TURBINES FOR FLEXIBLE POWER PLANT OPERATION

Mark Baker

St Edmund Hall

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Michaelmas 2021

Acknowledgements

Personal

Most of all I would like to thank my supervisor, Prof. Budimir Rosic, for his enthusiasm and guidance. It has a been a pleasure returning to Oxford to work with his group in the Thermofluids Institute.

Wesley Ramm deserves particular thanks for his assistance in instrumentation, design and development which supported many of the experimental rigs that contributed data towards this thesis. In addition, I would like to thank Faisal Shaikh, Jamie Saw and Oguzhan Murat for their input and discussion, along with all students and staff at the Institute.

I would like to thank my friends and family for their encouragement, especially my parents, who have always been there for support.

I must highlight the friends at st Edmund Hall and the spirit of the MCR that have made the past four years exceptionally enjoyable. Though I can't mention them all by name, special thanks goes to the cohort of 2017 and the housemates of Kingston Road.

Institutional

This work was part funded by Mitsubishi Heavy Industries. Regular communication with their turbine design team was exceptionally helpful in scoping this project and shaping the research direction.

In particular, thanks to Tanimoto Kouichi, Uchida Sumiu, Ito Eisaku, Kondo Yoshiyuki and Egami Ryo for their feedback and expertise.

Abstract

More than 70 percent of power in the world is generated by gas and steam turbines. Whilst renewables are desirable and continue to provide a growing contribution to the energy portfolio, turbine technology is projected to play an important role in power generation to 2060. The increased capacity from renewables is imposing new challenges and operational requirements on conventional power systems. Traditional designs, optimised for peak performance at constant load, must be adapted for load-levelling flexible operation, accepting more frequent and demanding start-stop cycles.

These challenging operating conditions are driving the need for advanced online diagnostic and monitoring tools. The harsh internal conditions of power turbines mean limited access and data is available to measure the thermal behaviour directly. These restrictions force the need for fast simulation methods to remotely assess the turbine condition. Detail knowledge of the thermal profile, and associated clearances, is essential for optimising transient control without compromising reliability.

Numerical methods for the fast simulation of thermal behaviour in 1D and 3D have been evaluated. New solution methods are presented to support fast 1D modelling of transient heat flow and allow the accuracy of traditional methods to be quantified. A novel hybrid methodology is developed, enabling data from multiple fidelity sources to be combined, thereby bridging the limitations in the independent analyses. New concepts in hybrid data transfer and thermal network modelling are demonstrated in the case of analysing temperatures in a Mitsubishi Heavy Industries steam turbine. A new multi-fidelity thermal analysis software is developed utilising plant measurements, thermal networks, neural networks and simulation data.

Validation cases and future developments are explored, highlighting the potential of the hybrid modelling concept. Demonstrated in the case of thermal analysis for flexible operation of power turbine, the hybrid methods offer new and exciting opportunities for rapid design and online diagnostic monitoring.

Contents

Appendices

List of Figures

Abbreviations

1D, 2D, 3D	One-, two- or three-dimensional (spatial)
Adam	Adaptive moment estimation optimiser
AI	Artificial Inteligence
ANN	Artificial Neural Network
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
ASTM	American Society for Testing and Materials
CAD	Computer Aided Design
CAGR	Compound Annual Growth Rate
CCS	Carbon Capture Sequestration
CFD	Computational Fluid Dynamics
CHT	Conjugate Heat Transfer
CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture
EAB	Element Array Buffer
FEA	Finite Element Analysis
FEM	Finite Element Model
GPU	Graphics Processing Unit
GTCC	Gas Turbine Combined Cycle
HP	High Pressure
HTC	Heat Transfer Coefficient
IP	Intermediate Pressure
IR	Infra-red
LHS	Left Hand Side
LP	Low Pressure

LSTM	Long Short Term Memory
LTI	Linear Time Invariant
MHI	Mitsubishi Heavy Industries
MSE	Mean Square Error
NARX	Nonlinear AutoRegressive eXogeneous
NGV	Nozzle Guide Vane
OEM	Original Equipment Manufacturer
OTI	Oxford Thermofluids Institute
PCA	Principal Component Analysis
ReLU	Rectified Linear Unit
RHS	Right Hand side
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
S2F	Solid to Fluid
S2S	Solid to Solid
SGD	Stochastic Gradient Descent optimiser
SLA	Stereolithography 3D printed plastic material
STL	Stereolithography file type
TNM	Thermal Network Model
TNMS	Thermal Network Model Software
VBO	Vertex Buffer Object

Nomenclature

Chapter 2: Literature Review

c_p	Specific heat capacity $[J/kgK]$
c_p^*	Modified specific heat capacity $[J/kgK]$
t	Solid domain time scale $[s]$
t^*	Modified solid domain time scale $[s]$
SF	Modification scale factor

Chapter 3: 1D Heat Transfer Analysis

x, y, z	Cartesian spatial dimensions $[m]$
t	Time $[s]$
s	Laplace variable
f_x	Sampling frequency of signal x $[Hz]$
T	Temperature, relative to initial ambient, in the time domain $[K]$
T^*	Modified temperature, in the time domain $[K]$
q	Heat flux, in the time domain $[W/m^2]$
q^*	Modified heat flux, in the time domain $[W/m^2]$
W	Thermal driving function, in the time domain $[K]$
ϕ	Temperature, in the Laplace domain $[K]$
ψ	Heat flux, in the Laplace domain $[W/m^2]$
Ω	Thermal driving function, in the Laplace domain $[K]$
F_{XY}	Laplace transfer function that maps X $\rightarrow$Y
δ	Dirac delta function
h	Heat transfer coefficient $[W/m^2K]$
α	Thermal diffusivity $[m^2/s]$
k	Thermal conductivity $[W/mK]$
ρ	Density $[kg/m^3]$

c	Specific heat capacity $[J/kgK]$
$\sqrt{\rho c k}$	Thermal product, $k/\sqrt{\alpha}$
λ	Laplace exponent, $\lambda = \sqrt{s/\alpha}$

Chapter 4: Multilayer 1D Heat Transfer Analysis

i, j, k	Integer indices
x, y, z	Cartesian spatial dimensions $[m]$
a	Layer thickness $[m]$
t	Time $[s]$
s	Laplace variable
T	Temperature, relative to initial ambient, in the time domain $[K]$
q	Heat flux, in the time domain $[W/m^2]$
ϕ	Temperature, in the Laplace domain $[K]$
ψ	Heat flux, in the Laplace domain $[W/m^2]$
F_{XY}	Laplace transfer function that maps X →Y
δ	Dirac delta function
h_x	Impulse response filter for the x layer model
α	Thermal diffusivity $[m^2/s]$
k	Thermal conductivity $[W/mK]$
ρ	Density $[kg/m^3]$
c	Specific heat capacity $[J/kgK]$
$\sqrt{\rho c k}$	Thermal product, $k/\sqrt{\alpha}$

Chapter 5: Cylindrical 1D Heat Transfer Analysis

x, y, z	Cartesian spatial dimensions $[m]$
r, θ, z	Cylindrical spatial dimensions $[m]$
t	Time $[s]$
s	Laplace variable
T	Temperature, relative to initial ambient, in the time domain $[K]$
q	Heat flux, in the time domain $[W/m^2]$
ϕ	Temperature, in the Laplace domain $[K]$
ψ	Heat flux, in the Laplace domain $[W/m^2]$

α	Thermal diffusivity $[m^2/s]$
k	Thermal conductivity $[W/mK]$
ρ	Density $[kg/m^3]$
c	Specific heat capacity $[J/kgK]$
n_i	Thermal product, $\sqrt{\rho_i c_i k_i} = k_i/\sqrt{\alpha_i}$
λ_i	Laplace exponent, $\sqrt{s/\alpha_i}$
$\sigma_{m,n}$	Laminate layer ratio $k_n\lambda_n/k_m\lambda_m$
$\gamma_{m,n}$	Laminate layer ratio $(1-\sigma_{m,n})/(1+\sigma_{m,n})$

Chapter 6: Neural Network Thermal Prediction

t	Time $[s]$
E	Error $[K]$
T	Temperature, relative to initial ambient $[K]$

Chapter 7: Hybrid Fidelity Geometry Representation, ...

X, Y, Z	Global Cartesian spatial dimensions $[m]$
x, y, z	Local Cartesian spatial dimensions $[m]$
r, θ, z	Local Cylindrical spatial dimensions $[m]$
t	Time $[s]$
E	Error $[K]$
T	Temperature, relative to initial ambient $[K]$
p_{ij}	Element (i, j) of matrix P
n_x	Number of elements in vector x

Chapter 8: TNMS, Development and Structure

i, j, k	Integer indices
x, y, z	Cartesian spatial dimensions $[m]$
u, v, w	Cartesian normal dimensions $[m]$
R, G, B	Red, Green, Blue 16-bit colour components [0-255]
f_i	Scale factor on index i
t	Time $[s]$
T	Temperature, relative to initial ambient $[K]$
q	Heat flux $[W/m^2]$

α	Thermal diffusivity $[m^2/s]$
k	Thermal conductivity $[W/mK]$
ρ	Density $[kg/m^3]$
c_p	Specific heat capacity $[J/kgK]$
h	Heat transfer coefficient $[W/m^2K]$
$\dot{m}$	Mass flow rate $[kg/s]$
L	Length $[m]$
A	Area $[m^2]$
V	Volume $[m^3]$
R	Thermal resistance $[K/W]$
C	Thermal capacitance $[J/K]$
$M_{c,v}$	Boolean mask heat transfer mode for conduction (c) or convection (v)
σ	StefanBoltzmann constant $\sigma = 5.670374419... \times 10^8 W m^2 K^4$
ϵ	Material radiation emissivity
F_{ij}	Radiation view factor between facets i and j
θ	Angle between vectors $[radians]$
τ_x	Time constant of simulation domain x
M_F	Fluid domain simulation matrix
M_S	Solid domain simulation matrix
S_F	Fluid domain radiation matrix
R_F	Decoupled boundary radiation heat flux tracking matrix
Q_F	Decoupled boundary convection heat flux tracking matrix
Q_S	Decoupled boundary heat flux solid summation matrix

Chapter 9: TNMS, Validation and Future Application

t	Time $[s]$
T	Temperature, relative to initial ambient $[K]$
q	Heat flux $[W/m^2]$
E	Error $[K]$
$y*$	Non-dimensional wall distance by turbulent kinetic energy
$u*$	Non-dimensional velocity by turbulent kinetic energy
x_i	neural network input i
h_i	neural network recurrent weight i
w_i	neural network output weight i
b_i	neural network bias i

1

Introduction

Demand for electricity is predicted to double by 2060 from $23.8TWh$ to $48.5TWh$ [1]. Traditionally rapid growth has been satisfied by fossil fuels however a drive towards clean and sustainable energy means the future must be different. The World Energy Council predicts a fundamental change in the energy industry, *The Grand Transition*, as focus shifts towards renewable sources.

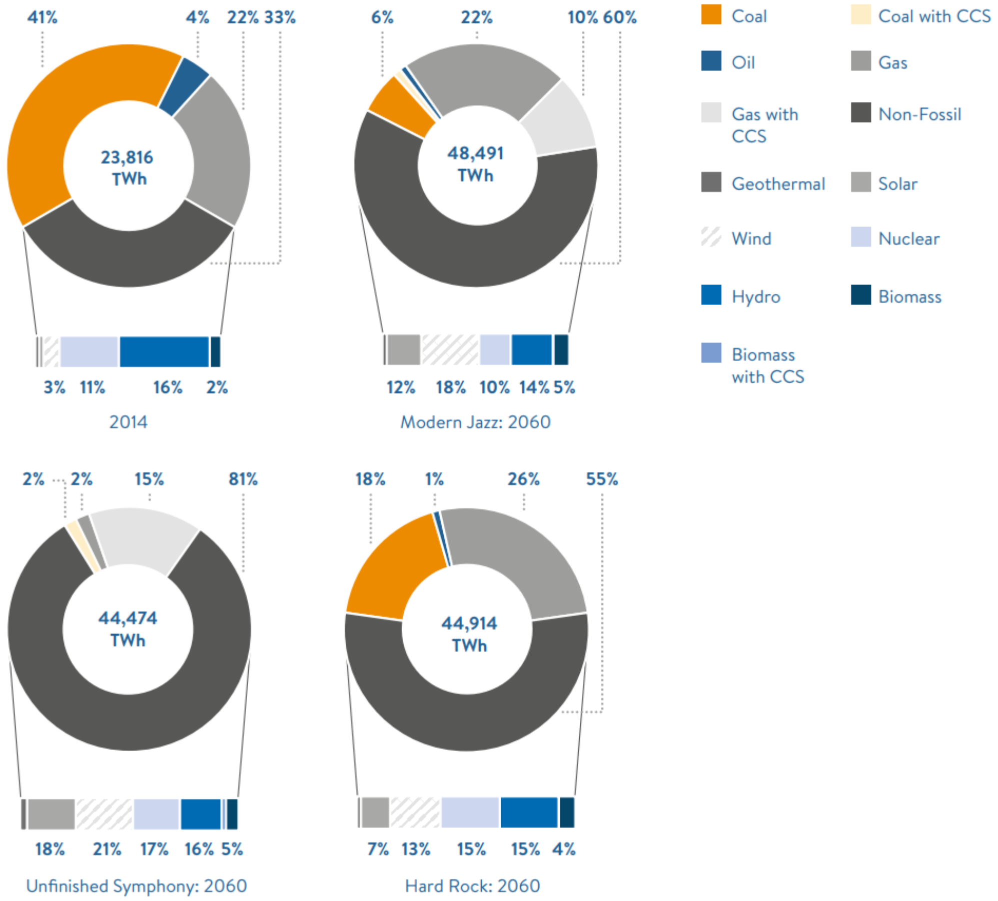


Figure 1.1: Forecast 2060 global electricity generation by resource and uptake of renewable energy [1].

The market dependence on future economic and political uncertainties leads to a broad range of predicted eventualities in 2060. Three proposed scenarios: *Modern Jazz*, *Unfinished Symphony* and *Hard Rock* cover the extremes of renewable uptake depending on the influence of geopolitical and technological advances. Other sources propose similar outcomes, generally predicting a mid-range uptake of renewables and a continued dependence on natural gas [2] [3].

1.1 The Challenge of Renewables

Investment in solar and wind technologies has seen steady growth in the installed capacity. UK renewable capacity increased to 40.5GW at the end of 2017 Q4, Figure 1.2, an increase of 13% on the previous year [4]. Higher capacity does not always translate to increased generation. Figure 1.3 shows dependence on unreliable environmental factors leads to unpredictability of renewable supply. Despite investment in 2016 Q4, UK renewable electricity generation fell 3.1TWh on the previous year due to lower average wind speeds affecting offshore wind (-23%) and onshore wind (-13%) [5].

Renewable load factors demonstrate strong seasonal effects which further complicate grid load balancing. Uptake of renewable energy is desirable; managing the unreliability is the key challenge for the future. Gas and steam turbines, along with advances in energy storage, are expected to stabilise the supply with load-levelling capacity. Research into the design and control of power turbines under this new application is essential to support the successful renewable transition.

Alongside load-levelling turbines, advances in energy storage will also play a vital role in balancing grid supply. Commercial 300MW battery installations have been demonstrated and will likely become common renewable support infrastructure. Large scale lithium-ion facilities are still a development technology and further work is required to understand and manage the risks [6][7][8]. Utilising load-levelling turbines has the advantage that much of the infrastructure already exists and they are more suitable for long duration operation and outage.

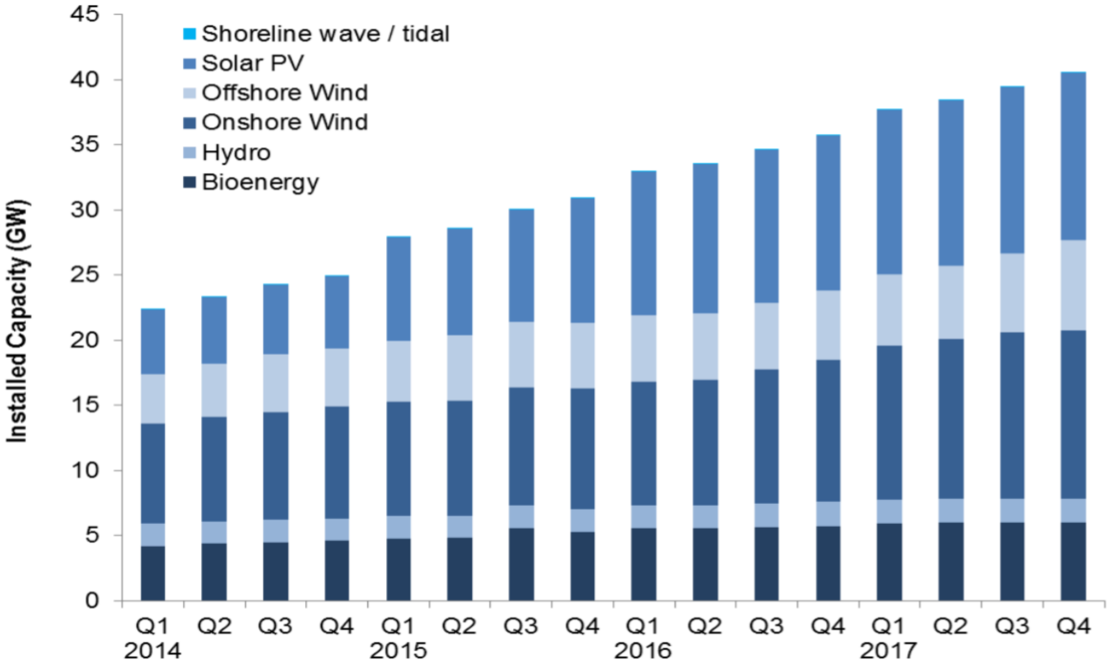


Figure 1.2: UK installed renewable capacity by energy source 2014 - 2017, showing the increased investment in renewable energy infrastructure [5].

