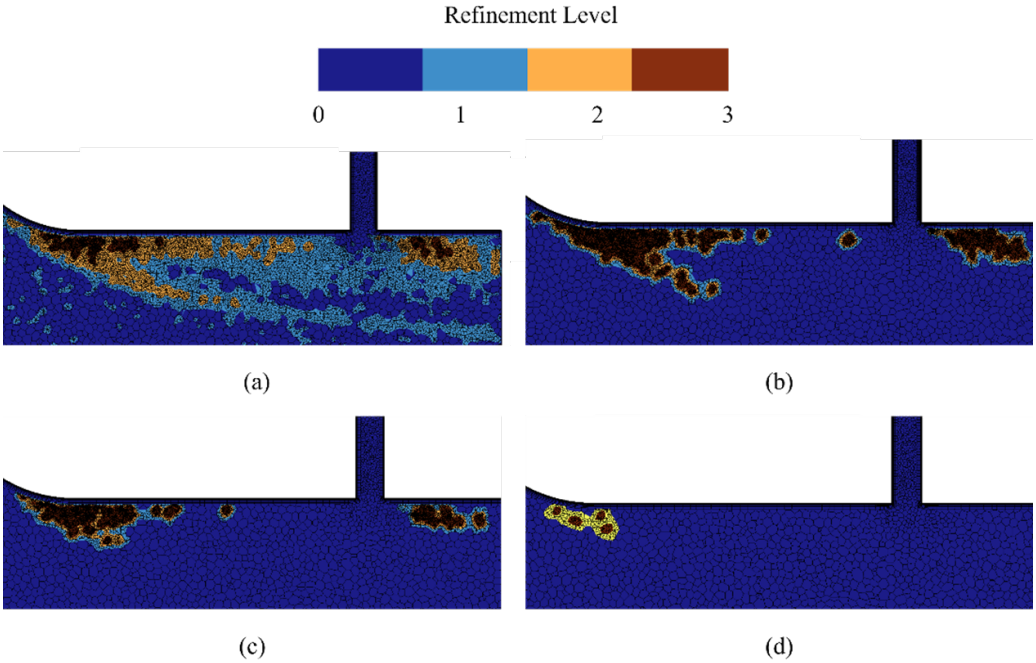


Figure 24: Refinement Levels based on various adaptive mesh refinement metric for (a) $r_{k,mean}$, (b) $r_{k,max}$, (c) $r_{k,total}$, (d) $r_{k,rms}$

The mean streamwise velocity profiles are plotted and represented in Figure 25, alongside velocity profiles generated with a high-fidelity mesh and wall-resolved LES data. Similar to

the analysis based on the refinement parameter $r_{k,grad}$ there are very obvious discrepancies between Tunstall et al's data and the data presented in this figure, mainly due to the fact that the adaptive mesh refinement suite is only compatible with RANS models within Star-CCM+. In the near wall regions at $-7.0D_b$, $r_{k,mean}$,$r_{k,max}$, $r_{k,total}$, $r_{k,rms}$ yield the same velocity profiles. However, they differ from the velocity profiles of the high-fidelity mesh, with the later experiencing a smaller deceleration in the bottom half of the flow as compared to the models with the refined mesh. As the flow proceeds further downstream to $-4.5D_b$, some similarities are seen between the velocity from the high-fidelity mesh and the refined meshes, where the point of inflexion (peak at the bottom half of the flow) is very similar. Notably, $r_{k,max}$ has the largest divergence from the other velocity profiles, mainly due to the coarse meshes away from the regions of interest. The velocity profiles for $r_{k,mean}$,$r_{k,max}$, $r_{k,total}$, $r_{k,rms}$ at $-2.0D_b$ shows some similarities with the ones at $-7.0D_b$ where each refinement parameter produces similar velocity profiles. The profiles produced by the refined mesh are 'closer' to the profile of the LES mesh at this location, particularly the point of inflexion. The high-fidelity mesh has its point of inflexion happening nearer towards the bottom of the flow. At $-0.5D_b$, the velocity profiles for $r_{k,max}$ and $r_{k,total}$ is not well represented, due to the regions of coarser mesh that are unable to accurately resolve the flow profiles. The profiles by the other refinement parameters show similar behaviour but they are still far off from the high-fidelity mesh. Interestingly, their point of inflexion occurs at a similar distance (in the middle of the flow), but its velocity values are understated. It can be said that using the refinement parameters $r_{k,mean}$ and $r_{k,rms}$ produces an overall mesh that offers very similar results to a high-fidelity model, at a much lower overall cell density. The other two refinement parameters $r_{k,max}$ and $r_{k,total}$ show otherwise.

As the flow progresses past the T-junction, at the locations $1.0D_b, 2.0D_b$ and $4.5D_b$, the velocity in the upper half of the flow away from the wall seems to be recovered relatively well with most of the refined models, except for $r_{k,mean}$,$r_{k,max}$. These velocity profiles closely match the velocity profiles generated by the high-fidelity mesh. It can be said that using the refinement parameters $r_{k,total}$ and$r_{k,rms}$ produces an overall mesh that offers very similar results to a high-fidelity model, at a much lower overall cell density. The other two refinement parameters $r_{k,mean}$,$r_{k,max}$ show otherwise.

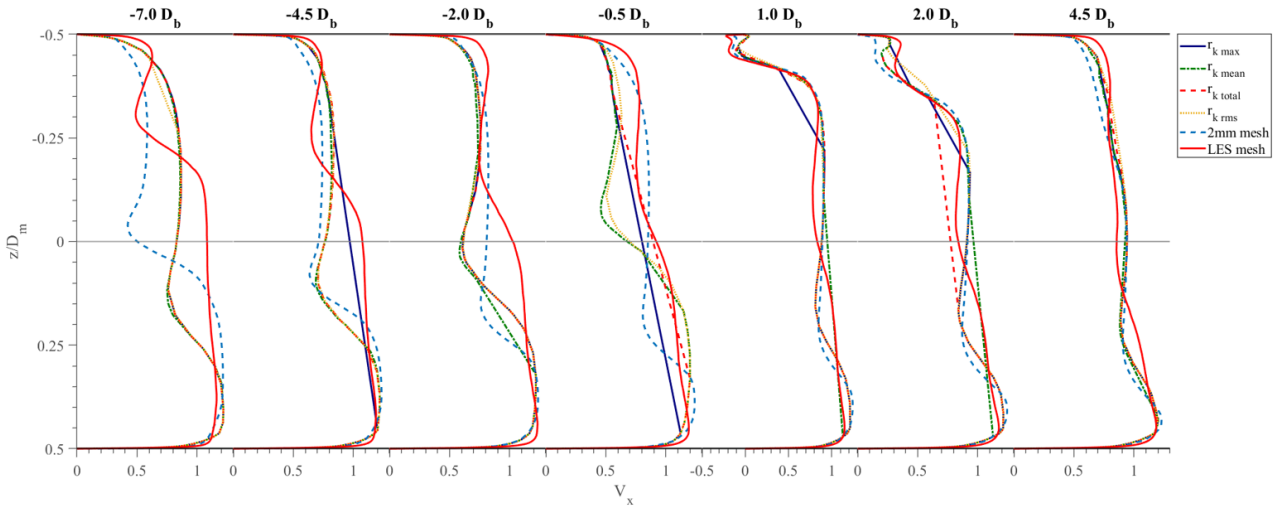


Figure 25: Streamwise velocity profiles obtained with various adaptive mesh refinement models, high fidelity model (2mm mesh) and LES model (LES mesh).

3.3.4 Results for Gradient-based Refinement (Temperature)

As the temperature profiles will invariantly be different across the entire flow domain, the locations of interest will be line probes carefully selected at certain distances away from the upper wall. This main reason for choosing the upper wall is due to the higher temperature fluctuations within this region. Figure 26 shows the location of these line probes.

Similar to the section above, the refinement parameter r_k, is chosen based on the gradient of temperature, grad(T) and is defined in a similar manner above. Figure 27(a) below shows the contour of magnitude of the temperature gradient on the YZ symmetry plane and the histogram of $r_{k,grad}$for the various computational regions can be found in Figure 27(b). There is a large number of cells seen with higher $r_{k,grad}$. This corresponds to the refinement levels at various regions seen in Figure 28 where is seen to originate from the elbow and the flow downstream of the T-junction gets more refined. This is further quantified in Table 6 where the number of cells within the computational domain are recorded before and after refinement, where the number of cells sees a 47% increase after the refinement.

Majority of the flow domain away from these regions do not get refined at all as they are away from the shear layer jet-in flow from the T-junction and from the upstream elbow. The first four refinement levels are concentrated around the thermal layer between the jet-in cross flow. They are evenly distributed throughout the computational domain, with more refined cells originating

from the upstream section (where the exit of the elbow is) and downstream of the T-junction. With a more refined mesh at the locations where most of the thermal mixing occurs, one can expect the accuracy of temperature values to be better and closer to the results generated by a high-fidelity model. Note that temperature data is not available from the LES benchmark used in the previous section, hence only data from the high-fidelity model is presented as the benchmark. The mean normalized temperature extracted from the line probes is presented in Figure 29. An immediate observation is the underprediction of temperature values downstream of the T-junction. This effect is more exacerbated as further away from the upper wall. While discrepancy exists, the results are commendable especially when the refined model contains 20 times less number of cells compared to the high-fidelity model (approximately 1 million versus 19 million cells).

Table 6: Cell count for temperature gradient-based refinement parameters

Refinement Metric	**Original Cell Count**	**New Cell Count**	**Percentage Change**
r_{k_grad}	805,880	1,184,800	47.2

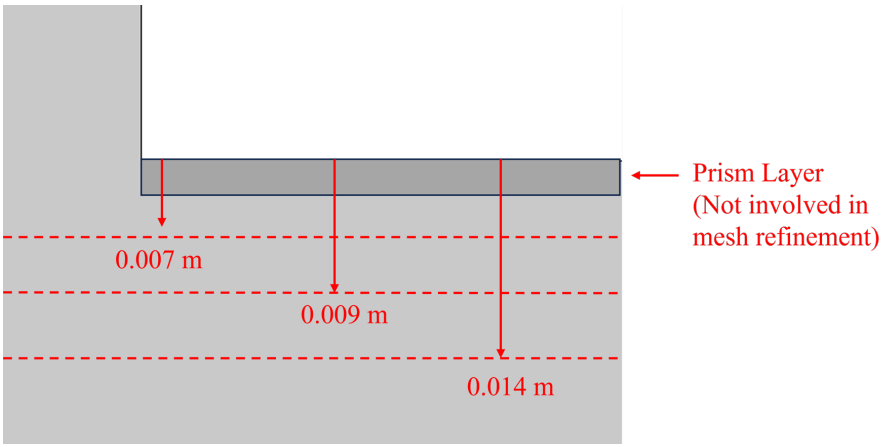
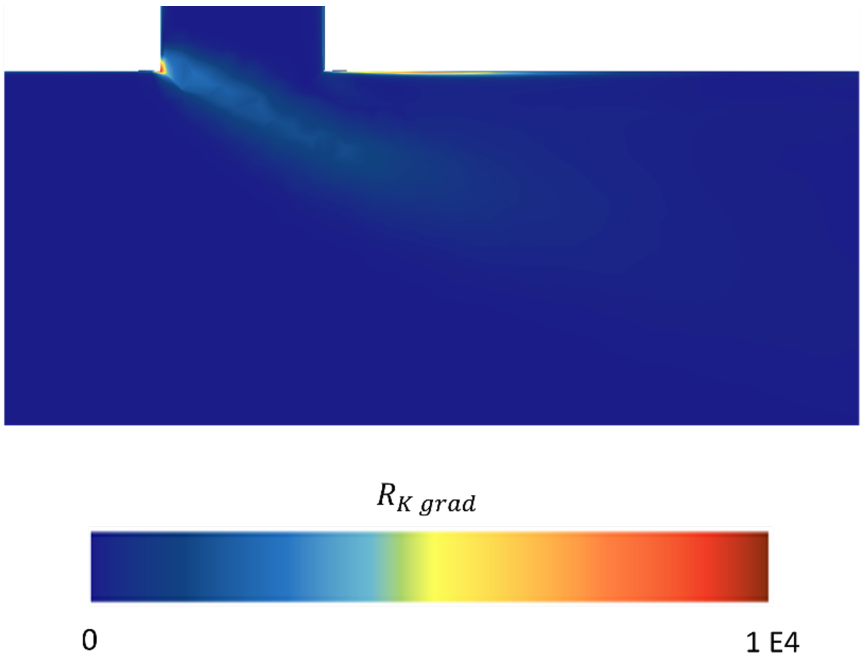
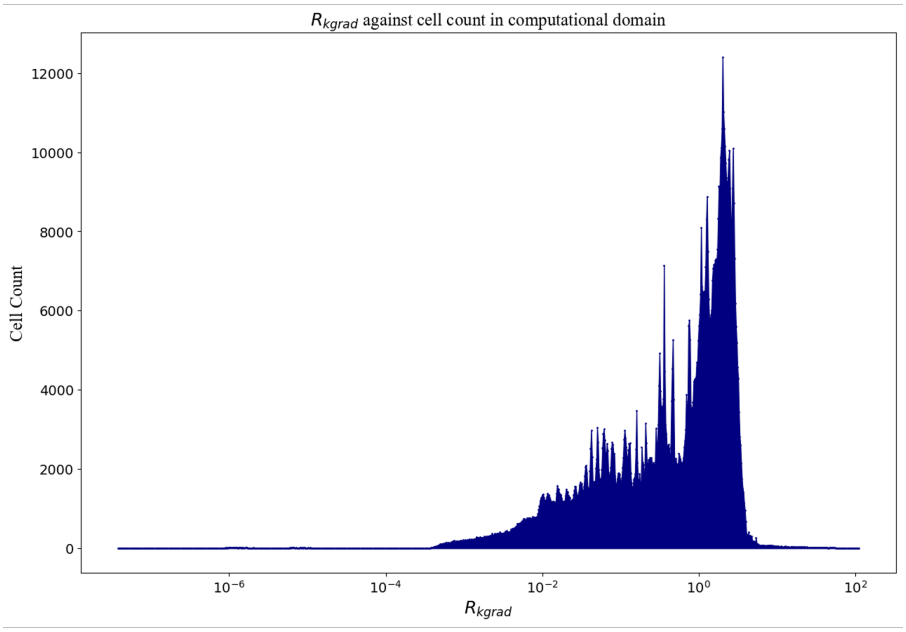


Figure 26: Locations of Line Probes Within Flow Domain



(a)



(b)

Figure 27: (a) Contour of magnitude of Grad of Temperature on the YZ symmetry plane, (b) Histogram of $r_{k,grad}$ against number of cells

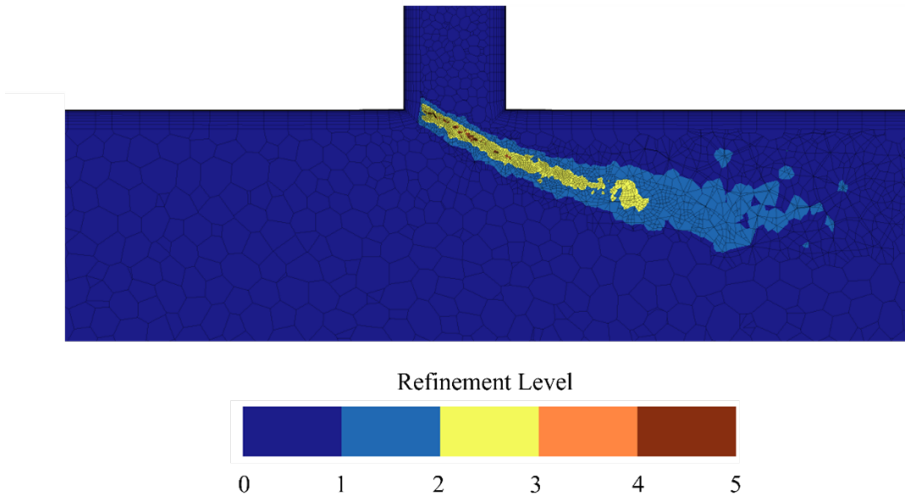


Figure 28: (a) Contour of magnitude of Grad of Temperature on the YZ symmetry Refinement Level for Gradient Based Refinement

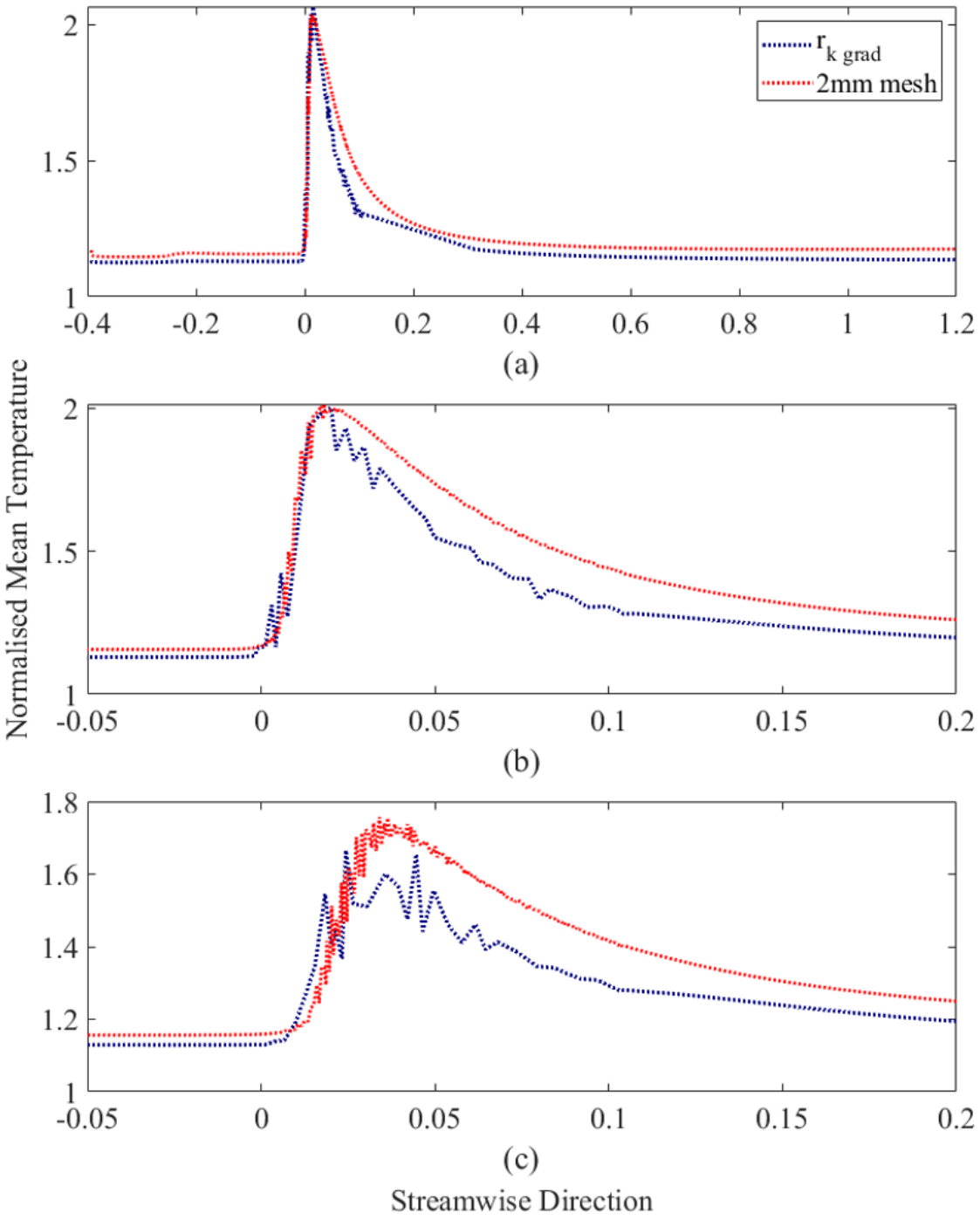


Figure 29: Mean Temperature plots along the X-Z Plane for probe at locations (a) 0.007m, (b) 0.009 and (c) 0.014m away from the upper wall

3.3.5 Results for Dual-weighted error-based Refinement (Temperature)

Similar to the analysis used in the previous section, the adjoint cost function of choice is defined based on the volume uniformity of Temperature, where $\psi = T$. Figure 30 depicts the contour of the adjoint error estimate of volume uniformity of temperature along the XY symmetry plane. Figure 31 shows the histograms of the different refinement parameters across the whole computational domain. Figure 21 shows that the adjoint error is the highest at the shear layer of the jet-in cross flow. This is represented by the red and blue regions in the same figure. A large number of cells with higher $r_{k,mean}$ is observed, compared to the other refinement parameters followed by $r_{k,mean}$ where the number of cells are more evenly distributed in the median value of $r_{k,total}$. Both $r_{k,max}$ and $r_{k,rms}$ have a similar distribution in general. The new cell counts based on the different refinement parameters can be found in Table 7. A stark difference between the original cell count and the new cell count for the refinement parameters is seen. $r_{k,mean}$,$r_{k,max}$ and $r_{k,total}$. $r_{k,rms}$ does not see a huge increase in the new cell count. All these are depicted in Figure 31 where a high number of cells is observed to possess a high $r_{k,mean}$ error.

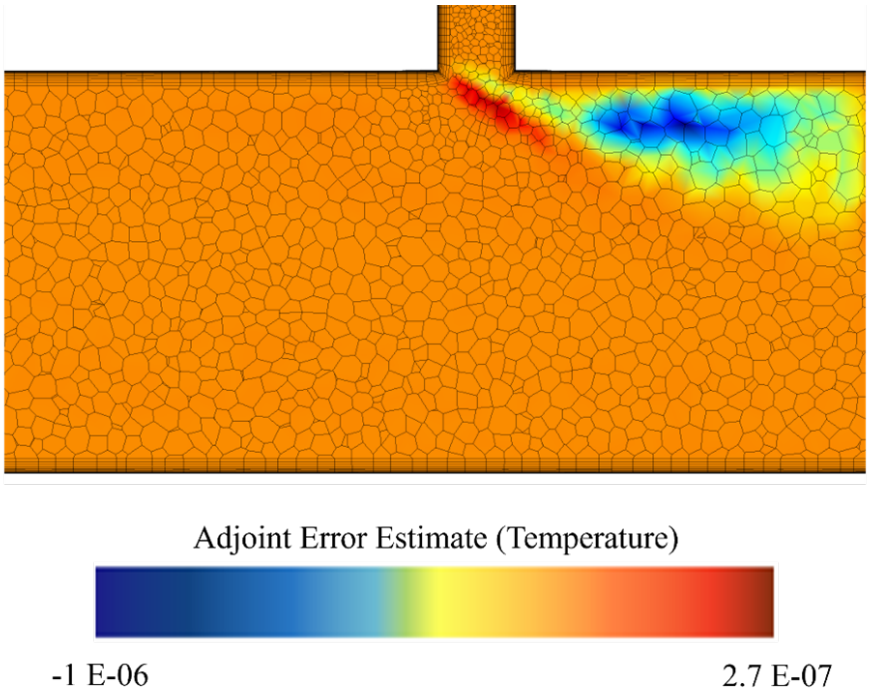


Figure 30: Contour of adjoint error estimate of volume uniformity of temperature on symmetry plane

Table 7: Cell counts for different refinement parameters based on dual-weighted error of temperature

Refinement Metric	Original Cell Count	New Cell Count	Percentage Change (%)
$r_{k\,mean}$	805,880	16,398,500	1934
$r_{k\,max}$	805,880	5,857,340	627
$r_{k\,total}$	805,880	8,136,310	909
$r_{k\,rms}$	805,880	809,047	0.5

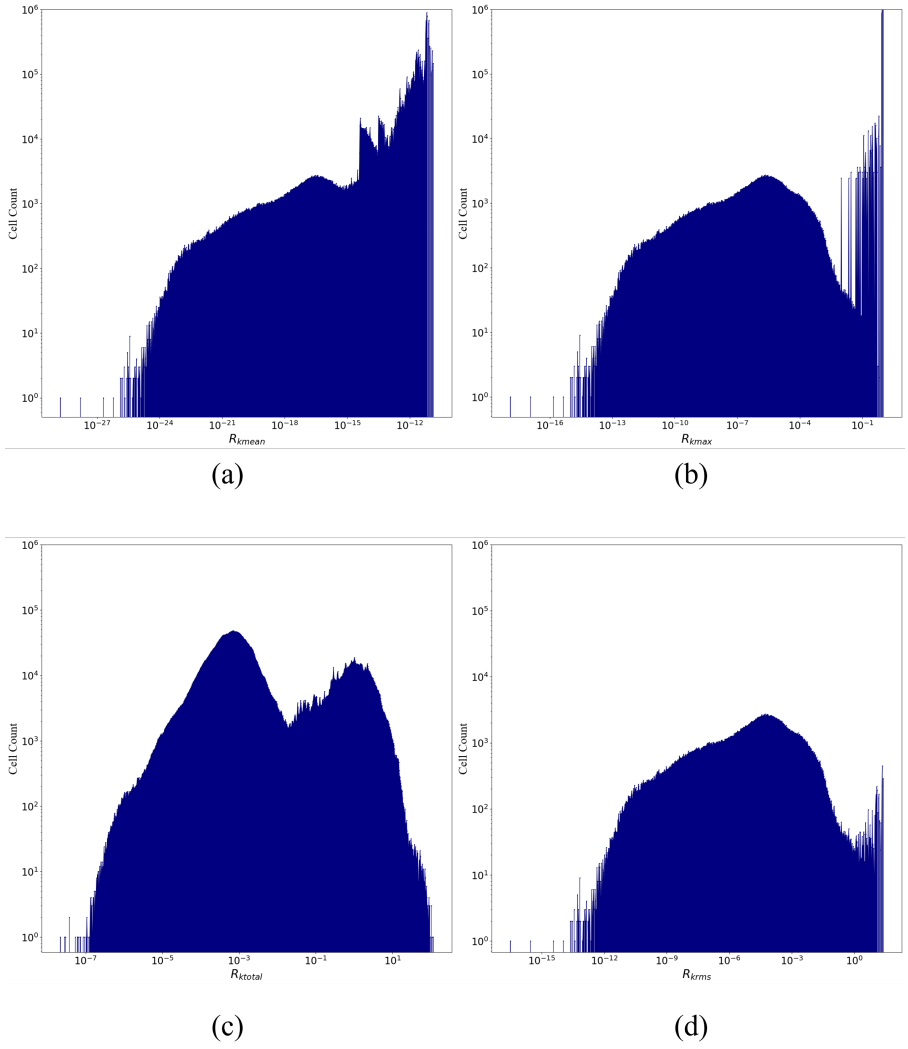


Figure 31: Histograms of different refinement parameters against number of cells (a)$r_{k,mean}$, (b)$r_{k,max}$ and (c) $r_{k,total}$, (d)$r_{k,rms}$

The cells with different refinement levels found in Figure 32 show a more locally distributed mesh refinement, which has also been observed in the previous section where velocity was used as the adjoint cost function. However, the region of refinement is seen to be more confined to a specific region downstream of the T-junction. There are many pockets of smaller regions where

further mesh refinement takes place. This can be seen from the color bar which represents the levels of refinement taking place. $r_{k,mean}$ shows a more spread-out region of mesh refinement and it has the highest level of refinement taking place in the near wall regions (indicated in dark brown). This is followed by $r_{k,total}$ where the region of refinement is more localized, away from the jet-in shear layer and closer to the upper wall. $r_{k,max}$ seems to only have a small region where a high level of refinement is taking place. while $r_{k,rms}$ is very spread out. The new cell count after refinement is the most for $r_{k,mean}$,$r_{k,max}$, $r_{k,total}$ and $r_{k,rms}$.

In a similar fashion to the previous sub-section, the mean normalized temperature extracted from the line probes is presented in Figure 33. This figure compares the normalized mean temperature profiles along the streamwise direction with the high-fidelity mesh serving as the benchmark data plotted against results based on the different refinement parameters. Among the refinement parameters $r_{k,mean}$and $r_{k,rms}$ provides the closest approximation to the high-fidelity data across the different locations. At locations further away from the upper wall, there are greater deviations from the high-fidelity data, which may suggest that even further mesh refinement is required to accurately capture the localized high gradient zones. While the results from $r_{k,rms}$ still deviate from the high-fidelity mesh, despite these limitations, it still demonstrates promising performance, particularly in smoother regions. This suggests that using $r_{k,rms}$ as the refinement parameter provides the best balance between added accuracy and computational efficiency.

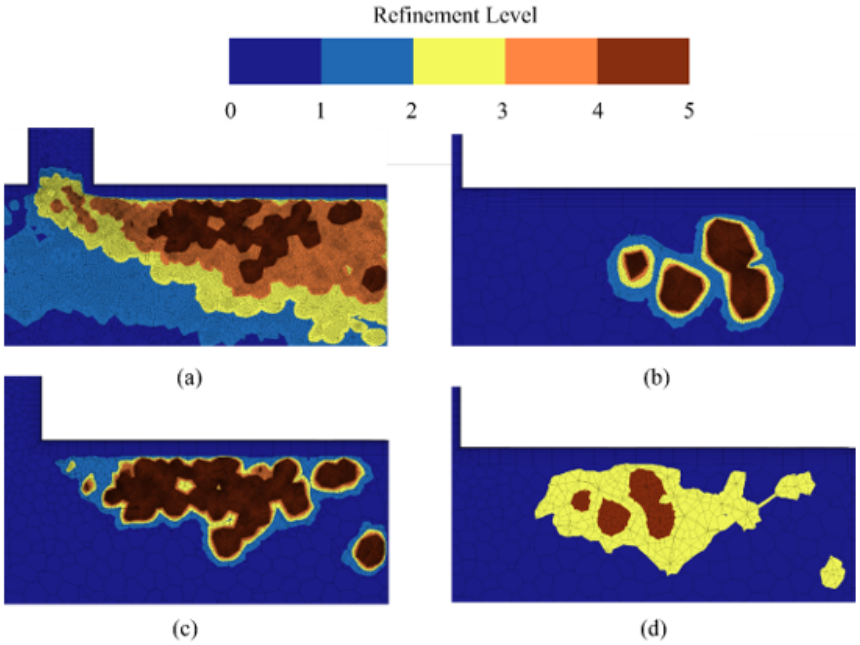


Figure 32: Refinement Levels based on various adaptive mesh refinement metric for (a)$r_{k,mean}$, (b)$r_{k,max}$ and (c) $r_{k,total}$, (d)$r_{k,rms}$

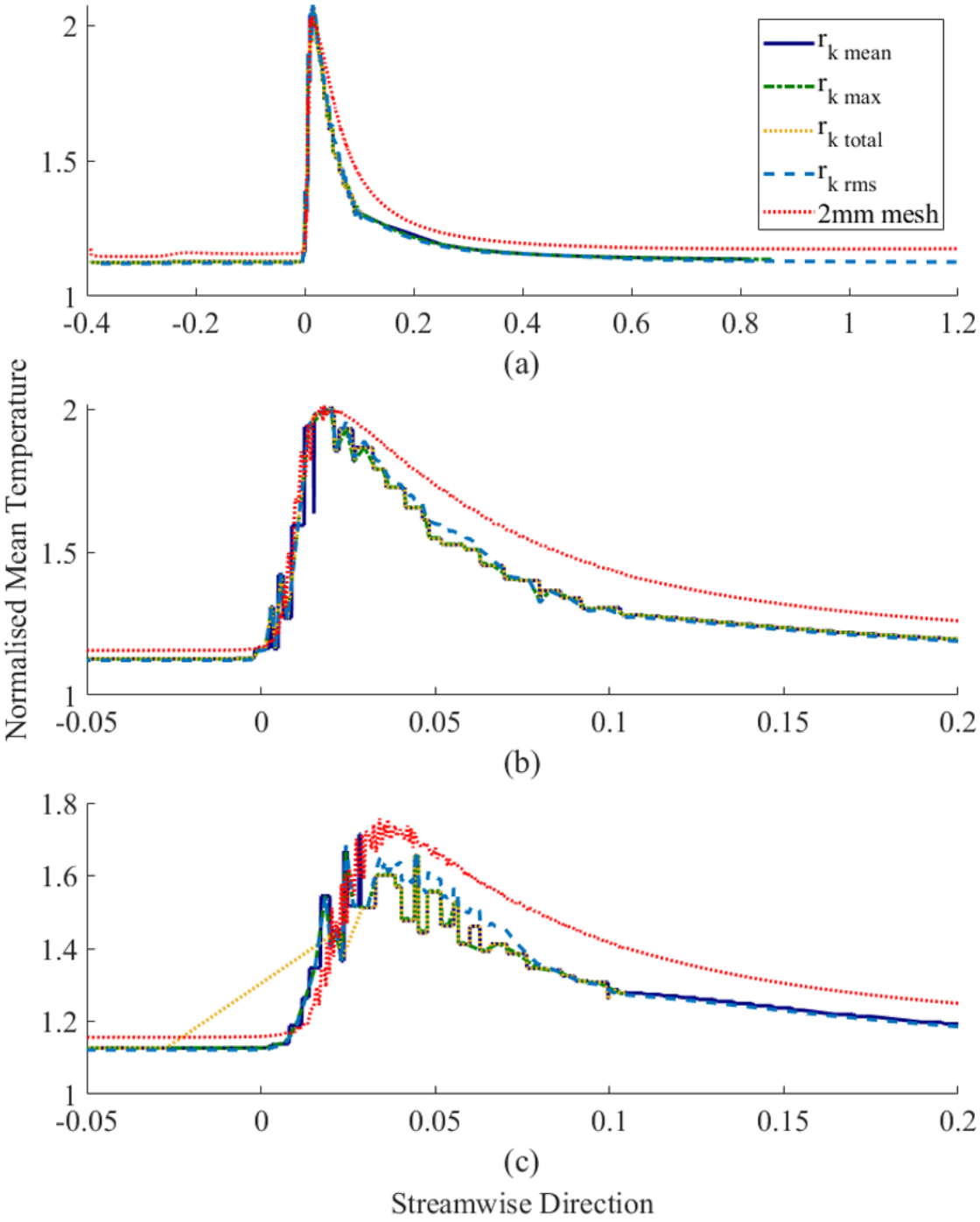


Figure 33: Mean Temperature plots along the X-Z Plane for probe at locations (a) 0.007m, (b) 0.009 and (c) 0.014m away from the upper wall

3.4 Conclusion

In this study, the effectiveness of different mesh refinement strategies was evaluated. They are namely, gradient-based and adjoint error-based, in capturing velocity and temperature fields

within a complex flow domain.

Utilizing the velocity gradient as the refinement criterion, significant mesh refinement is observed in regions with high velocity gradients, particularly near the upstream elbow and downstream of the T-junction. This approach led to a 44% increase in the total number of computational cells. The refined mesh accurately captured the velocity profiles in the upper half of the flow away from the wall; however, near-wall regions exhibited under predicted velocities. Focusing on the adjoint error estimates for velocity uniformity, mesh refinement was seen to predominantly localize around shear layers of the jet-in cross flow and the exit of the elbow. Among the refinement parameters tested, $r_{k,mean}$and $r_{k,rms}$ produced meshes that closely approximated high-fidelity results, especially in near-wall regions. Notably, $r_{k,mean}$resulted in the highest increase in cell count, indicating a more extensive refinement. Increasing mesh resolution enhances the accuracy of CFD simulations by capturing finer details of the flow field. However, this improvement comes with a significant rise in computational cost. When applying the temperature gradient for mesh refinement, similar patterns emerged. Regions with steep temperature gradients, especially downstream of the T-junction, experienced increased mesh density, resulting in a 47% rise in cell count. While the refined mesh improved temperature predictions, discrepancies remained in areas further from the upper wall, suggesting the need for additional refinement to capture localized high-gradient zones accurately. For temperature uniformity, the adjoint error-based refinement also concentrated on regions with significant thermal gradients. The $r_{k,mean}$ parameter led to a more widespread mesh refinement, particularly near the walls, and achieved temperature profiles that closely matched high-fidelity data. However, areas distant from the upper wall still exhibited deviations, highlighting the necessity for further mesh adaptation in those regions.

Both gradient-based and adjoint error-based mesh refinement techniques enhance the accuracy of velocity and temperature predictions in CFD simulations. Gradient-based methods effectively identify regions with high physical gradients, leading to targeted mesh refinement and improved flow feature resolution. Adjoint error-based approaches, on the other hand, offer a more systematic refinement by focusing on areas that significantly impact the quantity of interest, often resulting in meshes that balance accuracy and computational efficiency.

However, as seen in the results, careful consideration must be taken as a model with a relatively well resolved mesh may still outperform models that underwent adaptive mesh refinement.

This was observed to be true in the study. It also comes with the added computational cost arising from the increased model complexity. In practice, the choice between these methods should consider the specific simulation objectives, computational resources, and the desired balance between accuracy and efficiency. Hence, for the models used from chapter 5 onwards, computations will be based off a 2mm mesh base size.

Chapter 4

Description of Machine Learning Model

This chapter presents a detailed and concise overview of the various neural network types considered. Mathematical descriptions of the various neural networks are presented. The neural networks have been selected for reasons detailed in the following few subsections. An extensive assessment of how parameter selection affects the performance of a neural network is presented in chapter 7. The results and analysis for the performance of various neural networks as surrogate models is further discussed in chapter 8.

4.1 Deep Neural Network (DNN)

Machine learning (ML) and artificial intelligence (AI) algorithms have found widespread use within science and engineering for pattern recognition, object detection, and surrogate modelling. ML and AI algorithms are based upon DNNs and are inspired by the biology and physiology of the human brain (Szegedy et al. (2013); Nguyen et al. (2015); Cichy and Kaiser (2019)). This involves many simple processing units (neurons) that work in together via interconnected layers. When more layers are added or stacked, only then are they considered DNNs (Hinton (2007); LeCun et al. (2015)). Over the last decade there has been a significant in research studies focused on DNNs and their application within science and engineering (Sze et al. (2017)). As DNNs become deeper and more complex, it gets harder to interpret the decision-making process which ultimately leads to the 'black-box' problem and some computer

scientists have suggested that one should not be overly reliant on deep learning with neural networks (Buhrmester et al. (2021)). Some works have been done in this area to document this issue and suggested that a solution is to go beyond just a simple DNN (Castelvecchi (2016); Hussain (2019)). A typical DNN can be expressed mathematically as:

$$X_0 = t \tag{52}$$

$$X_1 = \sigma(W_1 X_0 + b_1)$$

$$X_2 = \sigma(W_2 X_1 + b_2)$$

$$X_3 = \sigma(W_3 X_2 + b_3)$$

$$X_4 = W_4 X_3 + b_4$$

The hidden variables of each hidden layer are denoted as X_k where k refers to the k_th layer. For each hidden layer, σ represents the non-linear activation function and W_k and b_k are the network weights and biases of the k_th layer. An example is found in Equation 52 where the hidden layers are seen in X_1, X_2 and X_3. The objective of a neural network is to train it such that the model's output $x(t)$ closely follows and approximates the true value (the data), $\hat{x}$ via minimising a loss function, L that consists of the data loss, L_{data}. An expression for this can be found in below, where θ represents the set of model parameters such as the set of weights W and biases b that gets iteratively updated during the training of the model, i.e $\theta = [W, b]$ Equation 53 expresses data loss in terms of error penalty between the model $x(t)$ and data $\hat{x}$. This loss is usually defined as the mean squared error, also known as the L_2 norm of the difference between the model and data divided by the number of training points, N.

$$L(\theta; t) = L_{data} \rightarrow min \tag{53}$$

$$L_{data} = \frac{1}{N} \sum (x(t) - \hat{x})^2 \tag{54}$$

To get a better understanding of how this is implemented in a code, Algorithm 1 below details the basic code structure of a DNN which can be applied to any language.