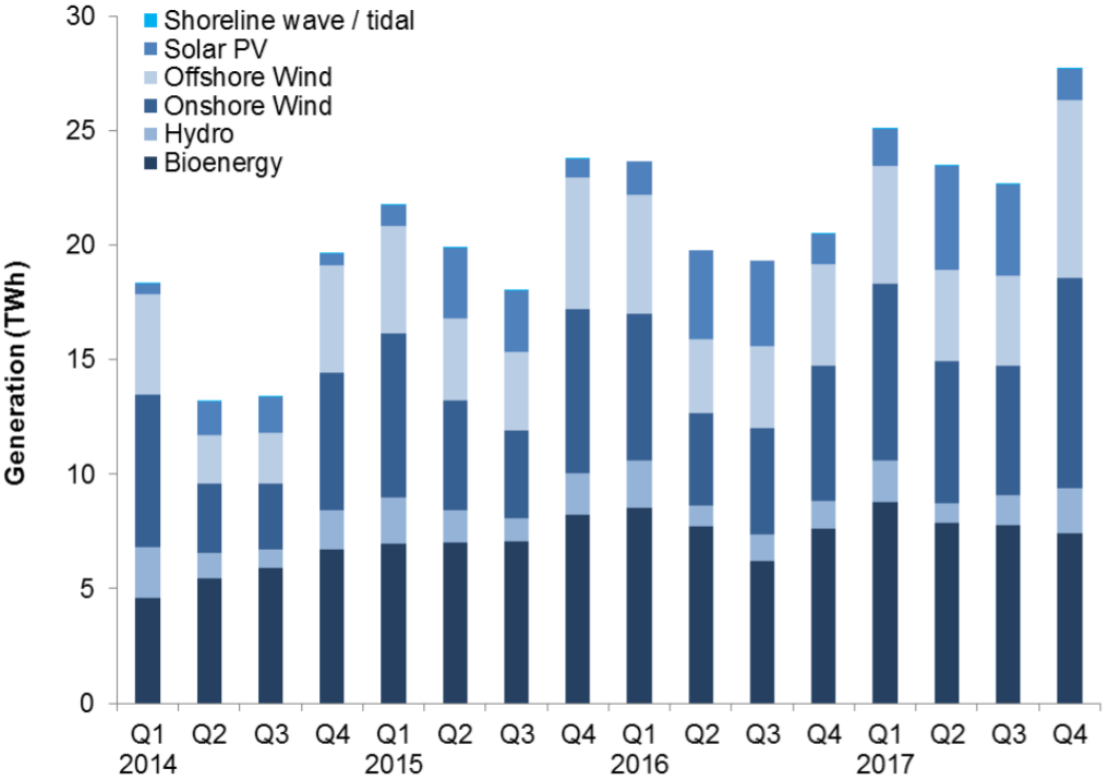


Figure 1.3: UK renewable generation by energy source 2014 - 2017, showing the disconnect between capacity and generation [5].

1.2 Future Operation of Power Turbines

Dependence on coal and oil is projected to be removed by 2060. Turbine technology will continue to play an important role in combined cycle gas, biomass and concentrated solar plants. The global steam turbine market is poised to grow at a CAGR of approximately 5.8% over the next decade and reach approximately $27.15 billion by 2025 [9]. The significant market change comes in the operational usage of the technology. Voelker stated that *"future operational profiles will be characterised by flexibility: fast start up times, with unlimited load changes while in operation. Steam units are not making the baseload as in the past"* [10]. Alongside these operational changes, demand for high thermal efficiency will continue to boost steam temperatures and pressures. Ultra-supercritical power plants now operate over 600^oC and manufacturers expect the need to push towards 650^oC inlet temperatures and longer term 700^oC [10]. The drive to higher temperatures, with increased flexibility, places high importance on understanding the thermal transients in these machines. Detailed knowledge of the thermal behaviour is needed to accurately control the component stresses, relative expansion and running clearances.

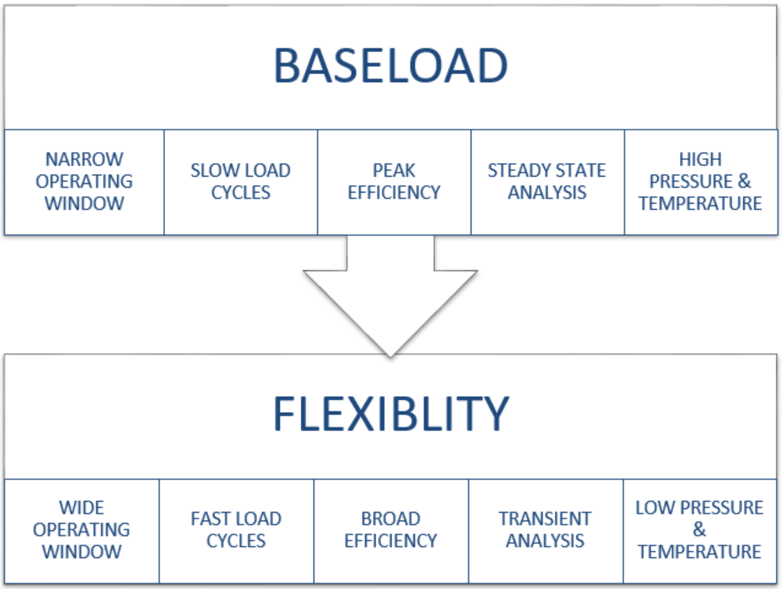


Figure 1.4: The change in design philosophy and requirements from traditional baseload to future load-levelling flexibility.

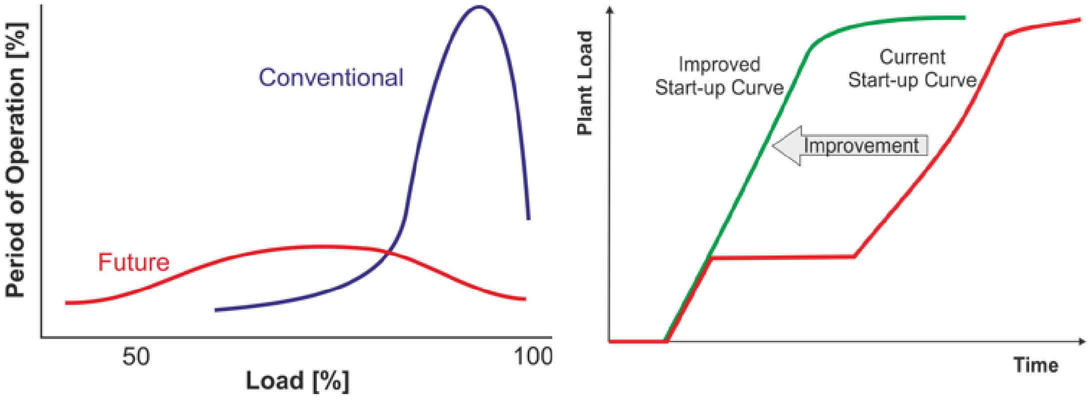


Figure 1.5: Expected future operating regime of load-levelling flexible turbines, showing the change in operational load and start-up requirements.

1.3 Turbine Start, Operation and Cooling Cycles

Traditional turbine starts are characterised as one of three types: cold-start, warm-start or hot-start. This allocation depends on the temperature of the High Pressure (HP) and Intermediate Pressure (IP) inner casings at the point of start-up. The thick wall casings retain heat from the previous operating cycle and may be either naturally or force cooled. Depending on the time since the previous shutdown, the temperature of the casings very significantly. The start-up procedure and associated start-up time is governed by the required pre-warming of the rotor and casing. These categories are traditionally defined by the following approximate temperature ranges [11].

- cold-start, HP and IP inner casing temperature is < 170 °C,
- warm-start, HP and IP inner casing temperature is < 430 °C,
- hot-start, HP and IP inner casing temperature is > 430 °C,

Nowak et al.[12] demonstrated that even within these categories there can be substantially different temperature distributions within the casing. Understanding how best to restart the turbine therefore requires a detail knowledge of the previous cooling procedure. Thus, it is highly advantageous to have fast thermal modelling tools or monitoring systems that are able to rapidly assess the thermal state of the machine. They showed that even after more than 100 hours of natural cooling, a uniform temperature field is not established within the rotor. Thermal differences across the shaft can still be in excess of 80°C and significant pre-stresses exist when steam is first fed to the turbine. This leads to a peak in rotor stress shortly after the initial feed. A second stress maxima occurs directly after the temperature levelling phase where the turbine is held for a long duration at constant steam input to achieve thermal stabilisation. This period is identified as a key area to reduce overall start-up time.

Traditionally, turbines are cooled naturally and these machines may take several days to reach safe handling temperature. In the case of planned maintenance, this delay causes a loss of productivity for the plant. Many turbine manufacturers now offer forced cooling systems, able to accelerate this process and reduce cool-down times up to 50% [13]. These systems however introduce additional transient cases that must be analysed and evaluated.

It should be noted that the gas and steam turbines in a combined cycle plant are not isolated components. Thermal stabilisation within the turbine is not the only factor for consideration in full plant operation. It is important to note that not all improvements in the turbine start-up will directly translate to the plant performance. Particular consideration is required to the boiler operation which falls outside the scope of this thesis. Nevertheless, a clear requirement exists for fast thermal modelling tools, capable of analysing a range of geometries. These need to handle long duration transient operation and support simulation over many hours.

1.4 Thesis Objectives

The market shift towards a wider portfolio of renewable energy increases the demand for flexible operation of both gas and steam turbines. Increasing focus is now placed on wider operating windows, load variability and fast-start rather than peak efficiency. Existing studies have focused on using commercial conjugate heat transfer methods in order to solve this problem. Whilst these tools provide good accuracy, simulations may take many hours or days and prevent real-time support for online monitoring.

Demand for flexibility presents a clear need for fast, approximate, transient heat analysis methods to support the design and operation of turbines. Existing monitoring systems track a low number of measurements and do not fully capture the thermal effects in the machine. It was set out to evaluate calculation methods suitable for supporting rapid development and online monitoring. Traditional numerical schemes compromise accuracy or resolution to achieve speed. Rather than simply accept these inaccuracies, it is proposed to evaluate a new data driven hybrid methodology.

Developed in this thesis, the hybrid concept will be used to directly couple real-time measurements with neural networks and low order simulations. The simultaneous use of numerical and measurement strategies allows their individual limitations to be bridged. Live data and neural networks prevent divergence whilst numerical methods enhance spatial resolution. Focus is given to developing all components of the hybrid system and enabling their integration. Each part is validated separately on a single test case to assess its suitability for the overarching concept. Heat transfer methods supporting both 1D and 3D across a range of fidelities will be evaluated to define the hybrid calculation strategy.

1.5 Thesis Overview

It was set out to evaluate fast, data-driven, transient heat analysis methods capable of supporting a wide fidelity range. A new hybrid methodology is presented to facilitate analysis and monitoring by combining: empirical studies, physical measurements, neural networks and low order numerical schemes. The components of the hybrid methodology have been developed, then individually tested in the analysis of a Mitsubishi Heavy Industries turbine.

Chapters 1 and 2. An introduction to the changing energy market is provided with a review of the relevant works completed in assessing the transient behaviour of power turbines. A short discussion on different numerical modelling techniques is included, along with a review of the experimental works on full size operational machines.

Chapter 3. The impulse response method, widely used for the fast post-processing of heat transfer data, is examined. The chapter provides a rigorous evaluation and better defines the assumptions, limits and accuracy of 1D semi-infinite analysis of convection boundaries. Improved post-processing methods are presented in the case of handling transient boundary conditions to experimentally determine the heat transfer coefficient for input to low order simulations.

Chapter 4. Analysis methods for heat transfer in multilayer laminate materials are evaluated. The example of a laboratory thin film sensor on a turbine nozzle guide vane is presented, along with upgraded impulse basis functions to handle multilayer systems. A one-dimensional numerical solver is developed, allowing fast heat transfer analysis whilst conserving flux at the laminate interfaces. The solver is used to validate the upgraded impulse method and will provide support for embedded heat flux measurements. High temperature heat flux sensors are critical to the future development of turbine monitoring systems.

Chapter 5. The capability of both the impulse and 1D numerical methods are extended. Traditional application of these tools is limited to a planer case and the effects of curvature are generally ignored. Response functions for a cylindrical system are presented and the error associated with the planar assumption evaluated. The method is validated in the case of analysing heat flux on a turbine nozzle guide vane. A new post-processing method is presented with improved accuracy, allowing better resolution of real turbine features and assessment of when curvature effects should not be ignored.

Chapter 6. Truncated time history is introduced by exploring the suitability of neural networks for thermal prediction. Conceptually similar to the impulse response, Long Short Term Memory (LSTM) neural networks make predictions based on weighted historical values of the input. A multi-input, multi-output, network is developed to track the thermal profile of a turbine during transient hot-start conditions. Data from a full scale operational machine, provided by Mitsubishi Heavy Industries, is used to train and validate the network. Pre- and post-processing methods are presented to improve the performance of the predictor-corrector network structure.

Chapter 7. The multi-fidelity hybrid concept is defined. Taken independently, experimental or numerical tools often have limitations in access, accuracy or spatial resolution. Hybrid calculations create a simultaneous portfolio of data, coupling both numerical and experimental resources. The combined fidelity aims to bridge the limitations in any one individual method. Supplementary calculation tools are presented to facilitate geometry and data mapping across the fidelity range. Implemented in Python and Tensorflow, these tools directly interface with the LSTM and numerical methods in earlier chapters.

Chapter 8. The hybrid functionality is extended by adding support for a three-dimensional thermal modelling tool: Thermal Network Model Software (TNMS). The novel mesh driven, lumped mass, electrical equivalence method allows a low order conjugate thermal model to be automatically built from experimental data, CAD and CFD results. The chapter outlines new concepts for automated blocking and lumped thermal property calculation. The GPU solver and post-processing algorithm are defined; written in Tensorflow, these allow native coupling to the neural network methods.

Chapter 9. The Thermal Network Model Software is validated against known analytic solutions and a commercial heat analysis package. Test cases in both blocked and full mesh mode are considered. A conjugate model of a full machine is tested to analyse the heat transfer in a Mitsubishi Heavy Industries steam turbine. An overview of the preliminary studies that branch from this thesis are discussed, including recommendations for future research and development.

The thesis closes with concluding remarks of the developed methodologies, including comments and suggestions of how best to apply these methods to the analysis of flexible operation of power turbines.

2

Literature Review

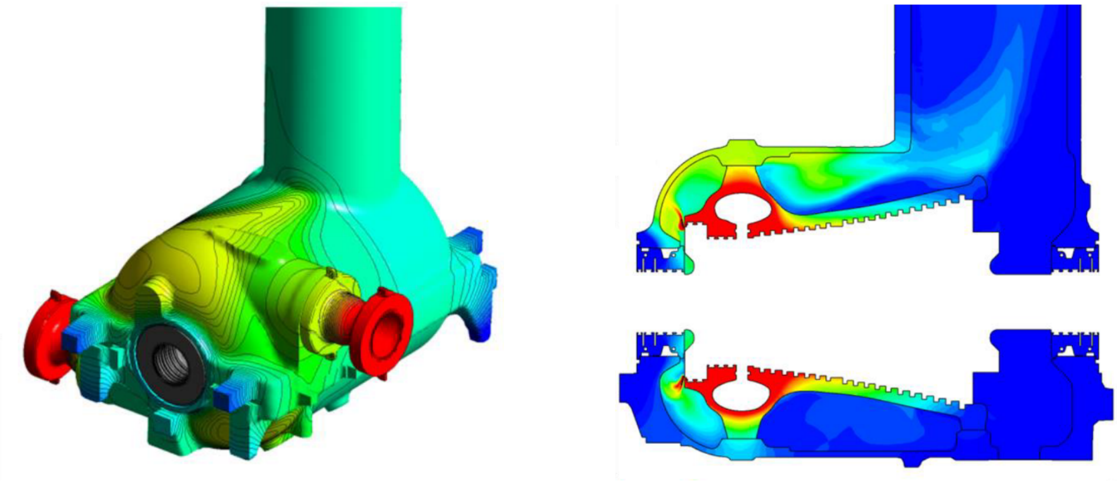
2.1 Market Specification

The market transition towards flexibility is well documented within industry. There is increasing commercial focus in this area with evermore publications regarding transient operation. Della Villa [14] translates the broad desire for transient operation into a detailed understanding of the market need; flexible operation is defined as the ability to *"start-up and run when renewable energy systems cant provide the load, and shutdown safely when they can"*. This equates to start-up times less than 10 minutes at ramp rates of 50MW per minute. The market expectation is fast-start, load-following, safe shutdown turbines that achieve high efficiency and low emissions across the load profile. The desire for flexibility does not detract from existing market Key Performance Indicators (KPI) and a premium is placed on mitigating uncertainty and operational risk. Thus, reliability and durability remain the highest importance. Increased online monitoring has seen the market tolerate lower levels of product validation. However, flexibility is not accepted at the cost of reliability and new systems must undergo extensive testing.