Algorithm 1 DNN Algorithm

Initialize: Weights W_k, Biases b_k for each layer X
Input: t consisting of an array of values
Output: y consisting of an array of values
for each input-output set **do**
    $X[0] \leftarrow t$ ▷ Input layer + normalization
    for $z = 1$ to k **do** ▷ Hidden layers
        $z[k] \leftarrow W[k] \cdot X[k-1] + b[k]$
        $X[k] \leftarrow \text{ReLU}(z[k])$
    end for
    $\bar{y} \leftarrow W[k+1] \cdot X[k] + b[k+1]$ ▷ Output layer
end for
Compute loss: $L_{\text{data}} = \text{MSE}(y, \bar{y})$
Minimize L_{data} using L-BFGS optimizer

4.2 Physics Informed Neural Network (PINN)

PINNs are a type of neural network that combines DNNs with physical laws or equations. They were designed to solve partial differential equations (PDEs) and other complex physical problems by incorporating these known governing equations as the loss function during training. This addition ensures that predictions and the learned solutions satisfy the governing equations as much as possible (Cai et al. (2021b)). The application of PINNs for various complex engineering problems are looked into with the aid of PINNs, such as power systems (Misyris et al. (2019)), characterizing failure modes, heat transfer problems and more relevantly, fluid flows (Mao et al. (2020); Cai et al. (2021a); Krishnapriyan et al. (2021); Eivazi et al. (2022)). This concept can be extended to the reconstruction of flow field, where a PINN is trained using a dataset of known input and output pairs and then makes predictions for another dataset of

known input points, but the outputs are the quantities of interest. There has been some work done on flow field reconstruction such as reconstructing a fully developed turbulent channel flow and the wake flow past a circular cylinder (Xu et al. (2022); Li et al. (2024)). Not much has been done on complex geometries and this forms the basis for the novelty of this paper. It is worth nothing that there are many different methods to construct a PINN in which the location of the constraints is implemented differently. The most straight forward method is to place the physical constraints within the loss function. In this case, a custom loss function is created instead of using the built-in loss functions offered by Keras, which is one of the various ML libraries offered under TensorFlow.

The governing equation for the flow in question is the incompressible Navier-Stokes equation where * represents non-dimensional terms $u^* = \frac{u}{U}$; $t^* = \frac{t}{T}$; $p^* = \frac{p}{P}$. Equations 55 and 56 show the Cartesian component forms of the non-dimensional Navier-Stokes momentum equation and continuity equation respectively. In these equations, i refers to the i-th component of non-dimensional velocity ($i = 1, 2, 3$ for x, y, z) and j represents the Cartesian coordinates ($x_1 = x, x_2 = y, x_3 = z$). Note that Re represents the Reynolds number ($Re = \frac{\rho U L}{\mu}$) while Fr represents the Froude number ($Fr = \frac{U}{\sqrt{gL}}$), where ρ refers to the fluid density, U refers to the characteristic velocity, L refers to the characteristic length, μ refers to the dynamic viscosity and g refers to the gravitational acceleration. For incompressible, isothermal flows, the energy equation, usually for temperature or internal energy, is not required to solve the velocity and pressure fields. The flow behavior is fully described by the Navier–Stokes equations and continuity equation. Energy is usually only considered if the problem requires solving for thermal fields or compressible flows. Classic PINN test cases like lid-driven cavity, flow past a cylinder, or Poiseuille flow are isothermal with no heat transfer. Hence, in this project, including an energy equation adds unnecessary complexity with no new information for the flow field.

$$\frac{\partial u_i^*}{\partial t} + u_j^* \frac{\partial u_i^*}{\partial x_j} = -\frac{\partial p^*}{\partial x_i} + \nu \frac{\partial^2 u_i^*}{\partial x_j \partial x_j} + \frac{1}{\mathrm{Fr}^2} g_i \tag{55}$$

$$\frac{\partial u_j^*}{\partial x_j} = 0 \tag{56}$$

Since this analysis is focused on a steady state flow within the lid-driven cavity, the time derivative term is equivalent to zero, aka vanishes. To further simplify the problem, the inertial force within the fluid is assumed is much higher than the force due to gravity. As a result, the term that includes the Froude number is also equivalent to zero and thus vanishes. Furthermore, the governing equations are normalised by setting $u = L = p = 1$. This leads to the resulting momentum and continuity equations below:

$$u^*\nabla^* = -\nabla^* p^* + \frac{1}{Re}\nabla^{*^2} \tag{57}$$

$$u^*.\nabla^* = 0$$

The loss for PINN follows a similar form where L_{data} is the standard cost function for the adaptation of weights and biases which penalises the deviation of the predictions from the actual value. L_c encrypts the continuity equation into the error while L_m encrypts the conservation of momentum into both horizontal and vertical components as seen in $L_{m,x}$ and $L_{m,y}$. These equations can be expressed in the following few equations below.

Equation 58 represents the overall loss function typically used in a PINN architecture where it is comprised of loss values based on the data and loss values from the physical laws (PDEs). The loss from the PDEs can be further broken down into Equations 59 and 60, which represents loss based on the continuity equation and momentum. Since Equation 60 consists of momentum values in both spanwise and streamwise directions, it can be further expressed in the form of Equations 61 and 62.

Figure 34 presents the schematic diagram of a classic PINN for a two-dimensional flow application. The diagram shows a typical feed-forward neural network consisting of an input layer, hidden layers and an output layer comprising of nodes where weights and biases are embedded

within each node. The PDEs of interest are enforced in the layer immediately after the output layer, which is where the physics loss values are computed and back-propagated back into the whole neural network.

$$L_{PINN} = L_{data} + L_c + L_m \tag{58}$$

$$L_c = \frac{1}{N}\sum (u_x - v_y)^2 \tag{59}$$

$$L_m = L_{m,x} + L_{m,y} \tag{60}$$

$$L_{m,x} = \frac{1}{N}\sum \left(u.u_x + v.u_y + p_x - \frac{1}{Re}(u_{xx} + u_{yy}) \right)^2 \tag{61}$$

$$L_{m,y} = \frac{1}{N}\sum \left(u.v_x + v.v_y + p_y - \frac{1}{Re}(v_{xx} + v_{yy}) \right)^2 \tag{62}$$

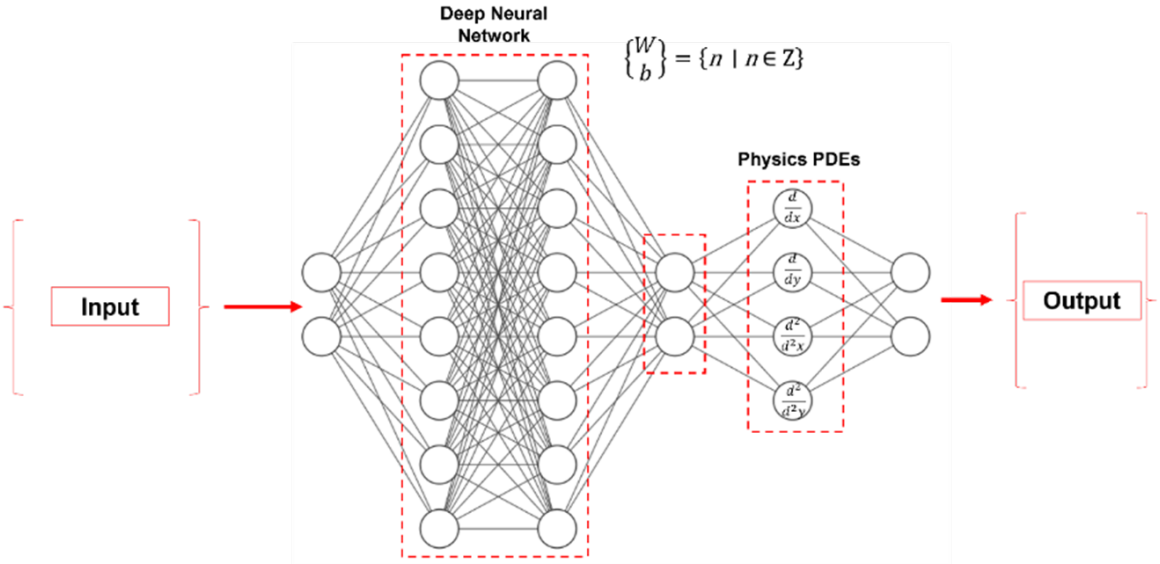


Figure 34: A Schematic Diagram of a PINN with the Navier Stokes Equation as the PDE of Choice.

Algorithm 2 below details the basic code structure of a PINN which can be applied to any language. The main difference from Algorithm 1 (DNN) is the loss function, where Algorithm 2 embeds the physical laws within the loss function.

Algorithm 2 PINN Algorithm

Initialize: Weights W_k, Biases b_k for each layer X
Input: t consisting of an array of values
Output: y consisting of an array of values
for each input-output set **do**
    $X[0] \leftarrow t$ ▷ Input layer + normalization
    for $z = 1$ to k **do** ▷ Hidden layers
        $z[k] \leftarrow W[k] \cdot X[k-1] + b[k]$
        $X[k] \leftarrow \text{ReLU}(z[k])$
    end for
    $\bar{y} \leftarrow W[k+1] \cdot X[k] + b[k+1]$ ▷ Output layer
end for
Compute data loss: $L_{\text{data}} = \text{MSE}(y, \bar{y})$
Compute physics loss: $L_{\text{physics}} = \text{MSE}(\mathcal{N}(\bar{y}), 0)$
Total loss: $L = L_{\text{data}} + L_{\text{physics}}$
Minimize L using L-BFGS optimizer

4.3 Bayesian Neural Network (BNN)

A Bayesian neural network (BNN) is a type of DNN that incorporates Bayesian methods for handling uncertainty in the model's parameters, which are the weights and biases. Traditional neural networks use point estimates for their parameters, which means they treat the weights and biases as fixed values (Goan and Fookes (2020); Jospin et al. (2022)). However, in many real-world scenarios, having uncertainty estimates is essential, especially when dealing with limited or noisy data. In contrast to traditional neural networks, Bayesian neural networks treat the weights and biases as random variables and place probability distributions (prior distributions) over them. Such networks have been extensively used in many practical applications such as near infrared spectroscopy, internet traffic classification and even in the field of astrophysics where BNNs are used for refining mass formulas for astrophysical applications (Thodberg (1996); Utama and Piekarewicz (2017)). For non-probabilistic models such as DNN and PINN mentioned in the previous section, the loss terms for each of the models consist of a mean squared error and the residuals from the physics respectively where the goal is to minimise the mean square error as this would indicate better agreement between the predicted and true data. For probabilistic models such as BNN and BPINN, the main objective is to

estimate the posterior parameter distributions $P(\theta|\hat{x})$ of a set of network parameters without any further information, such that the statistics of the neural network output $x(t)$ closely follows and approximates the true value (the data), $\hat{x}$. Bayes theorem is employed to estimate the posterior probability distribution of the network parameters θ, such that the output of the last layer seen in Equation 52 approximates to the true solution and agrees with the data $\hat{x}$. The posterior can be expressed below:

$$P(\theta|\hat{x}) = \frac{P(\hat{x}|\theta)P(\theta)}{P(\hat{x})} \tag{63}$$

where $P(x\theta)$ refers to the likelihood which is the conditional probability of the data $\hat{x}$ for a set of given fixed parameters θ, $P(\hat{x})$ refers to the evidence and $P(\theta)$ is the prior distribution of the network parameters. A normal distribution $N(\mu, \sigma)$ for the likelihood and the priors is chosen, where μ is the mean and σ is the standard deviation. The table (Table 8) below summarises the values selected as the mean and standard deviation for both the priors and likelihood.

Table 8: Likelihood and Priors Used for a Bayesian Neural Network

Likelihood	$P(\hat{x}\|\theta)$	$\hat{x}\|\theta N(\mu = x(t), \sigma = 0.05)$
Priors	$P(\theta)$	$\theta N(\mu = 0.00, \sigma = 2.00)$

The predictions generated from the model will then be used to calculate the mean and the standard deviation, which is crucial in getting the mean standardised log loss (MSLL). This is similar to how MSE is used the error metrics for predictions generated from the former two neural networks which do not have any Bayesian inference involved. The equation used to calculate the MSLL values can be found in the equation below:

$$L_{MSLL} = \frac{1}{2} log(2\pi x(t)^{*^2}) + \frac{(\hat{x} - \bar{x(t)})^2}{2x(t)^{*2}} \tag{64}$$

Where $\bar{x(t)}$ refers to the mean of the neural network output (predictions) while $x(t)^*$ is the standard deviation of $x(t)$.

Algorithm 3 below presents the main structure of code used for constructing a BNN. Compared to both Algorithm 1 and Algorithm 2, there are some differences. The main difference lies in how the model is trained and fitted. Unlike traditional PINNs or DNNs, the training involves maximising the log likelihood and prior.

Algorithm 3 BNN Algorithm

Initialize: Weights W_k, Biases b_k for each layer X
Input: t consisting of an array of values
Output: y consisting of an array of values
for each input-output set **do**
    $X[0] \leftarrow t$ ▷ Input layer + normalization
    for $z = 1$ to k **do** ▷ Hidden layers
        $z[k] \leftarrow W[k] \cdot X[k-1] + b[k]$
        $X[k] \leftarrow \text{ReLU}(z[X])$
    end for
    $\bar{y} \leftarrow W[k+1] \cdot X[k] + b[k+1]$ ▷ Output layer
end for
Compute likelihood distribution for data
Compute prior distribution for data
Compute posterior distribution (likelihood + prior)
Compute samples from HMC Sampler
for sample $= 1$ to N_{sample} **do**
    for each input-output set **do**
        $X[0] \leftarrow t$ ▷ Input layer + normalization
        for $z = 1$ to k **do**
            $z[k] \leftarrow W[k] \cdot X[k-1] + b[k]$
            $X[k] \leftarrow \text{ReLU}(z[X])$
        end for
        $\bar{y} \leftarrow W[k+1] \cdot X[k] + b[k+1]$
    end for
end for

4.4 Bayesian Physics Informed Neural Network (BPINN)

Similar to a BNN, the main objective of a Bayesian Physics Informed Neural Network (BPINN) is to estimate the posterior parameter distributions of both network and physics parameters. In a similar fashion to the equation presented in the previous section, the posterior probability distribution of the network parameters θ is estimated, such that the output of the last layer approximates to the true solution and agrees with both the data $\hat{x}$ and physics r. The posterior for a BPINN model can be expressed below:

$$P(\theta|\hat{x},r) = \frac{P(\hat{x},r|\theta)P(\theta)}{P(\hat{x})P(r)} = \frac{P(\hat{x}|\theta)P(r|\theta)}{P(\hat{x})P(r)} \tag{65}$$

where $P(\hat{x},r|\theta) = P(\hat{x}|\theta)P(r|\theta)$ refers to the two likelihoods which is the conditional probability of the data $\hat{x}$ and the conditional probability of the physics r for a set of given fixed parameters $\theta, P(\hat{x})P(r)$ refers to the marginal likelihood or evidence and $P(\theta|\hat{x},r)$ is the posterior. A similar normal distribution $N(\mu,\sigma)$ for the likelihood and the priors is chosen, where μ is the mean and σ is the standard deviation. A table (Table 9) below summarises the values selected as the mean and standard deviation for both the priors and likelihood. A pseudo code sample can be found in Algorithm 4 below, where the structure is similar to Algorithm 3. The major difference is the embedding of physical laws when both likelihood and prior distributions are calculated.

Table 9: Likelihood and Priors Used for a Bayesian Physics Informed Neural Network.

Data Likelihood	$P(\hat{x}\|\theta)$	$\hat{x}\|\theta N(\mu = x(t), \sigma = 0.05)$
Physics Likelihood	$P(r\|\theta)$	$r\|\theta N(\mu = x(t), \sigma = 0.05)$
Priors	$P(\theta)$	$\theta N(\mu = 0.00, \sigma = 2.00)$

Algorithm 4 BPINN Algorithm

Initialize: Weights W_k, Biases b_k for each layer X
Input: t consisting of an array of values
Output: y consisting of an array of values
for each input-output set **do**
  $X[0] \leftarrow t$ ▷ Input layer + normalization
  for $z = 1$ to k **do** ▷ Hidden layers
    $z[k] \leftarrow W[k] \cdot X[k-1] + b[k]$
    $X[k] \leftarrow \text{ReLU}(z[X])$
  end for
  $\bar{y} \leftarrow W[k+1] \cdot X[k] + b[k+1]$ ▷ Output layer
end for
Compute likelihood distribution for data and physics
Compute prior distribution for data and physics
Compute posterior distribution (likelihood + prior)
Compute samples from HMC Sampler
for sample $= 1$ to N_{sample} **do**
  for each input-output set **do**
    $X[0] \leftarrow t$ ▷ Input layer + normalization
    for $z = 1$ to k **do**
      $z[k] \leftarrow W[k] \cdot X[k-1] + b[k]$
      $X[k] \leftarrow \text{ReLU}(z[X])$
    end for
    $\bar{y} \leftarrow W[k+1] \cdot X[k] + b[k+1]$
  end for
end for

4.5 Concluding Remarks

In this chapter, a comprehensive overview of the neural network architectures considered for surrogate modelling was presented, covering both deterministic and probabilistic approaches. The deterministic models, including feedforward neural networks, were formulated with a focus on capturing nonlinear mappings between inputs and outputs in a data-driven manner. In parallel, probabilistic models such as Bayesian Neural Networks (BNNs) and Bayesian Physics-Informed Neural Networks (BPINNs) were introduced to incorporate uncertainty quantification and, where relevant, physics-based constraints. The mathematical formulations behind each model were derived in detail, and their training processes were articulated through algorithmic descriptions to provide clarity on implementation and convergence behaviour. Special attention was given to how each architecture handles generalization, overfitting, and the incorporation of physical priors or governing equations. This chapter lays the groundwork for the surrogate modelling strategies explored in this thesis, and the performance of these neural networks in terms of predictive accuracy and uncertainty representation will be critically evaluated in the chapters that follow.

Chapter 5

Effects of Momentum Ratio and Reynolds Number on Turbulent Flows

A large part of the work presented in this chapter has been reported in Wong et al (Wong et al. (2024b)). Preliminary work was first presented at UK Fluids Conference 2022 in Sheffield, United Kingdom. The primary aim of the research is to understand and quantify the impact of variations in the momentum ratio on turbulent fluid flow within T-junctions. This is done by either varying the branch pipe diameter while keeping the inlet velocity constant or adjusting the branch pipe inlet velocity while maintaining a constant diameter.

This chapter starts by exploring two consistent momentum ratios (98 and 66.4) classified as wall-jet and impinging-jet flows. Based on the description of both CFD models and boundary conditions found in chapter 3, simulations and validation are carried out using the IDDES-SST model. This is presented in section 5.1. The results and analysis from the simulations are discussed and shown in section 5.2. This chapter then concludes with a conclusion found in section 5.3.

5.1 Description of CFD Models and BCs

The computational model of the T-junction is based on the set-up in Hosseini's and Tunstall et al's work (Hosseini et al. (2008); Tunstall et al. (2016)). In this study, 2mm is used as the base size for meshing where the mesh sensitivity study can be found in chapter 3. A more detailed

diagram of the computational domain can be found in Figure 12.

The numerical simulations are conducted using the commercial CFD software Simcenter STAR-CCM+ (Siemens Digital Industries Software (Siemens 2023)). The turbulent flow in the fluid domain is modelled with the IDDES-SST turbulence model (Gritskevich et al. (2012)). This is a hybrid turbulence modelling approach that combines elements of Reynolds Average Navier Stokes (RANS) and Large Eddy Simulation (LES) models. The governing equation for the fluid flow is the Navier Stokes equation and it will be solved within the turbulence model. The boundary conditions and formulation used in IDDES-SST can be found in chapter 2, where they are described in further detail.

5.2 Results and Discussion

The numerical simulation results will be characterized in terms of averaged quantities-of-interest (QoI). It should be noted that the averaging process for the QoI takes place after the flow achieves convergence which takes approximately 18 seconds, namely 15 turnover time. Turnover time is the time it takes for a volume of fluid equivalent to that of the downstream region (after the T-junction exit) to make one pass through the system. The duration of a typical simulation run is set at 70s, with time steps of 0.001s for the computational domain. These averaged QoIs are normalised according to the equations below. The fluctuating components of the velocity vector are denoted as u, v and w. Note that ΔT refers to the difference in temperatures between the main pipe inlet and the inlet branch pipe.

$$T^*_{var} = \frac{\langle (T')^2 \rangle}{(\Delta T)^2} \tag{66}$$

$$T^*_{rms} = \frac{\sqrt{\langle (T')^2 \rangle}}{\Delta T} \tag{67}$$

$$\langle uu \rangle^* = \frac{\langle uu \rangle}{U^2_{bulk}} \tag{68}$$

$$\langle vv \rangle^* = \frac{\langle vv \rangle}{U_{bulk}^2} \tag{69}$$

$$\langle ww \rangle^* = \frac{\langle ww \rangle}{U_{bulk}^2} \tag{70}$$

$$\langle uw \rangle^* = \frac{\langle uw \rangle}{U_{bulk}^2} \tag{71}$$

$$\langle Tu \rangle^* = \frac{\langle T'u \rangle}{\Delta T.U_{bulk}} \tag{72}$$

$$\langle Tw \rangle^* = \frac{\langle T'w \rangle}{\Delta T.U_{bulk}} \tag{73}$$

$$TKE^* = \frac{\langle uu \rangle + \langle vv \rangle + \langle ww \rangle}{2U_{bulk}^2} \tag{74}$$

In this section, results of root mean square (RMS) temperature T_{rms} normalised by ΔT, resolved Reynolds normal stresses ($\langle uu \rangle, \langle vv \rangle, and \langle ww \rangle$) normalised by U_{bulk}, resolved turbulent heat flux ($\langle Tu \rangle^*, \langle Tw \rangle^*$) and resolved turbulent kinetic energy (TKE) are presented and analyzed. The resolved components of turbulent stresses and heat fluxes are computed directly from the instantaneous filtered (resolved-scale) velocity and temperature fields. Specifically, quantities like (Reynolds stresses) are approximated using fluctuations derived from the resolved velocity field. These resolved fluctuations are then used to calculate turbulent momentum and heat transport, while the subgrid-scale (SGS) components are modeled separately. For the IDDES-SST turbulence model within STAR-CCM+, the modelled components of turbulent stresses and heat fluxes are included in the RANS regions. These are computed using the underlying RANS turbulence model and fully account for the turbulent transport there, since no resolved turbulence is available. As the model transitions into the LES region, the resolved fluctuations begin to dominate, and the modeled contributions shrink to only cover the subgrid-scale motions. Note that the values for U_{bulk} is different in the regions upstream and downstream the T-junction, e.g. $U_{bulk,bef} = 0.89m/s$ is the bulk velocity from the upstream elbow, before the

T-junction while the bulk velocity after the T-junction is denoted as $U_{bulk,bef}$. This value is different for some of the cases, i.e. $U_{bulk,aft} = 0.89, 0.90, 0.91, 0.90, 0.90 m/s$ for Cases 1, 2, 3, 4 and 5 respectively. Therefore, the value of U_{bulk} will differ depending on the position of where the spatial points, x, are located.

The specific locations, $x/D_m = 0, 0.292, 1.0$ and 2.0 of the cross sections for the contours, the probe positions (that forms a line) for mean flow profiles, as well as the individual point probes for transient flow QoI, are visually represented in Figure 35(a) and (b). As shown in the figure, apart from the X-Z symmetric plane, multiple cross sections parallel to the Y-Z plane are designated to depict the mean flow behaviour downstream of the T-junction. More details concerning the mean flow behaviour downstream of the T-junction is determined by computing mean flow QoI along the line probes. Additionally, a frequency analysis is conducted for the transient flow QoI measured at the point probes.

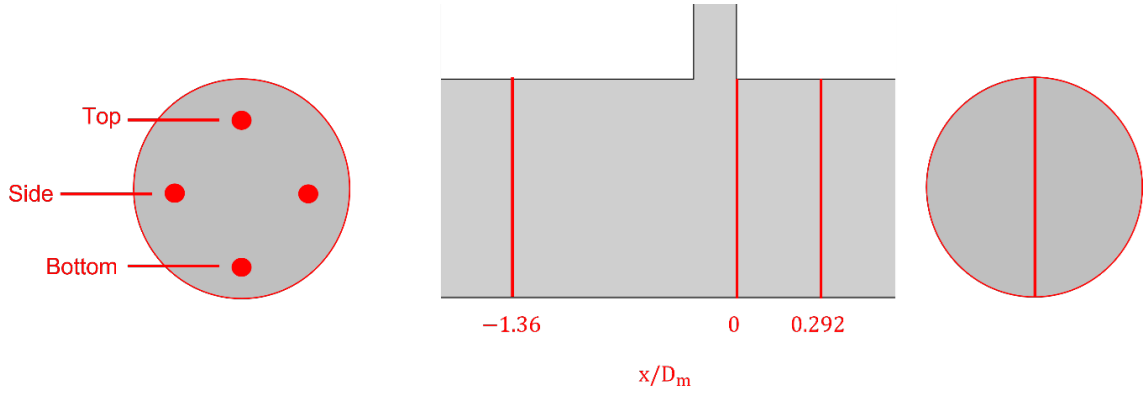


Figure 35: Selected locations of cross sections with (a) point probes used for the analysis (b) a string of linear probes spanning from the top to bottom of the flow is used for sampling the turbulent QoI.

5.2.1 Mean Flow Field

A contour map depicting the mean streamwise component of velocity on the X-Z symmetric plane is shown in Figure 36 for all test cases. Note that the diagrams shows a zoomed-in view of the flow at the T-junction exits and the region after the upstream bend. The faster-moving fluid from the main pipe dominates the bottom of the pipe while the remainder interacts with the slower-moving fluid entering via the hot inlet. This is because the curvature of the bend induces a pressure gradient which casues the fluid to move radially outward. Consequently, the

fluids likely concentrate near the outer wall of the bend, and results in faster-moving fluid in the vicinity of the outer wall of the bend. Due to continuity, the flow in the vicinity of the inner wall slows down. This trend continues until the flow redevelops after the bend. Since the T-junction is located close to the bend, the faster-moving fluid at the bottom of the horizontal part of the main pipe can be still observed in the main pipe.

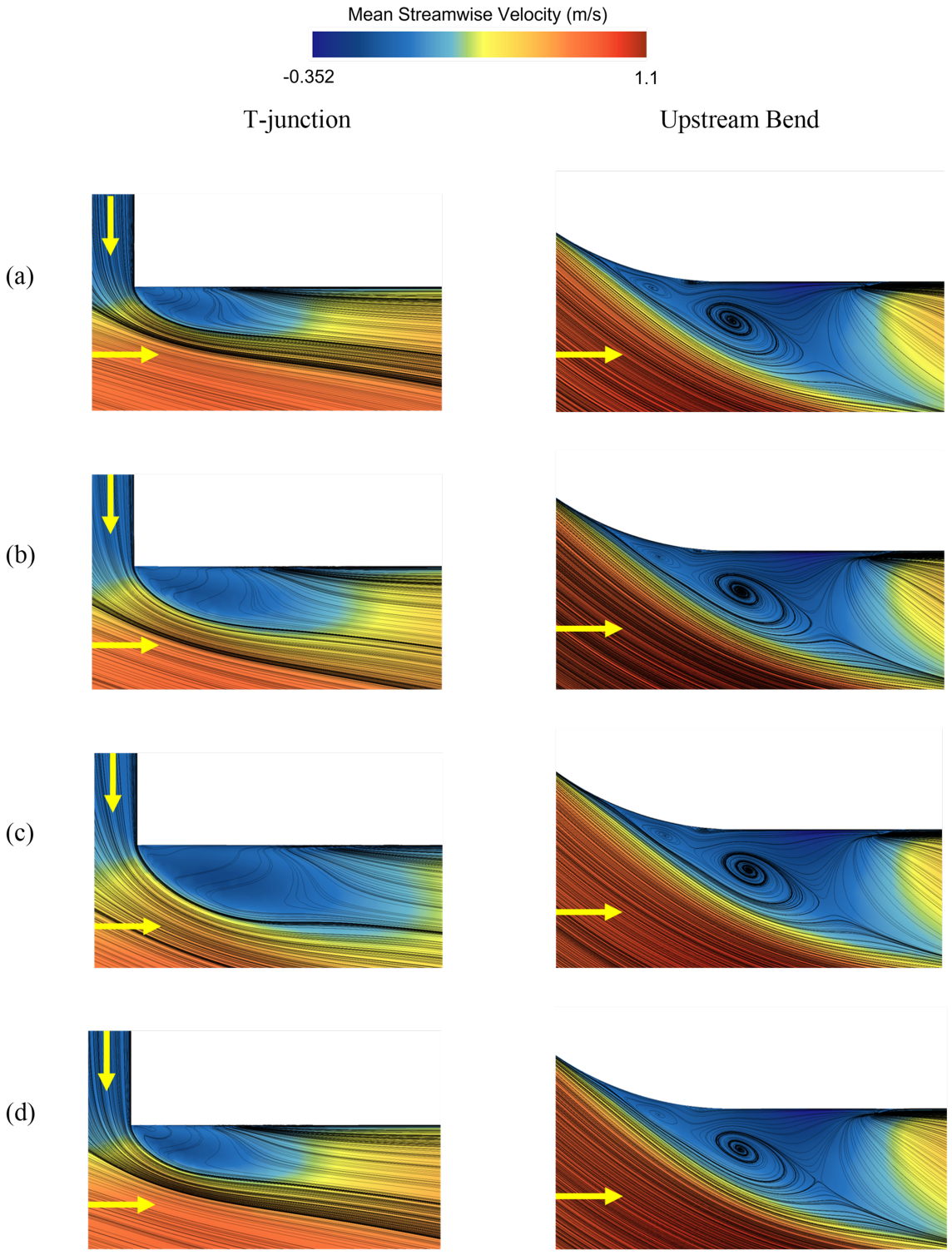
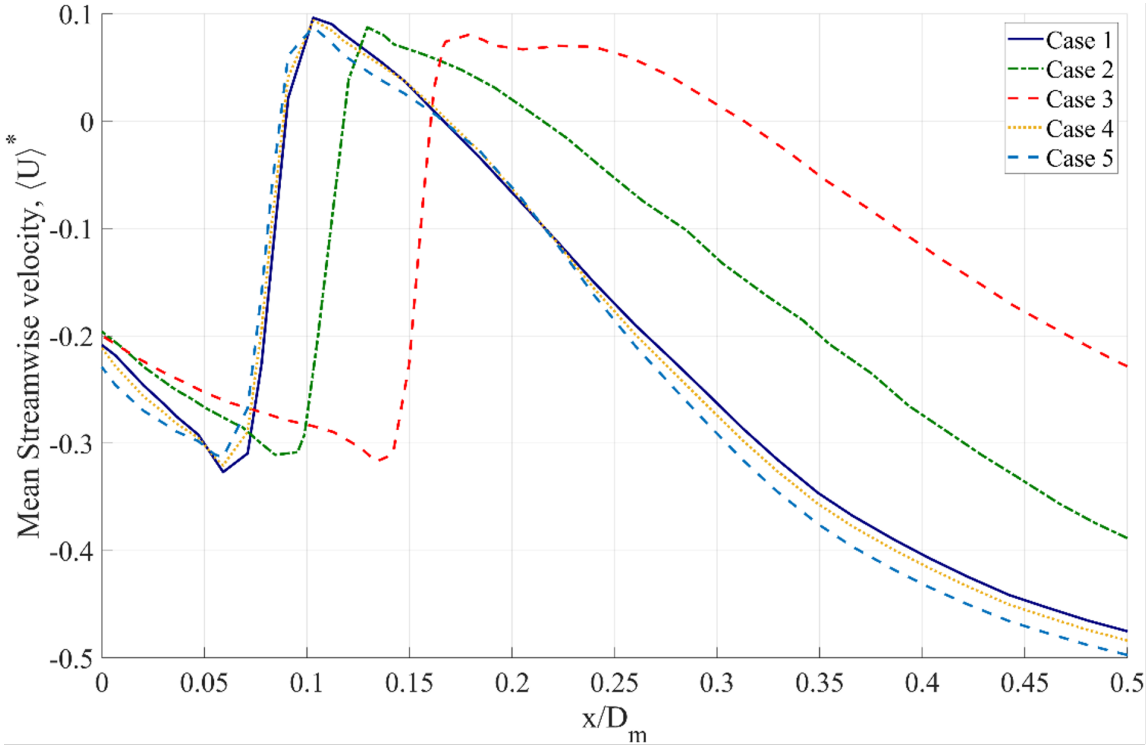


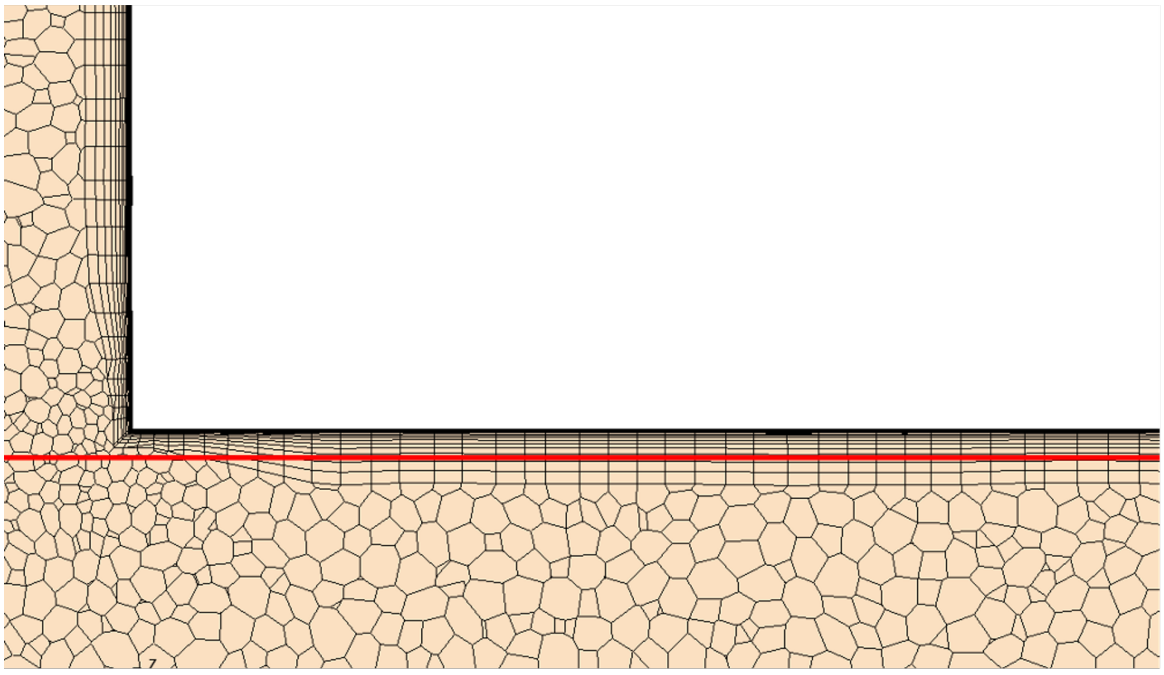
Figure 36: Contours of normalised mean streamwise velocity U (m/s) for Cases (a) 1, (b) 2, (c) 3, (d) 4, and (e) 5.

Similar to a jet in crossflow, a flow recirculation zone is formed immediately downstream of the branch pipe. The recirculation zone can be distinguished with a negative velocity with respect to the velocity of the main flow. The size of the recirculation zone can be distinguished by

the regions with normalised negative mean streamwise velocity $\langle V_x \rangle^*$ seen in Figure 37. It also be seen that the size of recirculation zone increases with a larger inlet branch diameter. One of the possible explanations is that the momentum of the hot inlet flow (M_b) being directly proportional to the volume of fluid entering the main pipe at any point in time, which agrees with the mass inlet flow rate seen in Table 3. With a smaller inlet branch diameter, the momentum of the hot inlet fluid is greatly reduced, and thus it does not interact and affect the flow of the cold inlet fluid as strongly.



(a)



(b)

Figure 37: (a) Normalised mean streamwise velocity along the X-Z plane to show where the recirculation zone starts, (b) Probe line (in red) within the prism layer used sampling. Note that this plot is zoomed-in and does not show the full range of velocity values.

The reattachment point is simply the location where the transition between forward and reverse flow occurs, essentially when wall shear stress transitions from the positive region to the negative region. Figure 38 graphically illustrates the wall shear stress close to the upper wall, with the point of origin located at the T-junction exit. Cases 1, 4, and 5 have similar locations for the recirculation point which happen to be the nearest from the point of origin amongst the cases. This is followed by Case 2 and then Case 3 in ascending order. Note that this phenomena could be dependent on the physical geometry of the models in each case, where for a larger D_b, the recirulcation point is found to be further away from the point of origin

The mean streamwise velocity $\langle U \rangle$ profiles are illustrated in Figure 39. These profiles have been normalised with respective $U_{bulk,aft}$ values unique to each case. The mass flow rate of the branch pipe flow for Cases 3, 2, 5 and 4 is in descending order. This means that the bulk flow coming from the branch pipe will interact more strongly with the main flow, which is seen in Case 3 compared to other cases. The acceleration of the fluid observed in Case 3 is located around the middle of the main pipe at $0 < x/D_m < 1.0$. In the range $0 < x/D_m < 1.0$, the deceleration of the fluid at the upper near-wall region of the main pipe is most profound in Case 3 then followed by Case 2. At $x/D_m = 0.292$ specifically, the velocity profile for Case 3 can be seen to extend into the negative domain, while the other profiles do not exhibit such behaviour. This complements the findings in the previous figure where the recirculation zone for all cases spans and originates at different distances from the T-junction exit. The acceleration of the fluid observed in Case 3 is located around the middle of the main pipe at $0 < x/D_m < 1.0$. It is reasonable to say that a larger momentum allows the branch flow to penetrate deeper into the main pipe flow and create a larger recirculation region. Although the momentum ratio in Cases 3 and 5 are similar, the mean streamwise velocity profile in the latter is closer to that of Case 1. Similar observations can be made when comparing the U of Case 4 to that of Case 2.

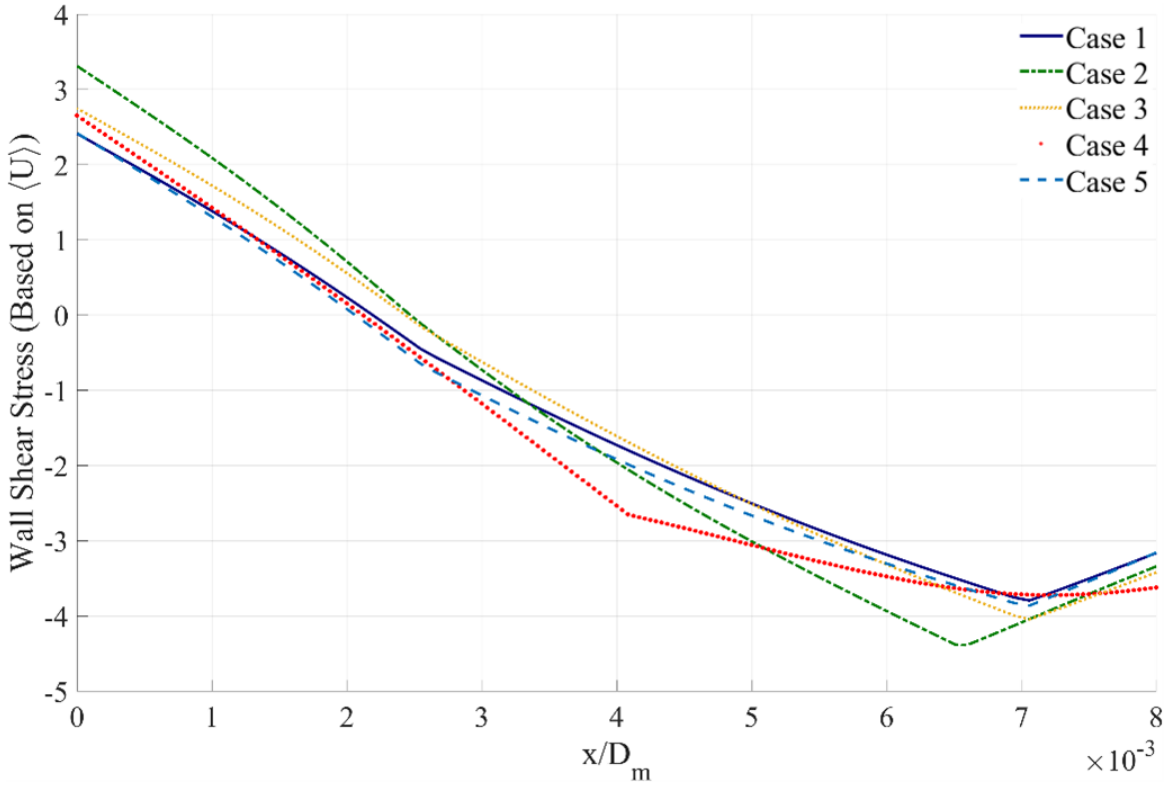


Figure 38: Streamwise wall shear stress in the upper wall. Note that $x/D_m = 0$ corresponds to the exit of the T-junction.

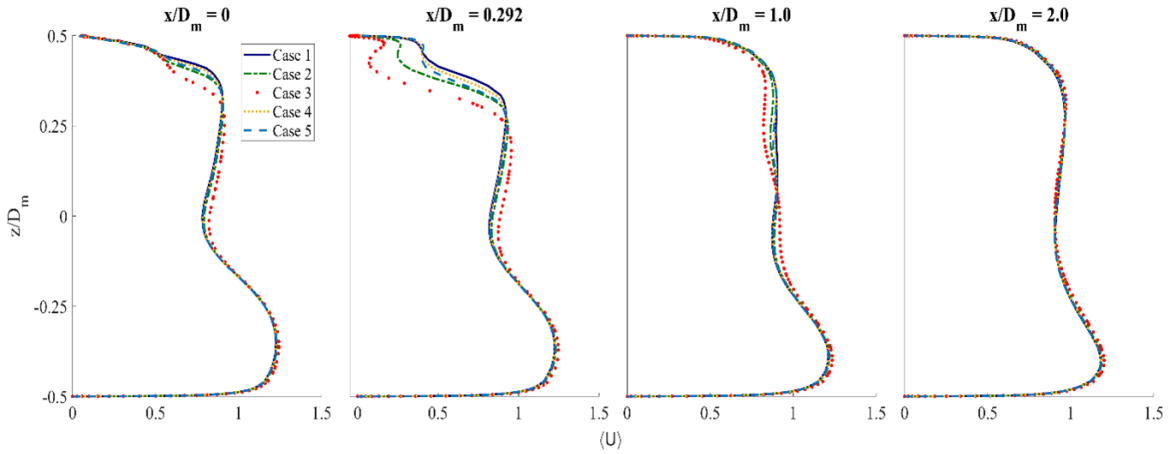


Figure 39: Profiles of mean streamwise velocity magnitude $\langle V_x \rangle$through the main pipe centre..

The mean secondary flow structures, shown by streamlines, and the root-mean-square temperatures at pre-defined cross-sections are illustrated in Figure 40. In all cases, there is a pair of smaller vortices that exist immediately after the T-junction, in addition to the main pair of counter-rotating vortices in the upper half of the flow. However, those smaller vortices disappear very quickly in the downstream locations seen in the Figure 40(b). Figure 40(c) represents the locations where snapshots for Figure 40(b) are taken from. A similar four-vortex system

is also reported in Tunstall et al's, however the results show that a four-vortex system exists further downstream at $x/D_m = 0.292$ (Tunstall et al. (2016)). One of the possible reasons is that the IDDES-SST solver exhibits a higher level of dissipation in comparison to LES, which could explain the faster damping of the additional vortex pair in this study. Additionally, the main flow is deflected around the injected fluid that leads to the second pair of vortices (Dean Vortices) that forms upstream of the T-junction exit to be weakened. The size of vortices are known to be related to their strength. This is observed and shown in Tunstall at al's study, where smaller secondary vortices coalesce with the original larger pair of vortices seen upstream and merge into a larger pair of vortices. This behaviour has also been reported in Khan et al where pairs of smaller scaled vortices coalesce were confirmed to form larger vortices (Khan et al. (2023)). The location further downstream of the T-junction has a dual vortex structure, similar to the structure seen upstream of the T-junction, such as $x/Dm = 0.292$. The upper two smaller vortices observed at the T-junction exit generally tend to weaken in terms of the size of vortices as they are convected downstream of the T-junction. This behaviour is seen to be consistent across all three cases.

T^*_{rms} at each cross section is plotted and overlaid with the velocity streamlines in the same diagram (see Figure 40(a)). This offers more information on the region with higher temperature fluctuations. As expected, regions with higher values of T_{rms} can be found in the the upper wall section, with a characteristic U-shape. The region where the majority of the temperature fluctuations occur can be found at the top, near-wall region of the pipe and this is consistent across all cases. At $x/D_m = 0, 0.292$, the extent of the temperature fluctuations spreads out, whereas at $x/D_m = 1.0$ smaller temperature fluctuation can be seen which is expected due to it being further away from the hot mixing of the inlet flows (from both the branch pipe and main pipe). Similar behaviour was observed in $x/D_m = 2.0$, hence it has been omitted. Interestingly, T_{rms} becomes nearly zero as $x/D_m > 1.0$ in Cases 1, 4, and 5. Meanwhile, T_{rms} at $x/D_m = 1.0$ in Case 2 is generally smaller than that in Case 3. This observation indicates that the temperature mixing process has been elongated by the larger branch pipe diameter and is inline with the observation of the positive relationship between the branch pipe diameter and size of the recirculation zone.

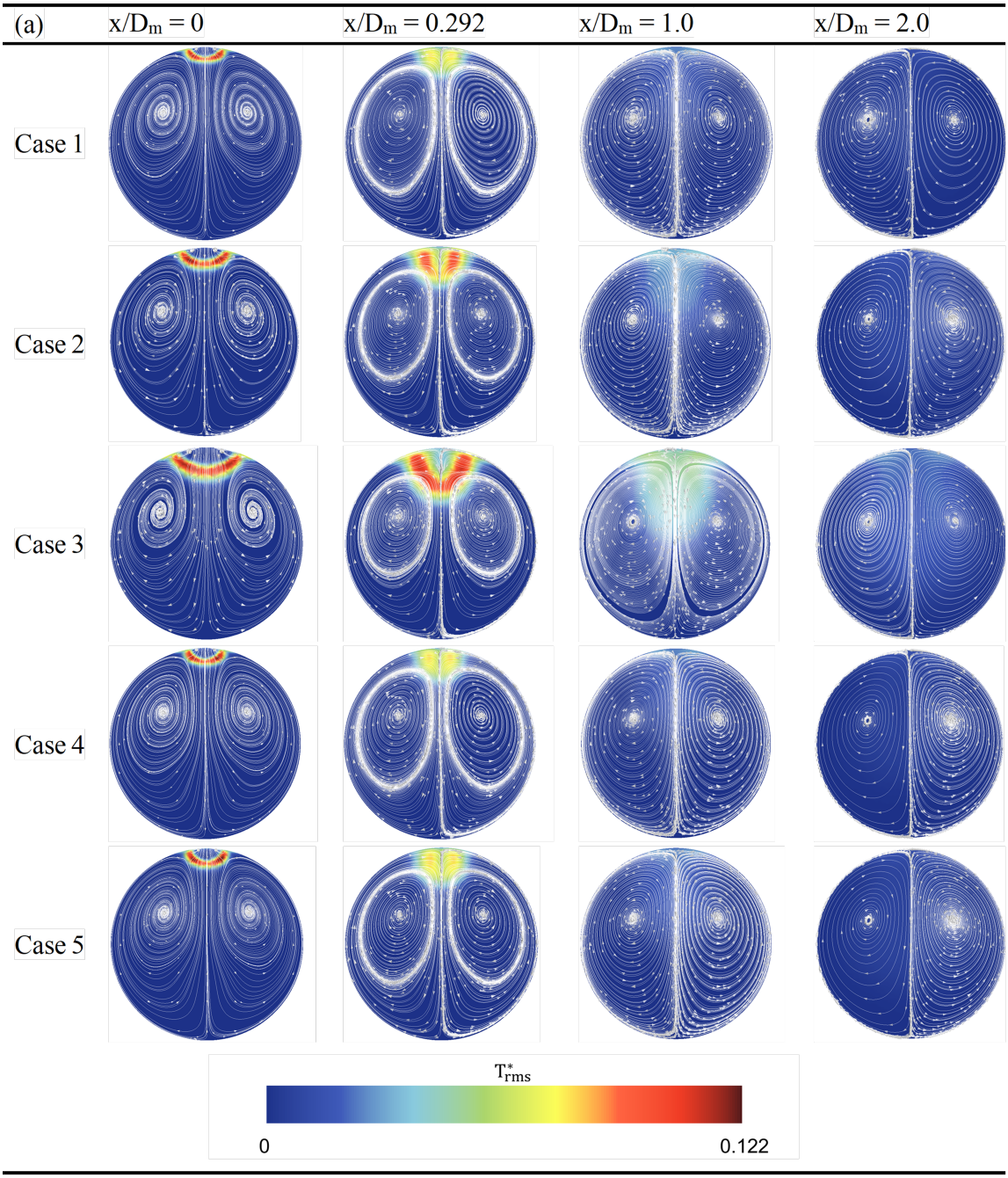
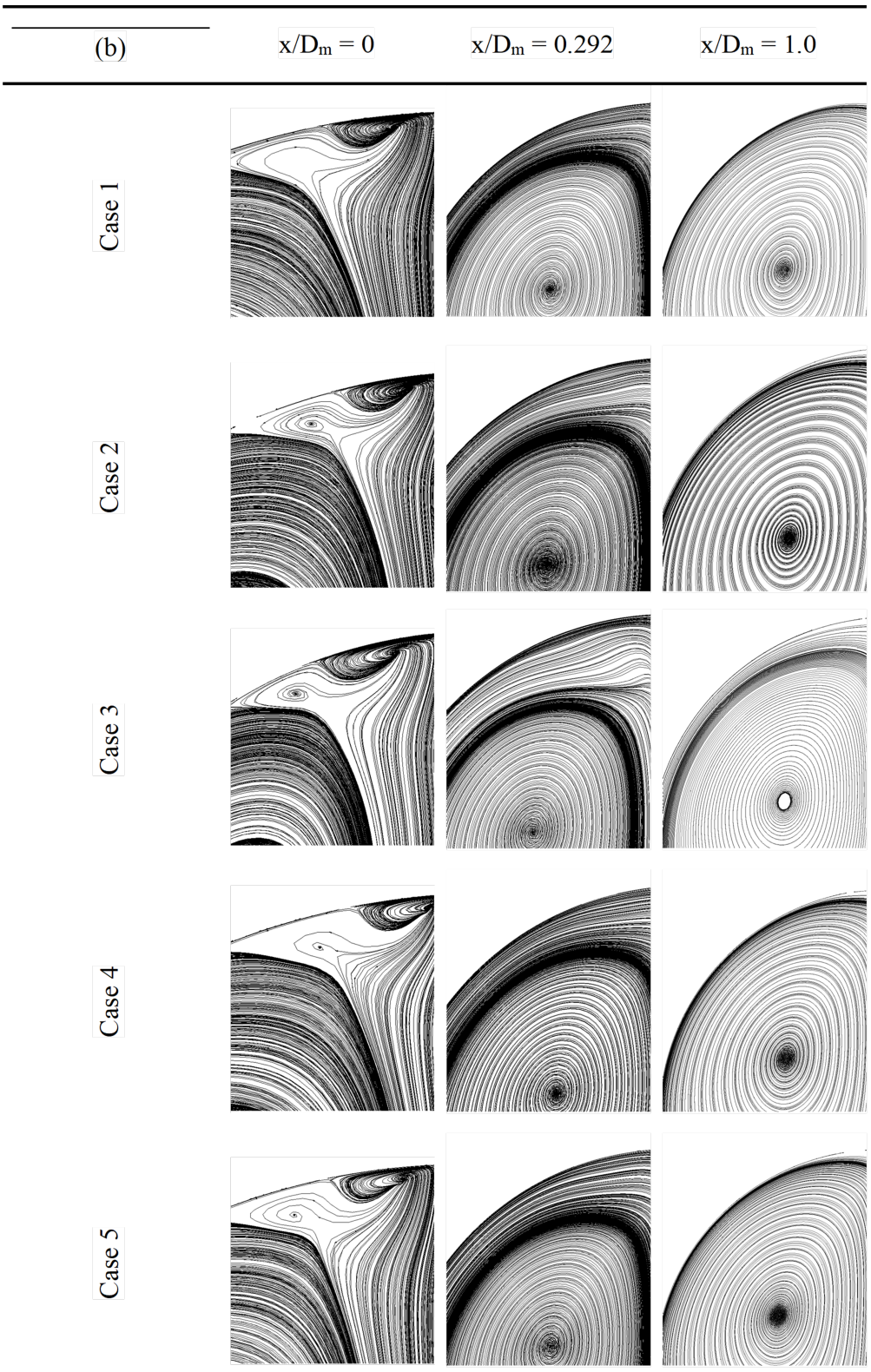
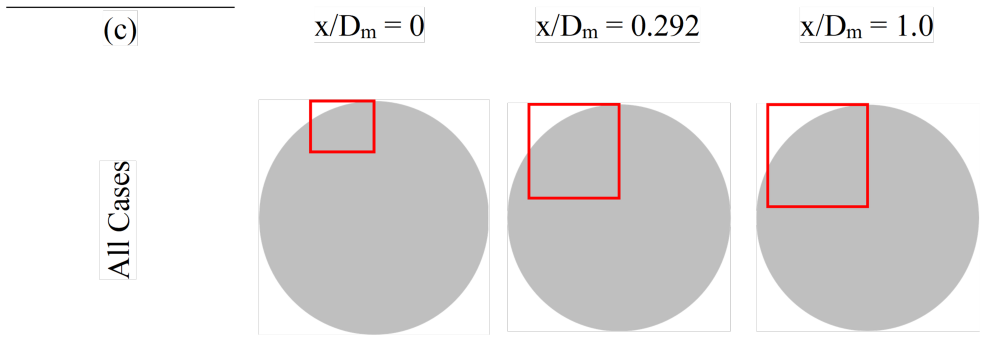


Figure 40: (a) Cross-sectional velocity streamlines at selected locations



(b) Zoomed-in streamlines at the upper wall for the location $x/D_m = 0, 0.2921.0$. Streamline plots for $x/D_m = 2.0$ have been omitted as they are generally similar to $x/D_m = 1.0$. Colours indicate the root mean squared temperature while streamlines represent the velocity vector field



(c) Location where snapshots for zoomed-in plots in (b) are taken from.

5.2.2 Turbulent QoIs

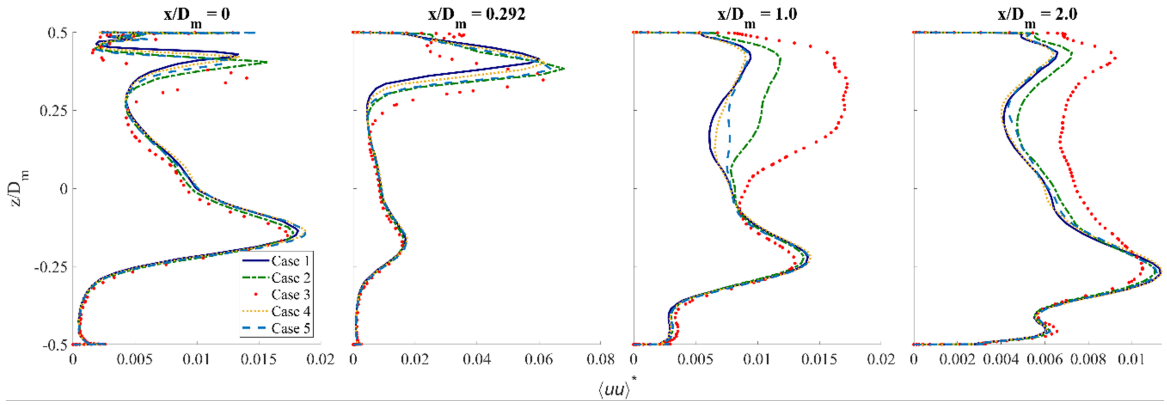
Figure 41 shows the profiles of the resolved components of $\langle uu \rangle^*$ and $\langle uw \rangle^*$ at the pre-defined line probes. Since Reynolds stress represents the transfer of momentum within the turbulent flow, the highly negative stress values seen in some of these components indicate a transfer of momentum in the positive direction while the opposite holds true for positive stress values seen in some of the plots below. As expected, the profiles of $\langle uu \rangle^*$ and $\langle uw \rangle^*$ in different cases is visibly different from each other in the upper region of the main pipe, while the $\langle uw \rangle^*$ in the lower region of the main pipe remains similar.

For $\langle uu \rangle^*$ at $x/D_m = 0$ and 0.292, seen in Figure 41(a), a peak is observed across all cases in the upper half of the pipe. This can be explained by the influx of fluid from the branch inlet and this promotes the transfer of momentum in the forward (positive) direction. In the lower half of the pipe, a secondary peak is also observed but it is much smaller than the peak observed in the upper half of the pipe. It can be deduced from this observation that there is a higher transfer of momentum in the upper half as compared to the lower half. Within the upper half of the flow, Cases 2 and 3 see some differences in the upper half region where the position of the primary peak is observed at a position nearer to the central pipe axis compared to the rest, although these differences observed are relatively small. At $x/D_m = 1.0$ and $x/D_m = 2.0$, Case 3 shows the largest deviation from the other cases. The Reynolds stress profiles for cases 1, 2, 4 and 5 all take on a generally similar trend at these two locations.

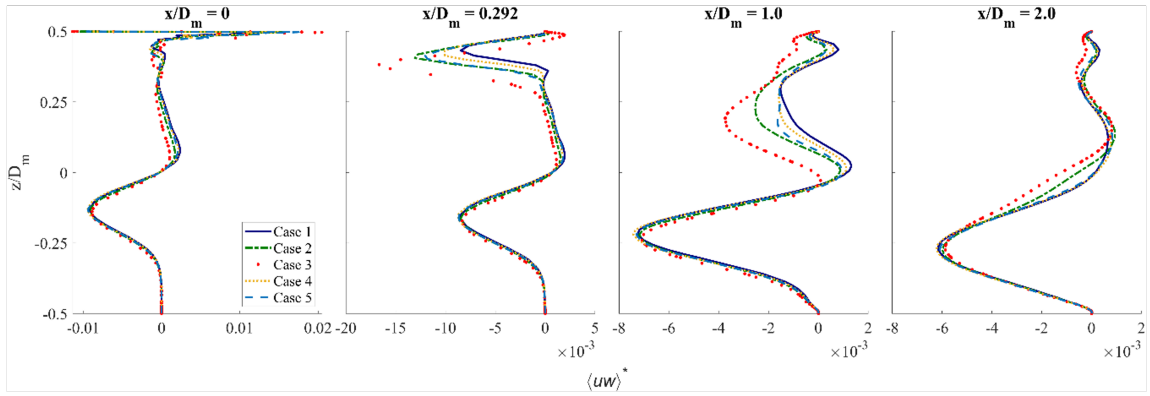
On the other hand, for $\langle uw \rangle^*$ at $x/D_m = 0$ and 0.292 shown in Figure 41(b), a peak is seen across all cases in the upper half of the pipe, but now in the negative direction. Conversely, in the lower section, a secondary peak is present, though notably smaller than the peak in the upper section. This is also in the negative direction, which suggests a greater momentum transfer in the upper section compared to the lower. However, Cases 2 and 3 exhibit variations in the lower section, with the secondary peak positioned closer to the central pipe axis. The downstream behaviour at $x/D_m = 1.0$ and $x/D_m = 2.0$ is very similar to the $\langle uu \rangle^*$ component of Reynolds stress, with Case 3 displaying the largest deviation, albeit less extreme, which has been mentioned above. The only major difference is the negative direction shown in the $\langle uw \rangle^*$ plots. the secondary peaks seen in the lower half of the plots at these locations are larger than the primary peaks in the upper half of the flow, which seems to shift as the fluid flows downstream, the bulk of momentum transfer gradually shifts to the lower half of the flow.

In both Reynolds stress plots, Case 3 deviates the most from the other cases and it is likely due to its significantly higher Reynolds number found within the branch inlet, even though the momentum ratio found in Case 3 is the same as Case 5. These results shows that the effect of the inlet branch diameter exhibits a more significant impact on the turbulence, as opposed to the influence of momentum ratio. This phenomena is more distinguishable further downstream of the T-junction.

The profiles of resolved TKE^* are illustrated in Figure 42. It is depicted in the figure that TKE^* is higher in the upper half of the pipe. This trend is consistent across all the locations. At $x/D_m = 0$, the turbulent kinetic energy plots are very similar for Cases 1, 2 and 4, with some differences seen in Cases 3 and 5. Note that these differences exist even though the momentum ratio is the same in Cases 3 and 5. At $x/D_m = 0.292$, the peaks seem to congregate in the upper half of the pipe with Case 5 having the highest peak within that region. This is followed by Case 3, 2, 4, 1 in descending order of TKE even though their differences are very small. This suggests that closer to the T-junction exit, the impact of the branch pipe diameter is insignificant, where more turbulence is observed in Case 5. Instead, the higher inlet branch velocity can potentially explain this behaviour. The results seen for the other cases are not too different from each other, which is most likely due to their minimal difference in inlet branch velocities. There is a lower inflexion point (where the peaks are) for the turbulent kinetic energy in this region in Cases 2 and 3, possibly due to the larger recirculation region for these two cases which can be expected to the influx of larger momentum from their respective branch inlets. At both $x/D_m = 1.0, 2.0$, the turbulent kinetic energy for Case 1 and 4 converges and take on very similar profiles, whereas Case 2, 3 and 5 shows more differences than similarities. The TKE for Case 3 shows the most deviation from the other cases. This could mean that if the flow is sufficiently far away from the T-junction exit, the impact of the diameter of the branch pipe on turbulent kinetic is more significant than the momentum ratio and its effects are more distinguishable within this region.



(a)



(b)

Figure 41: Reynolds Stress along the components (a) $\langle uu \rangle^*$ (b) $\langle uw \rangle^*$ at various locations.

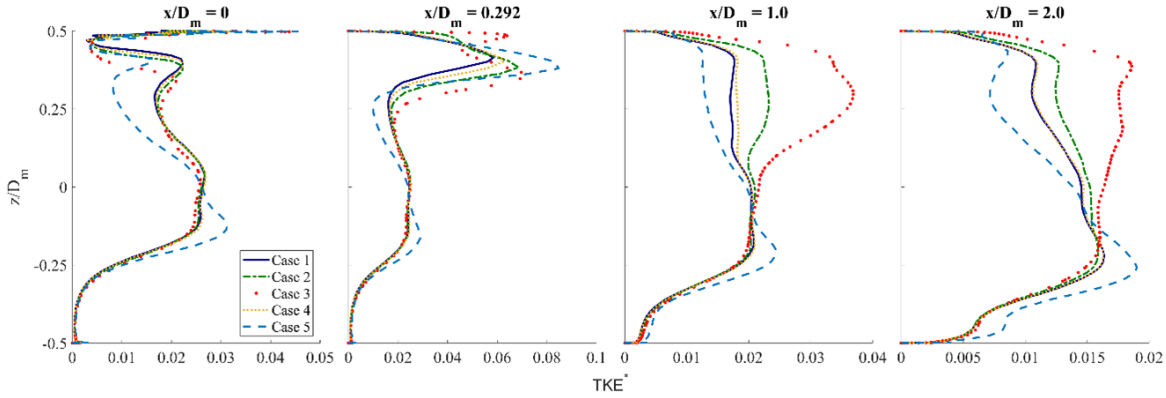
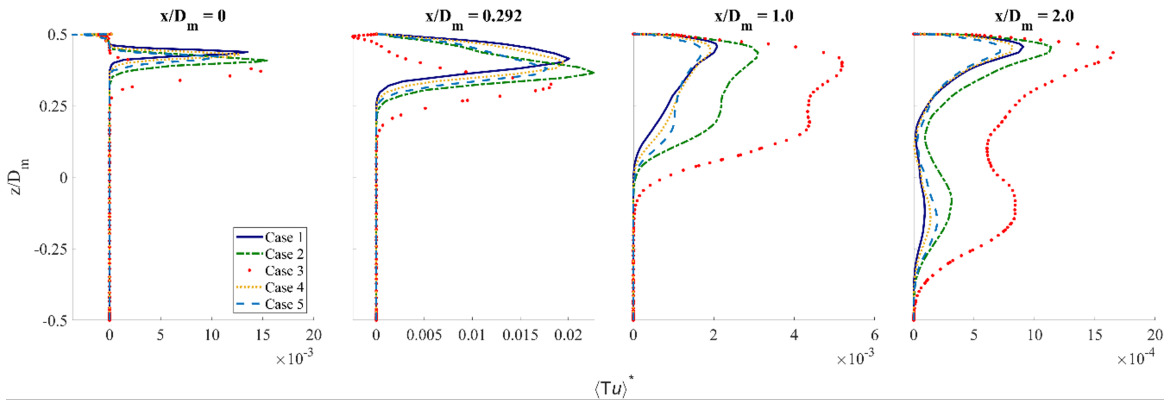


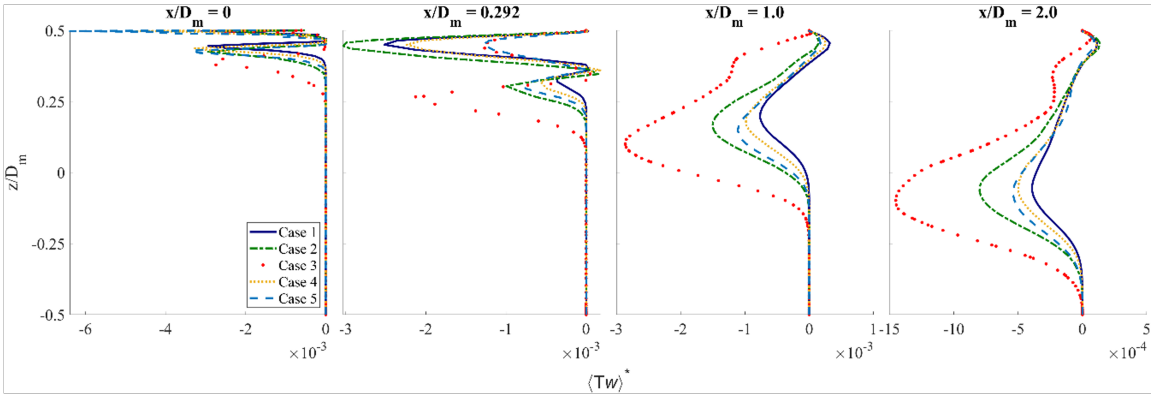
Figure 42: Normalised turbulent kinetic energy (TKE^*) at prescribed locations.

Figure 43 shows the normalised resolved turbulent heat flux profiles along the V_x and V_k component. The peaks for each case congregate within the same near-wall region in the upper half of the pipe at $x/D_m = 0$. Locations of these peaks plotted in Figure 43 correspond to the thermal mixing layer between the main flow and recirculation region. Shifting downstream of the T-junction, these peaks spread out and enters the lower half of the flow at $x/D_m = 1.0$ and

$x/D_m = 2.0$. This is consistent with the known fact that these locations are situated within the re-circulation region where eddies are observed. This is also known as regions with flow instabilities which intensifies the convective heat process. All these observations are similar across all cases. At both $x/D_m = 1.0, 2.0$, Case 3 has a much higher turbulent heat flux compared to the other cases, especially with regards to Case 5 for the same momentum ratio. While Case 2 and Case 5 have similar Reynold's number, the profiles of the turbulent heat flux show the observable differences, and such difference is much larger compared to the difference of the heat flux in Case 5 and Case 4. Given this observation and the fact that Case 3 has a higher Reynold's number, it could potentially support the hypothesis that the diameter of the branch pipe has a more significant impact on the turbulent heat flux than the momentum ratio or Reynolds number. In the streamwise direction, a higher turbulent heat flux is seen in the lower half of the flow in the upstream bend across all locations. This phenomenon is consistent with findings from Liberto et al (Liberto and Ciofalo (2013)) where they investigate the turbulent heat transfer in curved pipes using numerical simulation and the turbulent heat flux profiles is similar.



(a)



(b)

Figure 43: Turbulent heat flux in the streamwise and vertical spanwise directions at selected locations.

Figure 44 shows the circumpolar plot of temperature variance around the pipe wall, normalised T^*_{var} for each cross-sectional location. Since the type of jet found in cases 1, 2 and 4 is a wall-jet while the jet seen in cases 3 and 5 is a re-attached jet, there is zero temperature variance in the lower half of the flow which thus has been excluded from the plot. Also, the temperature profiles are symmetrical about the vertical axis, only the first quadrant has been displayed for comparison purpose. For the location at the T-junction exit, at $x/D_m = 0$, the profile peaks are slightly further away from the centre in Cases 4 and 5, while the other cases' peaks are very similar to each other. This corresponds to the maximum temperature variance plots seen that exists in Cases 4 and 5.

Further downstream at both $x/D_m = 1.0, 2.0$, the temperature variance is seen to spread out over a larger area, while keeping within the upper wall region between 0 to 30 degrees. This behaviour is consistent across all cases. The maximum temperature variance at these peaks sees a gradual decrease downstream of the flow (refer to temperature variance values along the

horizontal axis). Case 3 has a much higher temperature variance compared to the other cases, especially with regards to case 5 for the same momentum ratio. As case 3 has a higher Reynold's number than case 5, this potentially supports the hypothesis that Reynolds number has a more significant impact on the turbulent heat flux than the momentum ratio. Next, circumpolar plots of the normalised turbulent heat flux at each location for each case are plotted. These can be found in Figure 45(a) and (b). Along the V_x component in Figure 45(a), the profiles are seen to hover around the same values and shape which suggests that the heat flux along this component is isotropic. On the other hand, the profiles along the V_k component seen in Figure 45(b) exhibit differences in terms of their shapes and magnitude, suggesting that the heat flux along this component varies with direction and is anisotropic. While Figure 45(a) does not provide much discernible differences between each plot, it can be seen in Figure 43(b) that case 3 exhibits one of the highest turbulent heat fluxes at locations $x/D_m = 0.292, 1.0$ & 2.0. The turbulent heat fluxes seen for the other cases do not seem as highly varying as seen in case 3. This seems to support the hypothesis that with a higher Reynolds number, it has a much larger impact on the turbulent heat fluxes as compared to the effect of having a higher momentum ratio.

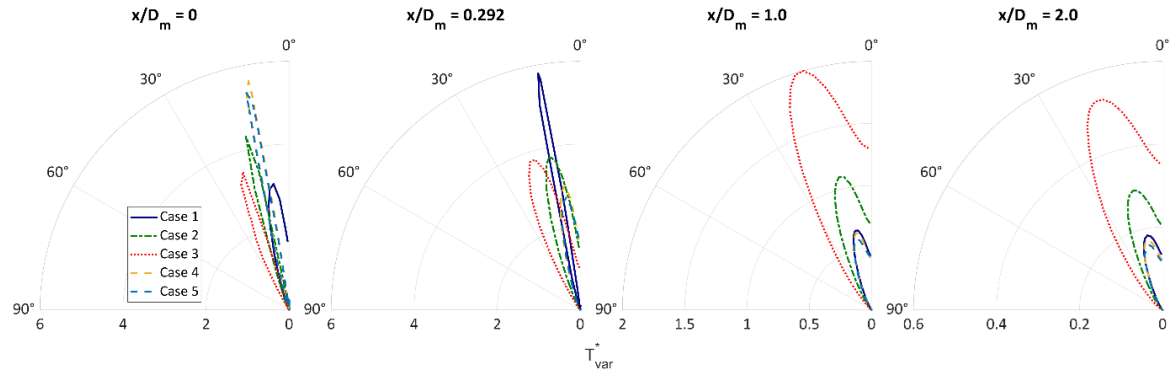
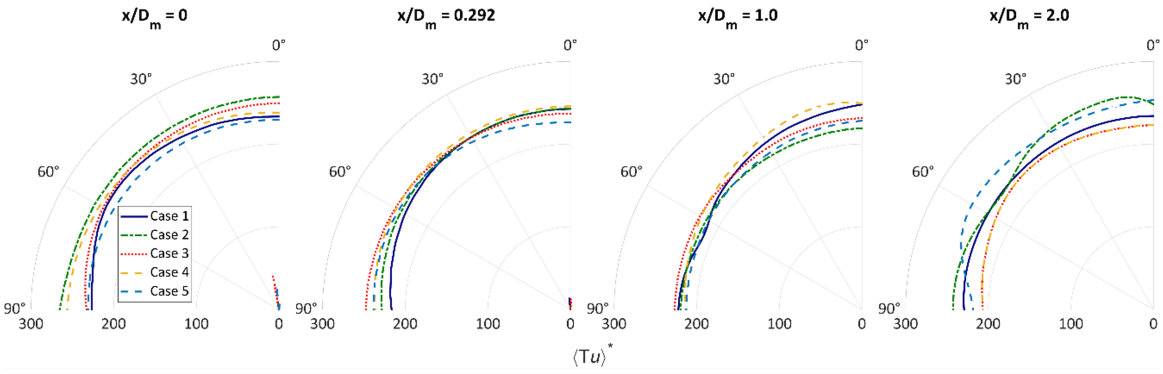
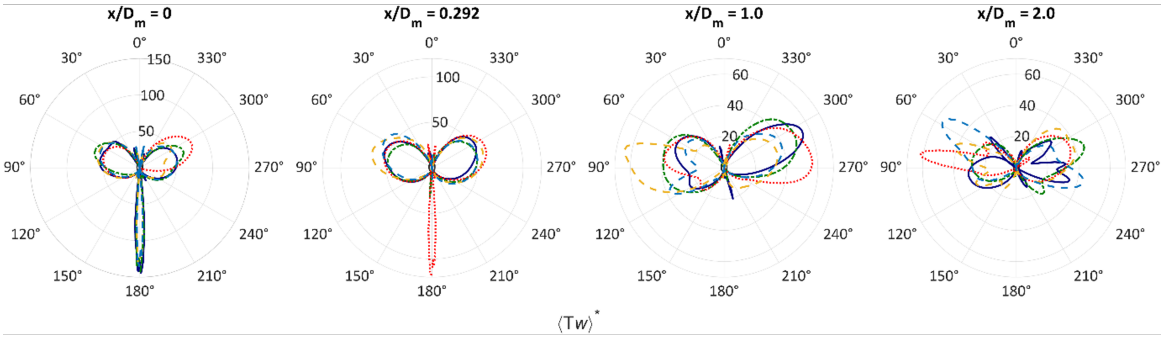


Figure 44: Circumpolar plots of temperature variance at selected locations.



(a)



(b)

Figure 45: Circumpolar plots of turbulent heat flux in the (a) streamwise and (b) vertical spanwise directions at selected locations.

5.2.3 Fast Fourier Transform (FFT) Frequency Analysis

The fluctuating component of the velocity magnitude is sampled every 0.01s for a total of 40s which starts right after the flow achieves a quasi-stable state. This equates to approximately 33 turnover time and gives us approximately 4000 snapshots in time and is done so at the prescribed points previously seen. Considering the Nyquist-Shannon sampling theorem, the Nyquist frequency, f_N is half the sampling rate, f_s where $f_s = 100$ Hz and therefore, $f_N = f_s/2 = 50$ Hz. The highest frequency component is well below 50 Hz and the sampling interval of 0.01s is more than adequate to capture the highest frequency in this signal. One additional point for consideration ($x/D_m = 0.1$) has been included for completeness as it is located in a large recirculation zone. PSD plots of the sample are plotted in Figure 46. It is demonstrated in the figure that a significant portion of the energy of the streamwise velocity fluctuations are well wihtin the range 0 Hz to 50 Hz in all cases. Within 0 Hz to 10 Hz, a dominant frequency can be identified from the peak(s) observed at all locations. This is can be seen in greater detail

via the zoomed-in views of the frequency between 0 Hz and 10 Hz. This dominant frequency is within the range of concern as reported in Chapuliot et al's study on the range of frequencies where fatigue is most likely to happen. It is found in that beyond 10 Hz, there is a significant reduction in the peak stress found within the solid domain, ultimately signalling that only frequencies lower than 10 Hz are meaningful for analysis (Chapuliot et al. (2005)). Referring back to the zoom plots, the peak frequency generally seems to be the same across all cases.