Greis [15] conducts a similar market study, analysing the load profile caused by fluctuating energy consumption in a growing population. The variable load pattern introduces new challenges in managing thermal stress. Greis reinforces the need to account for durability; critically the low cycle fatigue in thick-walled components. Thus, traditional starts with constant steam temperature ramp are being replaced. New constant life consumption starts are preferred and industrial focus is given to variable controlled temperature rate for thermal stress mitigation.

2.2 Conjugate Heat Transfer Methods

Conjugate Heat Transfer (CHT) methods consist of a coupled solution between the fluid convection and solid conduction heat transfer. These methods work well for steady state flow in the fluid domain. With unsteady flow, there is a large time scale disparity between the fluid convection and solid conduction which leads to extensive computational effort.



3D view of the outer casing

2D section view of the cavity flow

Figure 2.1: Steady state analysis of KA26-1 Combined Cycle 460MW turbine [16]

2.2.1 Unsteady Timescale Problem

The issue of time scale disparity is well described by He [17]. Fluid velocities inside the turbine can exist in the region of 300ms^{-1}. For a blade row with axial dimension of 0.1m, the fluid structures in the machine vary in the order of 0.0003 seconds. Temperatures in the solid change at a significantly slower rate with variations in the order of seconds. This gives a time scale disparity between the fluid and solid domain in the order of 10^4. He compares the stability requirement for a simple forward time 1D convection and conduction analysis with equivalent discretisation. Despite the difference in the rate of conduction and convection, the number of time steps for convergence is comparable. Decoupled, the computational effort to independently solve the two domains is the same. Once coupled, the mesh discretisation and time step must be sufficient to resolve the fluid, but the total time sufficient to ensure consistency with the solid. This requires a computational effort in the order of 10^4 compared to a decoupled approach. Approximations or empirical methods are often employed which sacrifice the direct coupling of the two domains to reduce the computational effort; these methods are further discussed below.

2.2.2 Equalised Timescale

The coupled approach seeks to achieve continuity of the temperature field and flux across the state boundary. The interface is defined by solving the energy equation and the time scales modified to equalise the behaviour in the two domains. The time scale in the solid is dependent on the material specific heat capacity and density. In this method, a pseudo specific heat capacity, ${c_\mathrm{p}}^*$, is introduced which is factored by SF to give an artificially reduced transient time scale in the solid, t^*.

$$c_p^* = c_p/SF \qquad\qquad t^* = t/SF \tag{2.1}$$

By introducing this factor on the material properties, conduction in the solid can be modelling in the same time frame as convection in the fluid. Care needs to be taken when factoring the solid properties and over factoring leads to inaccuracies or non-physical results [18]. This method is most suited for solving the steady state solution and is less applicable to the transient modelling due to the factored solid domain behaviour.

2.2.3 Decoupled Iterative Boundary

In this approach the fluid and solid domains are solved independently. An unsteady CFD simulation is used to approximate the surface heat transfer coefficient which then defines a boundary condition in the steady state FE analysis. The approach assumes constant heat transfer coefficient on the interface between the two domains which limits the accuracy of this method [18]. For improved results, the two simulations can be periodically aligned; transferring the boundary temperatures between the two in a stepwise approach. The simulation time is vastly reduced by decoupling the simulations but at the expense of accurate representation of the changing surface heat flux. The most commonly adopted method is that recommended by Giles [19], to use a Dirichlet boundary condition on the fluid and Neumann boundary condition on the solid domain.

2.2.4 Harmonic Frequency Domain

He [17] addresses the time scale mismatch in periodic unsteady problems by solving conduction in the frequency domain. This allowed steady flow acceleration techniques to be employed, removing the need for time integration in the slow reacting solid. The unsteady temperature is decomposed into a time mean and Nth order Fourier harmonic signal. Taking time derivatives and balancing the sine and cosine terms, one unsteady solution becomes equivalent to $2N$ steady state solutions. A high level of accuracy was achieved with this method. He demonstrated the need for additional meshing considerations when resolving high frequency flows. Implementation of this approach requires modification to the baseline steady state conduction solver. Not yet implemented commercially, use of this advanced technique is limited to custom codes.

2.3 Thermal Stress Modelling

Monitoring lifetime consumption has seen growing interest in thermal stress modelling during start-up and warm keeping. Analysis of the start-up stresses requires knowledge of the full temperature profile. The rotor temperature is critical for determining the limiting ramp rate and is highly dependent on the steam condition. Chmielniak [20] presented an empirical approximation of the Heat Transfer Coefficient (HTC) based on Prandtl and Nusselt number in differing regions of the machine. The corresponding rotor stresses, at intervals between 20% and 100% nominal load, were calculated and successfully used to validate a new configuration.

Beer, Propp and Voelker [21] employed a one-dimensional analytical model of the heat equation. A maximum allowed temperature step was defined by a permitted thermal strain limit. Boundary conditions were specified by quadratic interpolation and a simple constant value assumed for thermal conductivity. The temperature difference was calculated at the component neutral axis, modelling the casing as a one-dimensional plate and rotor as a one-dimensional cylinder. Compensation was introduced for the casing boreholes and performed well when compared to a finite element model. Both approaches demonstrate the value of a simplified analytical/empirical model in the evaluation of thermal stress. The basic interpolation methods have significant assumptions in the heat flux but were able to provide a valuable result.

2.4 Turbine Heat Transfer Studies

Several commercial and research facilities have undertaken Conjugate Heat Transfer (CHT) studies to characterise the thermal performance of turbines. By combining both the fluid and solid domain in a single analysis, a conjugate model gives better resolution of the fluid-structure thermal interface. This method removes the requirement for adaptive boundary conditions at the significant cost of computational effort. Many of the referenced studies utilise loosely-coupled conjugate models where information is only periodically transferred between the two domains. Piotr [18] and Toebben [22] employed a Hybrid-FEM to evaluate the hot air warm keeping of a steam turbine. Warm keeping maintains the turbine at a higher thermal equilibrium, allowing faster ramp to nominal load conditions without establishing large thermal gradients in the rotor. The method introduced by Toebben integrates an analytic Nusselt number correction to a simplified fluid model for heat transfer. Combined with a finite element model, this method showed comparable results to a full unsteady CHT ANSYS simulation, without the extensive calculation requirement.

Brilliant and Tolpadi [23] investigated the design of a typical General Electric (GE) high pressure turbine using a physics-based ANSYS model. The rotor, diaphragm and shells were investigated and the computation results compared to field temperature measurements. Steady state CFD with proprietary performance codes defined the thermal boundary conditions for ANSYS. This approach worked well for loaded operation but the method requires well defined flow in the cavity. Analysis at low flow conditions, common in transient start-up and shut down, were not well resolved and the accuracy of this approach deteriorated when thermal, rather than pressure gradients, define the flow.

Mukhopadhyay [24] extended the work by Brilliant and analysed the shell deflection in a GE representative design. Shell deflection is a well-known phenomenon and is caused by the relative slower rate of cooling in the upper half. This is mostly due to natural convention in the cavity but also comes from differing behaviour of the insulation on the two casing halves. The temperature difference is reinforced by the asymmetric mass distribution in the upper and lower half casing. A conjugate CFD model was presented using PATRAN P-Thermal tool and GEs in house code YFT. The simulation could predict the normalised temperature delta between the top and bottom half casings within 10% under steady state conditions.

Spelling and Jocker [25] analysed a SST-700RH Solar 50 MW reheat system comprised of high-pressure and low-pressure turbine units. A 2D axisymmetric, three-part analytical model was created consisting of: finite volume heat condition, steam thermodynamics and a gland steam network model. Results were compared to 96hrs of operational data and achieved average error below 5% with peak error (7.5%) occurring during main start-up transients. Potential design improvements to maintain turbine temperature during idle periods were analysed. The authors showed that additional insulation has little effect but heater blankets provide a low cost effective method to maintain the casing temperature. Neither of these methods had significant impact on the rotor which could only be heated by changing the gland steam admission.

Topel expanded the work, introducing a modular geometry solution [26], allowing an approximate and fast prediction of the thermal behaviour of different steam turbines. She showed that despite the low level of detail, a large degree of agreement was achieved. This work was then applied to the flexible optimisation of concentrating solar power plants [27]. The combined MATLAB COMSOL model was used to experiment with different start-up curves and assess the thermal stress. The research showed that start-up improvements of 9.5% could be achieved, using the back-pressure and rotational speed to control the casing temperature and gland system to affect the rotor.

Prior to the acquisition by GE, Alstom were highly active in the analysis of transient effects. A large study was completed, comprising an experimental set-up and several numerical papers on heat transfer in the KA26-1. Initially, Mohr and Ruffino [28] developed high temperature optical probes in their experimental analysis of the turbine. Above 230^oC, these flexible pyrometer probes have ± 1.5 degree accuracy at 100kHz sample frequency. However, they deteriorate rapidly at lower temperature with no useful data provided below 130^oC. The probe data was combined with a limited number of thermocouple measurements to record the temperature during natural cooling. A novel 3D-2D equivalence model was introduced. The circumferentially non-uniform geometry was converted to an equivalent 2D axisymmetric design. A modified thickness parameter was introduced to give representative thermal mass. A 2D finite element model was applied to numerically solve for natural cooling. An accuracy of ±15°C was achieved in the prediction of the thermal state during shutdown. The axisymmetric approach demonstrated good general performance at significantly lower computational effort.

Following on from Mohr and Ruffino, Marinescu et al. [29] further evaluated the thermal 2D equivalence model. Additional improvements in modelling steam ingestion were included using an Alstom and Rolls-Royce plug-in for thermodynamic properties. The position of the steam jet was shown to have little impact but the fluid buoyancy proved to be critical in calculating the cavity temperature. Similar to Mukhopadhyay the authors emphasise the importance of natural convection in the turbine cavity.

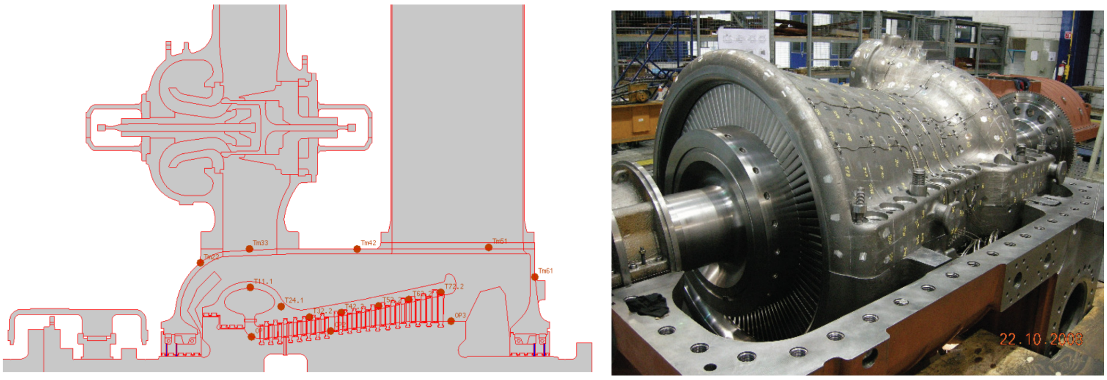
The key contribution from the work is the introduction of the over-conductivity method, used to capture the effects of local buoyancy in the cavity regions. The fluid thermal conductivity is replaced by a conductivity function, which scales the thermal conductivity of air. The $K(T)$ correction function has a quadratic temperature relation, $K(T) = a_1T^2 + a_2T + a_3$, with the coefficients (a_1, a_2, a_3) optimised to fit experimental observations. The fluid conductivity factor worked well when pressure gradients were negligible. The method failed to accurately model the behaviour of forced convection and becomes fundamentally challenged when modelling high velocity flows. This is the opposite issue encountered by Brilliant in his steady state CFD analysis. These works demonstrate that a complete numerical analysis of the turbine cavity requires a solver capable of resolving both pressure and temperature driven flow regimes.

Extending his work, Marinescu [30] included an upgraded model of gland flow, accounting for the changing flow direction as the relative pressures between the cavity and gland system vary. High loading was identified in the hot thick-walled components of the turbine rotor, valves and casings. The HP rotor was shown to be the most critical component. Stress raisers, such as blade grooves, concentrate the thermal stress and therefore the first blade groove of the HP rotor was identified as a critical point.

Marinescu concludes the research by validating the over-conductivity function on two additional turbine geometries [31]. The analysis was applied to the Alstom HP5X turbine (1100MW) and Alstom KA24-1 turbine (450MW). The results for all three geometries displayed similar accuracy, with temperature prediction in the range ±20°C after 8 and 60 hours of natural cooling. The similarity in the scatter for the calibration and test machines suggests the approach translates well across different designs. The author attributed this to a dependence only on the steam properties and temperature gradient rather than specific geometric features.

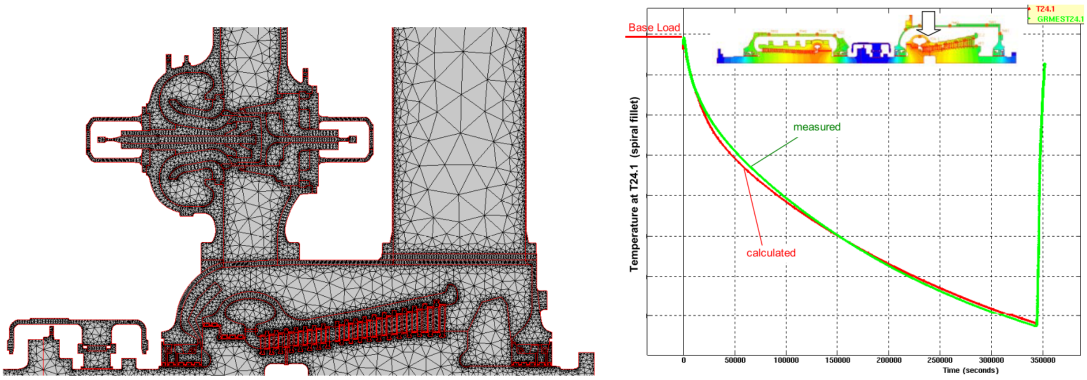
Born and Koch [16] continued research into the KA26-1 Combined Cycle 460MW steam turbine and validated a 3D numerical method to analyse the temperature and thermal stress. A steady state, baseload, multi-zone CHT was completed using ANSYS CFX. Gland and leakage flows were not included in the model and were replaced with an equivalent HTC. The cavity flow was not fully resolved in these regions. The steady state results were compared with the collated operational data and showed good overall agreement.

Transient behaviour was investigated but due to the differing time scales of the fluid and solid domain, a full 3D CHT model was not used. Instead, a 2D FE solver was adopted, modelling the fluid as a solid and factoring the conductivity. This method showed reasonable agreement however, scope exists to improve the analysis using a complete 3D study with resolved gland flows.



Location of experimental measurement probes

KA26-1 Combined Cycle 460MW



2D axisymmetric mesh used for simulation

Example calculation validation

Figure 2.2: Summary of work completed on the Alstom KA26-1 turbine, showing the measurement and configuration of the physical test facility, the axisymmetic model used for simulation and a comparison of the model's output to a validation temperature probe.

2.5 Turbine Real-Time Models

Artificial Neural Networks (ANN) are increasingly used to model the behaviour of complex systems. Provided with large data sets, these networks can self-characterise the design and provide accurate, fast approximations to more complex calculation techniques.

Kosowski [32] applied a three layer ANN to the design of a steam turbine cascade. CFD calculations were performed for over 600 geometries simulating pressure, velocity and enthalpy in the cascade. The ANN was trained to characterise the flow properties for a given geometry. The trained network could estimate all design parameters within 1% relative error. More importantly, it could run in near instantaneous time compared with hours for a comparable CFD simulation. This work demonstrates the speed of ANN and their ability, once trained, to replicate more complex analysis at greatly reduced computational effort.