At the location $x/D_m = -1.36$, the peak frequency found in the top probe for Case 1 is the same as the peak frequency found in Cases 2 to 5, found in the bottom and side probe. For the location $x/D_m = 0$, amongst the noise which the graphs captured, the peak frequency that can be distinguished from these plots generally lead up to 5.36 Hz and this observation is found in all cases as noted above, which highly suggests that it is case independent. For the location $x/D_m = 0.1$ which falls right within the large recirculation zone. Meanwhile, for location $x/D_m = 0.292$ the plots in each case have a much clearer definition with respect to each probe (top, side and bottom). The peak frequency of 5.36 Hz is generally observed in each case, with the bottom and side plots taking on very similar profiles. It is interesting to note that at $0.292D_m$, there is a possible presence of a harmonic frequency at 10.71 Hz (twice that of 5.36Hz).

In general, while the plot generated from the top probe across all cases and for each location does not show any correlation to the respective Reynolds number or momentum ratio found in each case, PSD plots generated for the bottom and side probes show a consistent peak frequency which all cases take on and can be taken as the dominant frequency in the flow domain. Converting the frequency to a non-dimensional term such as the Strouhal number, will not change the overall findings. In other words, it can be concluded that the overall dominant frequency is independent of the Reynolds number and momentum ratio, which are both non-dimensional terms.

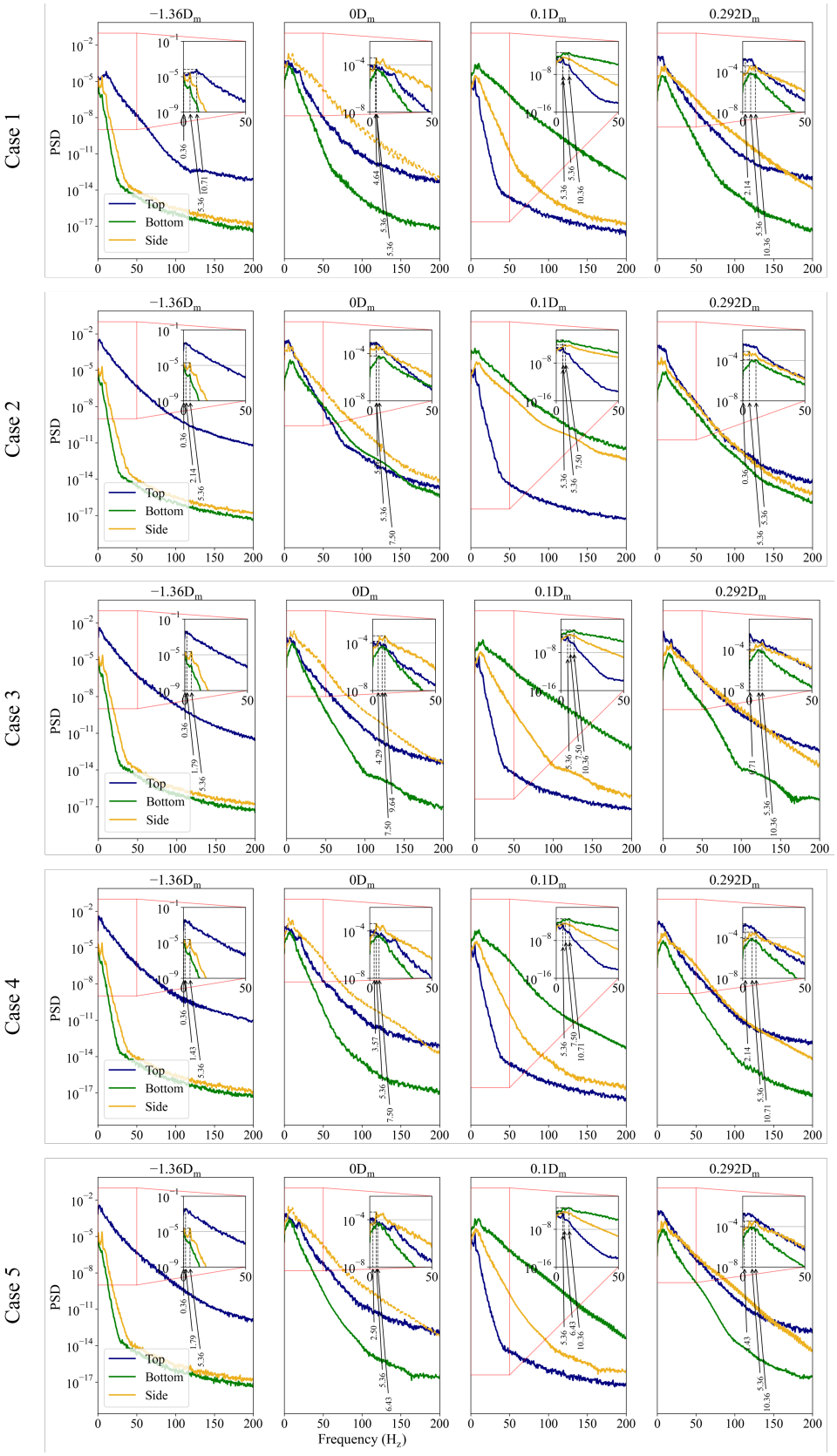


Figure 46: Power Spectral Density of the frequency based on fluctuating velocity magnitude downstream of the T-junction.

5.3 Conclusions

This study examined the mean velocity and turbulent QoI across various T-junction flow test cases with differing momentum ratios and Reynolds numbers. The momentum ratio of the T-junction varied by either adjusting the diameter of the branch pipe or altering the velocity of the branch pipe flow. Turbulent flow is modelled using the IDDES-SST turbulence model, where the suitability of the mesh was justified. The use of IDDES-SST in this chapter resolves coherent turbulence structures that govern hot–cold mixing, providing the necessary flow-field detail to capture the unsteady thermal loading patterns that ultimately drive fatigue initiation in T-junction walls. The numerical results indicated that neither fixing the momentum ratio nor maintaining similar Reynolds numbers for the branch pipe flow ensures consistent turbulent flow characteristics when the branch pipe diameter changes. Instead, it is observed that the diameter of the branch pipe significantly influences the size of the recirculation zone downstream of the T-junction Consequently, this affected the magnitude of turbulent QoI in this region. Specifically, it was found that the size of the recirculation zone is directly proportional to the diameter of the branch pipe. A larger branch pipe diameter resulted in a larger recirculation region downstream of the T-junction, leading to more significant turbulent mixing and higher turbulent quantities. Furthermore, it was noted that while the momentum ratio and Reynolds number ratio did have an impact on turbulent flow behaviours, their influence was secondary compared to the effect of branch pipe diameter. This was particularly evident when the diameter of the branch pipe flow remained constant. In the later section, the analysis of PSD plots generated for the bottom and side probes revealed a consistent peak frequency across all cases, indicating a dominant frequency within the flow domain. What was interesting is that this dominant frequency appears to be independent of the Reynolds number and momentum ratio. By linking dominant flow frequencies to coherent turbulent structures, this chapter identifies the key dynamic drivers of cyclic thermal loading, enabling more informed prediction of thermal fatigue excitation in T-junction geometries. This finding underscores the robustness and stability of the observed flow characteristics, offering valuable insights into the underlying dynamics.

To further substantiate this hypothesis, further investigations may focus on exploring the mechanisms driving this phenomenon and its implications for various applications. For example, an investigation on coherent structures along the cross-sectional planes spanning the X-Y and Y-Z direction could be done. This can be coupled with appropriate analysis tools such as Princi-

pal Component Analysis (PCA) methods. Additionally, it will significantly benefit the CFD community if there are more studies undertaken that explicitly verifies the numerical results generated from IDDES-SST turbulence models against numerical results produced from high-fidelity LES turbulence models. This forms the basis of the next chapter where such techniques are applied.

Chapter 6

A POD and SPOD Study on the Effects of Different Momentum Ratios and Reynolds Number in a T-junction with an Upstream Elbow

A large part of the work presented in this chapter is taken from Wong et al (Wong et al. (2025)). This chapter presents the results from utilising POD and SPOD methods to analyse the complex flow structures arising from various test cases for a PWR T-junction with an upstream elbow. Within this chapter, several objectives are explored and the findings from the respective simulations are then analysed and presented.

The first aim is to compare the flow structures produced in the case of different momentum ratios and/or Reynolds number of the inlet branch flow to determine which factor has most impact on the flow structures produced. The second aim is to compare the differences in the resulting flow modes produced by POD and SPOD. The T-junction test cases use different momentum ratios and Reynolds numbers. In this study these parameters are altered by changing the diameter of the branch pipe and inlet branch velocity. These test cases are similar to the ones used in the previous chapter (chapter 5).

As Reduced Order Modelling (ROM) methods are employed in this chapter, section 6.1 aims to

provide readers with a detailed but concise introduction to POD and SPOD. Note that the base case is recycled from the previous chapter, where a mesh sensitivity study has been performed. For that reason, the same mesh density is adopted and further explained in section 6.2. In subsection 6.3.1, the mean and instantaneous flow fields are presented and discussed. The mean flow field provides a baseline understanding of the overall structure and behaviour of the flow. It is a crucial step before delving into the flow decomposition methods, the Proper Orthogonal Decomposition (POD), in subsection 6.3.2 and the Spectral Proper Orthogonal Decomposition (SPOD) in subsection 6.3.3. This chapter then concludes with a conclusion found in section 6.4

6.1 A Primer on Proper Orthogonal Decomposition (POD)

Proper Orthogonal Decomposition (POD) is a type of dimensionality reduction technique that draws it roots from Principal Component Analysis (PCA). In this study, the POD approach is used for decomposing the turbulent fluctuating flow. The fluctuating velocity components are then decomposed as:

$$u'(x,t) = \sum_{k=1}^{\infty} a_k(t)\, {}_k(x) \tag{75}$$

In the equation above, fluctuating velocity field is decomposed as the sum of products of the deterministic spatial functions$_k(X)$ and its related time coefficients $a_k(t)$. For POD analysis in fluid dynamics, the snapshots represent the fluctuating velocity field sampled over a period of time. This study employs the Snapshot POD, which is an extension of the original above-mentioned POD method by Lumley (Lumley and Payne (1967); Lumley and Berkooz (2003)). This snapshot method was originally introduced by Sirovich where the decomposition involves deterministic temporal modes with random spatial coefficients (Sirovich (1987)). In the original POD method, the decomposition involves deterministic spatial modes and random time coefficients. In the snapshot POD method, the mean flow is removed from the snapshots of data, such that the decomposition is applied on the fluctuating component of velocity Lumley (1981, 1967). Let us denote the matrix of snapshots as S, which can be expressed as:

$$S = \begin{pmatrix} u'(\vec{x}_1, t_1) & \cdots & u'(\vec{x}_n, t_1) \\ \vdots & \ddots & \vdots \\ u'(\vec{x}_1, t_m) & \cdots & u'(\vec{x}_n, t_m) \end{pmatrix} \tag{76}$$

where $u'(\vec{x}_n, t_m) = u(\vec{x}_n, t_m) - \bar{u}(x_n)$ and $\bar{u}(x_n)$ is the fluctuating component of velocity at location $\vec{x}_n$ and m refers to the number of snapshots or time samples.

The covariance matrix C is calculated using Equation 77. In Equation 78, $S^T S$ captures the inner products between different snapshots and C provides a measure of how much the data varies across the snapshots. Eigenvalues $(\lambda_1, ..., \lambda_n)$ are the eigenvalues of C and reordered from largest to smallest, meanwhile the relevant eigenvectors are arranged as columns in an n×n matrix, .

$$C = \frac{1}{m-1} S^T S \tag{77}$$

$$\phi = \begin{pmatrix} {}_{1,1} & \cdots & {}_{1,n} \\ \vdots & \ddots & \vdots \\ {}_{n,1} & \cdots & {}_{n,n} \end{pmatrix} \tag{78}$$

Mathematically, the energetic content of each mode is determined by its corresponding eigenvalue, which is derived from the decomposition of the covariance matrix of the velocity field. Larger eigenvalues correspond to modes that contain more energy, meaning they capture the most significant flow structures. The first mode generally represents the dominant flow pattern and has the highest energy content, while subsequent modes represent smaller-scale structures or more localized fluctuations and contain progressively less energy. Hence by mathematical induction, the contribution to the energy from POD mode 1 is greater than POD mode 2 and this is true for $\lambda_n > \lambda_{n+1}$. The contribution of the POD modes can be described with respect to the sum of the energy contributed by all other modes in the expression below:

$$\text{Energy Percentage of Mode } i = \frac{\lambda_i}{\sum_{j=1}^{n} \lambda_j} \times 100\% \tag{79}$$

Spectral proper orthogonal decomposition (SPOD) is an extension of POD. While the mathematical basis is the same as Equations 75 and 76, the main difference is the covariance matrix that incorporates the spectral (frequency) information into the decomposition process (Schmidt and Colonius (2020)). The covariance matrix in SPOD, C_{spod} is not constructed directly from the time-domain snapshots but based on data that has been converted into the frequency domain using a Fast Fourier transform. This allows for C_{spod} to be formed based on the spectral domain. This can be found in Equation 80 below where $S(w)$ represents the snapshot matrix in the frequency domain at the frequency w. This lays the foundation for future work that involves POD as for reduced order models in turbulence modelling for this geometry. It is worth noting that some work has been done prior to this, where a POD-Galerkin based method which has been developed to model turbulence in straight T-junctions (Georgaka et al. (2020)).

$$C_{\text{SPOD}} = \frac{1}{m-1} S(w)^T S(w) \tag{80}$$

6.2 Description of CFD Models and BCs

As this section is an extension of the works cited in chapter 5, the CFD models and BCs used are the same. A more detailed diagram of the computational domain can be found in Figure 12 while the description of the cases used in this study can be found in Table 2, in chapter 3. The boundary conditions and formulation used in IDDES-SST can be found in chapter 2, where they are described in further detail.

6.3 Results and Analysis

6.3.1 Mean and Instantaneous Flow Structures

Figure 47 illustrates the streamlines formed in the mean flow, which are coloured with the mean streamline velocity magnitude. Within this figure, there are two areas where recirculation can be distinguished which are further detailed in Figure 47. It is clearly shown in Figure 47(a) that a large pair of counter rotating vortices (with a horseshoe pattern) can be found hovering in the upper wall region just downstream the elbow bend. It originates from the elbow bend itself due to the centrifugal force as well as the curvature of the pipe. Also, the small vortices and the counter rotating flow pattern also observed in the edges of the branch pipe seen in Figure 47(b) and Figure 47(c). There is also a vortex breakdown region seen at the edge of the T-junction exit. This phenomenon can be characterised as a straight vortex that experiences breakdown from the recirculation and flow reversal (Escudier (1988); Keller (1995); Brucker (1997)). It is interesting to note that these regions have been reported in straight T-junctions in the downstream region, where the pipe elbow is absent (Frank et al. (2010); Baranova et al. (2021); Lampunio et al. (2021, 2022); Wong et al. (2024b)).

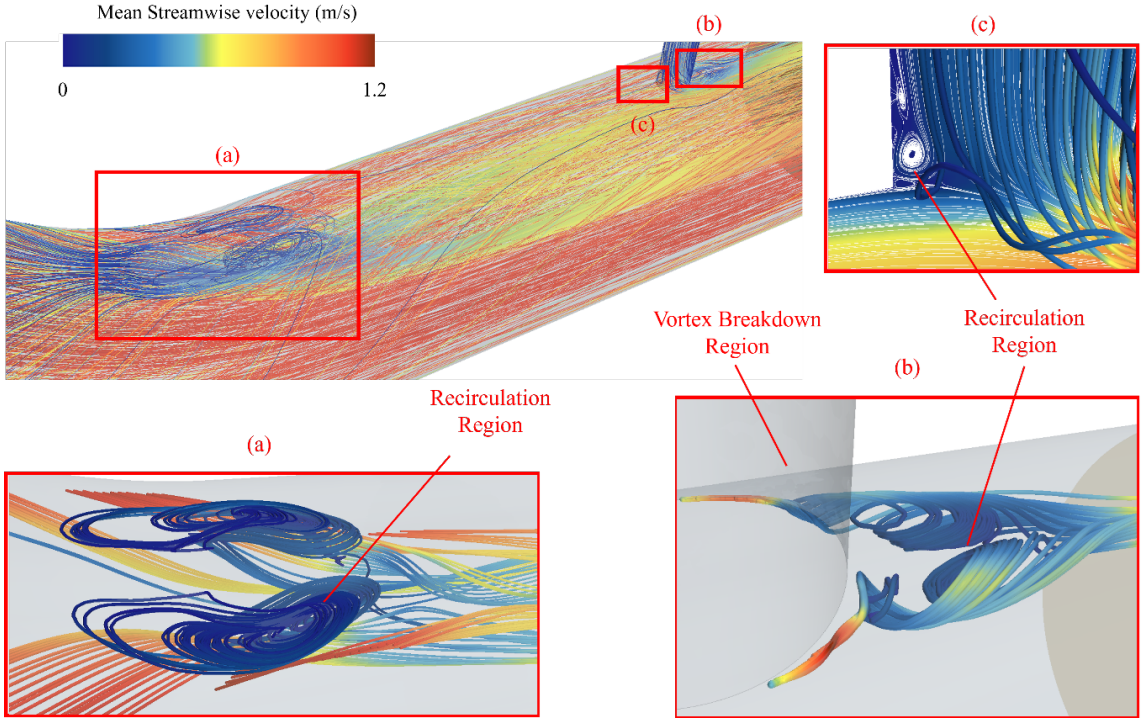


Figure 47: Mean flow structures at (a) a horseshoe-like recirculation at the exit of the elbow, (b) the recirculation region after the T-junction, (c) the recirculation region before the T-junction.

Figure 48 visualises the instantaneous flow field as well as the mean flow field on the X-Z symmetric plane and Y-Z cross-section located at $x/D_m = -1.36, 0, 0.292$. The instantaneous flow domain in the left column shows the chaotic nature in all cases, especially in the recirculating regions downstream of the elbow and the T-junction exit due to the flow separation and reattachment. After the fluid passes through the elbow, the instantaneous velocity field displays significant spatial variation. This indicates the presence of vortices, eddies, and other coherent structures that are constantly forming and dissipating, leading to a continuously changing flow field. The smaller cross-sectional views further highlight these fluctuations. It can be seen in that instantaneous flow at the cross-sections, that the counter rotating Dean-vortices originating from the bend are highly unstable and present the swirling features downstream of the bended pipe. It also exhibits larger perturbations from the branch pipe flow as the branch pipe diameter increases, where a larger branch pipe diameter leads to a higher Reynolds number of the inlet branch flow in this study.

On the other hand, the mean flow field shown on the right column in Figure 49 clearly illustrates the centrifugal effect of the bent pipe, where the mean velocity is deflected toward the outer side of the bend, forming a flow separation region near the bend exit. Subsequently, the flow velocity in the lower half of the pipe downstream of the bend exit is much higher than that in the upper half, while the counter-rotating Dean vortices also originate from the bend exit and propagate downstream. As fluid is injected into the main pipe from the upper T-junction, it generates a region of flow separation (also called a separation bubble). In the region downstream of the T-junction, there is a similar behaviour seen, but at a much smaller scale. The size of this region changes with the diameter of the branch pipe. Since the direction of the branch pipe flow here is perpendicular to the main pipe flow, it is akin to a jet in crossflow. It is noted that the jet from the branch pipe also creates multiple pairs of counter-rotating mean flow structures in the upper region, but these quickly dissipates. This corresponds to the findings in the previous chapter, chapter 5.

It is worth noting that because the branch pipe diameter is much smaller than the main pipe, the interaction between the fluid from the branch pipe and main pipe is akin to the jet-in the cross-flow near the exit of the T-junction. This is evidenced in Figure 48 where flow structures such as wall-jet and impinging jet are observed. For cases with a larger inlet branch pipe diameter, the flow from the main pipe interacts more strongly with the inlet flow from the

branch pipe. This is more evident from the instantaneous contour plots in the left column of Figure 48. On the right column of Figure 48, the mean streamwise velocity contour plots show the velocity field averaged over time and note the size of the recirculation region being dependent on the inlet branch pipe's diameter. Within Figure 49, Dean vortices are visualised before and after the elbow section. It is observed that the gradual evolution of the Dean vortices where they are plotted in line with the mean velocity magnitude at each cross section. These complex flow structures are originated from the region right before the elbow due to the curvature and centrifugal force in the bend. These counter rotating pair of vortices converge in the centre to form a kidney shaped pattern in the recirculation region and it is also seen in the region further downstream of the T-junction exit.

In Figure 50, a Q-criterion of $150s^{-2}$ is seen for each case where the general underlying flow structures can be found. Within these contour plots, it can be seen that as the flow progresses from Case 1 to Case 3, the flow downstream of the T-junction becomes more turbulent as the branch pipe diameters increase. In the first three cases, the vortices appear more coherent and compact as the inlet branch diameter increases while the flow structures seen in Case 4 and Case 5 seem to be as turbulent as Case 1, but not as chaotic as Case 3. This is seen from a more detailed close up in Figure 50(b), where intensified and more chaotic swirls are observed downstream of the T-junction in Case 3. It is here where the vortices become more elongated and dispersed, suggesting increased interaction and the onset of vortex breakdown. While this behaviour has not been explicitly observed in the previous work, note that the turbulent kinetic energy for Case 3 is generally the higher in the upper half of the flow in the downstream region.

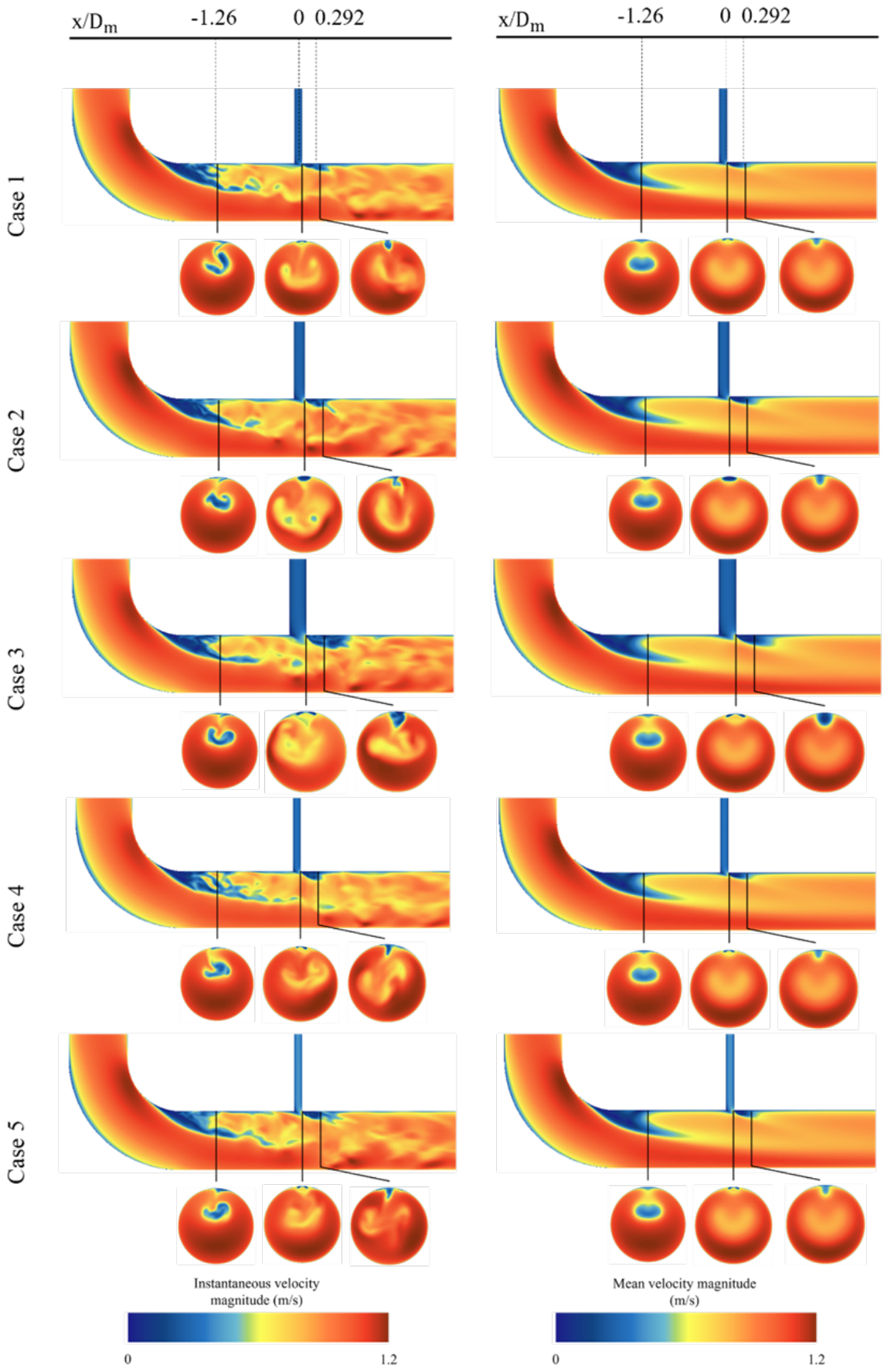


Figure 48: Instantaneous (left) and mean (right) flow field on the X-Z symmetric Plane (Side View) and Y-Z plane (cross-sectional view).

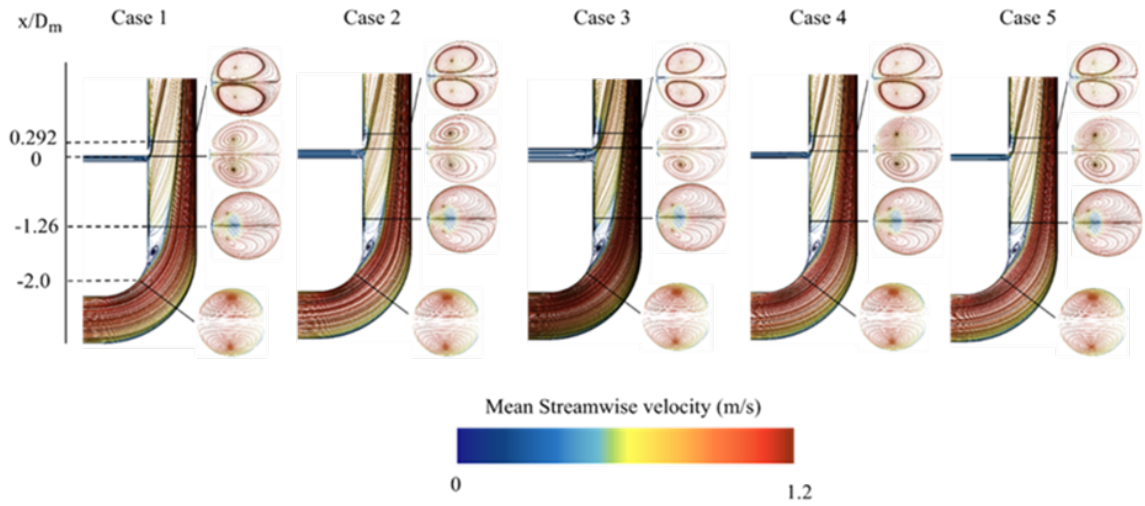


Figure 49: Evolution of Dean vortices based on velocity streamlines and mean velocity

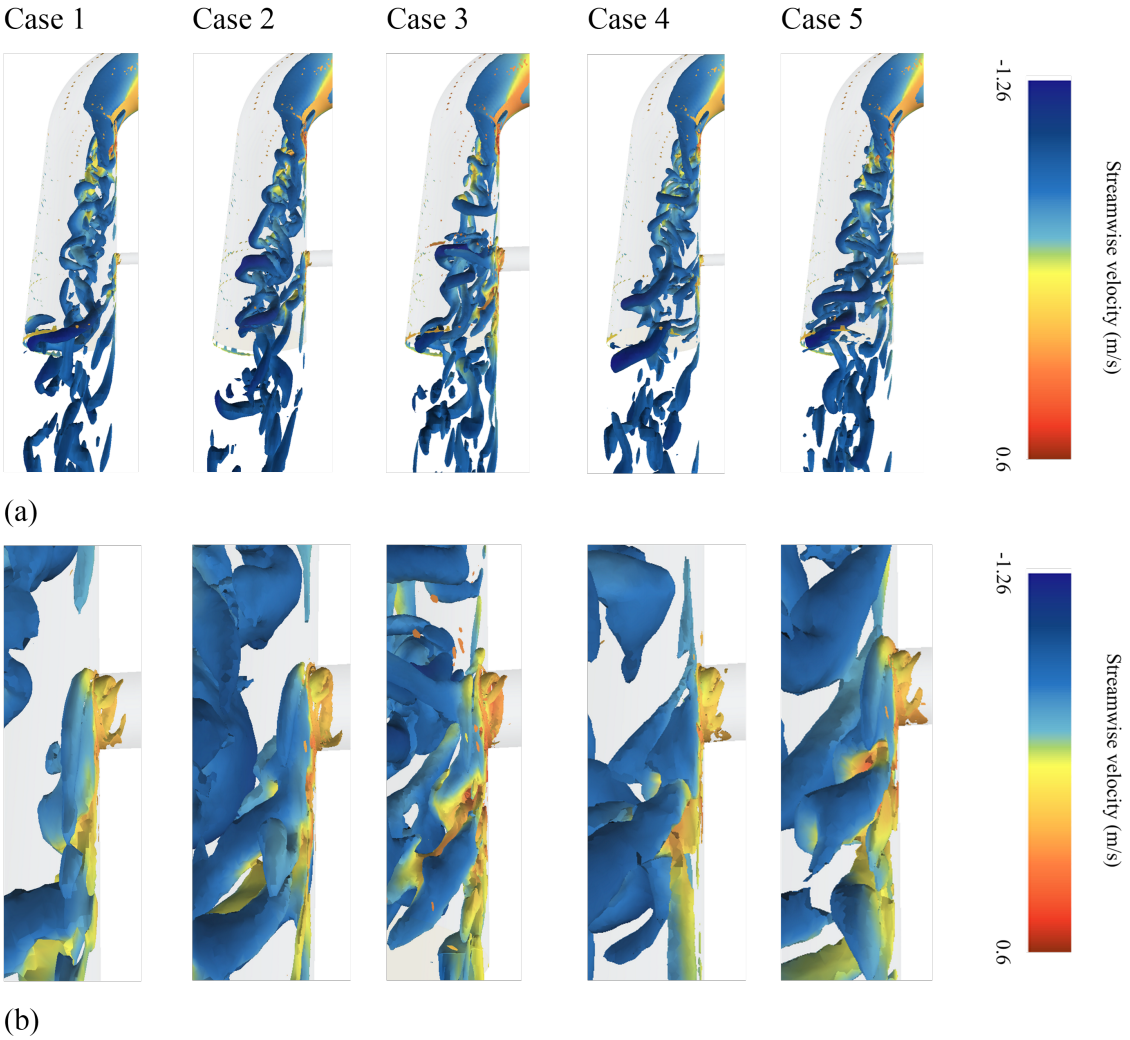


Figure 50: (a) Contours of $Q-2$ Criterion Value of $150s^{-2}$ Coloured with Streamwise Component of Velocity (b) Zoomed-in view at the T-junction exit for each case

6.3.2 POD analysis

In the POD analysis, the POD modes of the flow along two major planes, are shown from the side (X-Z plane) and the flow seen from the frontal position (Y-Z plane). For the reader's reference, please refer to Figure 12 for the reference plane. Along the X-Z plane, snapshots of the fluctuating velocities $V_x'$and V_z' are collected, while on Y-Z planes, snapshots of the fluctuating velocities V_y' and V_z' are recorded.

6.3.2.1 POD Y-Z in-plane analysis

The record of the snapshots commenced where it achieves time-convergence at around 18 s, namely 15 turnover time. Turnover time is the time it takes for a volume of fluid equivalent to that of the downstream region (after the T-junction exit) to make one pass through the system. The fluctuating velocities on the Y-Z plane are sampled every 0.01 s for 40 s, after the flow becomes quasi-stable. This gives us approximately 4000 snapshots in time. The locations of interests are similar to the ones detailed in Figure 35.

The power ratio of POD modes for each Y-Z plane are summarised in Figure 51, where the energetic content is plotted as a function of modes. The energetic content used here refers to the amount of kinetic energy captured by each mode in the decomposition. It indicates how much of the total energy of the flow is represented by that mode. Note that the percentage of energy that each individual mode has, is calculated as a fraction of the mode's energy with respect to the sum of the energy across all modes. This will be further illustrated in the results and analysis section. In a preliminary study, the plots generated forV_y' and V_z' are identical. Hence, the V_z' has been considered for this spectral analysis. In this diagram, the first 20 POD modes and their respective energy content are displayed, and the dataset includes the abovementioned cross sections across the three cases. It is evident that the first two modes accounts for at least 40-70% of the total accumulated energy and these two modes are considered for analysing underlying flow structures. There are expected flow features such as single or counter-rotating vortices. For the similar geometry, such flow features have been explored and reported in a couple of recent studies, where certain similarities are seen in the flow structures generated with POD (Joksimovic et al. (2023)).

Figure 52 presents a series of subplots that depict the PSD estimates obtained using Welch's

method for different cases and spatial locations. Each subplot illustrates the PSD of the first five time-coefficients derived from a POD analysis of V_z' for simplicity. Each subplot shows the first five time-coefficients from the POD analysis. These coefficients represent the most significant time patterns in the velocity field, with the first typically being the strongest.

Across all cases and locations, there appears to be a peak or notable feature near the 5.36 Hz mark, especially in the first time-coefficient (blue line). This suggests that this frequency corresponds to a dominant flow structure or instability that contributes significantly to the energy content of the system. The prominence of this peak in the first time-coefficient highlights its role in capturing the most energetically significant mode of the flow. The first time coefficient consistently shows the highest PSD values across all cases and spatial locations. This indicates that it captures the bulk of the energy in the flow, particularly at the lower frequencies. The dominance of this coefficient aligns with the fact that it represents the primary mode of the velocity field, which typically carries the largest energy content. Notably, the blue line exhibits the most pronounced peak around 5.36 Hz, reinforcing the importance of this frequency.

The remaining time coefficients (green, red, orange, and purple lines) generally display lower PSD magnitudes, indicating their progressively smaller contributions to the flow dynamics. Although they also exhibit peaks around the 5.36 Hz frequency, their influence is less pronounced than the first time-coefficient. This reflects the fact that these modes represent secondary flow structures or instabilities that contribute less energy. Comparing across the five cases, the general pattern of decreasing PSD with increasing frequency remains consistent. However, some cases exhibit more pronounced differences between the coefficients, indicating that the specific flow conditions such as varying momentum ratios or Reynolds numbers, influence the distribution of energy across the modes. For example, in Case 5 at $x/D_m = 0$, the spread between the first and subsequent coefficients is more noticeable than in other cases, suggesting a more dominant primary mode in that particular flow condition.

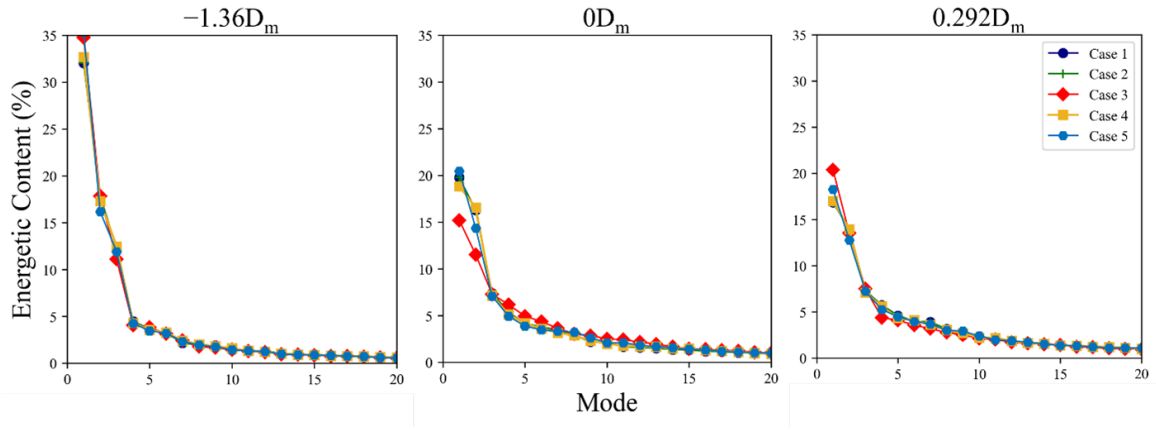


Figure 51: POD energy spectra based on V_z' at various cross-sectional locations; Vertical axis represents the energetic content in terms of percentage, horizontal axis represents the mode number.

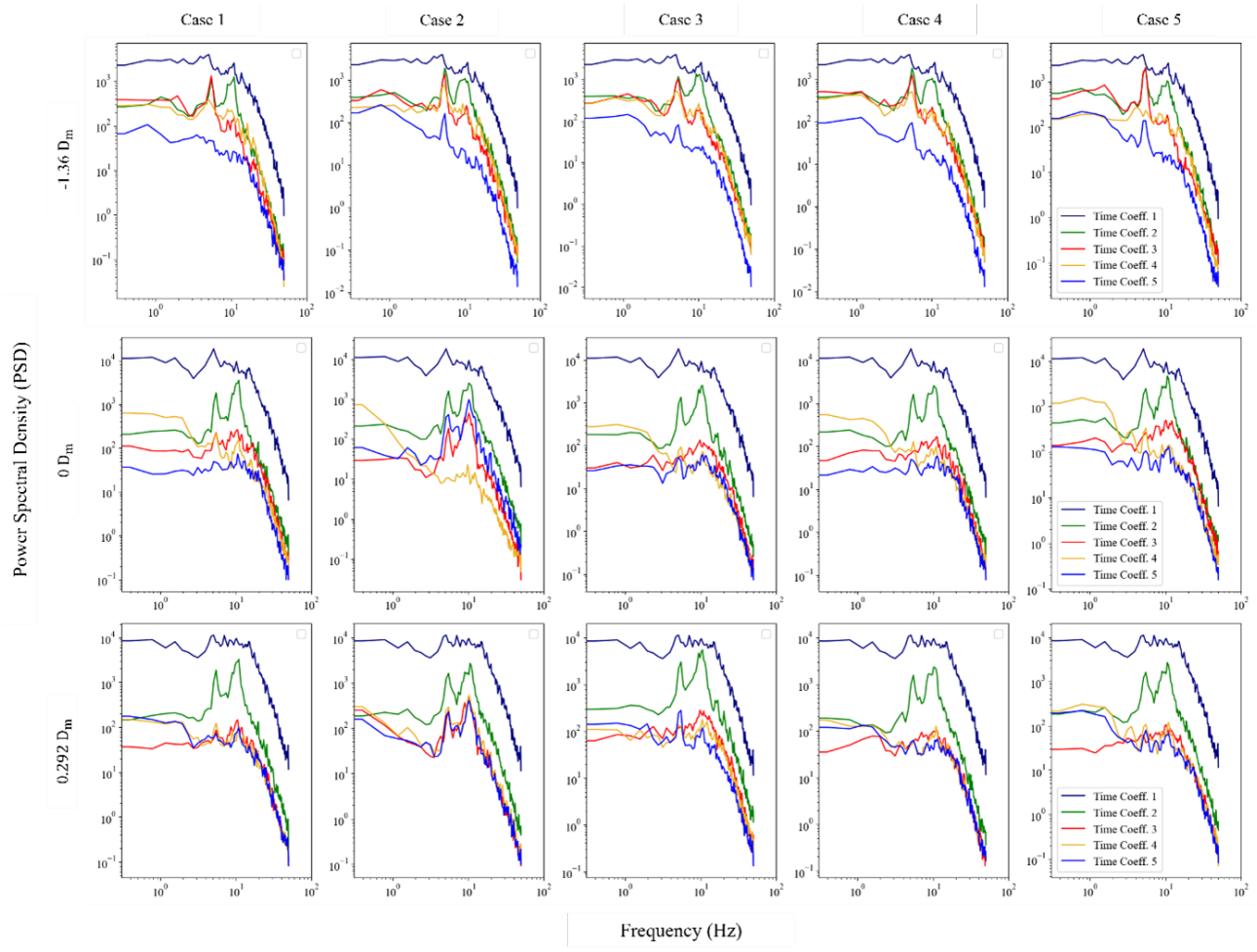


Figure 52: Welch's estimate of the power spectral density of the first 5 fluctuating modes of V_z at each location.

The first POD mode at each location is shown in Figure 53. At $x/D_m = -1.36$, a pair of

alternating positive and negative regions is seen to dominate the upper half of the flow, and this pattern is found across all models with little to no differences. This represents oscillatory behaviour in the flow originating from the upstream bend. The observed pattern corresponds to a large singular vortex observed in a study where a POD analysis of the fluctuating component of velocity was carried out (Joksimovic, Corbo, Hussong, & Jakirlic, 2023). In a previous work, this location is seen to exist within a recirculating region in wake of the upstream bend and corresponds to the peak frequency of 5.36 Hz. This frequency may potentially be the base for thermal fatigue at this location.

Downstream of the junction's exit at $x/D_m = 0$, a pair of alternating positive and negative regions is seen to dominate the centre of the flow, and this pattern is found across all models with some differences. This represents oscillatory behaviour in the flow originating from the T-junction exit. It is noted that the alternating regions of positive and negative energy is very similar across all cases and this phenomenon corresponds to the streamlines seen in the previous work. At this location, these oscillatory structures are linked to a pair of large counter rotating vortices seen in the upper half, closer to the centre of the flow. The counter rotating pair of vortices possibly corresponds to a frequency of 5.36 Hz observed in the previous study. There are also several other smaller peaks observed which reflects the quasi-periodic nature of the unsteadiness and this could account for the smaller vortices which are seen to be more spread out in the other cases. Further downstream of the T-junction at $x/D_m = 0.292$, it is observed a pair of alternating positive and negative regions that dominates both upper and lower regions of the flow. Some differences exist where the alternating regions in Case 3 is seen to be more energetic compared to the other cases. These alternating regions of energy corresponds to the vortex pair seen dominating the entire flow and this is located within the recirculating region of fluid injected by the branch pipe. It corresponds to the peak frequency of 5.36 Hz seen in the previous study, which may potentially be the base for thermal fatigue at this location.

While the 2nd POD mode is not the most energetic, it potentially offers some useful insights on the flow structures seen in this mode. This is presented in Figure 54 in a similar fashion to the previous figure. The main pattern seen at $x/D_m = -1.36$ is a trio of alternating regions (positive-negative-positive) in the upper half of the flow and this is the same across all cases. This is different from observations made from Mode 1. A similar structure has been reported in a recent study which also may also corresponds to the peak frequency of 5.36 Hz (Joksimovic

et al. (2023)). At $x/D_m = 0, 0.292$, the patterns found in both locations for each case can be generally described as a set of clover-shaped alternating regions where it consists of lobes that alternate between positive and negative values This pattern suggests a rotational symmetry and is often associated with oscillatory phenomena in the flow, such as vortex shedding, swirling flows, or other periodic structures. In Case 3, the alternating regions of positive and negative energy has slightly more energetic regions, followed by Cases 2 and 5 and Cases 1 and 4, where both case-pairs have similar intensities in the patterns observed.

The observations at the locations downstream of the T-junction provide compelling evidence that the influence of Reynolds number of the inlet branch flow outweighs that of the momentum ratio. This assertion is supported by the similarity between Case 2 and Case 5, which share identical Reynolds number in the inlet branch flow yet exhibit markedly different momentum ratios. Despite this discrepancy, their underlying flow patterns remain highly comparable, indicating that variations in the momentum ratio do not significantly alter coherent structures downstream of the T-junction. Conversely, cases with comparable Reynolds numbers demonstrate consistent underlying flow structures across all locations. The vortices observed at $x/D_m = 0.292$ rotate in a clockwise direction. This interesting observation can be attributed to a transition from counter-rotating vortices on the left and right side of the flow to counter-rotating vortices in the upper and lower half of the flow, happening at this location.

Whenever dean vortices change their size, intensity and centre-axis in a time-oscillatory manner, which is also known as swirl-switching and is one reason for mechanical pipe fatigue (Kalpakli (2014); Noorani and Schlatter (2016)). A partial reconstruction of the flow field from POD with the mean flow, the first and second mode, based on V_z', reveals certain swirl switching behaviour at the location after the bend, before the T-junction at $x/D_m = -1.36$. This is further illustrated in Figure 55 below. It is shown in this figure that the intensity and the centre-axis shift from the mean mode to mode 2. The size of these dean vortices also shrinks from the mean mode to mode 2. This is one of the various ways of depicting the swirl switching phenomena.

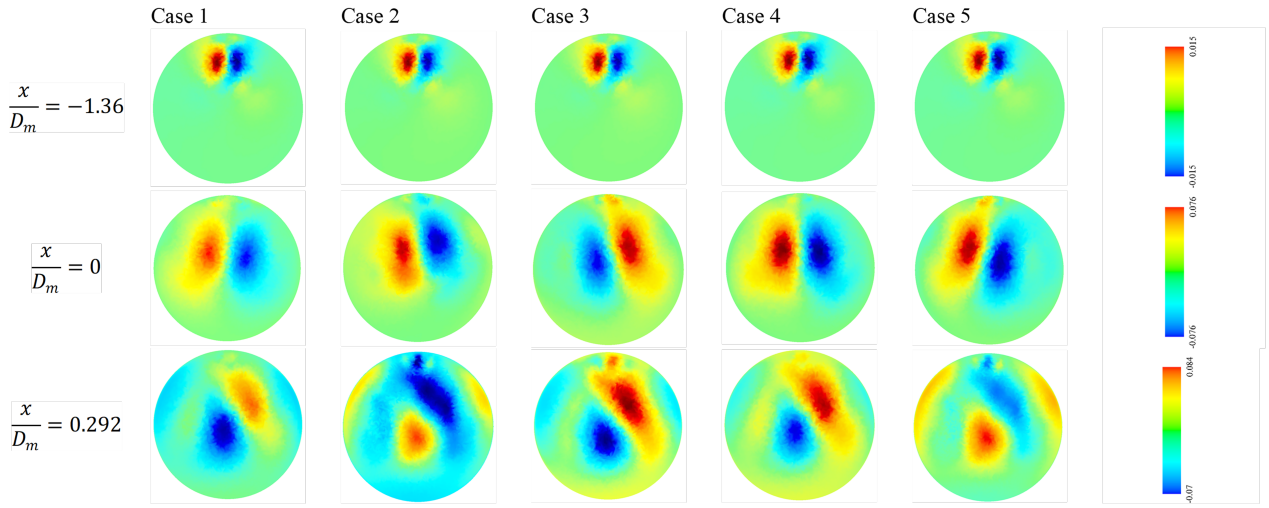


Figure 53: POD Mode 1 of Y-Z in-plane velocity components V_z' for $x/D_m = -1.36, 0, 0.292$; the contours represent the amplitude of mode 1 of V_z.

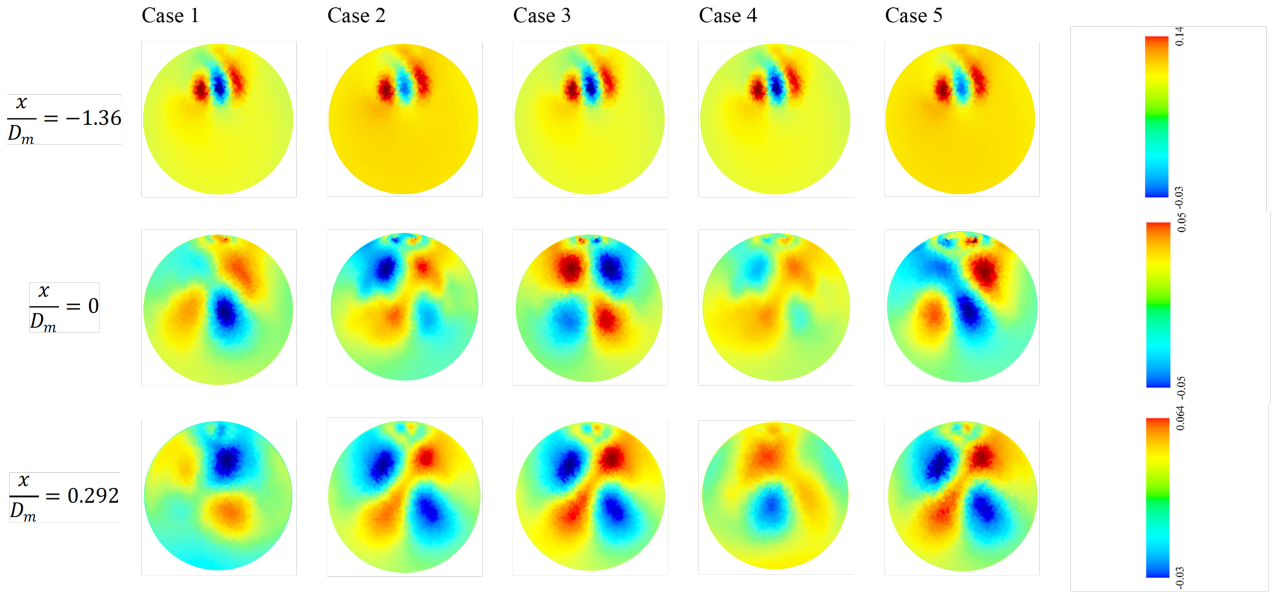


Figure 54: POD Mode 2 of Y-Z in-plane velocity components V_z for $x/D_m = -1.36, 0, 0.292$; the contours represent the amplitude of mode 1 of V_z

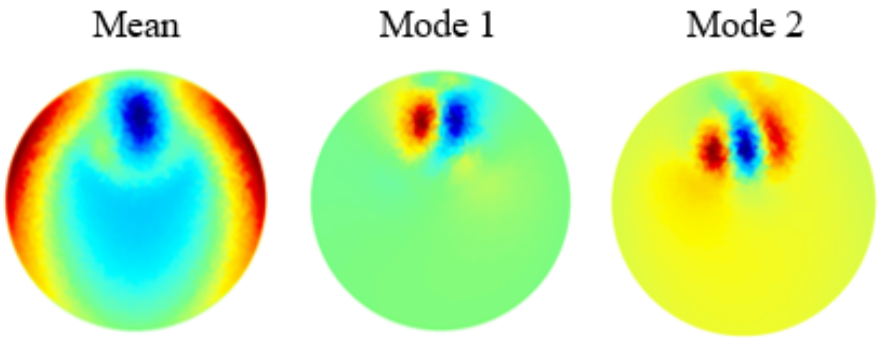


Figure 55: Three instantaneous views at different phase of the cross-sectional velocity after the bend from the recomposed velocity field.

6.3.2.2 POD X-Z in-plane analysis

In a preliminary study, it is found that plots for energy content against POD modes for V_x' and V_z' are also identical. Hence, for the sake of simplicity, only the POD analysis of V_x' is discussed in this section. In a similar fashion to the previous section, the energetic content (%) of POD modes on this planar view are summarised in Figure 56, where eigenvalues are plotted as a function of energetic content. In this diagram, the first five POD modes with the respective energy percentage content are displayed. An immediate observation is that the energetic content in terms of percentage starts to plateau after mode 4, which suggests that the first two modes accounts for majority of the total accumulated energy ($\sim 60\%$) and these modes can primarily be considered for analysing underlying flow structures.

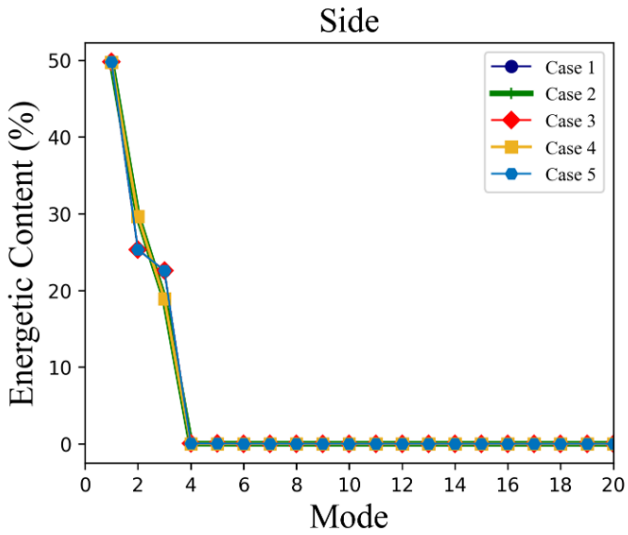


Figure 56: POD energy spectra along the X-Z plane based on V_x'; Vertical axis represents the energetic content in terms of percentage, horizontal axis represents the mode number.

The first two POD modes (Mode 1 and Mode 2) based on the magnitude are plotted in Figure 57 and Figure 58. At the exit of the T-junction and the elbow bend, there are several vortical structures that originate from that location. These structures manifest along the shear-layer emanating from the upstream elbow, a phenomenon consistent across all cases examined. In Figure 57 there are differences observed among cases in terms of the size of structures at the exit of the elbow (in the upstream section) and at the exit of the T-junction (in the downstream section). Vortex shedding from the shear layer at the upstream section is observed in each case. These structures manifest within the shear-layer jet's primary flow, originating from the upstream elbow. It is worth noting that this geometry induces secondary flow that originates at the upstream bend which has been reported in this study. It is also noted that the injection of the branch inlet impacts the flow behaviour in the downstream section (Tunstall et al. (2016)). It seems that the values of Re_b do not affect these structures upstream of the T-junction, but instead, they only very slightly affect the flow patterns downstream of the T-junction (at the exit). These can be discerned from the pockets of energy (in light green) see at the T-junction exits. This behaviour seems to be correlated to the inlet branch diameter where it gets more pronounced in Figure 58 where the second mode shows a larger region with respect to Cases 2 and 3 compared to Cases 1, 4 and 5. It is seen from these figures where the effects of the strength of secondary flow dominating the flow downstream on the underlying flow structures produced. With the above mentioned explanation, higher Re_b values lead to weakened secondary flow in the downstream region, producing a highly energetic region at the T-junction exit as the main underlying flow structure. On the other hand, cases with a much stronger secondary flow that dominates the main flow, there are more significant shredded vortical structures from the shear layer jet that exists as the main underlying flow structure.

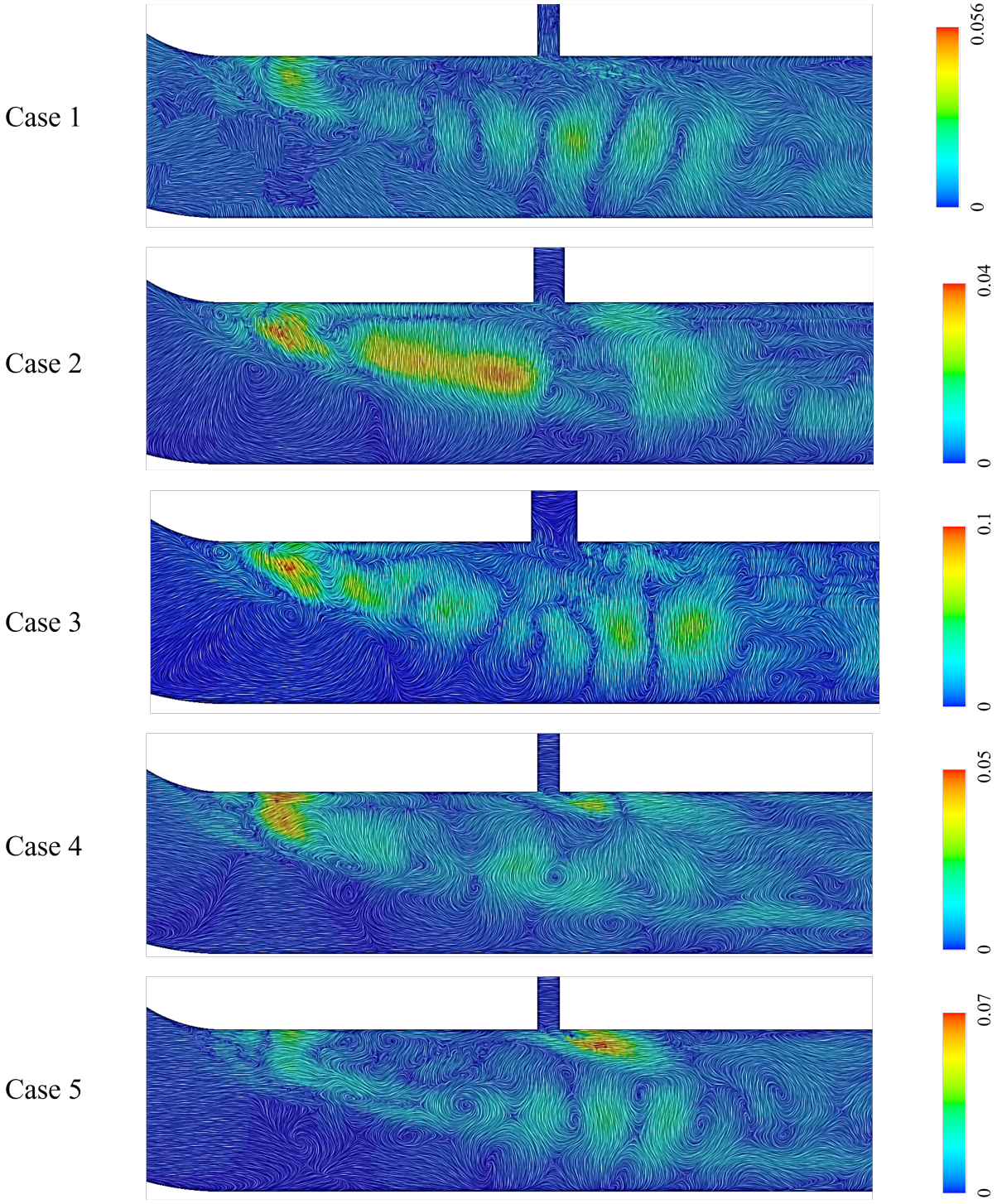


Figure 57: POD Mode 1 of X-Z in-plane velocity components of V_x and V_z; the contours represent the amplitude of mode 1 of V_z while the streamlines represent vectors consisting of V_x and V_z

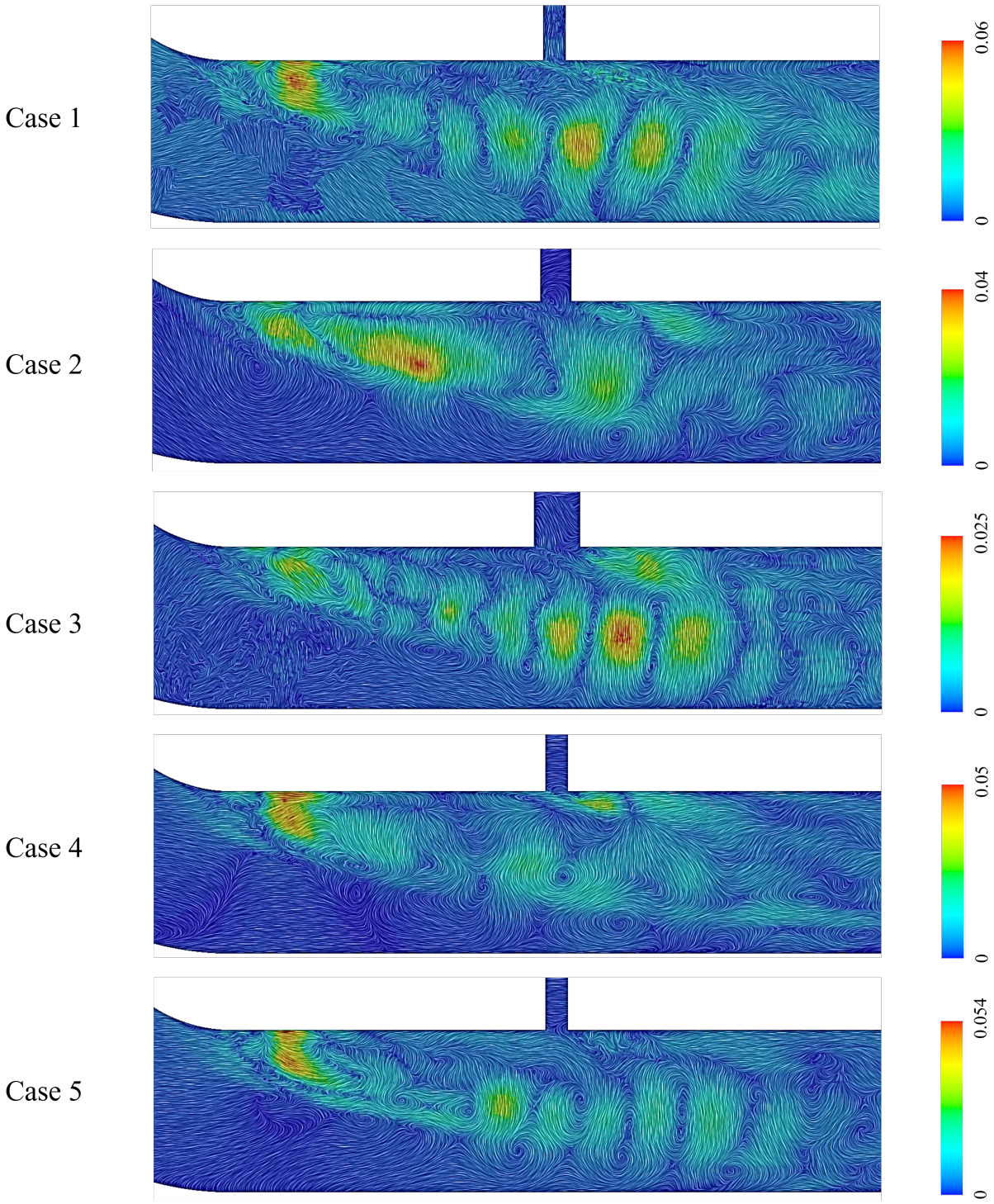


Figure 58: POD Mode 2 of X-Z in-plane velocity components of V_x and V_z; the contours represent the amplitude of mode 1 of V_z while the streamlines represent vectors consisting of V_x and V_z

6.3.3 SPOD analysis

In the SPOD analysis, the SPOD modes of the flow along two major planes in the flow, mainly the flow seen from the side (X-Z plane) and the flow seen from the frontal position (Y-Z plane). Along the X-Z plane, snapshots of the fluctuating component of fluctuating velocity are collected in their vertical and horizontal components, V_x', V_z', while along the Y-Z plane, snapshots of the fluctuating components of fluctuating velocity in their vertical and horizontal components, V_y', V_z'', which are perpendicular to the streamwise velocity. Sampling methodology used in this section is the same as subsection 6.3.2 to which the reader is advised to refer to.

6.3.3.1 SPOD Y-Z in-plane analysis

Similar to the previous section, SPOD will be applied on the turbulent flow along the two major planes: the flow seen from the side (X-Z plane) and the flow seen from the frontal position (Y-Z plane). The first four SPOD modes for each location are summarized in Figure 59 where eigenvalues are plotted as a function of frequency. It is noted in a preliminary analysis, the plots generated for V_y' and V_z' are identical. Hence for simplicity, only the plots generated for V_z' are included. In this diagram, the full spectrum of SPOD eigenvalues is displayed and the dataset includes the above mentioned cross sections across the three cases. In this case, a large separation between the eigenvalues can be discerned in the region around $4.0 \leq$ Hz ≤ 7.0. No peaks are observed for the regions spanning the top and bottom of the flow and it seems to be consistent across all locations. The leading mode is evidently more energetic than the suboptimal modes and captures more than 50% of the flow energy across all frequencies. The low ranked behaviour observed at 5.36 Hz which is indicated by a large gap between the first and second SPOD eigenvalues, persists in the range of $4.0 \leq$ Hz ≤ 7.0.

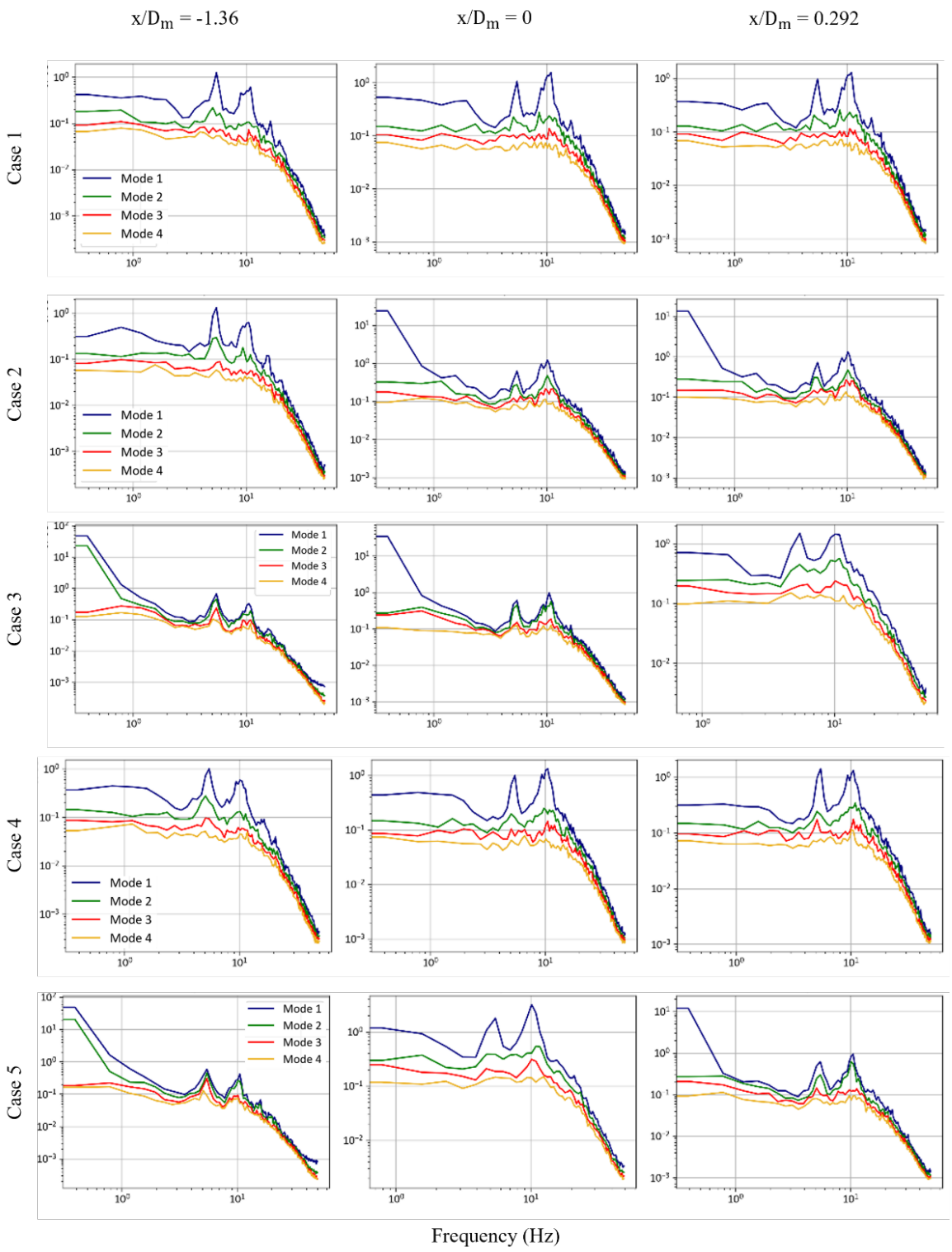


Figure 59: SPOD energy spectra for the first four modes based on V_z', at various cross-sectional locations

The first mode associated with the principal peak frequency (5.36 Hz) at each spatial location is depicted in Figure 60, where the energy spectra of V_y' & V_z' are visually represented and rendered as a colour-mapped contour plot. The wavelength and spatial extent of these structures exhibit

a strong dependency on frequency, with both quantities inversely proportional to the frequency. In this analysis, SPOD is only applied on V_z' since it is found that only the V_z' component of velocity contains dominant flow patterns.

In the upstream region of the elbow at $x/D_m = -1.36$, a pair of alternating positive and negative regions is seen to dominate the upper half of the flow, and this pattern is similar across all cases. There are no differences observed in both the V_y' and V_z' components respectively. This represents oscillatory behaviour in the flow originating from the upstream bend, along both vertical and horizontal directions. Note that this pattern is not seen with POD, which may suggest that SPOD provides a more detailed look into the underlying flow patterns. As reported in the previous section, there is a large vortex seen in all cases which represents the dominant flow structure at this location. This is due to the separated flow in the upper wall and lower wall, where the latter experiences acceleration as reported in figures found in Wong et al and Tunstall et al's study (Tunstall et al. (2016); Wong et al. (2024b)). This underlying flow structure corresponds to the peak frequency of 5.36 Hz, seen in Figure 59. As mentioned previously, this underlying flow structure can be a catalyst for potential failure in the future.

Downstream of the junction's exit at $x/D_m = 0$, a pair of alternating positive and negative regions dominates the centre of the flow, which is a pattern consistently observed across all models with some variations. This indicates oscillatory behaviour originating from the T-junction exit. The similarity in these alternating regions of positive and negative energy across all cases corresponds to the streamlines seen in the previous work. At this location, these oscillatory structures are associated with a pair of large counter-rotating vortices in the upper half, near the centre of the flow. It is worth noting that this pattern is also observed at the same location when POD was used for analysis. The counter rotating pair of vortices possibly corresponds to the peak seen in Figure 59 across all cases at a frequency of 5.36 Hz, highlighting a potential cause of failure at this location due to the observed underlying flow structures.

Further downstream of the T-junction at $x/D_m = 0.292$, The patterns observed in each case can generally be described as clover-shaped regions with alternating lobes of positive and negative energy. This pattern, indicating rotational symmetry, is often linked to oscillatory phenomena in the flow, such as vortex shedding, swirling flows, or other periodic structures. This has been highlighted in the POD analysis but is seen to exist at $x/D_m = 0, 0.292$. It is found that for Case 3, the alternating regions of positive and negative energy has slightly more energetic

regions, followed by Cases 2 and 5 and Cases 1 and 4, where both case-pairs have similar intensities in the patterns observed.

In a similar manner, the first SPOD mode associated with the other frequency that is thought to be its harmonic frequency at 10.71 Hz is plotted in Figure 61. It is found that only the V_k component of velocity contains dominant flow patterns which is also seen in the analysis above. Hence, the plots seen in this figure only involves the V_z component of velocity.

Harmonic frequencies are integer multiples of the fundamental frequency of any system, in this case is the turbulent flow within the T-junction. Since they represent resonance conditions, it allows us to know more about the oscillations that happen at this specific frequency which can potentially account for failure in the T-junction. The wavelength and spatial extent of these structures exhibit a strong dependency on frequency, with both quantities inversely proportional to the frequency. The value of 10.71 Hz is found to be coherent with the dominant frequency (at 5.36 Hz), by a factor of 2. The first mode associated with 10.71 Hz at $x/D_m = 0$ is represented in this figure where the energy spectra of V_y',V_z' are visually represented and rendered as a colour-mapped contour plot. For the SPOD analysis on V_k' , there are very similar structures seen at this harmonic frequency, where a pair of alternating positive and negative regions is seen to dominate the upper half of the flow, and this pattern is similar across all cases. This suggests that the structures are repeated at this frequency, further supporting the postulation that this is indeed the true harmonic frequency of the system. The only difference that exists is the V_y' component where the transition from red to blue and back to red energetic regions is spread out over a larger region in the lower half of the flow. This implies an oscillatory or periodic structure in the flow, which is a characteristic of phenomena such as vortex shedding or swirling flows. The somewhat circular distribution of these alternating regions suggests a degree of rotational symmetry in the flow, often associated with coherent structures like vortices which has been mentioned and reported in Tunstall et al's and Joksimovic et al's works (Tunstall et al. (2016); Joksimovic et al. (2023)).

Downstream of the junction's exit at $x/D_m = 0, 0.292$, a pair of alternating positive and negative regions dominates the center of the flow, a pattern consistently observed across all models with some variations. This indicates oscillatory behaviour originating from the T-junction exit. The similarity in these alternating regions of positive and negative energy across all cases corresponds to the streamlines seen in the previous work. At this location, these

oscillatory structures are associated with a pair of large counter-rotating vortices in the upper half, near the centre of the flow. Notably, this pattern is also observed at the same location when POD was used for analysis and is almost identical to the ones seen at the dominant frequency with very insignificant differences, if any.

In summary, the results seen in the previous two figures do provide somewhat substantial evidence that the influence of Reynolds number of the inlet branch flow within such flows outweighs that of the momentum ratio. In this case study, this is directly correlated to changing the inlet branch diameter or the inlet branch velocity. Cases with identical Reynolds numbers, such as Case 2 and Case 5, exhibit strikingly similar flow structures despite significant differences in momentum ratios. Conversely, cases with comparable Reynolds numbers demonstrate consistent flow structures across all locations. To corroborate these findings, X-Z in-plane analysis is employed to garner additional insights into coherent flow structures.

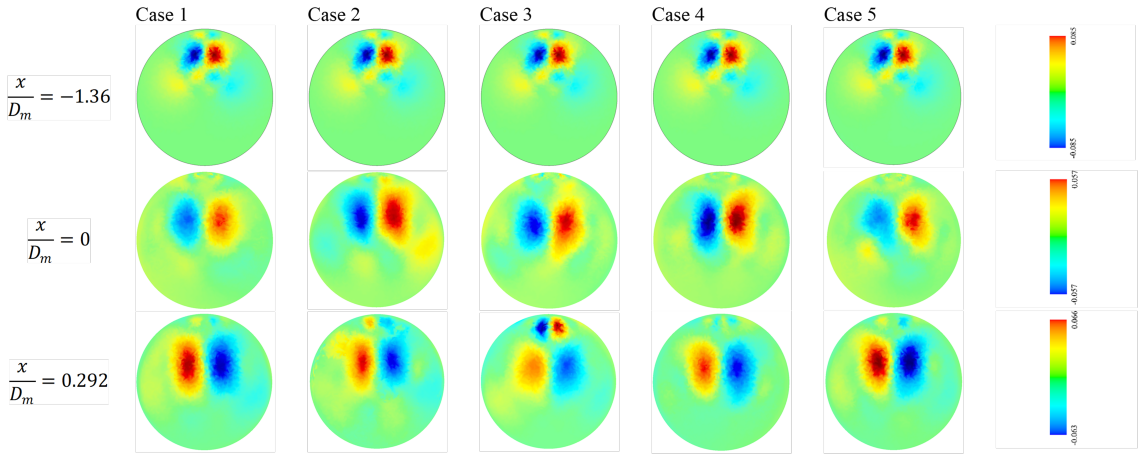


Figure 60: SPOD Mode 1 of Y-Z in-plane velocity components V_z for $x/D_m = -1.36, 0, 0.292$ at dominant frequency of 5.36 Hz

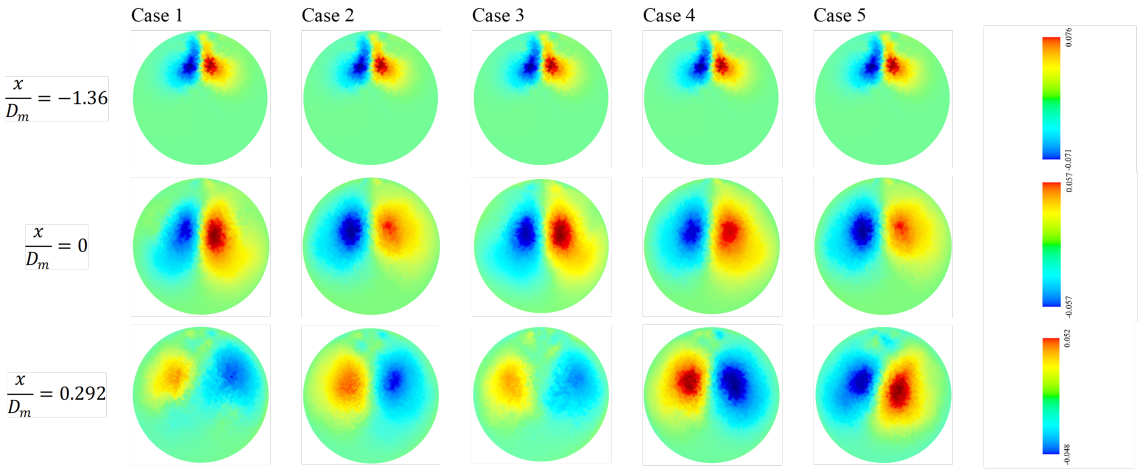


Figure 61: SPOD Mode 1 of Y-Z in-plane velocity components V_z for $x/D_m = -1.36, 0, 0.292$ at dominant frequency of 10.71 Hz

6.3.3.2 SPOD X-Z in-plane analysis

Similar to the analysis above, snapshots of the instantaneous velocity along the X-Z plane were collected and analysed using SPOD. It was found that the number of snapshots collected did not exhibit the same effects seen in the Y-Z plane analysis. For each of the cross-sectional planes, approximately 5000 snapshots are recorded every 0.01s spanning a total of 50s. The first SPOD mode based on the magnitude of the X-Z component contains the most energetic spectra and this is plotted in Figure 62. It is found that the dominant frequencies for all cases are within the range of approximately 5-10 Hz which still agrees with (Chapuliot et al. (2005)). Furthermore, the flow structures at the oscillatory frequency (10.71 Hz) are very similar to the flow structures seen at 5.36 Hz. Therefore, the analysis in this section will only consider the dominant frequency at 5.36 Hz.