Dominiczak et al. [33] investigated online prediction of temperature and stress in turbine components using neural networks. They trained a Nonlinear AutoRegressive eXogeneous (NARX) recurrent neural network to model real-time behaviour of the rotor in a high-pressure turbine. Like Marinescu, they identified the first blade groove as the critical point and focused only on this location for temperature and stress prediction. A three NARX network architecture was used, incorporating several exogenous inputs: rotor axial expansion, turbine speed, turbine load, steam temperature and pressure. Data from 168 2D axisymmetric rotor simulations was used to train the model. The network learned the blade root temperature and stress to maximum Root Mean Square Error (RMSE) of 5.4^oC and 18.8MPa respectively. All temperature predictions were within ±5% and all stress predictions within ±15%. The authors went on to evaluate an improved stress control system using future predictions from the NARX network. Reasonable predictions in stress and temperature were limited to 5mins in the future. However, this limited knowledge of the future is promising for predictive system control.

Goyal et al. [34] built a three part model, testing different statistical and neural network methods. Their work focused on performance prediction for the power and blade path temperature spread in a gas turbine. They introduced a hidden Markov model to segregate the data into three categories: turbine off, transition and steady state operation. This segregation allowed a targeted model to be build for each state and improved the overall performance of the combined prediction. Results for the turbine output power were very successful however, the accuracy of the blade path temperature spread prediction was notably lower; this was mostly observed in the low speed and off conditions.

Ibrahem et al. [35] presented a NARX ensemble method for the power, outlet pressure and output temperature prediction of a three spool aero-derivative gas turbine engine, SGT-A65. They generated an initial pool of 240 structures with varying neuron count and activation functions to identify the optimum architecture. This structure was then used to build an ensemble of eight models for each output parameter; trained separately on different datasets, these model were then combined to give a single output. Basic mean, weighted mean and point-wise best methods were tested for ensemble combination. Significant improvement was seen in the performance of the ensemble compared to a singular model output.

Panov and Cruz-Manzo [36], Siemens Industrial Turbomachinery, presented a paper on the practicalities of deploying real-time software for their digital twin field trial. Combining real-time differential equation models, an estimator, a Kalman tuner and sensor diagnostic blocks; their MATLAB Simulink model was tested for the performance and diagnostic monitoring of a small industrial gas turbine. Panov outlines how future control systems require a cyber-physical space with real machine data, virtual machine simulations and cloud system integration. The concept was demonstrated on a stand alone simulator with future field trials destined to validate remote connectivity, ensuring robust data transfer, safety and security.

2.6 Summary

Demand for fast-start, load-levelling, flexible operation has seen more focus given to transient thermal behaviour. Several works have targeted this issue, using loosely-coupled conjugate methods. Data driven plug-ins, simplified equivalent geometries and conductivity models were all shown to work well when applied to either forced or natural convection cases. The reviewed works suffered when handling the wider operating regime due to the physical assumptions in each solver set-up.

This suggests the need for a more data driven methodology, with less dependence on a purely physical model. Rather than apply a predefined empirical correction, the measurements can be directly coupled with the solver. A hybrid method of this type requires thermal sensor data to be embedded in the simulation. The accuracy of traditional heat flux measurements will therefore be reviewed, along with new mapping methods, to support the direct coupling of thermal data to a low order numerical solver.

The requirement to analyse long duration simulations, in excess of 100hrs, dictates the need for a fast approximate solver. ANN models have been shown to rapidly and reliably handle thermal turbine calculations. The previous works focused on the prediction of the thermal stress or temperature at a single location. Neural network methods will be further developed to track multiple locations and provide real-time boundary conditions for a high pressure steam turbine.

Restrictions in measurement access reduce the availability of known data locations for ANN training. A spatial model is therefore required to interpolate the sparse data and recover the thermal profile. Manual construction of a low order, full turbine, spatial model may take many weeks to build. User demanding tasks are required to approximate the full geometry, set the model connectivity and calibrate the associated thermal properties. Full automation of this process is desirable and will be tackled in this thesis, alongside high resolution reconstruction of the full turbine thermal profile.

3

1D Heat Transfer Analysis

3.1 Introduction

One-dimensional methods are commonly used to approximate or initialise heat transfer analyses. The simplified analytic solutions, along with experimental methods, are routinely applied to measure the heat flux and extract the heat transfer coefficient. This experimental data provides vital support to build and calibrate low order heat transfer models. The impulse response method is widely used, particularly in the analysis of high frequency transient convection boundaries. The method is routinely applied to laboratory analysis to extract the time-averaged heat transfer coefficient in complex flow structures. Often applied at the surface only, using a simplified unit step solution, traditional use of this technique is limited. The impact of these limitations are commonly misunderstood or ignored, leading to significant errors in the data post-processing. This chapter evaluates the impulse response method in depth, presenting new spatial-temporal solutions to better define the response in convection flow analysis. New techniques are introduced to better handle flow transients, determine the method's accuracy and define the limitations.

3.2 1D Heat Equation

Heat flow via conduction in an isotropic material is governed by the heat equation.

$$\frac{\partial T}{\partial t} = \alpha \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right) \tag{3.1}$$

Assuming the system can be considered one-dimensional and that the material is initially at uniform zero temperature, the Laplace transform $\phi(x, s)$ is defined by Equation 3.4. Greek letters are used to indicate functions in the Laplace domain.

$$\text{One} - \text{dimensional assumption} : \frac{\partial^2 T}{\partial x^2} >> \frac{\partial^2 T}{\partial y^2} \; ; \; \frac{\partial^2 T}{\partial z^2} \tag{3.2}$$

$$\text{Initial state assumption} : \phi(x,0) = 0 \tag{3.3}$$

$$\frac{\partial^2 \phi}{\partial x^2} - \frac{s}{\alpha} \; \phi(x,s) = 0 \tag{3.4}$$

This equation has the well-known general solution

$$\phi(x,s) = A(s) \; \mathrm{e}^{-\lambda x} + B(s) \; \mathrm{e}^{\lambda x} \tag{3.5}$$

In cases where the material properties and thickness prevent heat penetration from the front to the back surface, the material is assumed semi-infinite. This assumption requires $\phi(\infty, s) = 0$ and defines $B(s) = 0$. $A(s)$ is then defined by some function $F(s) = L\{f(t)\}$ where $F(s)$ is dependent on the Laplace variable only and is independent of x.

$$\phi(x,s) = F(s) \; \mathrm{e}^{-\lambda x} \qquad \textit{where} : \lambda = \sqrt{\frac{s}{\alpha}} \tag{3.6}$$

Setting $x = 0$ yields $\phi(0,s) = F(s)$ and identifies $F(s)$ as the surface temperature applied to the material. For any applied surface function, the time varying temperature profile $\phi(x,s)$ can be defined by the Laplace domain product with $\mathrm{e}^{-x\sqrt{s/\alpha}}$. Transformed to the time domain, this is equivalent to a convolution of $f(t)$ with $L^{-1}\{\mathrm{e}^{-x\sqrt{s/\alpha}}\}$. Equation 3.7 therefore gives the full domain material impulse response to any arbitrary surface forcing function.

$$L^{-1}\left\{\mathrm{e}^{-x\sqrt{s/\alpha}}\right\} = \frac{x}{\sqrt{4\pi\alpha t^3}} \; \exp\left(-\frac{x^2}{4\alpha t}\right) \tag{3.7}$$

The ability to solve the time varying solution using a product rather than a convolution means operating in the Laplace domain is preferable. Any Laplace function that can be factored by $\mathrm{e}^{-x\sqrt{s/\alpha}}$ is a solution to the 1D semi-infinite heat equation. If the transform and its inverse are both known, entire families of solutions can easily be found from Laplace transform tables. These are discussed further in Section 3.4

3.3 1D Heat Flux Equation

The heat flux in a 1D isotropic material is defined by Fourier's Law of Conduction.

$$q = -k \, \frac{dT}{dx} \tag{3.8}$$

Applied to the 1D heat solution discussed in the previous section, the heat flux in the Laplace domain $\psi(x,s)$ is defined by

$$\psi(x,s) = k \, \sqrt{\frac{s}{\alpha}} \, \phi(x,s) \tag{3.9}$$

$$\psi(x,s) = k \, \sqrt{\frac{s}{\alpha}} \, F(s) \, \mathrm{e}^{-x\sqrt{s/\alpha}} \tag{3.10}$$

Therefore, for any applied surface temperature function $F(s)$, the time varying heat flux profile at any position in the material $\psi(x,s)$ can be defined by the Laplace domain product with $k \, \sqrt{s/\alpha} \, \mathrm{e}^{-x\sqrt{s/\alpha}}$. Any Laplace function that can be factored in this way is a solution of the heat flux in a 1D semi-infinite material.

3.4 1D Temperature and Heat Flux Solutions

The above two sections yield general solutions for the temperature $\phi(x,s)$ and heat flux $\psi(x,s)$ in a 1D, semi-infinite, isotropic material. Abramowitz and Stegun's book [37] contains an extended list of known Laplace transforms, line items 29.3.80 to 29.3.97 list the solutions containing the $\mathrm{e}^{-x\sqrt{s/\alpha}}$ factor. Of most interest are surface temperature functions $F(s)$ that have known inverse transforms for both temperature (Equation 3.6) and heat flux (Equation 3.10). Two important cases are listed in Table 3.1.

Case	$F(s)$	$\phi(x,s)$	$\psi(x,s)$
(1)	$\frac{1}{s^{1+\frac{1}{2}n}}$ $(n = 0, 1, ...; \; x, \alpha \geq 0)$	$\frac{1}{s^{1+\frac{1}{2}n}} \; \mathrm{e}^{-x\sqrt{s/\alpha}}$	$\frac{k}{\sqrt{\alpha}} \; \frac{1}{s^{\frac{1}{2}(1+n)}} \; \mathrm{e}^{-x\sqrt{s/\alpha}}$
(2)	$\frac{a}{s(a+\sqrt{s})}$ $(x, \alpha \geq 0)$	$\frac{a}{s(a+\sqrt{s})} \; \mathrm{e}^{-x\sqrt{s/\alpha}}$	$\frac{k}{\sqrt{\alpha}} \; \frac{a}{\sqrt{s}(a+\sqrt{s})} \; \mathrm{e}^{-x\sqrt{s/\alpha}}$

Table 3.1: Laplace domain solutions of 1D temperature and heat flux.

In the case of constant parameters α and a, with uniform initial zero conditions, it should be noted the repeated integrals or differentials, with respect to t or x, are also valid solutions of the problem. Often heat transfer analysts focus only on the surface temperature, this approach means analytic data is lost regarding the subsurface behaviour. Knowledge of the full time-varying solution allows assumptions in both the impulse response and other numerical methods to be validated. The inversions of these general Laplace solutions are discussed in the following sections and used later in this thesis for numerical scheme validation.

3.4.1 Conduction Solution

Case 1 of Table 3.1 gives the general solution for an applied polynomial or parabolic surface temperature. Taking the inverse Laplace transform returns the time domain solution in terms of repeated integrals of the complimentary error function.

$$T(x,t) = (4t)^{\frac{1}{2}(n+1)} i^{n+1}\mathrm{erfc}\left(\frac{x}{\sqrt{4\alpha t}}\right);\ n \geq 0 \tag{3.11}$$

$$q(x,t) = \sqrt{\rho c k}\ (4t)^{\frac{1}{2}n} i^{n}\mathrm{erfc}\left(\frac{x}{\sqrt{4\alpha t}}\right);\ n \geq 0 \tag{3.12}$$

Calculating higher orders of this general solution, $n > 1$, the properties of the repeated integral functions $i^n erfc$ can be used to combine lower order solutions efficiently. Repeated integrals can be applied in the Laplace domain, factoring by 1/s for a time integral or $\sqrt{\alpha/s}$ for a spatial integral. Alternatively, substituting $x^* = x/\sqrt{4\alpha t}$ for the dimensionless depth allows simple calculation of the repeated integrals in the time domain using Equation 3.15.

$$i^{-1}\mathrm{erfc}(x^*) = \frac{2}{\sqrt{\pi}}\mathrm{e}^{-x^{*2}} \tag{3.13}$$

$$i^{0}\mathrm{erfc}(x^*) = \mathrm{erfc}(x^*) \tag{3.14}$$

$$i^{n}\mathrm{erfc}(x^*) = -\frac{x^*}{n} i^{n-1}\mathrm{erfc}(x^*) + \frac{1}{2n} i^{n-2}\mathrm{erfc}(x^*) \tag{3.15}$$

Unit Step Solution

Traditionally, work has focused on the unit step solution and looked at the material behaviour of the top surface only. Substituting $n = 0$ and $x = 0$ in Equation 3.11 and Equation 3.12, with the relation $k/\sqrt{\alpha} = \sqrt{\rho c k}$, returns the widely used unit step solutions [38].

$$T(0,t) = \frac{2}{\sqrt{\rho c k}}\sqrt{\frac{t}{\pi}} \tag{3.16}$$

$$q(0,t) = u(t) \tag{3.17}$$

This simplified form is often used for basis functions to generate the filter in the impulse response method. The availability of a more general polynomial basis set allows for a rigorous validation of the impulse response method and is discussed further in Section 3.5.

Unit Step Semi-infinite Limit

Substituting $n = 0$ for unit step heat flux, then taking the ratio of the surface and depth functions, returns the widely used semi-infinite formulae described by Schultz and Jones[39], Equations 3.18 and 3.19. This approach can be used for any value of n to determine the semi-infinite behaviour for any arbitrary surface function. It should be noted that this ratio is not constant and the value of n has a significant effect. This is discussed in detail when evaluating the semi-infinite criterion in Section 3.5.2.

$$\frac{T(x,t)}{T(0,t)} = \mathrm{e}^{-x^{*2}} - \sqrt{\pi}\, x^*\, \mathrm{erfc}(x^*) \tag{3.18}$$

$$\frac{q(x,t)}{q(0,t)} = \mathrm{erfc}(x^*) \tag{3.19}$$

Utilising polynomial best fit methods and superposition of the above solutions, one could derive an approximate analytic solution for any given surface forcing function. This methodology has not been applied in this thesis due to preference of the impulse response method but may be of use in other applications.

3.4.2 Convection Solution

Case 2 of Table 3.1 gives the convection solution for a constant temperature free-stream flow across the upper surface of a semi-infinite solid. Full definition of this solution requires an additional thermal driving function to be introduced that defines the effect of the free-stream flow, $W(x,t)$. It should be noted that the driving function has a complete Laplace domain solution, and although it defines the fluid at the surface of the domain, is fully defined throughout the entire spatial range of the solid, $\Omega(x,s) = L\{W(x,t)\}$. The convection solution also requires a heat transfer coefficient, h, which is assumed constant. Due to the additional variables, this second solution is slightly more complex than the previous conduction case.

$$\psi(x,s) = h\Big[\Omega(x,s) - \phi(x,s)\Big] \tag{3.20}$$

$$\Omega(x,s) = \frac{1}{s}\ \mathrm{e}^{-x\sqrt{s/\alpha}} \tag{3.21}$$

$$\phi(x,s) = \frac{a}{s(a+\sqrt{s})}\ \mathrm{e}^{-x\sqrt{s/\alpha}} \tag{3.22}$$

$$\psi(x,s) = \frac{k}{\sqrt{\alpha}}\ \frac{a}{\sqrt{s}(a+\sqrt{s})}\ \mathrm{e}^{-x\sqrt{s/\alpha}} \tag{3.23}$$

Applying Newton's Law of cooling, Equation 3.20, at the surface $x = 0$ allows for the constant a to be solved.

$$a = \frac{h}{\sqrt{\rho c k}} \tag{3.24}$$

For a step in free-stream temperature, $\Omega(0,s) = 1/s$, taking the inverse Laplace transform of $\Omega(x,s)$, $\phi(x,s)$, and $\psi(x,s)$ returns the time domain form for constant flow temperature. It can be seen that the driving function, $W(x,t)$, follows the polynomial $i^n erfc$ solution given in the previous section. As discussed, repeated spatial integrals of this solution can be used to generate any order temperature function. By applying the repeated integral process, one can derive the convection solution in the full material for any arbitrary surface flow temperature.

The first case corresponding to constant flow temperature is defined below.

$$q(x,t) = h\Big[W(x,t) - T(x,t)\Big] \tag{3.25}$$

$$W(x,t) = \operatorname{erfc}\left(\frac{x}{\sqrt{4\alpha t}}\right) \tag{3.26}$$

$$T(x,t) = \operatorname{erfc}\left(\frac{x}{\sqrt{4\alpha t}}\right) - \mathrm{e}^{(hx/k)}\ \mathrm{e}^{(h^2 t/\rho c k)}\ \operatorname{erfc}\left(\sqrt{\frac{h^2 t}{\rho c k}} + \frac{x}{\sqrt{4\alpha t}}\right) \tag{3.27}$$

$$q(x,t) = h\ \mathrm{e}^{(hx/k)}\ \mathrm{e}^{(h^2 t/\rho c k)}\ \operatorname{erfc}\left(\sqrt{\frac{h^2 t}{\rho c k}} + \frac{x}{\sqrt{4\alpha t}}\right) \tag{3.28}$$

Taking $x = 0$, the surface temperature and heat flux functions can be found for a constant free-stream temperature. Note, this is simply one of many possible solutions from the $i^n erfc$ family. Higher order solutions can also be used to handle transient free-stream temperatures.

3.5 Impulse Response Method

High speed linear cascades are often used to analyse aerodynamic performance of turbine blades, nozzle guide vanes and cooling systems. Compressed air is blown through a heater mesh to achieve uniform step change in temperature. The heated air is then passed through the test cascade, optionally with cooling flow, to conduct heat transfer measurements. The data is routinely post-processed using the impulse response method. A known analytic temperature and heat flux, T_a and q_a (commonly the unit step solution), are used to derive an impulse filter response for heat flux F_{Tq}. This filter is then applied to the measured temperature signal, T_m, to infer the surface heat flux, q_m. This is routinely done using Matlab $filter$ and $fftfilt$ functions but can equally be calculated using Python SciPy $lfitler$ and *convolve* [40].

$$F_{Tq} = filter\Big(q_a, T_a, \delta\Big) \tag{3.29}$$

$$q_m = fftfilt\Big(T_m, F_{Tq}\Big) \tag{3.30}$$

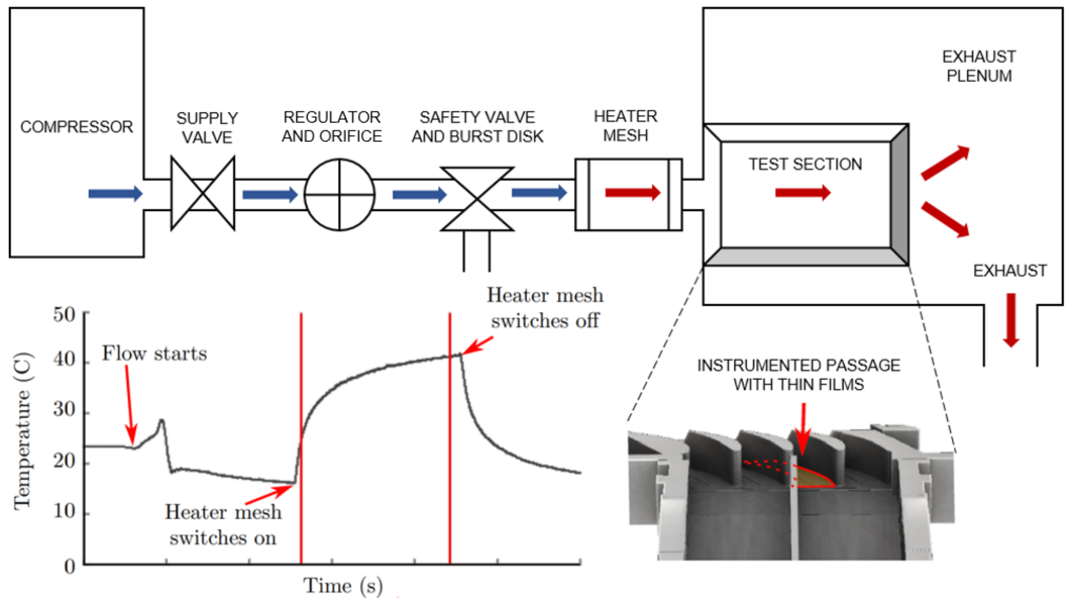


Figure 3.1: Example high speed linear cascade facility instrumented with thin film gauges on the passage endwall, with typical temperature measurement for heat flux calculation [41]

The impulse response method was first presented by Oldfield for the analysis of thin film data [38]. The method assumes the substrate under test is an isotropic, semi-infinite, 1D material with constant properties and can therefore be considered a Linear Time-Invariant (LTI) system. The response of any LTI system can be solved via a convolution integral with the impulse response of that system.

Oldfield's process uses the analytic solution for unit step heat flux. As discussed in his paper, this is simply one of many basis functions that could be used and, any of the polynomial or convection solutions defined in the previous section are equally valid. The heat flux impulse response for a semi-infinite material, independent of the $filter$ inputs, is given by $L^{-1}\{\sqrt{\rho c k}\sqrt{s}\}$. This expression is infinite at t=0 and has no continuous time solution in standard form. As highlighted by Oldfield, the discrete deconvolution is a finite approximation of the infinite continuous response. The finite approximation of ∞ at $t = 1/f_s$ is the only reason that this method works, and the initial time step must be set $t = 1/f_s$, ignoring $t = 0$, in order to obtain a valid filter with this technique.

It should be noted that for constant HTC convection solutions, one can equally define an impulse response between T and W, allowing both to be known for a given HTC. For example the surface temperature, T_m, for any arbitrary free-stream temperature, W_m, can be found by simply applying the F_{WT} impulse response to W_m. In the case of a heater mesh with exponential free-stream heating, discussed in Chapter 2 of Gillespie's thesis [42], the direct analytic solution becomes quite involved, requiring complex erf functions. This result can be more easily replicated using the impulse response, setting $W_m = T_g[1 - e^{-t/\tau}]$.

$$F_{WT} = filter\Big(T_a, W_a, \delta\Big) \tag{3.31}$$

$$T_m = fftfilt\Big(W_m, F_{WT}\Big) \tag{3.32}$$

3.5.1 Accuracy of the Impulse Response

Oldfield analysed the numerical accuracy of the impulse response method, Figure 6 of his publication [38] shows the error to be within $\pm 6 \times 10^{-14}$ for a parabola in T. Note, that the published accuracy being valid for a parabola is important. The basis temperature function for the unit step case is parabolic. This means that the functions used to derive the impulse response and validate the method are of the same form. This error only shows the numerical accuracy of the Matlab *filter* and *fftfilt* functions, not the true accuracy of the method as a whole. A more rigorous validation requires different basis and test functions be used. This analysis was performed using the general analytic solutions presented in Section 3.4.

Figure 3.2 shows the functions used in this analysis. The analytic temperature and heat flux pairs were used to compute an impulse response for each reference case, mathematically these responses should be identical. Each response was then applied to all other surface temperatures to return the calculated heat flux when using the different filters. This was repeated at three different sampling frequencies for cases with and without noise.

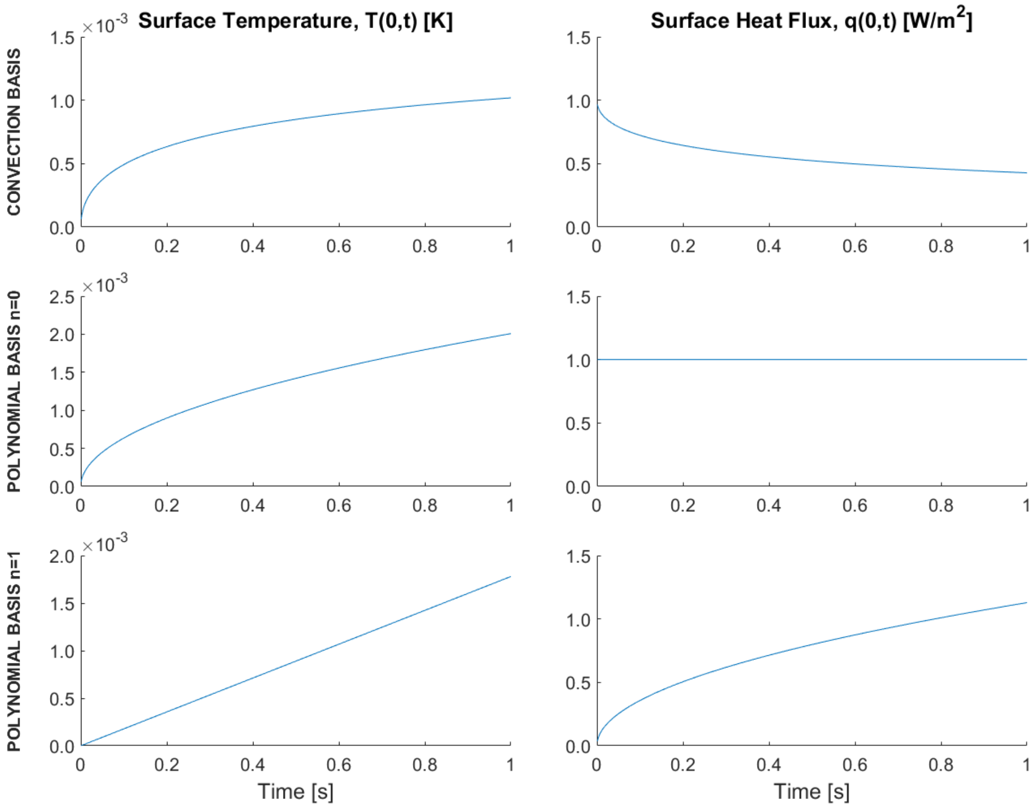


Figure 3.2: Impulse response basis functions for the first convection and first two polynomial cases, used in the validation of impulse response accuracy.

Figure 3.3 shows the results of the analysis and is assembled in grid format. Horizontal rows are results using the same response filter, identified by the left most y-axis. Vertical columns are results of each filter, identified by the column title. All graphs show the calculated heat flux, $q(0,t)[W/m^2]$, with the evaluated Root Mean Square Error (RMSE) compared to the true analytic solution.

When the basis and test functions are the same, i.e. on the grid diagonals, the accuracy without noise compares well with the published result by Oldfield. However, in the off-diagonals when the basis and test functions differ, the accuracy is several orders of magnitude worse. In all practical applications, the test function does not match exactly the basis function, the RMSE in these cases is of the order 10^{-3}. This error is still very low and likely sufficient for most heat transfer analysis tests. Oldfield stated that the main cause of uncertainty is the physical measurement of the material thermal properties; this is likely still true but the numerical errors are of comparable significance.

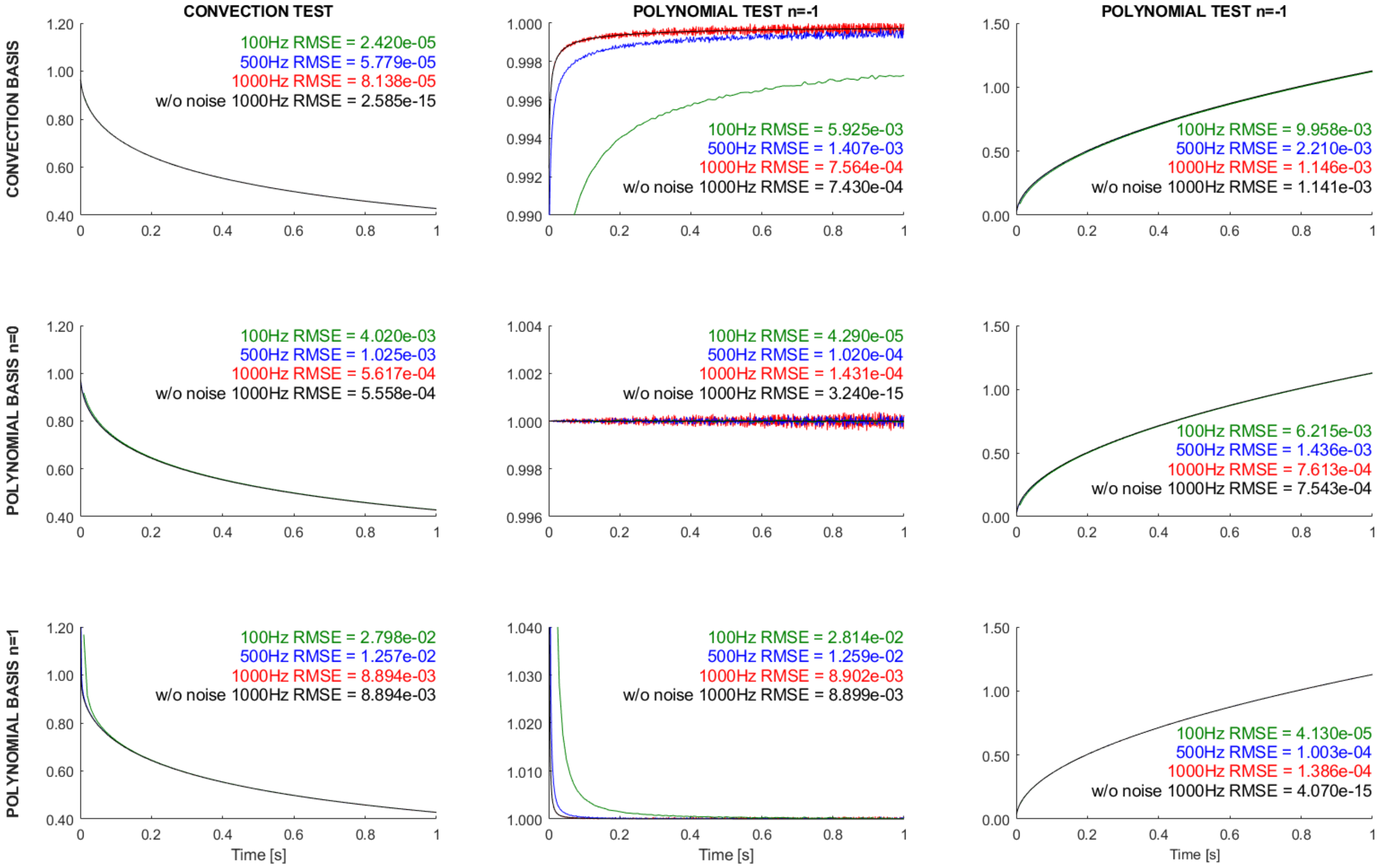


Figure 3.3: The calculated heat flux $[W/m^2]$, using the impulse response method, for all combinations of three different basis and test functions. Showing the effect of noise and sampling rate on the root mean square error of the method.

The analysis highlights the importance of matching the basis and test functions. The absolute accuracy is improved when the most similar basis set is used to derive the impulse response. Even in the case of noisy data, the best results are seen on the grid diagonals. Therefore, when analysing heat transfer via convection, it is recommended that the impulse response be derived from the convection solutions presented in Section 3.4.2, rather than the more commonly used unit step.

Comparison of the different sampling frequencies in Figure 3.3 shows the impact of the data acquisition system. For cases on the off-diagonal, most similar to a genuine use case, it can be seen that the expected accuracy is greatly improved when the sampling frequency is increased. This is caused by better discrete approximation of the response function at the first sample point, $t = 1/f_s$. Faster sampling rates are also advantageous for resolving high frequency flow phenomena, commonly of interest in heat transfer analysis. However, care should be taken when using high frequency data; the faster response increases the resolution of noise and the signal should first be bandpass filtered to ensure only physical unsteadiness is analysed.

This effect is clearly seen on the grid diagonal, which shows a negative impact of higher sampling frequency. Unlike the mismatched cases, the best performance is now seen at low frequency not high frequency. This is entirely caused by the better noise resolution in the high frequency case. The faster sample rate captures more peak noise events and acts to expand the variance of the heat flux data, increasing the associated RMSE. Assuming the signal has first had bandwidths associated with sensor noise removed, users should always opt for a higher sampling rate to better resolve the true physical behaviour.

The nature of the impulse response convolution means this technique generally responds well to noisy inputs. The absolute accuracy for the noisy test cases was very good, with both the 500Hz and 1000Hz cases proving suitable. Comparing the 1000Hz test cases in Figure 3.3, with and without noise, it can be seen that the addition of noise leads to a negligible decrease in accuracy.

Evaluation

The validation graphs show that the RMSE in the impulse response method is comparable to those achieved by direct numerical methods discussed later in this thesis. Spatial numerical methods have the advantage that additional boundary conditions can be specified and a time-varying system can be modelled. This allows material properties that are time or temperature dependent, which is not possible with the impulse response method. In cases where the assumptions of LTI are suitable,

- the material is semi-infinite
- the material is isotropic
- the material properties remain constant
- the system is initially at uniform zero temperature

the impulse response method offers a very computationally efficient method to analyse heat transfer data. The derivation of the response functions assumes uniform absolute zero temperature as the initial condition throughout the body. This condition can be satisfied by ensuring the test article has cooled to a uniform temperature then this value subtracted from all measurements. It is recommended that applications use a high sampling frequency, bandpass filtering the data to remove sources of experimental noise. The method is most accurate when the basis and test functions are of the same form. Users are encouraged to use the correct basis set, rather than simply relying on the simplified traditional unit step.

3.5.2 Semi-infinite Assumption

The limits of the semi-infinite assumption are important when applying the impulse response method. The analytic solution that defines the basis functions is solved via Equation 3.5 which uses the semi-infinite assumption to state $B(s) = 0$ and ensure finite temperature. Therefore, this analysis method is only valid when heat penetration does not reach the back surface of the test article.

Schultz and Jones derived the formula for dimensionless penetration depth under unit step surface heat flux in Equation 3.18 [39]. They presented a rule of thumb, in terms of the ratio of front to back surface flux, such that the material can be considered semi-infinite. This rule states that the back surface flux should be less than 1% of the front surface value. Combined with the penetration depth formula, this rule defines the semi-infinite test duration. Beyond this time, the heat penetration is too great for the assumptions in the analytic basis functions to hold true. The Schultz-Jones recommend maximum test duration is given by Equation 3.33.

$$t = x^2/16\alpha \tag{3.33}$$

This rule, based on the unit step only, is widely applied independent of the actual surface function. This is incorrect and one should calculate the penetration depth for the actual conditions of the test. Using the general solution for polynomials and convection, presented in Section 3.4, it can be seen that the semi-infinite time varies significantly depending on the test type.