Generally, each plot in the figure below reveals several vortical structures that are observed to repeat periodically. This happens at the shear-layer jet main flow that originates from the upstream elbow. This behaviour is consistent across all the different cases. There is a handful of smaller vortices which are seen to shed from the upstream edge of the jet entrance. These shedding phenomena follow the curved path of the jet in the region before the T-junction. The size of these shredded vortices grow as the flow passes the T-junction.

Additionally, downstream of the T-junction at the T-junction exit within the jet-in flow, there

is a hint of vortex separation in the underlying flow structure and this effect is more pronounced in Case 2 and Case 3 and 5, where more energetic and/or shredded vortical structures are found that originates from the T-junction exit. In the previous section for the POD analysis, a similar behaviour is observed which is possibly due to the dependence on Re_b values of the inlet branch flow. This is coupled with the weakened impact of the secondary flow that originates from the upstream elbow, due to the injection of the inlet branch flow. In summary, a higher Reynolds number of the inlet branch flow impacts the overall underlying flow structure due to its impact on the secondary flow from the upstream elbow.

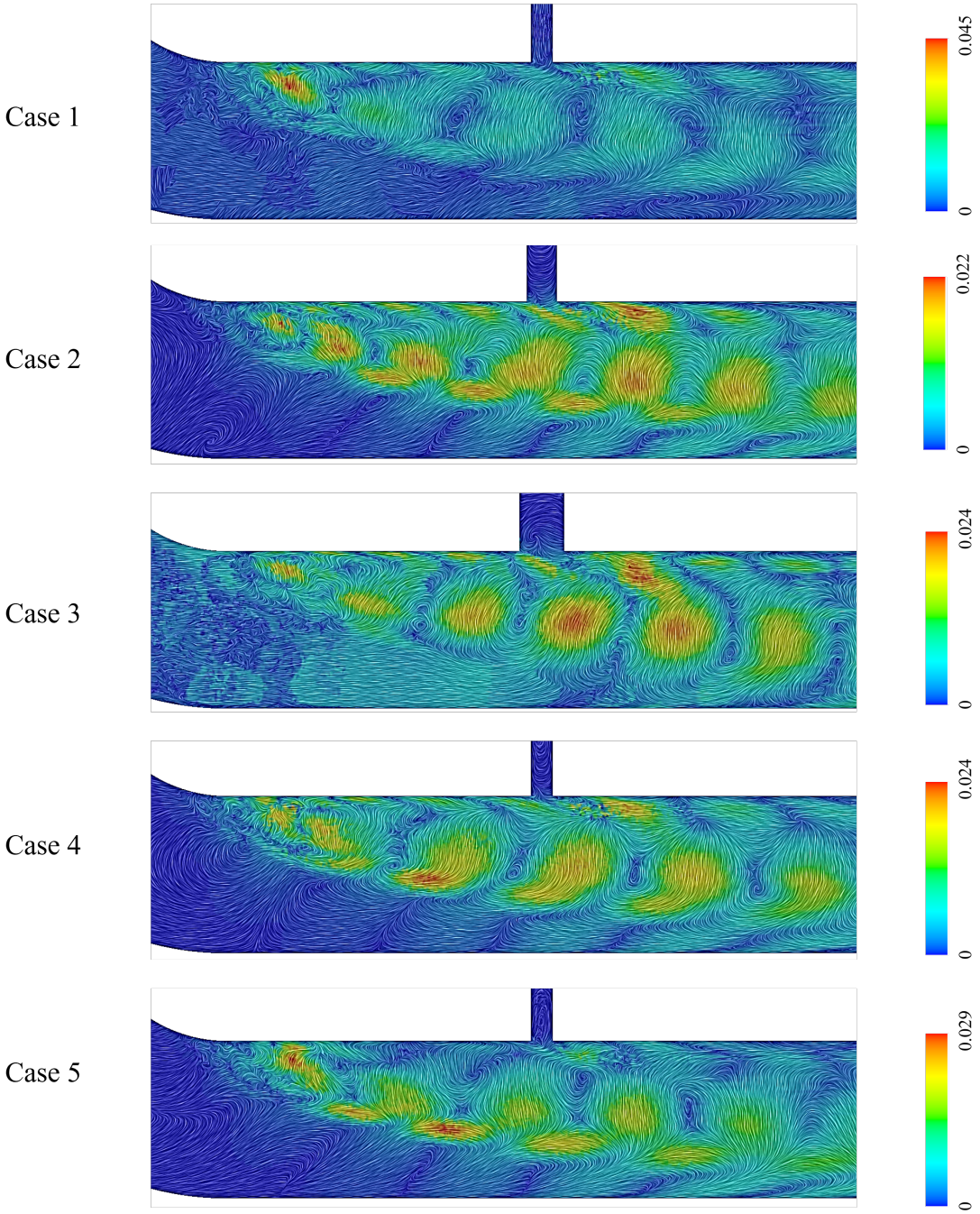


Figure 62: SPOD Mode 1 of X-Z in-plane velocity components of V_x and V_z at 5.36 Hz; the contours represent the amplitude of mode 1 of V_z while the streamlines represent vectors consisting of V_x and V_z

6.4 Summary and Conclusions

This study examined the underlying dominant flow structure across various T-junction flow test cases with differing momentum ratios and Reynolds numbers. The momentum ratio of the T-

junction was varied by either adjusting the diameter of the branch pipe or altering the velocity of the branch pipe flow. Turbulent flow was modelled using the IDES-SST turbulence model. In this study of cross-sectional fluid flow through a T-junction, a 180° rotation of vortex patterns is observed in the POD plots, which can be attributed to an intriguing transition in the flow structure. Specifically, the vortices at $x/D_m = 0.292$ are rotated in a clockwise direction. This transition highlights the complexity of the flow dynamics and the inherent symmetry in the POD modes, which can represent physical structures in different orientations without altering their underlying significance.

The general finding is that Reynolds number exerts a stronger influence than momentum ratio on turbulence structures has direct fatigue implications, as these structures dictate the mixing dynamics and resulting wall temperature fluctuation patterns that cause cyclic thermal stresses. For example, an in-depth analysis reveals that the Reynolds number of the branch flow does not significantly affect the flow structures upstream of the T-junction, while only slightly influencing the flow patterns downstream at the exit. Instead, the inlet branch diameter plays a more critical role, where the second mode shows a larger region in Cases 2 and 3 compared to Cases 1, 4, and 5. This suggests that the diameter of the inlet branch has a more pronounced impact on the downstream flow structures, indicating that geometrical factors may be more influential than the Reynolds number in certain regions of the flow. Additionally, this study highlights the dominant effect of secondary flow strength on the flow structures downstream of the T-junction. Higher Reynolds numbers in the inlet branch flow tend to weaken the secondary flow in the downstream region, resulting in a highly energetic region at the T-junction exit as the primary underlying flow structure. In contrast, in cases where the secondary flow is stronger and dominates the main flow, more significant shedding of vortical structures from the shear layer jet become the main underlying flow structure. This suggests that the secondary flow's strength, influenced by the inlet branch diameter and flow conditions, plays a crucial role in shaping the downstream flow patterns. When comparing POD and SPOD analyses, it becomes clear that the flow structures and patterns generated by each method are somewhat similar. In the cross-sectional analysis, it was observed that the underlying structures are similar for $x/D_m = -1.36\&0$. However, at $x/D_m = 0.292$, major differences were seen in terms of the patterns for POD and SPOD.

One interesting observation for the analysis in the X-Z plane is the larger differences in flow

structures generated from POD and SPOD, as compared to the cross-sectional analysis. For example, more intricate details of vortex shedding are noted in the SPOD plots versus the POD plots, where vortex shedding phenomena was observed but with much less definition. While POD is widely used and valuable, particularly for identifying dominant flow structures in many applications, SPOD's ability to account for frequency-dependent dynamics may offer some advantage, especially in the study of turbulent flows. SPOD reveals the nuances and complexities of the underlying flow structures more clearly, making it a more powerful tool for analysing flows where oscillatory behaviour and intricate vortical structures are important.

Chapter 7

Assessment of Neural Network Parameter Selection

The work presented in this chapter is taken from a conference paper by Wong et al (Wong et al. (2024a)). This was presented during Advances in Thermal Hydraulics (November 17th -21st 2024) by the American Nuclear Society, held in Orlando, Florida. This chapter presents the results of how tuning neural network hyperparameters can affect its performance. The study in this chapter involves a PINN to solve the well-established lid-driven cavity flow CFD verification benchmark problem. The main motivation is to derive a robust set of hyperparameters that can be reused for the eventual neural network-based surrogate model in chapter 8.

While this problem is relatively simple, it exhibits complex flow characteristics, making it an ideal test case for evaluating the numerical accuracy and computational efficiency of neural network-based models. Unlike conventional DNNs which rely solely on data, PINNs incorporate governing physics equations into the training process. This added complexity makes PINNs particularly sensitive to network settings, providing a unique opportunity to assess how architectural choices and hyperparameters influence model performance. By leveraging PINNs, this study systematically examines these dependencies while maintaining a physically consistent solution space. Additionally, capability of PINNs is explored to produce accurate results with sparse training datasets, distinguishing this approach from traditional neural networks that typically require large amounts of labelled data. This study considers a Reynolds number of 10,000 to assess PINN performance in the turbulent flow regime and investigate the role of

transfer learning (TL) in improving accuracy and efficiency. In section 7.1, a description of the well-established lid-driven cavity flow CFD verification benchmark problem is presented. It is supplemented alongside a detailed overview and mathematical induction of activation functions in section 7.2. In section 7.3, performance based on the number of layers, number of neurons per hidden layer, activation functions and number of samples are analysed and discussed. This chapter then concludes with a conclusion found in section 7.4.

7.1 Description of Lid Driven Cavity Problem

As mentioned above, the main example used here is a classic two-dimensional (2D) flow problem which is used widely within the field of CFD as a verification benchmark. The lid-driven cavity problem, despite its relatively simplistic geometry, exhibits a variety of complex fluid flow phenomena, including recirculating vortices and shear layers, which can be challenging to model accurately. It is for this very reason that this problem is selected to stress test the sensitivities of various network parameters. Figure 63 shows the computational domain of the problem constructed using the commercial CFD software STAR-CCM+. A turbulent fluid flow regime with a Reynolds number of 10000 is selected for the CFD model in Simcenter STAR-CCM+ (Siemens Digital Industries Software (Siemens 2023)). It is understood that a higher Reynolds number will lead to the formation of smaller eddies away from the centre of turbulence (Britter and Hanna (2003)) and these regions can be generally found in corners or areas with sharp edges/boundaries.

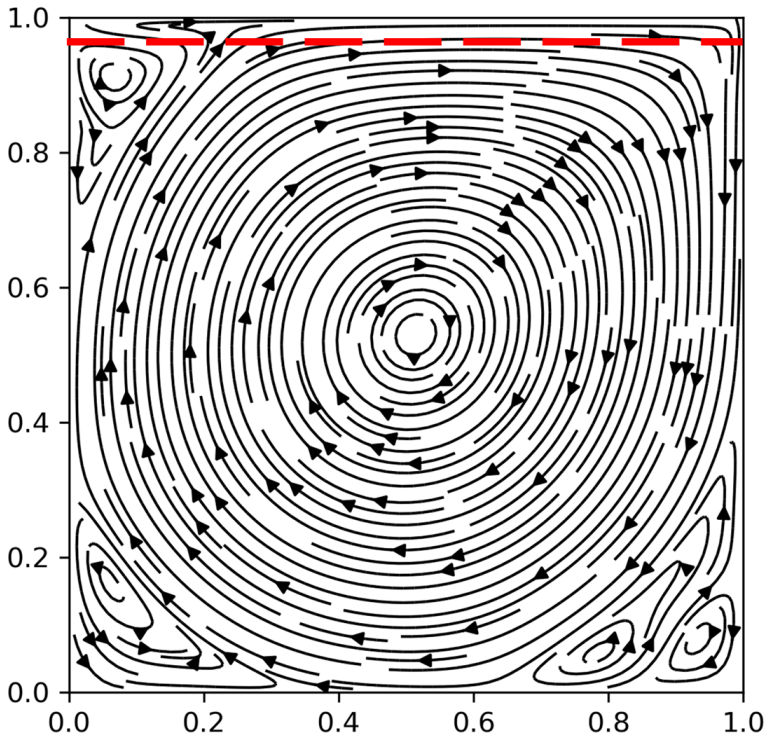


Figure 63: Geometry of the Lid-Driven Cavity and its Solution Comprised of Streamlines (Red-dashed Line Represents the Location of Interest for Analysis in Section 3)

7.2 Activation Functions

The performance of a neural network is influenced by numerous factors, and it is widely acknowledged that there is no single set of ideal parameters that guarantees optimal performance for every application. Each neural network must be tailored to the specific complexity and requirements of the problem it aims to solve. In this section, various applications will be examined and analysis on how different parameter settings impact the performance of the respective neural networks is carried out.

While the number of layers and neurons in a neural network is crucial, the choice of activation functions is often overlooked yet pivotal. Activation functions enter the analysis by introducing nonlinearity into the network, which is essential for learning complex mappings. During training, they influence the gradient flow because their derivatives appear in the chain rule used in backpropagation. This means the choice of activation function affects how effectively the network learns, and whether issues like vanishing or exploding gradients occur. Without them, networks are confined to linear transformations akin to simple linear regression models. As problems grow in complexity and networks expand, non-linear activation functions become indispensable, allowing networks to capture intricate functions essential for effective problem-solving. There are a few popular choices which are widely used in training neural networks (Apicella et al. (2021); Sibi et al. (2013); Rasamoelina et al. (2020)) and in this study, a variety of activation functions are considered, such as 'tanh', 'soft-plus', 'sigmoid', 'SILU', 'SELU' and 'ReLU'. A graphical description of these activation functions can be found in Figure 64 below. Their mathematical expressions can be found in the Table 10, accompanied by the reason(s) for their selection.

Table 10: Activation Functions and their Mathematical Expressions

Type	**Mathematical Formula**	**Reasons for Selection**
Tanh	$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	It is suitable for problems with data that includes a median of zero. Moreover, due to its symmetrical characteristics, it helps with gradient descent optimization.
Soft-plus	$f(x) = \ln(1 + e^x)$	Softplus is commonly used in probabilistic models where smooth gradients are required. This helps to mitigate the issues of wrongly unweighted neurons seen in ReLU.
Sigmoid	$f(x) = \frac{1}{1+e^{-x}}$	In binary classification, Sigmoid is commonly used as it re-scales large inputs into small values. These can be interpreted as probabilities. However, it risks facing the vanishing gradient due to its nature.
SILU	$f(x) = \frac{x}{1+e^{-x}}$	SILU (or Swish) provides better gradient propagation than ReLU and can potentially perform better at some tasks.
SELU	$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha e^x - \alpha & \text{if } x \leq 0 \end{cases}$	It is designed for self-normalizing networks where many layers are involved.
ReLU	$f(x) = \max(0, x)$	ReLU is the simplest of all activation functions, where it is less likely to face the vanishing gradient problem. However, as mentioned above, weights for certain neurons will stop updating when $x \leq 0$.

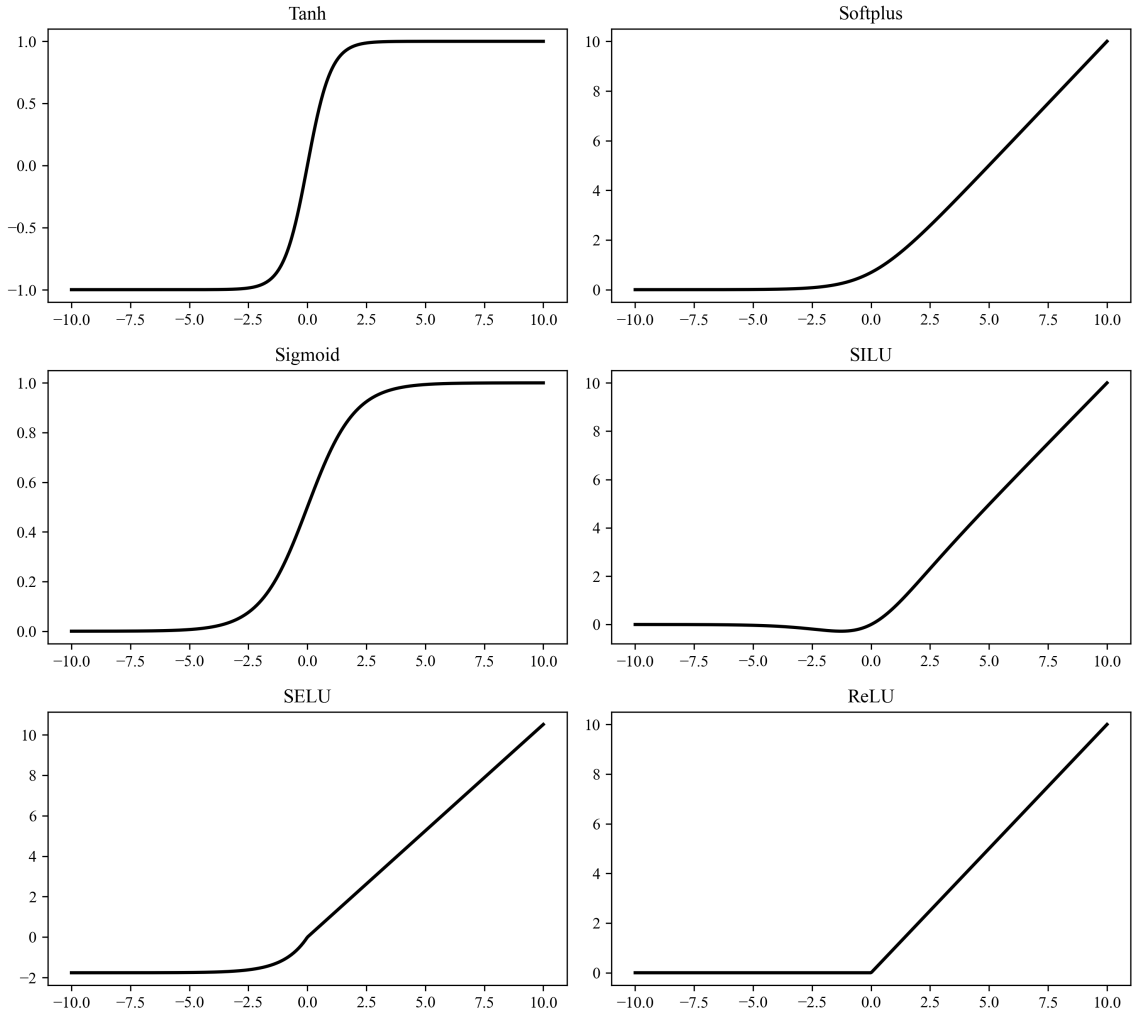


Figure 64: Graphical Representation of Activation Functions

It is a common practice to configure the number of layers and neurons in each hidden layer manually. Deep and wide networks have been proven effective, but they come at a great computational cost which increases their impracticality (Adil et al. (2022); Alvarez and Salzmann (2016)). The current challenge here, is knowing the number of hidden layers and their neurons which some works have been done in the hopes of reducing the number of parameters while retaining the network accuracy (Sheela and Deepa (2013); Vujicic et al. (2016)). These values are expected to be different for different and more complex problems, especially for this case which involves certain complexities. This is carried out on a Reynolds number value of 10000, mainly because turbulent flows are ubiquitous in real-world engineering and scientific problems, such as aerodynamics, weather forecasting, and industrial processes. Successfully modelling turbulent flows can have substantial practical implications due to their complex, chaotic, and multi-scale behaviours, which pose significant challenges for traditional CFD methods. These character-

istics make turbulent flows an ideal testbed for advanced neural network approaches. In this sensitivity analysis, ADAM is used as the main choice of stochastic optimization (Kingma and Ba (2015)).

7.3 Network Performance Analysis

In this section, an approach is presented to solve the Navier Stokes equations in two-dimensional lid driven cavity problem. To quantify the performance and accuracy of TL-PINNs, predictions are compared with results from the solution obtained with Simcenter STAR-CCM+ (Siemens Digital Industries Software (Siemens 2023)). In the following few figures, data is extracted from a 'slice', located at $y = 0.00975$ in the near-wall region. The main reason is due to the presence of smaller eddies located at the corners (where $x = 0$ and $x = 1$) where one can see how the different PINN network settings affect the predictive performance.

7.3.1 Number of Hidden Layers

For an accurate measure of the performance of the neural networks based on their sensitivities to the number of hidden layers, specific parameters of the network and training settings are kept constant such setting limiting the number of training iterations to 10000, using 60 neurons per hidden layer, setting the learning rate for ADAM = 0.005 and 'softplus' as the activation function. The accuracy of predictions generated by models are assessed, with a range of hidden layers, varying between 2 and 12 hidden layers. The resulting L^2 error plot during model training for each neural network with different number of hidden layers are plotted and shown in Figure 65. In general, the speed of convergence is the fastest for models with less hidden layers. For example, the model with 12 hidden layers only achieved convergence at a much later iteration while the model with just 2 hidden layers converged to the same level of error much earlier. Note that the error threshold is the highest for 2 hidden layers while the error threshold is approximately the same for more than 2 hidden layers used. These observations can be seen from the error value where it starts to plateau.

However, this cannot be used as a primary metric for evaluating the models' predictive performance. The study uses two-dimensional velocity profiles to provide a clearer comparison between the models' predictions against CFD data. These are presented in Figure 67(a). The model with 2 hidden layers performed poorly, as it severely under predicts velocity values in

the near wall regions. Models with 6 and 8 hidden layers used demonstrated very similar near wall performance with slight under prediction at the corners. Models with 10 and 12 hidden layers performed as poorly as the model with 2 hidden layers, where it underpredicted velocity in the near-wall region. This is potentially caused by overfitting with a higher number of hidden layers. The model with 4 hidden layers showed a slightly poorer performance compared to models with higher number of hidden layers used, where it showed signs of under prediction towards the middle of the flow field (at $x = 0.5$).

7.3.2 Number of Neurons per Hidden Layers

Another factor that is known to affect the performance of neural networks is the number of neurons used within each hidden layer. To evaluate how the neural networks' performance varies with the number of neurons per hidden layer, specific parameters of the network and training settings are kept constant such setting limiting the number of training iterations to 10000, using 6 hidden layers, setting the learning rate for ADAM = 0.005 and 'softplus' as the activation function. Figure 67(b) presents two-dimensional velocity profiles of predictions against the data. The model with 2 neurons demonstrated the worst performance, where velocity across all spatial points is severely underpredicted. Note that the performance is the best with 80 neurons used. Conversely, models with 20, 40, 60 and 100 neurons severely underpredicts at the right most corner in the near-wall region.

7.3.3 Choice of Activation Functions

While the number of hidden layers and neurons used per hidden layer affects the neural network's performance, activation functions are also essential for precise and numerically accurate results. The use of a wide range of different activation functions found within Tensorflow Keras' library is compared in this work. Certain network settings are kept constant such as limiting the number of training iterations to 10000, using 6 hidden layers with 60 neurons per hidden layer and setting the learning rate for ADAM = 0.005. Velocity line plots are presented in Figure 67(c) where it is shown that 'SELU' and 'ReLU' are exceptionally bad at predicting velocities in the right most corner of the flow field. This can be seen from the underpredicted values at $x = 1$ in the plot. The best performance came from using 'softplus'.

7.3.4 Number of Sample Points Required for Recovery

One noteworthy area for consideration is the ideal number of sample points required to obtain reasonable accurate predictions. While the number of sample points used in training affect how well the model is trained, the number of sample points used in flow recovery also matters. The range of samples points used for prediction are further explored in the following few figures. Figure 66 shows the locations of the sample points used in this study, for different cases. A smaller number of sample points leads scatter points within the control volume that are more spaced out while a higher number of samplers leads to a denser set of scatter points located closer from each other. More sample points generally allow for better predictions, but it can also lead to a case of overfitting. As seen in Figure 67, a smaller number of sample points led to poor predictions. Note that the predictions generated with 3000 sample points led to the best results while the others failed to generate consistent predictions.

7.3.5 Training Time and Quantitative Error Metric

The main advantage of using a neural network in place of numerical solvers is the cheaper computational cost and the overall efficiency. In this example, the time required to generate a converged flow from the numerical solver takes $300s$ and the choice of using this example is to rigorously test the validity of the PINN that involves incompressible Navier Stokes equations. Figure 68 shows the training time required to train each network with their respective settings. The x-axis represents the number of parameters considered per performance metric. Generally, it is expected that a denser neural network (high number of neurons and high number of hidden layers) lead to higher training times. As seen in the previous section, a higher number of neurons and/or number of hidden layers do not necessarily guarantee better performance, which does not justify the longer training times required. The choice of activation functions used also impacts the training time, where using SILU, SELU and ReLU require a shorter training time. However, much is left to be desired regarding their predictive accuracy. Using Softplus gave the best results, albeit the slightly longer training time required.

Here, the predictive accuracy of different model configurations is also assessed, by calculating various types of errors between the predicted values and the reference data obtained from the finite element solver. These values are subsequently visualized using plots in Figure 69 to provide a clear and comparative analysis of the models' performance. This approach allows

for a quantitative evaluation of using different settings that affects the models' capabilities in approximating the underlying physical phenomena. The first plot illustrates the Root Mean Square Error (RMSE) across different network settings. The RMSE decreases slightly with an increase in the number of hidden layers, stabilizing around 6 layers. Regarding the number of neurons, there is a significant drop in RMSE when increasing from 2 to 20 neurons, with the error stabilizing thereafter. The activation function influences RMSE to a moderate extent, with 'tanh' and 'SELU' showing better performance. Increasing the number of samples leads to a reduction in RMSE, with a notable improvement seen when moving from 100 to 500 samples. While literature indicates that Physics-Informed Neural Networks (PINNs) can achieve RMSE values below 0.05, the RMSE value of around 0.1 in this study is attributed to the preliminary stage of investigation into training parameters such as learning rate and choice of optimizers. This research primarily aims to demonstrate the effectiveness of network settings for transfer learning in PINNs, rather than optimizing all training parameters for minimal error. The focus is on showcasing how adjustments in network architecture, such as the number of hidden layers and neurons, along with the selection of activation functions and sample sizes, can significantly influence the model's performance when transferring learned knowledge to new conditions. This emphasis on network settings provides a foundational understanding of their impact, laying the groundwork for future research that can fine-tune these parameters for even better accuracy.

The second plot shows the Mean Absolute Error (MAE). As the number of hidden layers increases, MAE decreases, stabilizing around 6 layers. A dramatic reduction in MAE is observed when the number of neurons is increased up to 20, after which the error levels off. The choice of activation function moderately affects MAE, with 'tanh' and 'SELU' providing better results. Additionally, increasing the sample size results in a decrease in MAE, with significant improvement observed up to 500 samples. The Mean Absolute Percentage Error (MAPE) plot indicates that increasing the number of hidden layers leads to a reduction in MAPE, which stabilizes around 6 layers. There is a sharp drop in MAPE when increasing the number of neurons to 20, after which the error stabilizes. The activation function significantly impacts MAPE, with 'tanh' and 'SELU' yielding better performance. The error decreases as the number of samples increases, particularly showing improvement up to 500 samples. The Root Mean Square Percentage Error (RMSPE) plot shows that increasing the number of hidden layers results in a decrease in RMSPE, which stabilizes around 6 layers. A significant reduction in RMSPE is observed with an increase in the number of neurons up to 20, after which the error stabilizes.

The activation function affects RMSPE significantly, with 'tanh' and 'SELU' showing better results. Increasing the number of samples leads to a reduction in RMSPE, especially improving up to 500 samples. The Log Loss plot demonstrates that an increase in the number of hidden layers slightly decreases Log Loss, stabilizing around 6 layers. There is a significant drop in Log Loss when the number of neurons increases up to 20, after which it stabilizes. The choice of activation function impacts Log Loss moderately, with 'tanh' and 'SELU' showing better performance. Increasing the number of samples from 100 to 500 significantly reduces Log Loss, indicating improved model performance.

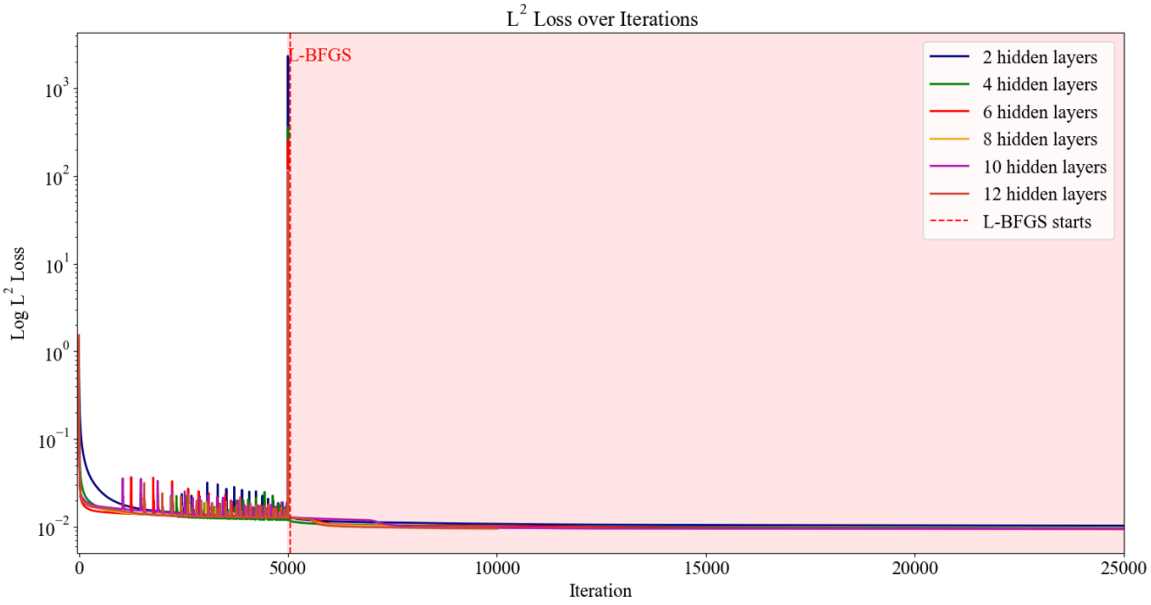


Figure 65: L^2 Error (Loss) Plot During Training for Different Number of Hidden Layers; (note that this is similar for other assessment metrics where L-BFGS greatly reduces the loss after 5000 iterations).

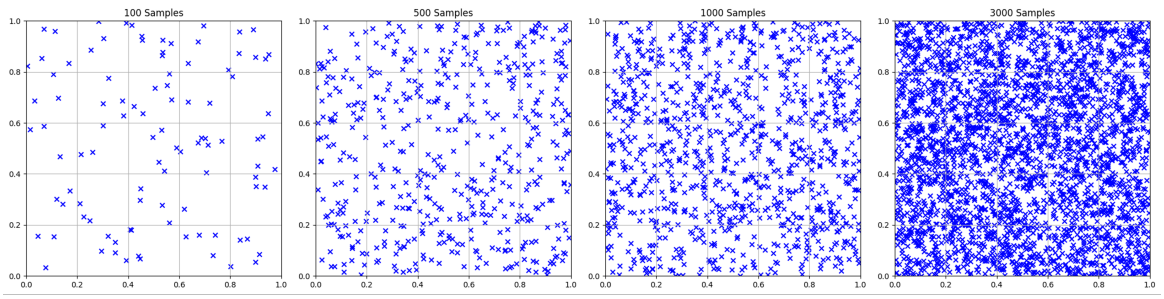


Figure 66: Scatter Plot Using (a) 100, (b) 500, (c) 1000, (d) 3000 Sample Points for Testing.

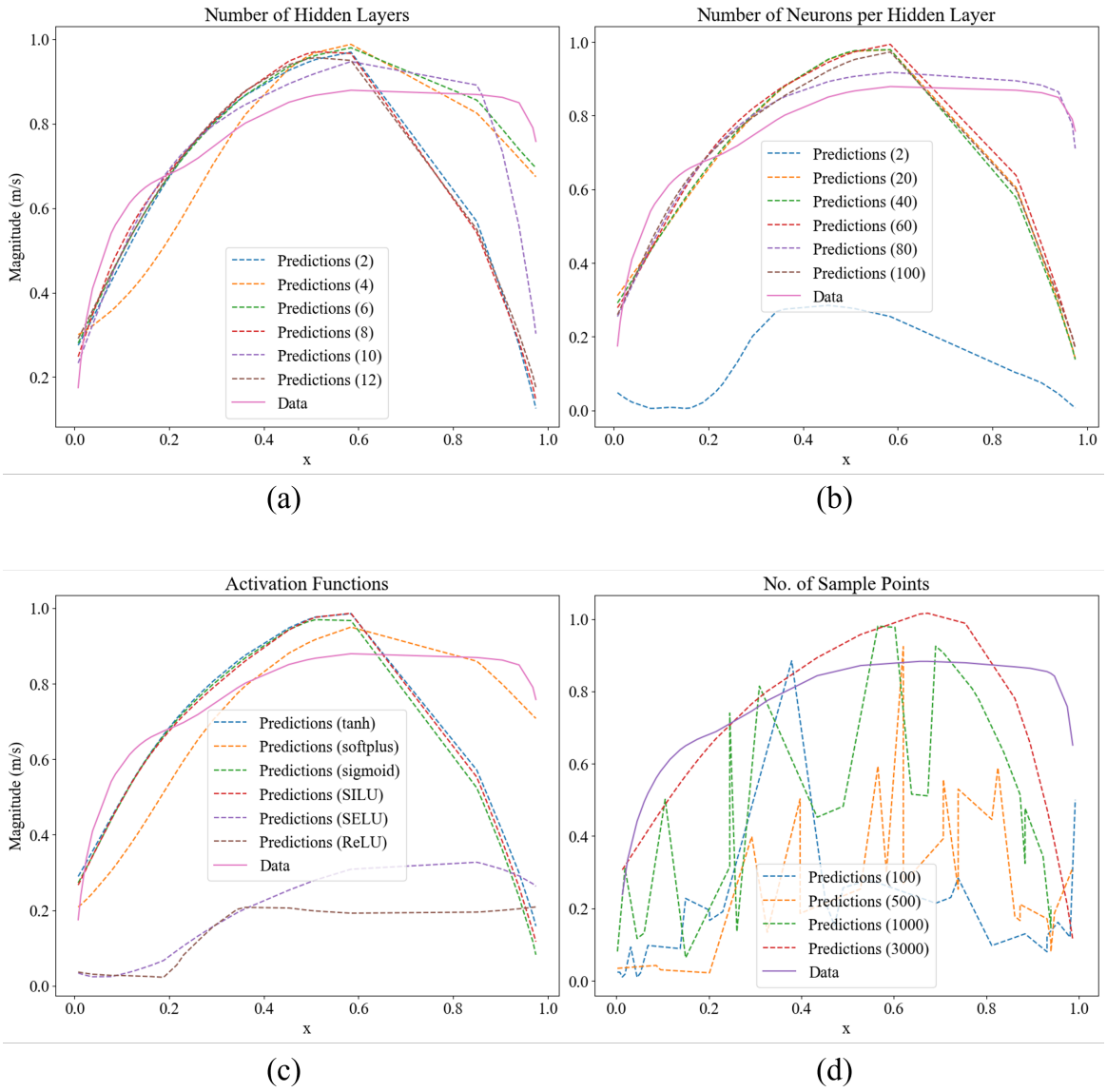


Figure 67: Velocity Profile Plots Based on Predictions for (a) Different Number of Hidden Layers, (b) Number of Neurons per Hidden Layer, (c) Activation Functions, (d) Number of Sample Points

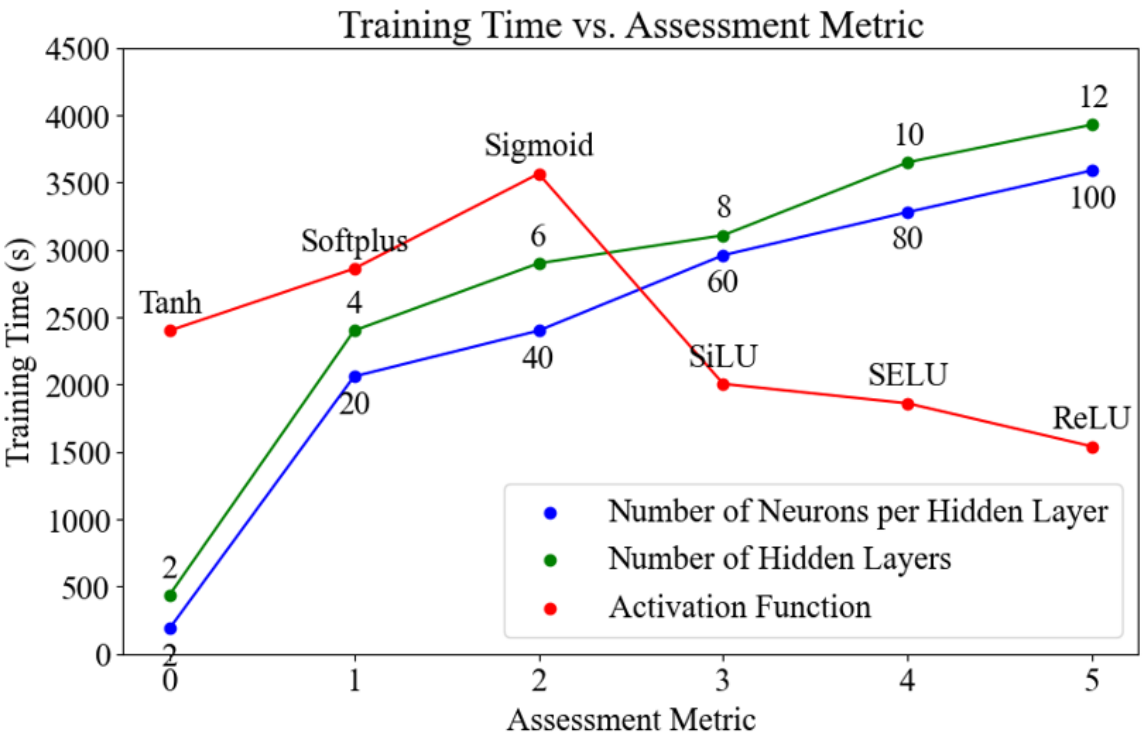


Figure 68: Training Time (s) for Different Network Settings

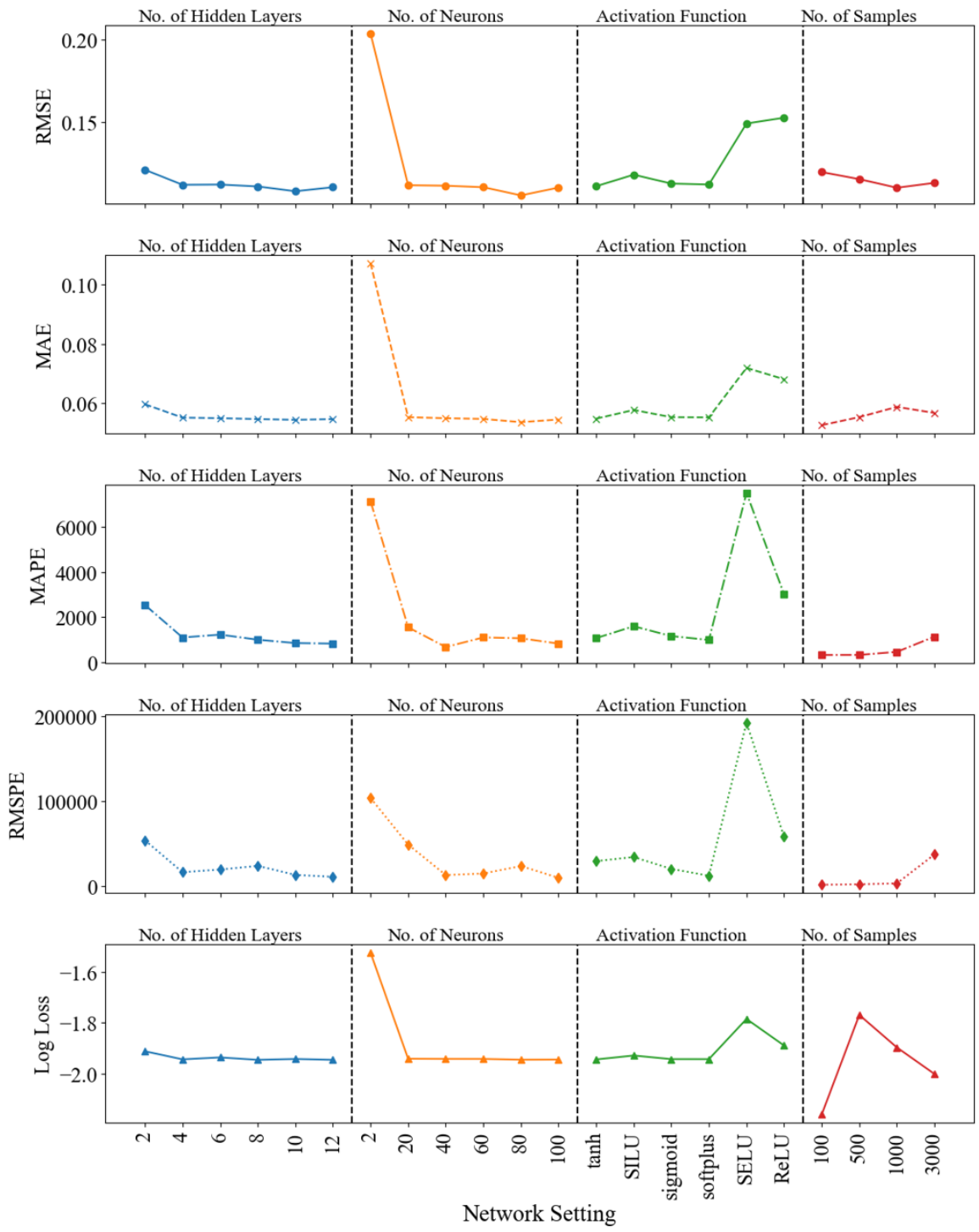


Figure 69: Error Metric Plots for Different Network Settings (Quantitative)

7.4 Conclusion

This study evaluates the performance of Physics-Informed Neural Networks (PINNs) against conventional CFD approaches. More specifically, the PINN based surrogate model is applied

to the well-established lid-driven cavity benchmark verification fluid flow problem. This work is novel as it provides the first comprehensive assessment of the computationally efficient and numerical accuracy of PINN based surrogate models for this problem, particularly focusing on a version of PINNs that require minimal pre-existing data for training. This approach differentiates the current study from conventional PINN based surrogate models, which typically need large datasets to achieve numerically accurate predictions. A Reynolds number of 10000 was considered, which encompasses the turbulent flow regime. The sensitivity analysis of PINN hyperparameters conducted here directly impacts thermal fatigue prediction, as optimal configurations improve the model's ability to reconstruct turbulence features responsible for mixing-driven temperature fluctuations at fatigue-critical wall regions. The results demonstrate the critical importance of sensitivity analysis in neural network architecture, particularly concerning the number of hidden layers, the number of neurons per hidden layer, the selection of activation functions, and the minimum number of samples required for numerically accurate predictions. It was found that using more than 6 hidden layers and using more than 60 neurons per hidden layer led to a poorer performance. Although most of the training times hovered within the magnitude of 1000 seconds, note that this is dependent on the number of training points and certain network settings used. In other words, should this be applied to a more complex problem that requires numerical solver runtimes in higher order of magnitudes, the training time required for the neural network will very likely remain unchanged. The quantitative analysis of error plots reveals that the optimal neural network settings involve around 6 hidden layers and 20 neurons per layer across various error metrics. The 'tanh' and 'SELU' activation functions consistently provide better results, indicating their suitability for this particular model. Additionally, increasing the sample size to beyond 500 samples significantly enhance model performance across all error metrics. These findings highlight the importance of selecting appropriate network settings to minimize error rates and optimize the performance of neural networks.