To increase the allowable duration of tests, it is common to select materials with deliberately poor thermal properties. This slows heat penetration through the material and increases the time that the underlying analysis assumptions are valid. Before applying the impulse response method, the semi-infinite duration must always be calculated for the actual materials and conditions of the test.

Test facilities at the Oxford Thermofluids Institute generally use heater mesh technology designed to introduce a unit step in free-stream temperature. This corresponds to the $n = 0$ convection solution and has the shortest semi-infinite duration. When running longer duration tests, especially with thin wall geometry, this limit is often reached. If the material properties cannot be changed, and a longer duration test is required, modifying the surface function provides an alternative route to extending the test.

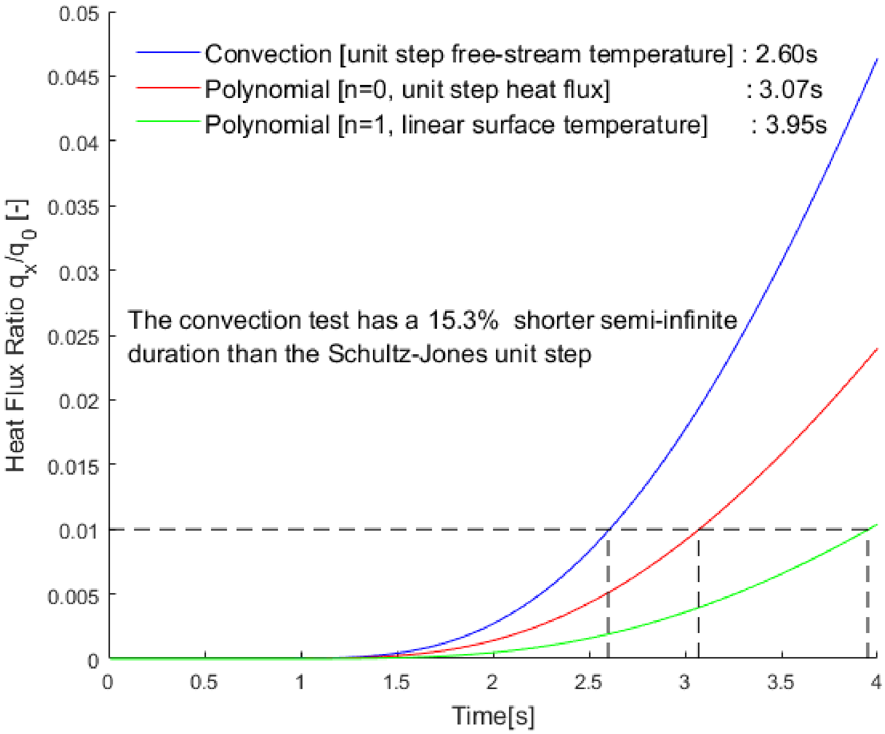


Figure 3.4: Semi-infinite duration for three different surface functions applied to a 2mm single material substrate of SLA 7870. The convection heat transfer coefficient is taken as $\sqrt{\rho c k}$.

Defining the allowable test time using the Schultz-Jones penetration depth is not applicable in the case of convection testing. The formula below can instead be used for standard heater mesh technology. The result depends on the heat transfer coefficient and would therefore need to be re-evaluated after each test. The equation can be solved graphically as shown in Figure 3.4, where x is the thickness of the test article substrate.

$$\frac{q(x,t)}{q(0,t)} = \frac{\mathrm{e}^{(hx/k)}\ \mathrm{e}^{(h^2t/\rho ck)}\ \mathrm{erfc}\Bigg(\sqrt{\frac{h^2t}{\rho ck}} + \frac{x}{\sqrt{4\alpha t}}\Bigg)}{\mathrm{e}^{(h^2t/\rho ck)}\ \mathrm{erfc}\Bigg(\sqrt{\frac{h^2t}{\rho ck}}\Bigg)} \tag{3.34}$$

Given that the semi-infinite limit should be calculated on a test case basis rather than once per facility; it is desirable to have a faster, more robust method to calculate this. The back surface heat flux can be directly computed using a depth filter, F_d, with the impulse response. Taking the conduction $i^n erfc$ solutions, the filter for temperature or flux at any position in the material can be computed. Setting x as the thickness of the substrate returns the effective back surface value, equivalent to that used in the Schultz-Jones criteria.

$$F_d = filter\Bigg(\mathrm{erfc}\Big(\frac{x}{\sqrt{4\alpha t}}\Big), \mathrm{erfc}(0), \delta\Bigg) \tag{3.35}$$

Note all of these methods assume a semi-infinite body and therefore do not fully account for the true behaviour of a back boundary in thin wall geometry. Nevertheless, applying the impulse response in this way does account for the test type variation and surface function, thus providing a more reliable definition than the traditional Schultz-Jones approach. The semi-infinite limit is then best defined by the following expression, taking the heat flux ratio for the criterion as this tends to impose an earlier limit than the temperature ratio.

$$0.01 \leq \frac{\Big|F_d * F_{Tq} * T(0,t)\Big|}{\Big|F_{Tq} * T(0,t)\Big|} \tag{3.36}$$

Taking this more robust definition, the semi-infinite duration of different free-stream temperatures can be evaluated and the facility designed to achieve maximum allowable test duration. Theoretically, an alternating stepped heating and cooling cycle can support an infinite duration test. This does however necessitate a below ambient temperature test gas with fast switching heater, possibly not viable in practice.

3.6 Solving Heat Transfer Coefficient

Most aerothermal tests are concerned with measuring the effect of different flows on the heat transfer and thus focus on calculating the HTC. Due to the mixing of mainstream and cooling flows, the true local fluid temperature $W(0,t)$ is unknown. Hot wire anemometry methods are available however, due to the expense, flow intrusion and the delicate nature of these probes, they are rarely employed within high speed cascades. The heat transfer coefficient h must therefore be evaluated graphically using the measured wall heat flux q, the local wall temperature T, and Newton's law of cooling, Equation 3.37.

$$q = h\Big(W - T\Big) \tag{3.37}$$

The law assumes h is independent of temperature and that W and the convective flow behaviour remain constant. In cases where the fluid properties or aerodynamics are temperature dependent e.g. buoyancy driven convection, care should be taken when applying this method and the impact of the constant h assumption considered.

Discussed by Maffulli [43], the heat transfer coefficient on a nozzle guide vane can be notably dependent on the wall temperature. His work evaluated several CFD cases at different wall temperature ratios, T/W. In both 2D and 3D, he showed there was a clear dependence of h on the temperature ratio, caused by temperature induced changes in the boundary layer, passage and secondary flows. He presented a linear HTC scaling method, applied point-wise to the domain, dependent on the temperature ratio. The linearity accommodates the changing flow behaviour and better fits a wide temperature ratio of 0.65 to 0.99, $T_\infty = 444K$.

His results extend the conclusions of Kays [44] and Reynolds [45] who first described this phenomenon on a flat plate. In the laminar boundary layer, seen near the leading edge of a turbine blade, little dependence is seen on the temperature ratio and changes in the physical properties dominate, Equation 3.38 [44]. Towards the trailing edge of a turbine blade, in the turbulent region, a strong non-linear effect of temperature ratio is additionally observed, Equation 3.39 [45].

$$Nu_x = 0.332\ Pr^{1/3}\ Re_x^{1/2} \tag{3.38}$$

$$Nu_x = 0.0296\ Pr^{0.6}\ Re_x^{0.8} \left(\frac{T_w}{T_\infty}\right)^{-0.4} \tag{3.39}$$

It is important to note that in cases of large temperature variations, both the physical properties and flow dynamics change, affecting the HTC. Experimental calculation should therefore be limited to controlled temperature ranges where the assumptions of constant HTC are more reliable. A new method, utilising the $i^n erfc$ properties, is presented to enable the handling of these controlled transients in the free-stream flow.

3.6.1 Legacy HTC Method

If the flow temperature can be maintained at a constant value, a plot of the measured wall temperature T, and heat flux q, gives a straight line with gradient equal to $-h$ and x-axis intercept equal to the flow temperature W. This method is widely used and assumes the HTC is independent of the wall and free-stream flow temperatures. This assumption is only suitable for a small temperature range and should be validated on a case by case basis.

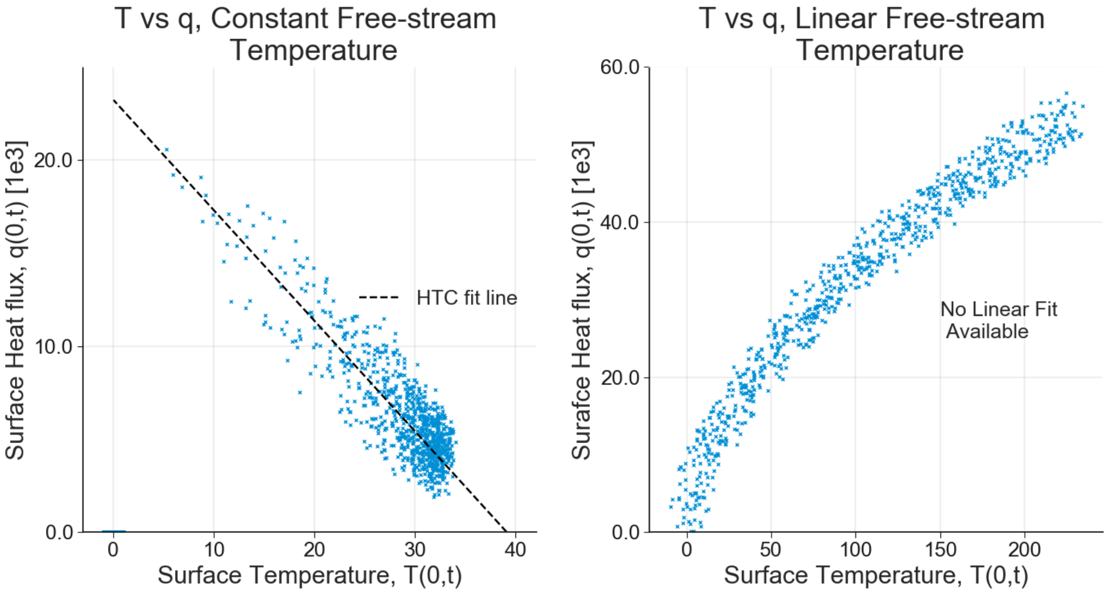


Figure 3.5: Example T-q plane fit method for HTC calculation for constant free-stream temperature and constant HTC of 562 W/m^2K. Demonstrating the limitation of this method and the inability to handle linear free-stream temperature.

To achieve the linear fit, designers go to great lengths to develop test facilities and heaters to achieve a near constant step in the free-stream temperature. An ideal convection test then provides data as shown in Figure 3.5. The Gillespie mesh design is widely used and provides a near ideal temperature step change [42].

The fit method breaks down if transients are introduced in the free-stream. Even simple cases, such as linear flow heating, lead to a time-varying flow variable $W(0,t)$, causing a changing x-axis intercept in the T-q plot and preventing the fit being solved graphically. The constant stream temperature requirement is solely a limitation of the legacy graphical method. Seen in the previous semi-infinite discussion, linear or increasing flow temperatures are preferable in long duration testing. If the flow properties remain approximately similar and the flow structure is not temperature dependent, having the ability to analyse transient tests opens new possibilities in heat transfer facility design. A new method is presented to handle both constant and time-varying free-stream flows.

3.6.2 Integral Superposition HTC Method

Working with the full domain (x, t) rather than simply the surface functions, the effect of the free-stream can be removed without knowledge of the absolute temperature value. A novel superposition methodology making use of the repeated integral properties is introduced. The full domain free-stream function $W(x, t)$ follows the conduction solution of the complementary error function integrals, $i^n erfc$. Assuming a constant heat transfer coefficient, h, the repeated integrals can be used to remove the unknown free-stream. This process is easiest done using repeated Laplace domain spatial integrals with respect to x, simply taking the exponent factor $-\sqrt{\alpha/s}$.

The assumption of constant h still limits the analysis. However, this limitation is also inherent to the legacy method. Working here with an integral solution, the constant h assumption is better defined as a constant time averaged h. This is likely achieved in linear cascade facilities which are designed to have stable flow velocity with high frequency unsteadiness about an approximately constant mean. Care should be taken when applying any of these methods to cases where velocity and HTC have large scale transients over the test duration. The process discussed below can be used to solve any bulk varying free-stream temperature function, with approximately constant wall HTC. The general free-stream function is given by Equation 3.40.

$$W(x,t) = (4t)^{\frac{1}{2}n} i^n \mathrm{erfc}\left(\frac{x}{\sqrt{4\alpha t}}\right);\ n \geq 0 \tag{3.40}$$

This follows the $i^n erfc$ family of the conduction solutions which has a repeated integral property with respect to z

$$i^m \mathrm{erfc}(z) = -\frac{z}{m} i^{m-1}\mathrm{erfc}(z) + \frac{1}{2m} i^{m-2}\mathrm{erfc}(z) \tag{3.41}$$

$z = x/\sqrt{4\alpha t}$ can be substituted to find the repeated integral property with respect to x, correcting for the new integration constant

$$i^{m}\mathrm{erfc}(\frac{x}{\sqrt{4\alpha t}}) = -\frac{x/\sqrt{\alpha}}{m} i^{m-1}\mathrm{erfc}(\frac{x}{\sqrt{4\alpha t}}) + \frac{2t}{m} i^{m-2}\mathrm{erfc}(\frac{x}{\sqrt{4\alpha t}}) \tag{3.42}$$

The above formula gives a relation for a high order integral to the two previous integrals. This equality is only true in the case of $i^{n}erfc$ and is only true for the free-stream function. The solid wall temperature and heat flux have a different form and therefore do not satisfy this equality. By subtracting the right hand side from the left, a superposition of different integrals can be found that eradicates the free-stream, whilst leaving a non-zero function for the temperature and heat flux.

There are two ways to implement this solution: differentiate the signals twice, or integrate the signals twice. Most heat transfer measurements have high frequency noise or physical flow unsteadiness that form part of the measured signal. Differential methods are therefore less suitable due to the high localised gradients in the signal. Thus, it is advantageous to use the integral option which acts to naturally smooth the high frequency components. The repeated integral superposition to remove W is therefore best applied with the measured signal as the lowest order term in the formula, i.e. corresponding to the last term on the right hand side $i^{m-2}erfc$.

The only requirement for this method is to know the expected shape of the free-stream function, set by the value n, defining the free-steam shape as constant, parabolic, linear or quadratic. The absolute scale does not need to be known, this factor propagates through the integrals and naturally cancels. The constant, m, is simply given by $n + 2$ and is also set by the free-stream temperature shape only. Performance can be further improved by integrating the measured signal once before applying the superposition, this acts to smooth the raw data. High frequency information can be recovered later by back substitution so no information is lost in applying this smoothing step.

Taking the case for constant free-stream temperature, $W = \mathrm{erfc}(x/\sqrt{4\alpha t})$, at the surface $x = 0$, using one pre-integration smoothing step, the repeated integral reduces to the simplified form

$$3i^3\mathrm{erfc}(0) - 2t\ i^1\mathrm{erfc}(0) = 0 \tag{3.43}$$

Applying this superposition to $q = h(W - T)$

$$\begin{aligned} 3\iiint q(x,t)dx^3\Big|_0 - 2t\int q(x,t)dx\Big|_0 &= \\ h\left[3i^3\mathrm{erfc}(0) - 2ti^1\mathrm{erfc}(0) - 3\iiint T(x,t)dx^3\Big|_0 + 2t\int T(x,t)dx\Big|_0\right]& \end{aligned} \tag{3.44}$$

Removing the zero components from the driving function $i^n erfc$ leaves

$$3\iiint q(x,t)dx^3\Big|_0 - 2t\int q(x,t)dx\Big|_0 = h\left[2t\int T(x,t)dx\Big|_0 - 3\iiint T(x,t)dx^3\Big|_0\right] \tag{3.45}$$

The repeated integrals look daunting but are easily solved using an impulse response. The spatial integration generates a repeated factor $-\sqrt{\alpha/s}$ in the Laplace domain. This multiplication in Laplace space is equivalent to a convolution in the time domain and behaves identically to an impulse response method. An integrator filter can therefore be found that automatically computes the spatial integration. Using the known solutions of the $i^n erfc$ family, integrator filters F_n can be computed easily using conventional Matlab $filter$ functions with different order $i^n erfc$ for the inputs. These filters are then applied with $fftfilt$ to quickly compute the spatial integrations. The repeated integrals are then efficiently replaced and the above equation simplifies.

$$3F_3 * q(0,t) - 2t\left(F_1 * q(0,t)\right) = h\left[2t\left(F_1 * T(0,t)\right) - 3F_3 * T(0,t)\right]$$

where F_n is the integrator filter defined by (3.46)

$$i^1\mathrm{erfc}(0) = F_1 * i^0\mathrm{erfc}(0)$$
$$i^2\mathrm{erfc}(0) = F_2 * i^0\mathrm{erfc}(0)$$
$$i^3\mathrm{erfc}(0) = F_3 * i^0\mathrm{erfc}(0)$$

Noting that $q(x,t)$ is solved by a filter response from $T(x,t)$, Equation 3.30, the above is simply a combination of filters applied to the measurement signal $T(0,t)$. The filter convolutions $F_n * F_{Tq}$ may be combined using convolution associativity to give F_{nTq}, or equally applied in series as shown in Equation 3.47.

$$3F_3 * F_{Tq} * T(0,t) - 2t\left(F_1 * F_{Tq} * T(0,t)\right) = h\left[2t\left(F_1 * T(0,t)\right) - 3F_3 * T(0,t)\right] \tag{3.47}$$

Substituting q^* and T^* for the superposition of the flux and temperature, a direct linear relationship is found with constant of proportionality given by the time-averaged heat transfer coefficient, h. Note the linearity is dependent on assumptions of near constant HTC over the duration of the test.

$$q^* = h\,T^* \tag{3.48}$$

It is recommended that the method be applied at the surface, $x = 0$, because this simplifies the equations and also prevents damping of high frequency flow phenomena which are usually of interest. However, this is not a requirement of the method and any value of x may be used if the temperature is measured at that location, i.e. subsurface measurements from embedded probes may also be used.

The ability to handle data from subsurface probes is another advantage of this new approach over the legacy method. The traditional T-q plot fails in a subsurface application due to the non-constant $W(x,t)$ at depth and therefore time varying x-axis intercept. To demonstrate this feature of the new integral method the solutions at various depths into the material have been included in the validation Figure 3.6.

Assuming the W^* components to be negligible, one could compute h directly using the ratio q^*/T^*. In practice, due to unsteadiness in the HTC caused by turbulence and mixing flow, h should not be computed in this manner. Similar to the legacy method, q^* and T^* should be plotted and a best fit line used to determine time averaged h. The integral method fit is preferable to the legacy T-q fit because it is known that the line should pass through (0,0), giving additional constraint to the fit.

The new HTC calculation process is more generalised and enables the handling of higher order flow temperature functions and embedded sensors. More importantly, the new integral method supports the use of periodic functions, such as a square wave or sawtooth, by additionally introducing a temporal offset superposition in t. In such cases, the factor of the final term in the repeated integral equation, $2t/m$, is more complicated and is best computed indirectly from another signal. This reference signal should be as similar as possible in shape to the supposed flow temperature, but may have any magnitude value.

In the case of a classical step change heater mesh, using the control clock or heater on/off signal, to give a unit square wave, may provide a suitable reference signal. The heater mesh power, assumed linearly proportional to the temperature, or even the upstream flow temperature itself may potentially provide a better reference signal. The measured power or upstream temperature likely capture any additional ramp or decay heating effects due to the hardware; this hypothesis has not been validated.

The example of a periodic square wave reference R, taken from the heater on/off signal is demonstrated below. This reference is very similar in shape to the step change flow temperature but has unit magnitude [0,1]. The periodic calculation is shown to achieve the same linear dependence of q^* and T^*, on both the upward and downward step, enabling multiple step changes to be evaluated in the same test run.

$$R(x,t) = \frac{1}{2}\Big[sgn\Big(cos(2\pi f t)\Big) + 1\Big] \ \mathrm{erfc}\Big(\frac{x}{\sqrt{4\alpha t}}\Big) \tag{3.49}$$

$$r = \frac{F_3 * R(0,t)}{F_1 * R(0,t)} \tag{3.50}$$

$$F_3 * F_{Tq} * T(0,t) - r\Bigg(F_1 * F_{Tq} * T(0,t)\Bigg) = h \ \Bigg[r\Bigg(F_1 * T(0,t)\Bigg) - F_3 * T(0,t)\Bigg] \tag{3.51}$$

$$q^*(0,t) = h \ T^*(0,t) \tag{3.52}$$

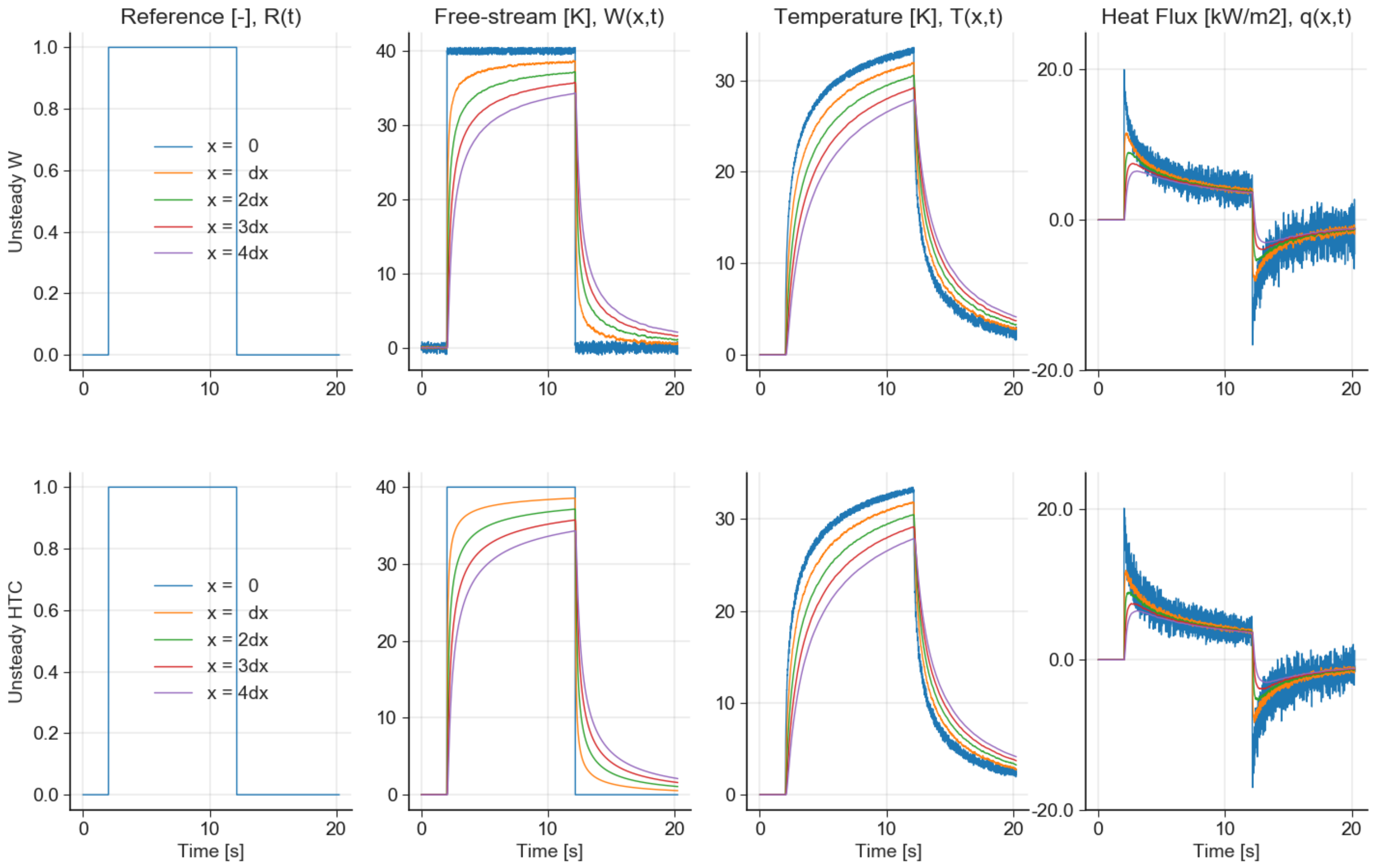


Figure 3.6: Plots of the square wave reference function, free-stream step temperature, solid temperature and solid heat flux, for the two cases of unsteady free-stream temperature (W) and unsteady HTC. Solutions at several depths, dx_i, are shown where dx_i is given by the time step penetration depth $dx_i = \sqrt{4\alpha t_i}$.

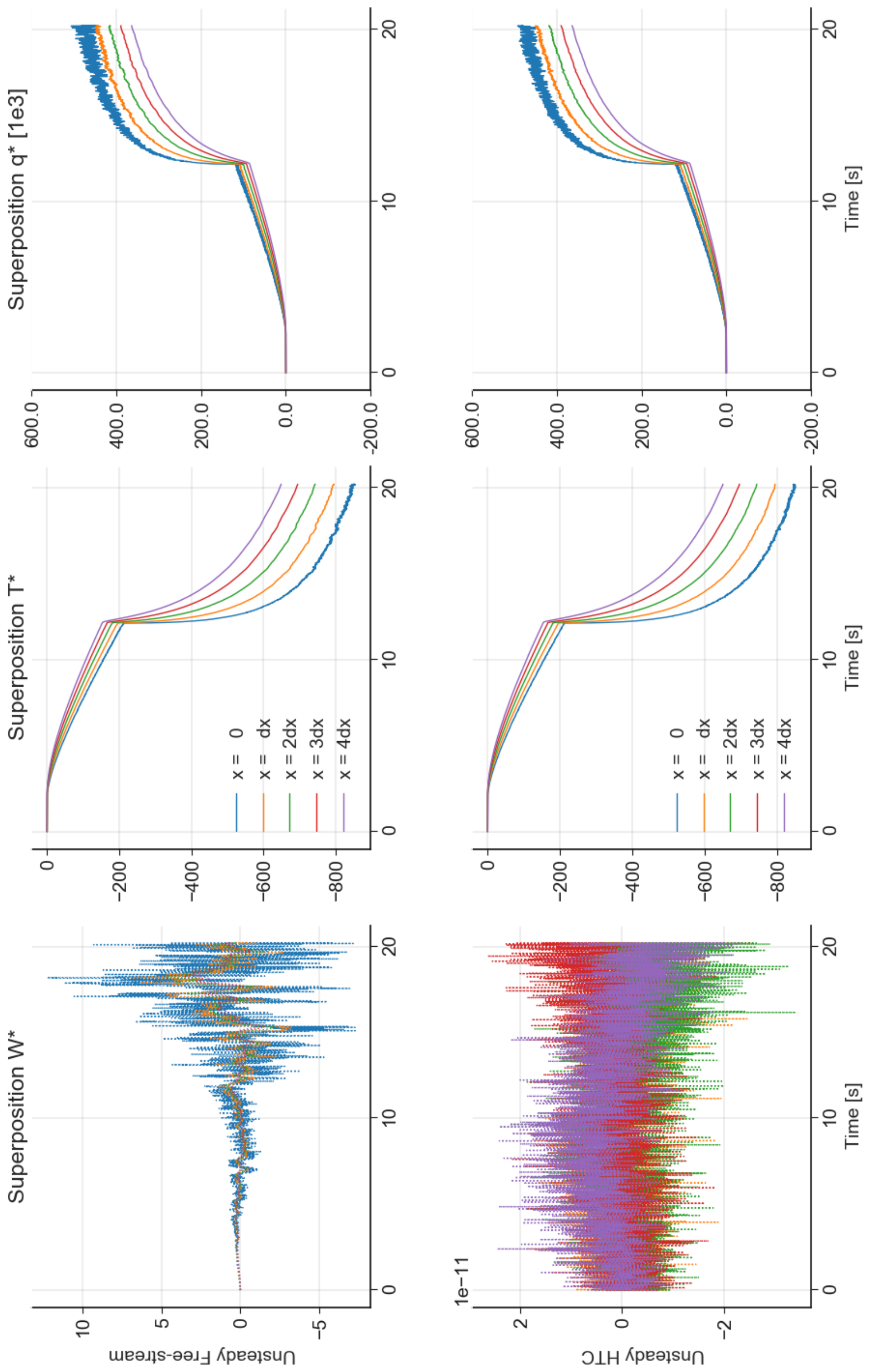


Figure 3.7: Plots of the superposition functions for the free-stream step temperature, solid temperature and solid heat flux, for the two cases of unsteady free-stream temperature (W) and unsteady HTC, demonstrating that W* can be ignored in the HTC calculation. Solutions at several depths, dx_i, are shown where dx_i is given by the time step penetration depth $dx = \sqrt{4\alpha t_i}$.

The new integral superposition method has been validated and compared to the legacy $T - q$ graphical fit method for two artificial test signals. The artificial signals were preferred for validation because this enabled unsteadiness in HTC and flow temperature to be decoupled, their relative impacts assessed and a full comparison of the legacy and new process on a known HTC. The first signal had constant HTC with unsteady square wave free-stream temperature, the second had uniform square wave flow temperature with unsteady HTC. The results are displayed in Figure 3.8 and shows in both cases that the integral superposition outperforms the legacy $T - q$ fit. The temperature and flux signals of the two cases are almost identical and it is not possible from these measurements alone to determine the source of unsteadiness. Care should be taken when interpreting raw $T - q$ plot results, especially when inferring unsteady HTC behaviour without full knowledge of the local fluid temperature stability.

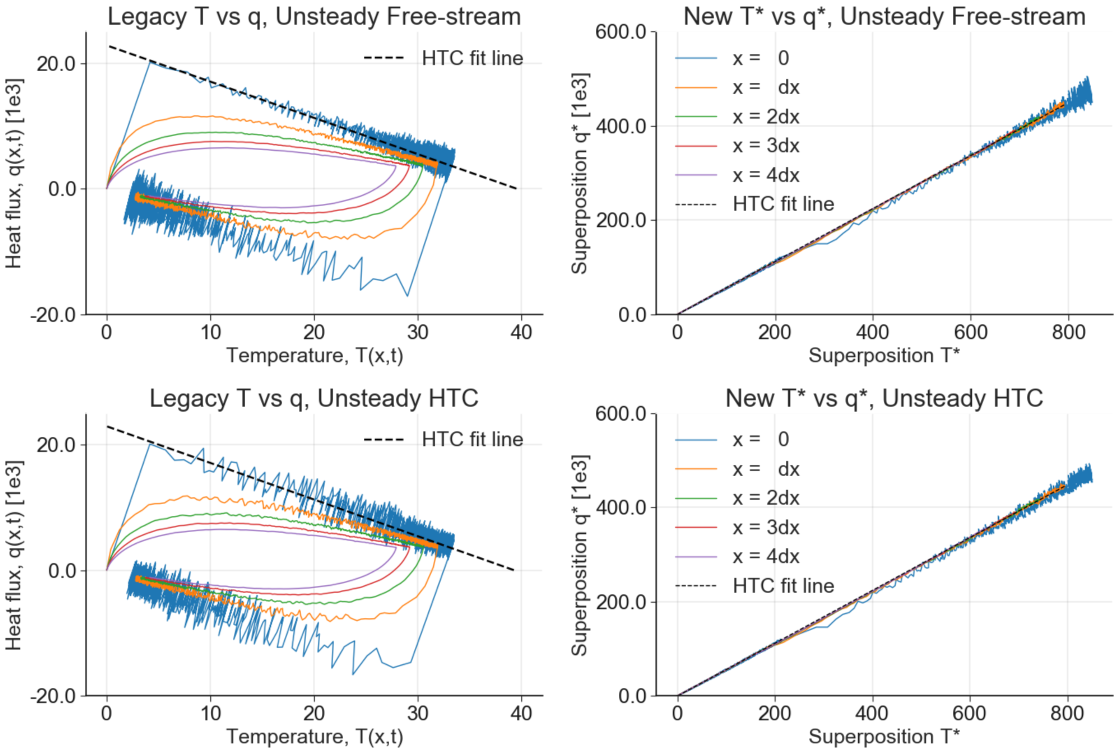


Figure 3.8: Comparison of the legacy T-q plot method and the new T*-q* method, for the data in Figure 3.6, demonstrating the effectiveness of the new method in handling subsurface data and the improved HTC line fit that passes through (0,0).

3.6.3 Heating and Cooling HTC Line Fit

Traditionally only the positive step is analysed, taking data after the heater mesh is turned on, to when the adiabatic temperature or semi-infinite limit is reached. The negative step, corresponding to the heater mesh being turned off, is generally ignored. For thick walled geometry, 5-10mm of 3D printed SLA, the test is often stopped before the semi-infinite limit.

As the wall approaches the adiabatic temperature, the heat flux and change in temperature is very low. The data points in the T-q plane become clustered and the additional data has minimal value. In these cases, early stopping of the heating cycle and using the cooling cycle can provide better results. The negative step change will partially affect the physical flow properties and boundary layer development. Care should be taken to ensure these effects do not impact the constant HTC assumption; which can be verified by checking the linearity of the HTC fit line.