Regarding boundary and initial conditions, retraining the PINN becomes necessary as they are not always the same when a higher model complexity is involved. Changing these conditions effectively alters the problem the network was trained to solve, necessitating an adjustment to the new conditions. Different conditions can lead to different behaviours and solution structures, requiring the network to explore and find the appropriate solution for the new setup. Despite the need for retraining, PINNs remain highly valuable and useful. They can quickly solve

complex PDEs, making them attractive even if retraining is required. Their flexibility and adaptability allow them to handle a wide variety of PDEs, including those with complex and non-linear behaviours. This versatility makes them indispensable tools in scientific computing. Furthermore, in some cases, transfer learning can be employed to reduce retraining time. If the new conditions are not drastically different from the original ones, the network can be fine-tuned instead of trained from scratch, leveraging the knowledge already learned.

Chapter 8

Deep Neural Networks as Surrogate Models for Flow Field Reconstruction

The work in this chapter is taken from a conference presentation by Wong et al. It was the first ERCOFTAC Workshop on Machine Learning for Fluid Dynamics that took place from March 6–8, 2024, at Sorbonne University's Pierre and Marie Curie (Jussieu) Campus in Paris, France. This chapter presents the assessment of various machine learning algorithms, with a specific focus on PINN, BNN and BPINN for representing complex turbulent flow fields.

In previous chapter, the network sensitivities based on a PINN was discussed and the main reason for selecting a PINN for the comparison is due to its complexity. However, that did not directly compare how a PINN would perform with respect to other neural networks. Hence, in this chapter, an explicit comparison on how various types of neural networks perform against each other is carried out. This is achieved by using a metric that has recently been gaining traction in the field of machine learning and fluid dynamics: Flow field reconstruction. It is the process of reconstructing missing or incomplete fluid flow data using computational methods, often leveraging physics-based models, interpolation techniques, or machine learning. In many practical applications, experimental and numerical simulations of fluid dynamics are limited by spatial or temporal resolution constraints, sensor placement restrictions, or computational cost. Flow field reconstruction addresses these challenges by inferring missing flow information, enabling the generation of high-fidelity flow fields from sparse or noisy data. This is particularly valuable in turbulence modelling, where resolving all scales of motion is computationally pro-

hibitive, and in experimental fluid dynamics, where reconstructing full-field velocity or pressure distributions from limited measurements can enhance analysis. More recently, ML algorithms such as PINNS, BNNs and BPINNs have been developed to supplement traditional high-fidelity CFD simulations (Linka et al. (2022); Stock et al. (2024); Xing et al. (2024)). ML algorithms, such as PINNs and BPINNs, integrate the governing PDEs, initial conditions (ICs) and prescribed boundary conditions (BCs) into loss functions. PINN and BPINN algorithms establish theoretical constraints and biases and can be used to supplement any available experimental data. These ML algorithms can be applied to both supervised and unsupervised learning tasks while requiring substantially less data than most other deep learning algorithms as their computational performance is not directly related to the quantity of available training data. A description of the computational models used is presented in section 8.1, one of which is similar to the CFD benchmark case seen in chapter 7, and the other being an extended geometry based off this model. Settings used for the neural network is described in section 8.2 and in section 8.3, a basic network sensitivity analysis is explored. These form the basis for section 8.4 where the predictive capabilities of various neural networks are compared and discussed. This chapter then concludes with a conclusion found in section 8.5.

8.1 Description of Computational Model

This chapter utilises the same two-dimensional (2D) lid-driven cavity flow problem discussed in the chapter 7, but with the addition of a more complex geometry to further assess the performance of the model. This added complexity to the classic flow problem can be found in many real-life scenarios, from convection tiles used on rooftops to small imperfections found on seemingly smooth surfaces. As these phenomena can be challenging to model accurately, it makes the problem ideal for testing the sensitivities of various network parameters. Figure 70 shows the additional computational test case considered, which represents a multi-cavity flow. This is also constructed using the commercial CFD software STAR-CCM+. The same turbulent fluid flow regime with a Reynolds number of 10,000 is selected for the CFD model in STAR-CCM+.

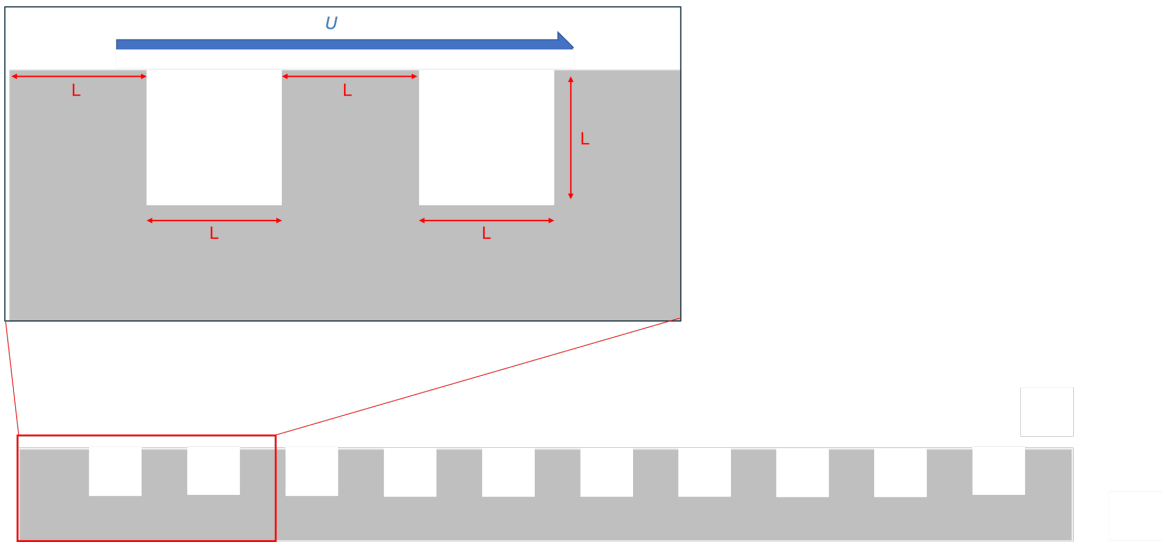


Figure 70: Computational Domain of Multi-cavitated Flow.

8.2 Neural Network Settings

For a fair comparison of performance, the neural network settings were standardized for deterministic networks (DNN and PINN) and probabilistic networks (BNN and BPINN). For DNNs and PINNs, the models were trained over 1000 epochs using the ADAM optimizer with a learning rate of 0.005. An epoch is one full pass through the entire training dataset during the training of a machine learning model. After each epoch, the model updates its parameters to reduce prediction errors based on the data it has seen. As for the number of sample points used for training, the training dataset consisted of 2000 points, while 1000 points were reserved for testing. These fixed settings ensure a consistent basis for evaluating the performance of deterministic networks. In contrast, probabilistic networks (BNN and BPINN) require a different approach due to their inherent uncertainty quantification. Instead of relying solely on deterministic training, these networks utilize an adaptive Hamiltonian Monte Carlo (HMC) sampler to draw 2000 samples from the posterior distribution. HMC is a method for efficiently sampling from complex probability distributions based on the concept of momentum. It treats the sampling process like a particle moving through a landscape, guided by gradients to take long, informed steps. This helps it explore the space more thoroughly and avoid getting stuck, especially in high-dimensional problems like BNNs in this case. The adaptive HMC is a state-of-the-art Markov Chain Monte Carlo (MCMC) algorithm designed to efficiently sample from complex, high-dimensional probability distributions. Traditional HMC requires manually

setting the trajectory length (or number of leapfrog steps) and the integration step size. Adaptive HMC eliminates the need for this manual tuning by dynamically adapting the trajectory length during sampling and automatically tuning the step size during a warm-up phase. It works by introducing an auxiliary momentum variable for each parameter, simulating Hamiltonian dynamics to propose new states. Unlike standard HMC, adaptive HMC avoids inefficient backtracking by building a binary tree of candidate states, exploring both forward and backward directions in time until the simulated path begins heading back toward its origin. The algorithm then samples one of the valid candidate states from this trajectory. During an initial adaptation phase, it also uses a dual-averaging scheme to tune the integration step size so that the acceptance probability stays close to a desired target and often estimates a diagonal or full mass matrix to better match the geometry of the target distribution. These adaptations enable the adaptive HMC to perform well out of the box in a wide range of applications without the need for expert tuning, while still preserving detailed balance and ergodicity.The algorithm behind the adaptive HMC sampler used for both BNN and BPINN can be found in Algorithm 5 below. This probabilistic sampling is essential for capturing uncertainty in the predictions, a key feature of BNNs and BPINNs, and cannot be directly compared to the training process of deterministic networks. Thus, the differences in settings reflect the distinct objectives and methodologies of deterministic versus probabilistic networks. To get a better idea of how the codes are implemented for each of these neural networks, example pseudocodes for each neural network can be found in chapter 4, section 4.4.

Algorithm 5 Adaptive HMC

Initialize: Step size ϵ, Mass matrix M
for $t = 1$ to number of samples **do**
  Sample momentum from normal distribution $p^{(0)} \sim \mathcal{N}(0, M)$
  Simulate forward and backward in time using leapfrog steps (build trajectory/tree)
  Stop if U-turn in trajectory is detected
  Pick a valid state from the tree
  Accept and append new samples
  if $t \leq$ warmup steps **then**
    Adapt step size ϵ
    Adapt mass matrix M
    $y \leftarrow W[k+1] \cdot X[k] + b[k+1]$
  end if
end for

The neural network configurations employed in this study were systematically designed to investigate the influence of activation functions, network depth, and width on model performance,

while adhering to specific computational constraints. Three activation functions: Tanh, Sigmoid, and ReLU, were evaluated due to their differing impacts on gradient propagation and convergence during training. The Tanh function, being centered at zero, was the default activation function for most configurations and served as the baseline for comparison. Sigmoid, known for its smooth gradients but potential for vanishing gradient issues, was also tested. ReLU, a widely used activation function for its computational simplicity and resistance to vanishing gradients, was included for assessing its suitability in this context. The variety of activation functions ensured a robust understanding of how non-linear transformations influence learning behavior and convergence rates.

The network's width, defined by the number of neurons per hidden layer, was varied across five configurations: 2, 20, 40, 60, and 80 neurons per layer. This variation enabled a systematic exploration of how increasing or decreasing network capacity affects the ability to capture complex patterns in the data. To isolate the effects of width, the number of hidden layers was fixed at 4 during these tests, using Tanh as the activation function. Similarly, the network's depth was examined by varying the number of hidden layers while maintaining a fixed width of 40 neurons per layer. The configurations tested included networks with 2, 4, 6, and 8 hidden layers, which allowed for the assessment of how deeper architectures impact training stability and predictive accuracy. These numerical experiments aimed to identify optimal configurations that balance model complexity and computational efficiency.

The computational environment was carefully chosen to ensure reproducibility and compatibility with the tools required for the study. All simulations were conducted using Python 3.11.15. This version of Python was specifically selected because Python 3.12.x is incompatible with certain TensorFlow 2.0 libraries used in the implementation. The hardware setup included an Intel Xeon Gold 6442Y processor with a clock speed of 2.6 GHz and a maximum allowable RAM of 32GB. These specifications provided sufficient computational resources for training networks of varying depth and width while preventing memory bottlenecks during large-scale experiments.

8.3 Network Sensitivity Analysis

8.3.1 Quantitative Loss

In Figure 71 , the Mean Squared Error (MSE) is plotted for both DNN and PINN, while the Mean Squared Log Loss (MSLL) is plotted for both BNN and BPINN. These are all plotted with respect for different activation functions, number of hidden layers and the number of neurons used.

One can see that performance consistently dips after surpassing certain thresholds, such as more than six hidden layers or 60 neurons per hidden layer. This decline is likely due to overparameterization, which increases model complexity unnecessarily, leading to overfitting and instability during training. Deeper and wider networks also demand more computational resources, exacerbating numerical instability and making convergence more challenging under constrained environments. As for the performance based on the choice of activation functions, Tanh emerges as the consistently superior activation function across all neural network configurations for this problem. Its ability to maintain zero-centered outputs facilitates faster and more stable convergence during training by avoiding bias shifts in gradients. While ReLU, favoured for its computational efficiency and resistance to vanishing gradients, performs well in many tasks, its unbounded nature introduces instability in deeper networks for this specific problem. Similarly, Sigmoid underperforms due to gradient saturation, particularly in deeper architectures. The robust performance of Tanh across different configurations underscores its effectiveness in handling complex flow dynamics, such as those present in the lid-driven cavity flow problem. For probabilistic models, BPINN and BNN show strikingly similar mean standardized log loss (MSLL) values once the architecture exceeds 40 neurons per hidden layer and 4 hidden layers. This suggests that increasing network capacity beyond these thresholds yields minimal additional benefits in terms of predictive performance or uncertainty quantification. BPINNs, which incorporate physics-based constraints, are sensitive to architectural choices due to their reliance on the governing equations during training. In contrast, BNNs, which focus on uncertainty quantification through posterior sampling, achieve adequate representational power at these thresholds, rendering further increases in complexity unnecessary. The convergence of MSLL values indicates that both probabilistic models reach optimal performance with moderate network architectures, beyond which diminishing returns set in.

These observations emphasize the need to balance model complexity with computational efficiency. While increasing depth and width initially improves performance, exceeding certain limits introduces instability and overfitting. Overfitting happens when a model learns the training data too well, including its noise and random fluctuations. This makes it perform poorly on new, unseen data because it has essentially memorized rather than generalized. The consistently strong performance of Tanh demonstrates the importance of activation function selection in maintaining stability and accuracy, especially in physics-informed neural networks. Additionally, the similarity in BPINN and BNN performance at specific architectural thresholds provides valuable guidance for designing probabilistic models that balance computational cost and predictive capability effectively.

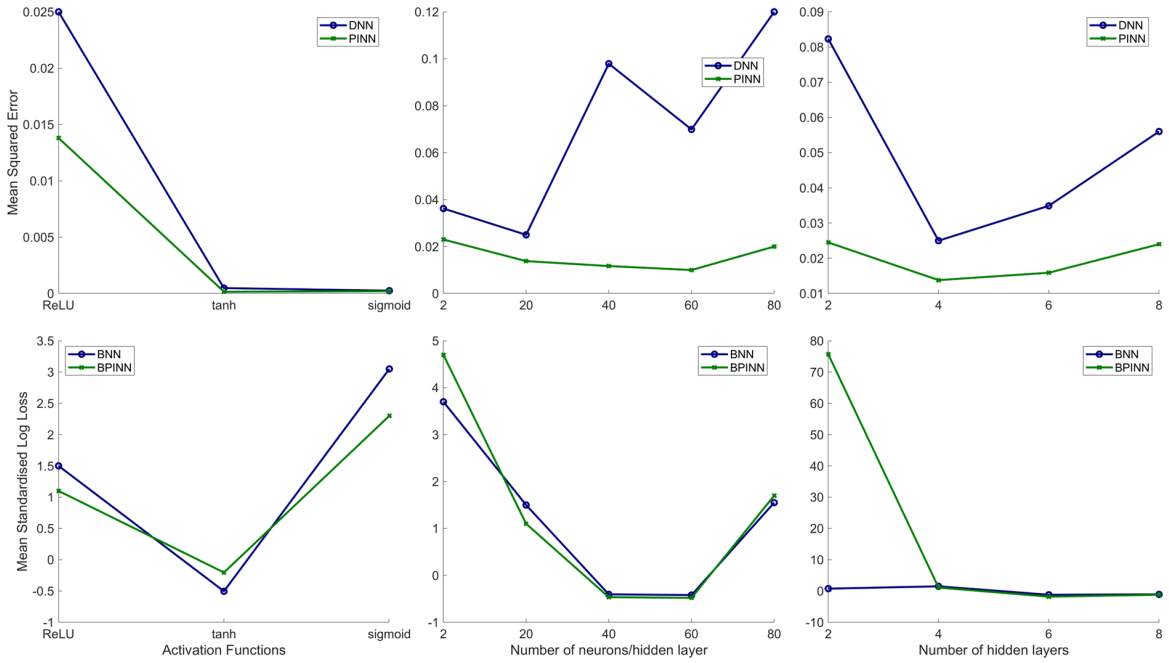


Figure 71: Loss Values for Various Network Settings.

8.3.2 Training Time

Figure 72 below illustrates the training time required for different network settings. An immediate observation is that ReLU consistently achieves the fastest training times across all neural networks due to its computational simplicity. Unlike other activation functions, ReLU involves minimal operations per forward pass and avoids the gradient saturation issues associated with Tanh and Sigmoid, resulting in quicker convergence and reduced computational overhead. This efficiency makes ReLU particularly appealing for applications where training speed is critical,

although its suitability must be balanced with accuracy considerations.

DNNs exhibit the shortest training times overall. Their relatively simple architecture, without the added constraints or probabilistic mechanisms of more advanced models, allows them to train efficiently. The absence of physics-informed constraints or Bayesian sampling processes simplifies the optimization process, making DNNs the most computationally efficient in terms of time requirements. This simplicity, however, comes at the cost of limited accuracy and uncertainty quantification, depending on the problem's complexity. BNNs, despite incorporating uncertainty quantification, require shorter training times than PINNs. This is primarily because BNNs focus on posterior sampling, such as Hamiltonian Monte Carlo, which operates within a defined framework once the architecture is set. In contrast, PINNs must enforce additional physics-informed constraints during training, which significantly increases the computational burden. The need to solve partial differential equations alongside standard neural network optimization makes PINNs inherently slower. BPINNs take the longest time to train among all the network types. This is due to the combination of physics-based constraints and the added complexity of Bayesian inference, which compounds the computational requirements. While BPINNs provide comprehensive uncertainty quantification alongside physics-based accuracy, their extended training times raise questions about feasibility, particularly for time-sensitive or large-scale problems. The practicality of BPINNs ultimately depends on the specific application and whether the additional accuracy and uncertainty quantification justify the increased computational expense.

While simpler architectures like DNNs and BNNs excel in training speed, more complex models such as PINNs and BPINNs prioritize accuracy and uncertainty quantification at the expense of training times. The rapid training enabled by ReLU demonstrates its utility in accelerating optimization, though the choice of activation function must align with the problem requirements. The extended training times of BPINNs, while a concern for feasibility, reflect their ability to address highly complex problems, emphasizing the need for careful consideration of computational resources in selecting the appropriate network type.

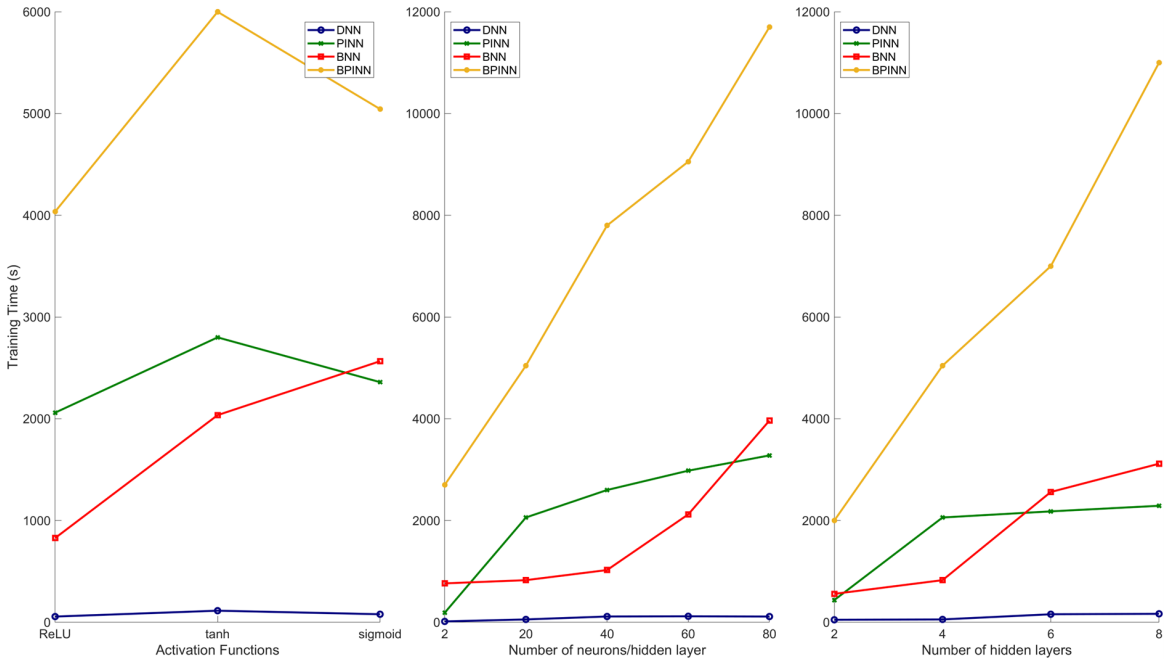


Figure 72: Training Time for Various Network Settings.

8.4 Flow Field Recovery Comparison for Different Neural Networks

Based on the findings from the previous section, the most optimal set of network settings that balances quantitative error and computational efficiency, has been identified. These settings offer the best performance with a reasonable training time and will be adopted in the following analysis to ensure a fair and consistent comparison across the DNN, PINN, BNN, and BPINN models, as shown in Table 11 below.

Table 11: Hyperparameters for each model and case

Case	Hyperparameters	DNN	PINN	BNN	BPINN
Lid Driven Cavity	Number of epochs	1000	10000	-	-
	Number of samples	-	-	3000	3000
	Training Points	2000	2000	2000	2000
	Test Points	1000	1000	1000	1000
	Leapfrog Steps	-	-	30	50
	Step Size	-	-	0.01	0.005
Multi-cavitated Flow	Number of epochs	3000	20000	-	-
	Number of samples	-	-	6000	6000
	Training Points	6000	6000	6000	6000
	Test Points	3000	3000	3000	3000
	Leapfrog Steps	-	-	30	50
	Step Size	-	-	0.01	0.005

8.4.1 Lid Driven Cavity Results

Figure 73 below illustrates the predictions made by the various neural networks based on the hyperparameter settings mentioned in the previous subsection. These predictions are then interpolated and plotted as streamlines within the flow domain, where it is compared to the solution generated by the CFD. It is quite evident that both BNN and BPINN effectively recover eddies in the upper left region of the flow. This success can be attributed to their probabilistic nature, which allows for improved representation of uncertainty and more robust flow reconstructions in regions dominated by well-defined, localized vortices. The Bayesian framework enables these models to explore a wider range of possible solutions, leading to a higher likelihood of capturing intricate flow patterns in this region. However, the two probabilistic models diverge in their performance in other areas. BNN is capable of recovering eddies in the bottom left region, a feat BPINN struggles to replicate. This discrepancy could be due to BPINN's reliance on physics-based constraints, which, while beneficial in certain cases, might overly restrict the model's flexibility in regions where the flow exhibits high variability or complex interactions. In contrast, BNN's purely data-driven approach allows it to better adapt to the nuances of this region, albeit at the cost of not explicitly enforcing physical consistency.

Conversely, BPINN outperforms BNN in the bottom right region, where it successfully recovers interacting eddies. This advantage arises from the incorporation of physics-informed constraints, which enhance the model's ability to capture interactions between eddies in regions where physical principles, such as conservation laws, play a dominant role. By embedding the

governing equations of fluid dynamics into its training process, BPINN demonstrates superior performance in modelling these highly interactive flow structures, where data alone might not be sufficient to guide accurate predictions. DNN, on the other hand, performs poorly across the board, failing to recover most of the eddies. The lack of physical constraints and probabilistic mechanisms limits its ability to generalize effectively to the complexities of the flow. Without the added capacity to account for either physical laws or uncertainty, the DNN struggles with the intricate flow patterns, resulting in suboptimal performance. PINN, while better than DNN, exhibits only marginal improvement. It barely recovers any eddies, with some weak representation in the bottom right region. Although the inclusion of physics-based constraints enhances PINN's capacity to enforce consistency with the governing equations, its reliance on sparse datasets and the added computational complexity often hinders its ability to fully capture the intricate flow details. This limitation becomes particularly evident in turbulent regions with interacting eddies, where the model's sensitivity to hyperparameters and network settings becomes a bottleneck.

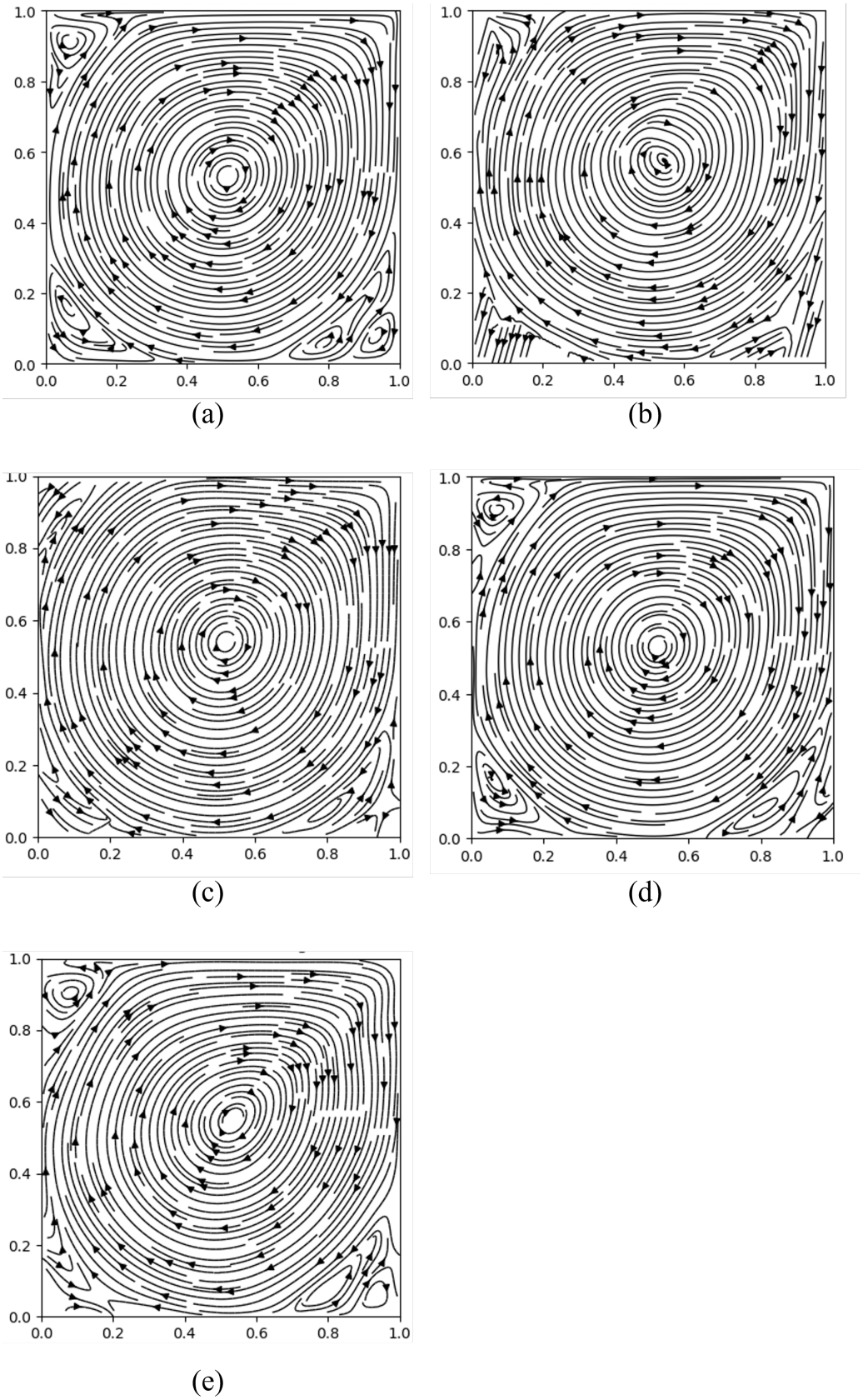


Figure 73: Recovered Flow Field a Lid Driven Cavity Based on Predictions by (b) DNN, (c) PINN, (d) BNN & (e) BPINN, Compared Against (a) CFD Solution

8.4.2 Multi Cavitated Flow

As explained in the previous section, a multi cavitated flow is inherently more complex in terms of recovering the flow field. This allows for a more detailed insight into how the different neural networks handle such added complexity when it comes to generating predictions in the additional corners of the flow domain. In a similar fashion to the previous figure, Figure 74 illustrates these predictions which are interpolated and visualized as streamlines across the flow domain, providing a basis for comparison with the CFD solution. Firstly, the BNN was the only network capable of capturing the fine-scale eddies formed in the bottom-right corner of the second cavity. This result highlights the strengths of Bayesian neural networks in handling uncertainty and representing intricate flow structures. The probabilistic framework of BNNs allows them to better capture the variability and subtle features in the data, particularly in regions where the flow dynamics are dominated by smaller-scale phenomena, such as eddies. By sampling from the posterior distribution, the BNN effectively accounts for the uncertainties inherent in the flow physics, enabling it to reconstruct these challenging features with higher fidelity. In contrast, the PINN performed poorly, likely due to the complexity involved in enforcing boundary conditions within the physics-informed framework. PINNs rely on the governing equations of fluid flow, such as the Navier-Stokes equations, to guide the learning process. However, accurately enforcing boundary conditions especially in regions with sharp gradients or complex geometries, can be computationally challenging. This difficulty can lead to inaccurate predictions in regions where the boundary conditions play a critical role, such as the formation of eddies near corners or boundaries. The inherent trade-off between enforcing physical laws and achieving numerical stability may have hindered the PINN's ability to capture these smaller-scale features.

The other neural networks, including the DNN and BPINN, were able to reconstruct the general flow field but failed to capture the finer details, such as the eddies. These models successfully learned the broader flow patterns but lacked the sensitivity or representational capacity to resolve small-scale phenomena. This limitation can be attributed to their architecture and training objectives. While they can approximate the flow field in regions dominated by larger-scale dynamics, they struggle to represent localized features like eddies, which require finer spatial resolution and a more detailed understanding of flow physics.

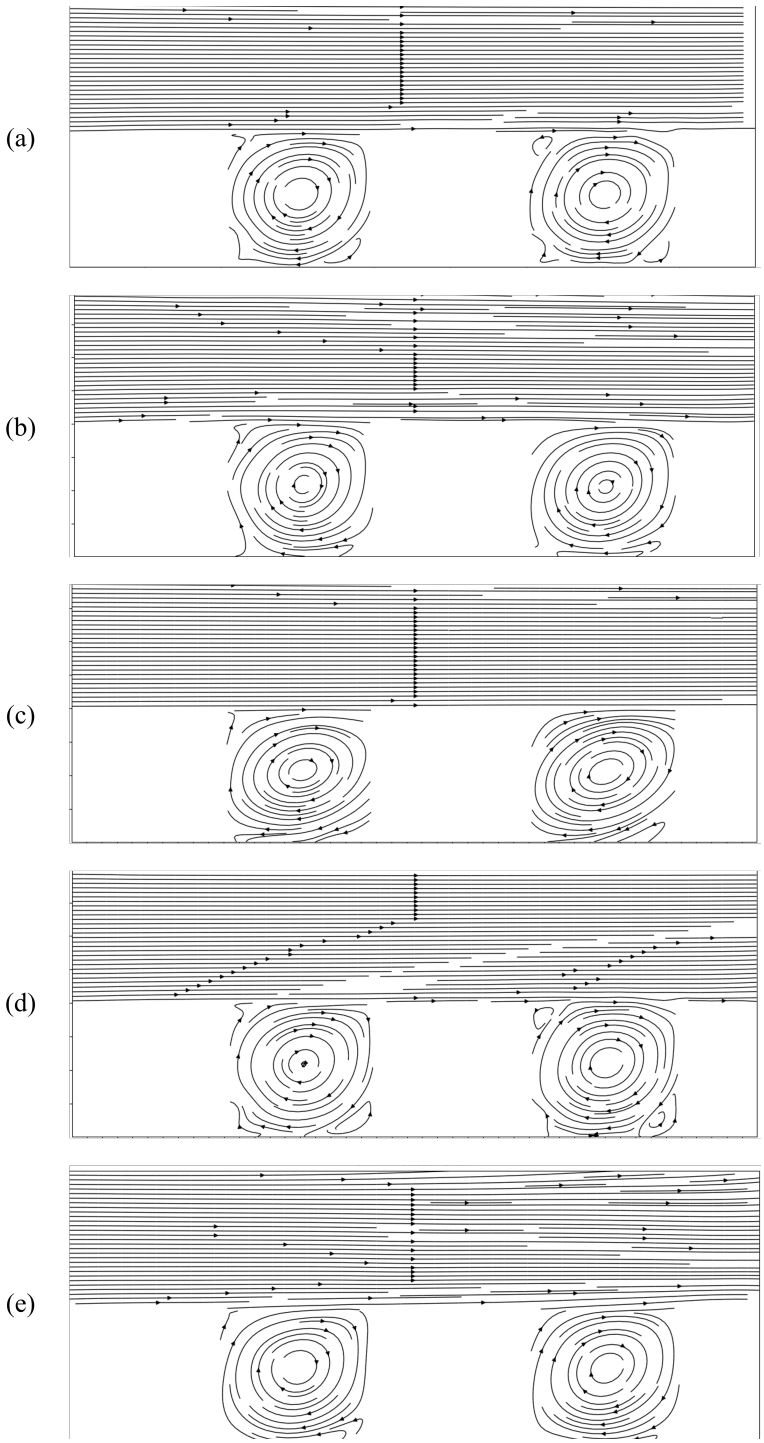


Figure 74: Recovered Flow Field for Flow Over Multiple Cavities Based on Predictions by (b) DNN, (c) PINN, (d) BNN & (e) BPINN, Compared Against (a) CFD Solution

8.5 Conclusion

For relatively simple problems, DNNs emerge as an efficient and practical choice due to their simplicity and minimal computational requirements. However, their performance declines as

the complexity of the problem increases, making them less effective for intricate flow scenarios. PINNs, while grounded in physical principles and capable of embedding governing equations like the Navier-Stokes equations into their training process, demonstrate significant challenges in enforcing boundary conditions for complex geometries. As such, they may not be appropriate for problems requiring high precision in turbulent regimes or intricate flow features. BPINNs, which combine physics-informed constraints with Bayesian uncertainty quantification, performs slightly better than BNNs for simpler geometries. However, for more complex geometries, BNNs prove superior due to their probabilistic nature, allowing them to better resolve fine-scale flow details and account for uncertainties. Despite this, the heavy computational demands of BPINNs make their usage impractical for most standard workstations. Their potential is best realized on high-powered computational systems, which could unlock their capability to tackle challenging flow field problems.

These results demonstrate that the BNN is a robust and effective surrogate modelling approach, particularly well-suited for applications requiring both predictive accuracy and reliable UQ. Unlike deterministic models, the BNN inherently accounts for epistemic uncertainty through its Bayesian framework, enabling it to provide not only point estimates but also confidence intervals around its predictions. This probabilistic treatment is critical when propagating uncertainties arising from variable input parameters (parametric uncertainty) as well as from the model structure itself (model-form uncertainty). The ability of BNNs to both reconstruct fine-scale turbulence structures and quantify predictive uncertainty enhances confidence in thermal fatigue assessments by ensuring that predictions of wall temperature fluctuation statistics are both accurate and accompanied by reliable confidence bounds. Furthermore, the BNN offers significant computational advantages over full-resolution CFD simulations, allowing for scalable integration into large-scale UQ studies and potentially enabling real-time risk-informed decision-making frameworks.

Chapter 9

Future Work

9.1 Coupling Surrogate Models with Whole-Plant Simulators

From a research perspective, a long-term goal is the full integration of CFD–surrogate modelling frameworks with structural fatigue life prediction tools, enabling direct computation of fatigue crack initiation and growth from predicted wall temperature fluctuation histories. This would close the loop from flow simulation to structural integrity assessment, requiring the coupling of high-fidelity thermal–hydraulic predictions with validated thermo-mechanical models that incorporate realistic material properties and cyclic stress–strain responses.

At the plant scale, whole-plant simulators and system codes such as RELAP5-3D are widely used for design and safety verification. However, their accuracy is constrained by simplified component models. For example, T-junctions are often represented by only a handful of computational cells, which fails to capture their complex thermo-fluid behaviour. A natural evolution is to couple RELAP5-3D with high-fidelity CFD for transient events, but this is computationally prohibitive for parametric studies, sensitivity analysis, or UQ.

Surrogate models, particularly BNNs trained on transient CFD data across a range of inlet conditions, offer a viable solution. Replacing low-fidelity T-junction components in RELAP5-3D with a BNN surrogate could achieve CFD-level accuracy for wall temperature fluctuations

at a fraction of the cost, enabling near-real-time transient simulation and probabilistic fatigue risk assessment. This hybrid approach would bridge the gap between high-fidelity component modelling and whole-plant operational decision-making.

9.2 Uncertainty Quantification (UQ) and Experimental Validation

A grand challenge in advancing thermal fatigue prediction is validation under reactor-representative conditions. Large-scale experimental campaigns that measure both velocity fields and wall-temperature distributions in prototypical T-junctions are essential to benchmark CFD models, AMR strategies, and surrogate approaches. These datasets would also enable rigorous verification of UQ predictions, ensuring that model confidence intervals reflect physical reality and can be trusted in design and safety evaluations.

In nuclear thermal–hydraulics, uncertainty quantification is increasingly critical due to the sensitivity of safety margins to modelling assumptions, parameter uncertainties, and measurement limitations. In plant-scale simulations with tools such as RELAP5, both aleatoric (stochastic) and epistemic (knowledge-based) uncertainties can propagate through the system, affecting operational decisions. While Monte Carlo Simulation (MCS) provides a general framework for UQ, its computational cost is prohibitive when each transient simulation takes hours.

Surrogate-enabled UQ frameworks, particularly those using BNNs, can offer probabilistic predictions at a fraction of the computational cost, making real-time or near-real-time fatigue risk assessment feasible. The combination of high-quality experimental validation with surrogate-enabled UQ would significantly improve predictive reliability, providing a robust foundation for integrating fatigue risk monitoring into plant operation.

9.3 Further Development in Surrogate Modelling

On the surrogate modelling front, there is significant potential to develop physics-aware, uncertainty-quantifying surrogates that operate in real-time within plant monitoring systems. Such models could ingest live plant data and dynamically update fatigue risk predictions, supporting predictive maintenance strategies that optimise safety and operational efficiency.

From an architectural standpoint, time-dependent neural networks will be central for transient simulation surrogates. While initial work explored Recurrent Neural Networks (RNNs) for their sequential modelling capabilities, and Long Short-Term Memory (LSTM) networks to overcome short-term memory limitations, LSTMs are optimised for predicting future sequences rather than interpolating within existing transient intervals. Alternatives such as Transformers and temporal CNNs may be better suited for RELAP5 coupling, offering improved performance for irregular dependencies and enabling parallelised computation.

In terms of physics integration, this thesis applied BPINNs using velocity–pressure formulations of the Navier–Stokes equations. Future work could explore embedding the Cauchy–Stress tensor directly into the physics-informed loss function, providing a more direct link to stress-related outputs relevant to fatigue. Preliminary testing of a Cauchy–Stress tensor-based PINN showed mixed performance at high Reynolds numbers, but combining this formulation with Bayesian inference may improve stability and uncertainty characterisation in these challenging regimes.

Finally, the development of standardised workflows for goal-oriented mesh refinement and uncertainty propagation, supported by open benchmark datasets, will be critical for reproducibility and widespread adoption. Such standards would not only facilitate direct comparison between research groups but also accelerate the translation of advanced surrogate methods into industrial nuclear safety practice.

Chapter 10

Conclusion

10.1 Summary of Research Efforts and Concluding Remarks

This project explored a range of research interests, uncovering several notable findings, each contributing to a better understanding of turbulent thermal mixing in nuclear T-junctions and its link to thermal fatigue prediction.

Chapter 3 marks a pioneering effort in the study of traditional meshing methods alongside AMR techniques, underpinned by a dual-weighted error estimation strategy. Here, AMR was assessed as a tool to demonstrate the applicability of advanced mesh refinement to the T-junction geometry. While AMR models demonstrated improvements over low-resolution, low-fidelity approaches, they were very slight and the findings suggest that well-resolved high-fidelity meshes may still outperform AMR under certain conditions governed by the dual-weighted error estimates. This critical evaluation provides future researchers with a clear understanding of the situations in which AMR can be applied effectively to target fatigue-relevant flow and thermal features without excessive computational overhead.

Building on these insights, Chapter 4 and Chapter 5 employed a high-fidelity IDDES-SST approach which utilises a hybrid LES-RANS method to simulate turbulent flow within a T-junction. Compared to DNS or conventional LES, IDDES-SST offers a practical balance, delivering detailed resolution of turbulence structures at a fraction of the computational cost.

The analysis extended to the influence of geometry and operational parameters, with Chapter 5 specifically addressing how Reynolds number compares to momentum ratio in shaping turbulent structures. The results indicate that Reynolds number exerts a more substantial influence on the resulting flow features, which directly govern hot–cold fluid mixing and, in turn, the spatial and temporal pattern of wall temperature fluctuations driving thermal fatigue. This finding is of practical value for designers aiming to mitigate fatigue risk through flow control or operating parameter selection.

Chapter 6 bridged the gap between turbulence dynamics and fatigue-relevant thermal loads by identifying specific flow frequencies using ROM techniques such as POD and SPOD. These frequencies, associated with coherent structures like eddies and oscillating vortices, are a direct driver of temperature fluctuation cycles at the wall. While both POD and SPOD captured the main energetic structures, SPOD provided a clearer link between flow features and their dominant frequencies, which could be used to anticipate thermal fatigue excitation mechanisms in future predictive workflows.

Chapter 7 examined surrogate modelling in depth, focusing on how hyperparameter settings influence the performance of PINNs for turbulent flow field recovery. This work fills a notable gap in the literature, where most hyperparameter studies are confined to simplified PDEs. By targeting a complex, high-Reynolds-number flow relevant to thermal fatigue problems, this chapter provides systematic guidance on how tuning affects the ability of surrogate models to resolve flow features that control mixing and temperature fluctuation patterns.

Finally, Chapter 8 systematically evaluated several neural network architectures such as DNNs, PINNs, BNNs, and BPINNs on benchmark problems such as the lid-driven cavity and multi-cavity flows. Insights from this pre-study were then applied to fatigue-relevant configurations. While all architectures showed baseline predictive capabilities, BNNs consistently outperformed others in capturing fine-scale flow structures and, crucially, quantifying predictive uncertainty. This capability is essential for thermal fatigue assessment, where confidence in predicted wall temperature fluctuation statistics directly affects safety margins.

Taken together, the work demonstrates that thermal fatigue in nuclear T-junctions can be better predicted when the underlying turbulent structures that drive hot–cold mixing are accurately resolved or reconstructed. This thesis contributes to that goal by: (i) critically evaluating AMR

for its suitability in complex fatigue-relevant geometries, (ii) using high-fidelity CFD to quantify parameter effects on turbulence structures linked to fatigue, (iii) identifying dominant flow frequencies associated with wall thermal loading, and (iv) advancing uncertainty-aware surrogate models capable of reproducing these features efficiently. These developments strengthen the predictive link between CFD-derived flow features and fatigue-relevant thermal loads, providing a foundation for future coupling with structural fatigue life models and experimental validation under reactor-representative conditions.

Bibliography

Adil, M., Ullah, R., Noor, S., Gohar, N., 2022. Effect of number of neurons and layers in an artificial neural network for generalized concrete mix design. Neural Computing and Applications 34, 1–9.

Ainsworth, M., Oden, J.T., 1997. A posteriori error estimation in finite element analysis. Computer methods in applied mechanics and engineering 142, 1–88.

Alvarez, J., Salzmann, M., 2016. Learning the number of neurons in deep networks, in: Proceedings on the 30th Conference on Neural Information Processing Systems (NIPS), 5-10th December, Computer Science, Barcelona, Spain. pp. 1–9.

Andersson, U., Westin, J., Eriksson, J., 2006. Thermal mixing in a T-junction. Technical Report U 06-66. Vattenfall Research and Development AB, Stockholm, Sweden.

ANS, 1999. JAPC provides tsuruga-2 leak findings to nuclear saftey commission (NSC) from the American Nuclear Society Nuclear News.

Apicella, A., Donnarumma, F., Isgrò, F., Prevete, R., 2021. A survey on modern trainable activation functions. Neural Networks 138, 14–32.

Babuška, I., Rheinboldt, W.C., 1978. A-posteriori error estimates for the finite element method. International Journal for Numerical Methods in Engineering 12, 1597–1615.

Baranova, T., Zhukova, Y., Chorny, A., Skrypnik, A., Popov, I., 2021. Non-isothermal vortex flow in the T-junction pipe. Energies 14, 1–17.

Boser, B., Guyon, I., Vapnik, V., 1992. A training algorithm for optimal margin classifiers, in: COLT: Proceedings of the fifth annual workshop on Computational learning theory, 27-29th July, ACM Digital Library, Pittsburgh, Pennsylvania, USA. p. 144–152.

Braillard, R., Howard, R., Angele, K., Shams, A., Edh, N., 2018. Thermal mixing in a T-junction: Novel CFD-grade measurements of the fluctuating temperature in the solid wall. Nuclear Engineering and Design 330, 377–390.

Brillard, O., Quemere, P., Lorch, V., 2007. Thermal fatigue in the mixing tees impacted by turbulent flows at large gap of temprature: The father experiment and the numerical simulation, in: ICONE15: 15th International Conference on Nuclear Engineering, 22-26th April, The Japan Society of Mechanical Engineers, Nagoya, Japan. pp. 1–12.

Britter, R., Hanna, S., 2003. Flow and dispersion in urban areas. Annual Review of Fluid Mechanics 35, 469–496.

Brucker, C., 1997. Study of the three-dimensional flow in a T-junction using a dual-scanning method for three-dimensional scanning-particle-image velocimetry (3-D SPIV). Experimental thermal and fluid science 14, 35–44.

Brunings, J.E., 1982. LMFBR thermal-striping evaluation. Technical Report EPRI-NP-2672.

Buhrmester, V., Munch, D., Arens, M., 2021. Analysis of explainers of black box deep neural networks for computer vision: A survey. Machine Learning Knowledge Extraction 3, 966–989.

Cai, S., Mao, Z., Wang, Z., Yin, M., Karniadakis, G.E., 2021a. Physics-informed neural networks (PINNs) for fluid mechanics: a review. Acta Mechanica Sinica 37, 1727–1738.

Cai, S., Wang, Z., Wang, S., Perdikaris, P., 2021b. Physics-informed neural networks (PINNs) for heat transfer problems. Journal of Heat Transfer 143, 1–15.

Carelli, M., Ingersoll, D. (Eds.), 2021. Handbook of Small Modular Nuclear Reactors. Woodhead Publishing Series in Energy. Second ed., Woodhead Publishing (Imprint of Elsevier), Oxford, UK , ISBN 978-0-12-823916-2.

Castelvecchi, D., 2016. Can we open the black box of AI? Nature 538, 20–23.

Chandy, A., Frankel, S., 2009. Regularization-based sub-grid scale (SGS) models for large eddy simulations (LES) of high-re decaying isotropic turbulence. Journal of Turbulence 10, 1–22.

Chapuliot, S., Gourdin, C., Payen, T., Magnaud, J., Monavon, A., 2005. Hydro-thermal-mechanical analysis of thermal fatigue in a mixing tee. Nuclear Engineering and Design 235, 575–596.

Cichy, R., Kaiser, D., 2019. Deep neural networks as scientific models. Trends in Cognitive Science 23, 305–317.

Cipiere, M., Le Duff, J., 2002. Thermal fatigue experience in french piping: Influence of surface condition and weld local geometry. Welding in the World 46, 23–27.

Coffin Jr, L., 2022. A study of the effects of cyclic thermal stresses on a ductile metal. Journal of Fluids Engineering 76, 931–949.

Coroneo, M., Montante, G., Paglianti, A., Magelli, F., 2011. CFD prediction of fluid flow and mixing in stirred tanks: Numerical issues about the RANS simulations. Computers Chemical Engineering 35, 1959–1968.

Courtin, S., 2013. High cycle thermal fatigue damage prediction in mixing zones of nuclear power plants: Engineering issues illustrated on the FATHER case. Procedia Engineering 66, 240–249.

Craft, T.and Gant, S., Gerasimov, A., Iacovides, H., Launder, B., 2002. Wall-function strategies for use in turbulent flow CFD, in: International Heat Transfer Conference 12, 18-23rd August, International Heat Transfer Conference (IHTC) Digital Library, Grenoble, France. pp. 1–5.

Dahlberg, M., Nilsson, K., Taylor, N., Faidy, C., Wilke, U., Chapuliot, S., Catalano, J., 2011. Development of a European Procedure for Assessment of High Cycle Thermal Fatigue in Light Water Reactors: Final Report of the NESC-Thermal Fatigue Project, 22nd December. Technical Report No. RIV000048.

Demkowicz, L., Oden, J.T., Strouboulis, T., 1984. Adaptive finite elements for flow problems with moving boundaries. part i: Variational principles and a posteriori estimates. Computer Methods in Applied Mechanics and Engineering 46, 217–251.

Dincer, I., Ezan, M.A., 2018. System Modeling and Analysis. Springer International Publishing, Cham. pp. 137–182.

Eivazi, H., Tahani, M., Schlatter, P., Vinuesa, R., 2022. Physics-informed neural networks for solving reynolds-averaged navier–stokes equations. Physics of Fluids 34, 1–10.

Elliott, J., Peraire, J., 1997. Practical three-dimensional aerodynamic design and optimization using unstructured meshes. American Institute of Aeronautics and Astronautics (AIAA) 35, 1479–1485.

(EMA), 2023. Singapore energy statistics 2023: Energy consumption retreived from the energy market authority.

Eriksson, K., Johnson, C., 1991. Adaptive finite element methods for parabolic problems i: A linear model problem. SIAM Journal on Numerical Analysis 28, 43–77.

Eriksson, K., Johnson, C., 1995. Adaptive finite element methods for parabolic problems ii: optimal error estimates. SIAM Journal on Numerical Analysis 32, 706–740.

Escudier, M., 1988. Vortex breakdown: observations and explanations. Progress in Aerospace Sciences 25, 189–229.

Fabiano, E., Mavriplis, D., 2016. Adjoint-based aerodynamic design on unstructured meshes, in: 54th AIAA Aerospace Sciences Meeting, 4-8th January, Aerospace Research Central (ARC) Digital Library, San Diego, California, USA. pp. 1–11.

Ferziger, J.H., Perić, M., 2002. Solution of the Navier-Stokes equations in Computational methods for fluid dynamics, 157-216. Springer, Berlin, Heidelberg, ISBN 978-3-642-56026-2.

Fidkowski, K., Darmofal, D., 2009-01. Output-based error estimation and mesh adaptation in computational fluid dynamics: Overview and recent results., in: 47th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, 5-8th January, Aerospace Research Central (ARC) Digital Library, Orlando, Florida, USA. pp. 1–10.

Frank, T., Lifante, C., Prasser, H., Menter, F., 2010. Simulation of turbulent and thermal mixing in T-junctions using URANS and scale-resolving turbulence models in ANSYS CFX. Nuclear Engineering and Design 240, 2313–2328.

Gelineau, O., Escaravage, C., Simoneau, J., Faidy, C., 2001. High cycle thermal fatigue - experience and state of the art in french LMFR, in: Structural Mechanics in Reactor Technology (SMiRT 16, 12-17th August, International American Associations for Structural Mechanics in Reactor Technology (IASMiRT), Washington, DC, USA. pp. 1–7.

Gelineau, O., Sperandio, M., Martin, P., Ricard, P., Martin, J., Bougault, A., 1994. Thermal fluctuation problems encountered in LMFR, in: IAEA-IWGFR Specialists' meeting on correlation, 22-24th November, International Atomic Energy Agency (IAEA), Aix-en-Provence, France. p. 38–46.

Georgaka, S., Stabile, G., Star, K., Rozza, G., Bluck, M.J., 2020. A hybrid reduced order method for modelling turbulent heat transfer problems. Computers and Fluids 208, 1–13.

Goan, E., Fookes, C., 2020. Bayesian neural networks: An introduction and survey in Case Studies in Applied Bayesian Data Science. Springer Nature , Cham, Switzerland, ISBN 978-3-030-42552-4.

Gritskevich, M., Garbarruk, A., Schutze, J., Menter, F., 2012. Development of DDES and IDDES formulations for the k- shear stress transport case. Flow, Turbulence and Combustion 88, 431–449.

Guseva, E., Garbaruk, A., Strelets, M., 2016. Application of DDES and IDDES with shear layer adapted subgrid length-scale to separated flows. Journal of Physics: Conference Series 769, 1–6.

Haldar, A., Mahadevan, S., 2000. Reliability assessment using stochastic finite element analysis. First ed., John Wiley & Sons, New York, NY, ISBN 978-0-471-36961-5.

Hannink, M., Kuczaj, A., Blom, F., Church, J., Komen, E., 2008. A coupled CFD-FEM strategy to predict thermal fatigue in mixing tees of nuclear reactors, presented at the 2008 Eurosafe Forum, p 1-13.

Hartmann, R., Held, J., Leicht, T., 2011. Adjoint-based error estimation and adaptive mesh refinement for the RANS and k- turbulence model equations. Journal of Computational Physics 230, 4268–4284.

Heywood, J.G., Rannacher, R., 1982. Finite element approximation of the nonstationary navier–stokes problem. i. regularity of solutions and second-order error estimates for spatial discretization. SIAM Journal on Numerical Analysis 19, 275–311.

Hinton, G.E., 2007. Learning multiple layers of representation. TRENDS in Cognitive Sciences 11, 429–434.

Hirsch, C., 2007. Numerical Computation of Internal and External Flows from The Fundamentals of Computational Fluid Dynamics. Second ed., Butterworth-Heinemann publishing (imprint of Elsevier), Oxford, UK, ISBN 978-0-7506-6594-0.

Hosseini, S., Yuki, K., Hashizume, H., 2008. Classification of turbulent jets in a T-junction area with a 90-degree bend upstream. International Journal of Heat Mass Transfer 51, 2444–2454.

Huseyin, A., Sokmen, C., 2013. CFD modelling of thermal mixing in T-junction: Effect of branch pipe diameter ratio., in: The 15th International Topical Meeting on Nuclear Reactor Thermal Hydraulics, 12-17th May, NURETH, Pisa, Italy. pp. 1–12.

Hussain, J., 2019. Deep Learning Black Box Problem. Ph.D. thesis. Uppsala Universitet.

IAEA, 1999. Report on the Preliminary Fact Finding Mission Following the Accident at the Nuclear Fuel Processing Facility in Tokaimura, Japan. Technical Report. International Atomic Energy Agency (IAEA).

IAEA, 2016. Advances in small modular reactor technology developments. a supplement to: IAEA advanced reactors information system (ARIS), a booklet by international atomic energy agency (IAEA).

Igarashi, M., Tanaka, M., Kawashima, S., Kamide, H., 2009. Experimental study on fluid mixing for evaluation of thermal striping in T-Pipe junction, in: 10th International Conference on Nuclear Engineering, 14-18th April, American Society of Mechanical Engineers (ASME), Arlington, Virginia, USA. p. 383–390.

Ingersoll, D., 2009. Deliberately small reactors and the second nuclear era. Progress in Nuclear Energy 51, 589–603.

Ioffe, S., Szegedy, C., 2015. Batch normalization: accelerating deep network training by reducing internal covariate shift, in: ICML15: Proceedings of the 32nd International Conference on International Conference on Machine Learning, 6-11th July, ACM Digital Library, Lille, France. p. 448–456.

Iqbal, M., Chan, A., 2015. A study of the effect of element types on flow and turbulence characteristics around an isolated high-reise building, in: Eleventh International Conference on CFD in the Minerals and Process Industries, 7-9th December, (CSIRO) Australia, Melbourne, Australia. pp. 1–9.

Jameson, A., Martinelli, L., . Aerodynamic shape optimization techniques based on control theory, in: 29th AIAA, Fluid Dynamics Conference, 15-19th June, American Institute of Aeronautics and Astronautics (AIAA), Albuquerque, NM, USA. pp. 1–15.

Joksimovic, I., Corbo, T., Hussong, J., Jakirlic, S., 2023. Secondary flow and thermal field dynamics in a T-junction with an upstream elbow: A POD-aided, eddy-sensitized reynolds-stress modeling study. International Journal of Heat and Fluid Flow 104, 1–15.

Jones, F.R., 2003. The effects of aggressive environments on long-term behaviour. Fatigue in Composites: Science and Technology of the Fatigue Response of Fibre-Reinforced Plastics 4, 117–146.

Jospin, L., Laga, H., Boussaid, F., Buntine, W., Bennamoun, M., 2022. Hands-on bayesian neural networks—a tutorial for deep learning users. IEEE Computational Intelligence Magazine 17, 29–48.

Kalpakli, A., 2014. Vortices in turbulent curved pipe flow—rocking, rolling and pulsating motions. Ph.D. thesis. Royal Institute of Technology, KTH Mechanics. Stockholm, Sweden.

Kamaya, M., Nakamura, A., 2011. Thermal stress analysis for fatigue damage evaluation at a mixing tee. Nuclear Engineering and Design 241, 2674–2687.

Karol, M., John, T., Zhao, J., 2015. Small and medium sized reactors (SMR): a review of technology. Renewable and Sustainable Energy Reviews 44, 643–656.

Kasagi, N., 1998. Progress in direct numerical simulation of turbulent transport and its control. International Journal of Heat and Fluid Flow 19, 125–134.

Keller, J., 1995. On the interpretation of vortex breakdown. Physics of Fluids 7, 1695–1702.

Khan, I., Sunil, P., Bhattacharyya, S., Yadav, R., Poddar, K., Kumar, S., 2023. Flow past two rotationally oscillating cylinders. Journal of Fluid Mechanics 969, 1–41.

Kimura, N., Kobayashi, J., Kameyama, Y., Nagasawa, K., Ezure, T., Ono, A., Kamid, H., 2014. Study on high cycle thermal fatigue in mixing tee: Evaluation of transfer characteristics of temperature fluctuation from fluid to structure by JAEA Research. Technical Report 2014-009.

Kimura, N., Ogawa, H., Kamide, H., 2010. Experimental study on fluid mixing phenomena in t-pipe junction with upstream elbow. Nuclear Engineering and Design 240, 3055–3066.

Kingma, D., Ba, J., 2015. ADAM: A method for stochastic optimization, in: 3rd International Conference on Learning Representations, 7-9th May, International Conference on Learning Representations (ICLR), San Diego, CA, USA. pp. 1–15.

Kolmogorov, A., 1941. The local structure of turbulence in incompressible viscous fluid for very large reynolds. Proceedings of Mathematical and Physical Sciences 434, 9–13.

Konakli, K., Sudret, B., 2016. Polynomial meta-models with canonical low-rank approximations: Numerical insights and comparison to sparse polynomial chaos expansions. Journal of Computational Physics 321, 1144–1169.

Kotov, D., Yee, H., Wray, A., Hadjadj, A., Sjogreen, B., 2016. High order numerical methods for the dynamic SGS model of turbulent flows with shocks. Communications in Computational Physics 19, 273–300.

Krishnapriyan, A., Gholami, A., Zhe, S., Kirby, R., Mahoney, M., 2021. Characterizing possible failure modes in physics-informed neural networks. Advances in Neural Information Processing Systems 34, 1–22.

Kuczaj, A., Komen, E., Loginov, M., 2010. Large-eddy simulation study of turbulent mixing in a T-junction. Nuclear Engineering and Design 240, 2116–2122.

Kuhn, S., Braillard, O., Niceno, B., Prasser, H., 2010. Computational study of conjugate heat transfer in T-junctions. Nuclear Engineering and Design 240, 1548–1557.

Laan, M., Andersen, S., 2018. The turbulence scales of a wind turbine wake: A revisit of extended k-epsilon models. Journal of Physics: Conference Series , 1–10.

Lampunio, L., Duan, Y., Eaton, M.D., 2022. Mean flow, turbulent structures, and SPOD analysis of thermal mixing in a T-junction with variation of the inlet flow profile. Energies 15, 1–36.

Lampunio, L., Duan, Y., Eaton, M.D., Bluck, M.J., 2021. The effect of inlet flow conditions upon thermal mixing and conjugate heat transfer within the wall of a T-junction. Nuclear Engineering and Design 385, 1–20.

Laribou, H., Elbasset, A., 2021. Numerical study of the thermo-mechanical behavior of 304l stainless steel pipeline junctions. International Journal of Physics 9, 155–168.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444.

LeCun, Y., Bosen, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., 1989. Backpropagation applied to handwritten zip code recognition. Neural Computation 1, 541–551.

Lee, H., Kim, J., Yoo, B., 1999. Assessment of fatigue and fracture on a tee-junction of LMFBR piping under thermal striping phenomenon. Journal of Korean Nuclear Society 30, 267–275.

Li, R., Song, B., Chen, Y., Jin, X., Zhou, D., Han, Z., Cao, Y., 2024. Deep learning reconstruction of high-reynolds-number turbulent flow field around a cylinder based on limited sensors. Ocean Engineering 304, 1–20.

Liberto, M., Ciofalo, M., 2013. A study of turbulent heat transfer in curved pipes by numerical simulation. International Journal of Heat Mass Transfer 59, 112–125.

Linka, K., Schafer, A., Meng, X., Zou, Z., Karniadakis, G.E., Kuhl, E., 2022. Bayesian physics informed neural networks for real-world nonlinear dynamical systems. Computer Methods in Applied Mechanics and Engineering 402, 1–22.

Locatelli, G., Mancini, M., Todeschini, N., 2013. Generation IV nuclear reactors: current status and future prospects. Energy Policy 61, 1503–1520.

Lohner, R., Soto, O., Yang, C., 2003. An adjoint-based design methodology for CFD optimization problems. 41st Aerospace Sciences Meeting and Exhibit , 1–299.

Lumley, J. L., H.P.J., Berkooz, G., 2003. The proper orthogonal decomposition in the analysis of turbulent flows. Annual Review of Fluid Mechanics 25, 539–575.

Lumley, J.L., 1967. The structure of inhomogeneous turbulent flows. Atmospheric turbulence and radio wave propagation , 166–178.

Lumley, J.L., 1981. Coherent structures in turbulence, in: Transition and Turbulence. Academic Press, pp. 215–242.

Lumley, J.L., Payne, F.R., 1967. Large eddy structure of the turbulent wake behind a circular cylinder. The Physics of Fluids 10, 194–196.

Mao, Z., Jagtap, A., Karniadakis, G.E., 2020. Physics-informed neural networks for high-speed flows. Computer Methods in Applied Mechanics and Engineering 360, 1–25.

Maronati, G., Petrovic, B., Wyk, J., Kelley, M., White, C., 2016. Impact of testing activities on small modular reactor total capital investment cost, in: 24th international conference on nuclear engineering ICONE24, 26-30th June, American Society of Mechanical Engineers (ASME), Charlotte, North Carolina, USA. pp. 1–20.

Michalcová, V., Kotrasová, K., 2020. The numerical diffusion effect on the cfd simulation accuracy of velocity and temperature field for the application of sustainable architecture methodology. Sustainability 12.

Mignacca, B., Locatelli, G., 2010. Competitiveness of small-medium, new generation reactors: a comparative study. Journal of Engineering for Gas, Turbines and Power 132, 1–10.

Misyris, G., Venzke, A., Chatzivasileiadis, S., 2019. Physics-informed neural networks for power systems, in: 2020 IEEE Power Energy Society General Meeting (PESGM), 2-6th August, American Society of Mechanical Engineers (ASME), Montreal, Quebec, Canada.. pp. 1–7.

Moin, P., Mahesh, K., 1998. Direct numerical simulation: A tool in turbulence research. Annual Review in Fluid Mechanics 30, 539–578.

Nguyen, A., Yosinski, J., Clune, J., 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images, in: 2015 IEEE Conference on Computer Communications (INFOCOM), 7-12th June, ACM Digital Library, Kowloon, Hong Kong. pp. 1–12.

Nielsen, E.J., Anderson, W.K., 2002. Recent improvements in aerodynamic design optimization on unstructured meshes. American Institute of Aeronautics and Astronautics (AIAA) 40, 1155–1163.

Noorani, A., Schlatter, P., 2016. Swirl-switching phenomenon in turbulent flow through toroidal pipes. International Journal of Heat and Fluid Flow 61, 108–116.

Odemark, Y., Green, T., Angele, K., Westin, J., Alavyoon, F., Lundstrom, S., 2009. High-cycle thermal fatigue in mixing tees: New large-eddy simulations valiyeard against new data obtained by PIV in the vattenfall experiment, in: 17th International Conference on Nuclear Engineering (ICONE17), 12-16th July, The American Society of Mechanical Engineers (ASME) Digital Collection, Brussels, Belgium. pp. 1–14.

Polson, N., Sokolov, V., 2017. Deep learning: a bayesian perspective. Bayesian Analysis 12, 1275–1304.

Raissi, M., Perdikaris, P., Karniadakis, G.E., 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics 378, 686–707.

Rangavajhala, S., Sura, V.S., Hombal, V.K., Mahadevan, S., 2011. Discretization error estimation in multidisciplinary simulations. American Institute of Aeronautics and Astronautics (AIAA) 49, 2673–2683.

Rasamoelina, A., Adjailia, F., Sinčák, P., 2020. A review of activation function for artificial neural network, in: 2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI), 23-25th January, IEEE Xplore, Herlany, Slovakia.

Rasmussen, C.E., Williams, K.I., 2005. Gaussian processes for machine learning. The MIT Press, Massachusetts Institute of Technology, Cambridge, MA, ISBN 026218253X.

Rumelhart, D., Hinton, G., Williams, R., 1986. Learning representations by back-propagating errors. Nature 323, 533–536.

Santis, D., Shams, A., 2018. Calibration of a reference dns to study thermal fatigue effects in a T-junction configuration, in: 27th International Conference Nuclear Energy for New Europe, 10-13th September, Slovenia. pp. 1–11.

Santner, T., Williams, B., Notz, W., 2023. The Design and Analysis of Computer Experiments. First ed., Springer New York, NY, ISBN 978-1-4757-3799-8.

Schmidt, O., Colonius, T., 2020. Guide to spectral proper orthogonal decomposition. American Institute of Aeronautics and Astronautics 58, 1024–1033.

Sharma, N., Patankar, N., 2005. A fast computation technique for the direct numerical simulation of rigid particulate flows. Journal of Computational Physics 205, 439–457.

Sheela, K.G., Deepa, S.N., 2013. Review on methods to fix number of hidden neurons in neural networks. Mathematical Problems in Engineering 2013, 425740.

Sibi, P., Jones, S., Siddarth, P., 2013. Analysis of different activation functions. Journal of Theoretical and Applied Information Technology 47, 1–25.

Sibil, R., 2024. Analysis of open channel flow with various layered vegetation using CFD, considering different near-wall treatment methods. International Journal of Numerical Methods for Heat Fluid Flow 34, 1780–1810.

Siemens Digital Industries Software, Siemens 2023. Simcenter STAR-CCM+, version 2023.1.

Sirovich, L., 1987. Turbulence and the dynamics of coherent structures. Quarterly of applied mathematics 45, 561–571.

Smola, A., Scholkopf, B., 2004. A tutorial on support vector regression. Statistics and Computing 14, 199–222.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A simple way to prevent neural networks from. Journal of Machine Learning Research 15, 1929–1958.

Stock, S., Babazadeh, D., Becker, C., Chatzivasileiadis, S., 2024. Bayesian physics-informed neural networks for system identification of inverter-dominated power systems. Electric Power Systems Research 235, 1–10.

Suryavanshi, A., Tawade, S., 2015. A review on thermal stress in mixing tee. International Journal of Innovative Research in Science, Engineering and Technology 4, 2420–2427.

Sze, V., Chen, Y., Yang, T., 2017. Efficient processing of deep neural networks: A tutorial and survey. Proceedings of the IEEE 105, 2295–2329.

Szegedy, C., Toshev, A., Erhan, D., 2013. Deep neural networks for object detection. Advances in Neural Information Processing Systems 26.

Taheri, S., Julan, E., Tran, X., Robert, N., 2017. Impacts of weld residual stresses and fatigue crack growth threshold on crack arrest under high-cycle thermal fluctuations. Nuclear Engineering and Design 311, 16–27.

Tanaka, M., Myake, Y., 2015. Numerical simulation of thermal striping phenomena in a T-junction piping system for fundamental validation and uncertainty quantification by GCI estimation. Mechanical Engineering Journal 48, 1–25.

Tanaka, M., Ohshima, H., Monji, H., 2010. Thermal mixing in T-Junction piping system related to high-cycle thermal fatigue in structure. Journal of Nuclear Science and Technology 47, 790–801.

Thodberg, H., 1996. A review of bayesian neural networks with an application to near infrared spectroscopy. IEEE transactions on Neural Networks 7, 56–72.

Tunstall, R., Laurence, D., Prosser, R., Skillen, A., 2016. A large eddy simulation of a T-Junction with upstream elbow: The role of dean vortices in thermal fatigue. Applied Thermal Engineering 107, 672–680.

Utama, R., Piekarewicz, J., 2017. Refining mass formulas for astrophysical applications: A bayesian neural network approach. Physical Review C 96, 1–10.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Polosukhin, I., 2017. Attention is all you need, in: 31st Conference on Neural Information Processing Systems (NIPS 2017), 4-9th December, ACM Digital Library, Long Beach, California, USA. pp. 1–14.

Vujicic, T., Matijevic, T., Ljucovic, J., Balota, A., Sevarac, Z., 2016. Comparative analysis of methods for determining number of hidden neurons in artificial neural network, in: Central European Conference on Information and Intelligent Systems, 21-23rd September, ACM Digital Library, Varaždin, Croatia. pp. 1–11.

Wakamatsu, M., Nei, H., Hashiguchi, K., 1995. Attenuation of temperature fluctuations in thermal striping. Journal of Nuclear Science Technology 32, 752–762.

Wong, Y.H., Duan, Y., Lampunio, L., Eaton, M.D., Bluck, M.J., 2025. A proper orthogonal decomposition (POD) and spectral proper orthogonal decomposition (SPOD) study on the effects of different momentum ratios and reynolds number in a T-junction with an upstream elbow. International Journal of Thermofluids 27, 1–20.

Wong, Y.H., Lampunio, L., Duan, Y., Eaton, M.D., Bluck, M.J., 2024a. Assessment of physics-informed neural networks (PINNs) with transfer learning (TL) for. advances in thermal hydarulics, in: Advances in Thermal Hydraulics (ATH 24), 17-21st November, American Nuclear Society, Orlando, Florida, USA. pp. 1–14.

Wong, Y.H., Lampunio, L., Duan, Y., Eaton, M.D., Bluck, M.J., 2024b. Effects of different momentum ratios and reynolds number in a T-junction with an upstream elbow. Nuclear Engineering and Design 428, 1–17.

Xing, C., Wang, H., Xu, Z., 2024. Probabilistic prediction on seismic responses of a long-span high-speed railway bridge using BPINN. International Journal of Structural Stability and Dynamics 24, 1–16.

Xu, S., Sun, Z., Huang, R., Guo, D., Yang, G., Ju, S., 2022. A practical approach to flow field reconstruction with sparse or incomplete data through physics informed neural network. Acta Mechanica Sinica 39, 1–15.

Zhao, Y., Akolekar, H., Weatheritt, J., Michelassi, V., Sandberg, R., 2020. RANS turbulence model development using CFD-driven machine learning. Journal of Computational Physics 411, 1–13.

Zhou, M., 2018. Turbulence Flow Mechanisms to Cause High-Cycle Thermal Fatigue near a Horizontal T-Junction. Ph.D. thesis. Universität Stuttgart.

Zhu, H., Deng, Y., Zhu, R., He, X., 2016-09. Fear of nuclear power? evidence from fukushima nuclear accident and land markets in china. Regional Science and Urban Economics 60, 139–154.