Switching to a relative negative free-stream temperature (a temperature below the initial ambient of the rig), resets the peak surface heat flux and extends the range of the T-q plot. For the same flow, the negative step has the same gradient in the T-q plane. Using both the positive and negative steps provides data over a wider range and a more reliable linear fit. Care must be taken to evaluate the semi-infinite limit. Switching the heater off, to a temperature above the initial ambient of the rig, does not give the desired effect and acts instead to reduce the allowable semi-infinite duration.

The best results are achieved when the step heating and cooling cycles are analysed independently. Due to the non-ideal step, the cooling phase has some fast decay transients which affect the integral calculation. Data should first be zeroed and analysed for the heat-up step, then separately zeroed and analysed for the cool-down step, thereby artificially removing these discontinuity effects. The resulting data for both the hot and cold regions has a high degree of linear correlation, suggesting the constant HTC assumption is valid for the temperature range of the experiment.

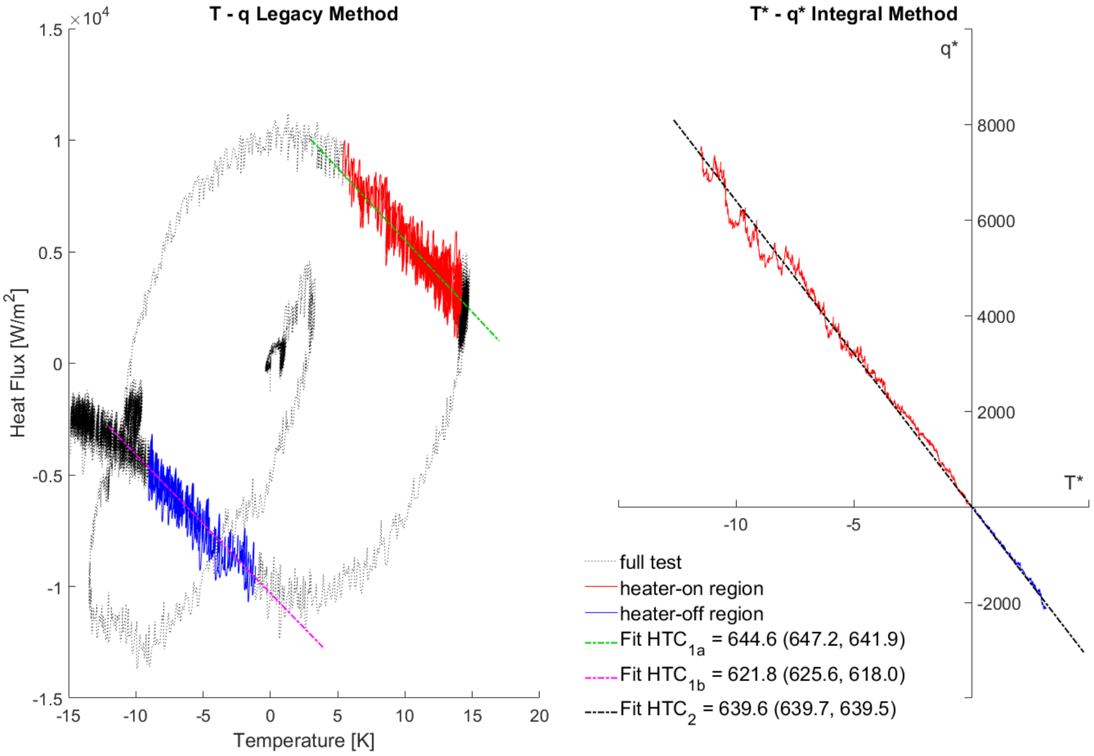


Figure 3.9: Experimental data comparison of the legacy T - q plane method (HTC_{1a} and HTC_{1b}) and the new integral method (HTC_2), highlighting the heater-on and heater-off regions of the test. The calculated HTC values for each fit line are shown in the legend, with the 95% confidence bounds of the fit. The integral T* - q* method results in the highest confidence fit, passing through (0,0) and both regions of the data.

3.6.4 Differential Method

In steady flow applications, the time derivative of the boundary reduces to $\dot{q} = -h\dot{T}$. The steady HTC can then be found from the plot of $\dot{q}$ and $\dot{T}$. The convection time derivatives are given below and can be used to create an impulse response for $\dot{T}$ and $\dot{q}$ directly. In practice, this differential solution is rarely applicable due to the high frequency noise present on most laboratory temperature measurements.

$$\frac{\partial T(x,t)}{\partial t} = \left[\frac{1}{\sqrt{\pi t}}\,\mathrm{e}^{x/\sqrt{4\alpha t}} - \frac{h}{\sqrt{\rho c k}}\mathrm{e}^{(h^2 t/\rho c k)}\,\mathrm{erfc}\left(\sqrt{\frac{h^2 t}{\rho c k}} + \frac{x}{\sqrt{4\alpha t}}\right)\right] \tag{3.53}$$

$$\frac{\partial q(x,t)}{\partial t} = h\left[\frac{h}{\sqrt{\rho c k}}\mathrm{e}^{(h^2 t/\rho c k)}\,\mathrm{erfc}\left(\sqrt{\frac{h^2 t}{\rho c k}} + \frac{x}{\sqrt{4\alpha t}}\right) - \frac{1}{\sqrt{\pi t}}\,\mathrm{e}^{x/\sqrt{4\alpha t}}\right] \tag{3.54}$$

3.7 Practical Test Limitations

Thin film gauges are widely used in the heat transfer community and are used extensively in the Oxford Thermofluids Institute for evaluating surface heat flux. Adhered to the surface of a test geometry, they are designed to measure the temperature with minimal intrusion to the flow. The resistance of the sensing element, usually platinum, varies with temperature and when employed in a Wheatstone bridge can be monitored throughout the test.

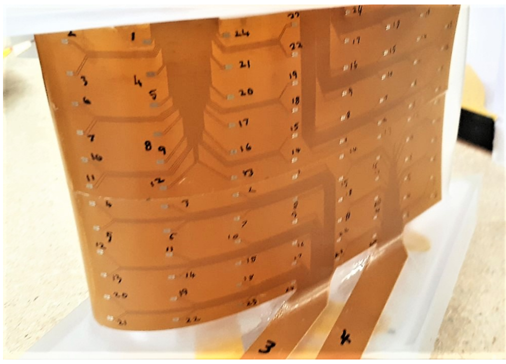


Figure 3.10: Thin film gauge installed on a nozzle guide vane.

Due to the very thin nature of these active elements, 250nm, high frequency behaviour in the order of 100kHz can be captured. The time varying resistance can be converted to surface temperature using the gauge calibration information. In turbine applications, accurate measurement of the temperature can be used to design blade cooling or examine flow phenomena and passage vortices. Real world testing introduces additional physical limitations and inaccuracies caused by the experimental design. The following effects are discussed: flow initialisation, non-ideal heater meshes, compressed gas cooling and unsteady free-stream flow.

3.7.1 Flow Initialisation

At the start of a test, all air in the system is stationary at ambient temperature. When the compressed gas valve is opened, a pressure wave passes through the test section before the flow is established. This effect settles before the heater mesh is turned on but some cooling is introduced due to the lower compressed gas temperature. It is common to omit this part of the run from the T-q plot because it is not included in the legacy gradient HTC line calculation. However, this data must not be ignored and incorrect handling severely affects the HTC fit in both the legacy and new integral method.

A critical assumption of the impulse response is that the system is initially at uniform zero state. The post-processing must therefore start from this condition. The full time history of the measured temperature must be used and the full time history of the heat flux calculated. The measured temperature must not be zeroed to remove the initialisation transients before applying the impulse response.

Once the full time history is known for both temperature and heat flux, these signals should be independently zeroed to remove the start-up transient. This method results in a vertical and horizontal shift in the T-q plane only, affecting the adiabatic temperature but not the HTC line gradient. This zeroing method is essential when applying the integral HTC calculation. Assumptions in the method do not hold during flow initialisation and this data must be zeroed before applying the integral.

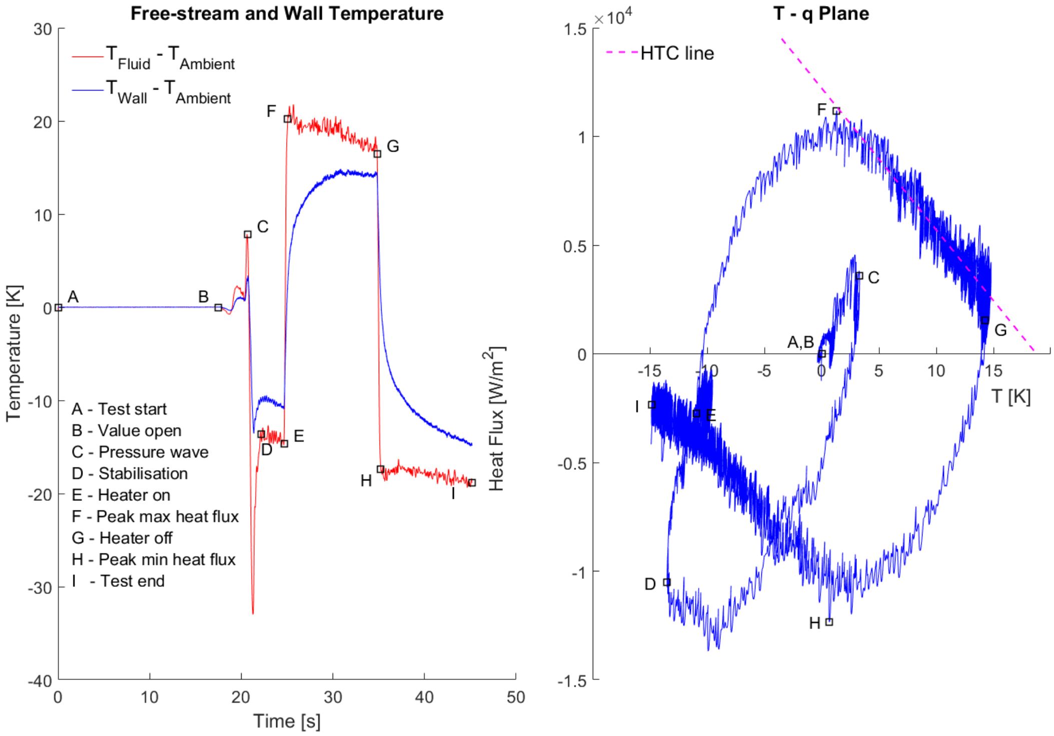


Figure 3.11: Measured wall temperature with simulated fluid temperature from a convection test with non-ideal step change and free-stream cooling.

3.7.2 Non-Ideal Heater Mesh

Although the Oxford heater mesh design produces a near unit step in free-stream temperature, it is not ideal. The power takes time to ramp up and the maximum temperature is approached asymptotically [42]. Figure 3.12 looks at the these effects, which cause a rounding of the T-q plot. This non-linearity clips the length of usable data that fits the HTC calculation line.

An assumption of the integral method is that the free-stream temperature is a known polynomial. For non-ideal heaters, this assumption is not true and the non-linear heating phase does effect the calculation. This can be handled in two ways. If the true heater profile is known, this can be used as the reference function to set the factor r in Equation 3.50. In practice the exact heater profile may be difficult to measure, in this case the data can be zeroed to a point after the heater is already at maximum temperature. This generates a pseudo ideal step in the data, thereby satisfying the assumptions without affecting the HTC calculation.

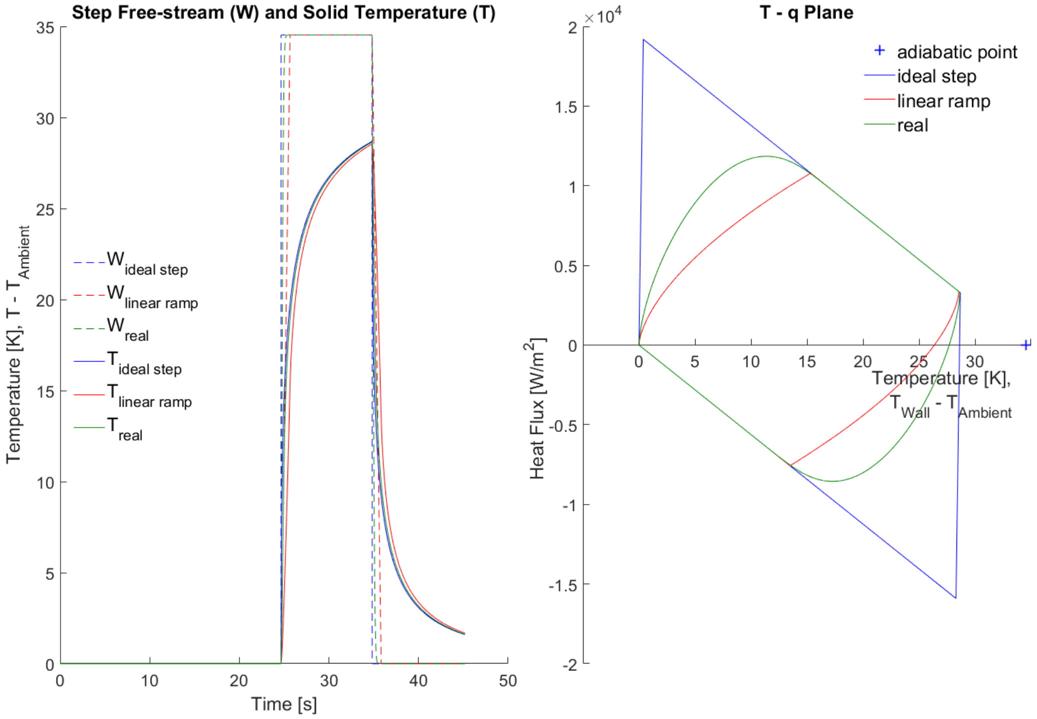


Figure 3.12: Simulated data from a convection test with ideal and non-ideal step change in local fluid temperature.

3.7.3 Compressed Gas Cooling

The use of compressed gas as the upstream feed for the main flow has two effects on the test.

1. The main compressor is housed separately to the test rig, often at a lower temperature. When the ambient air inside the rig is blown out by the test gas, a colder main flow is established. This leads to a translation in the T-q plane; the intersection of the HTC line with the x-axis is changed and the perceived adiabatic temperature is reduced.

2. For long duration tests, the air in the compressor expands and cools, slowly reducing the free-stream supply temperature. This effect causes the plot to curve downwards, effecting the late time test data. The legacy line fit thus over-predicts the HTC gradient and under-predicts the adiabatic temperature. This effect is most clearly seen at the end of a test when the heater is turned off. The T-q plot drifts towards negative values as demonstrated in Figure 3.13 and seen in the experimental data in Figure 3.11.

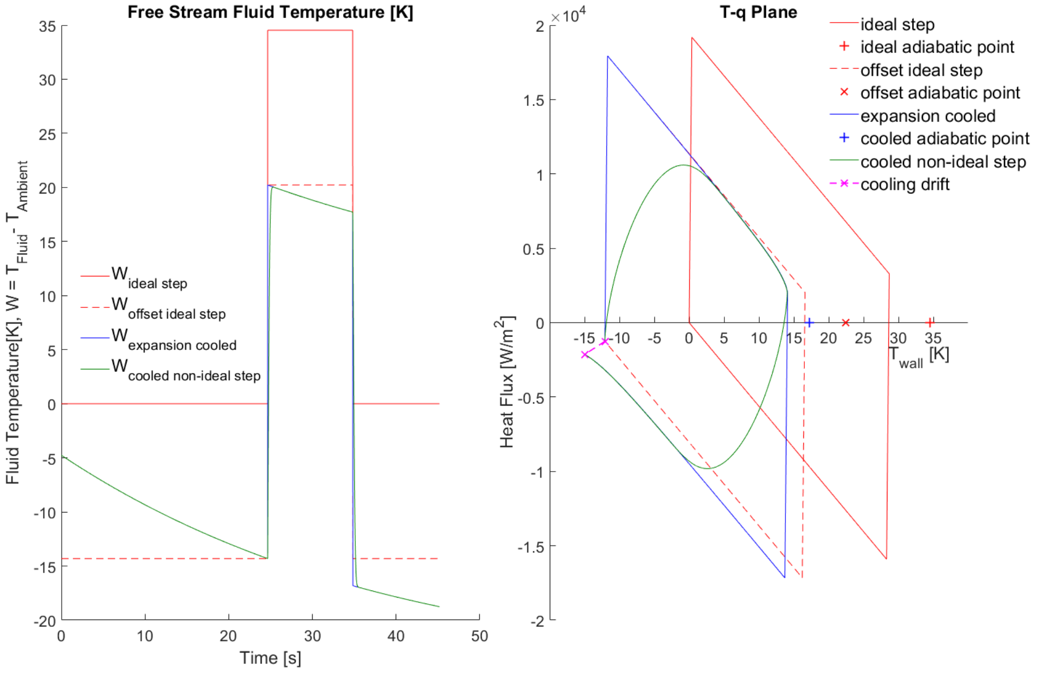


Figure 3.13: Simulated data from a convection test with different free-stream cooling effects, showing the impact on the wall temperature and heat flux in the T - q plane.

3.7.4 Unsteady Free-stream Flow

The analytic convection solutions are defined for constant constant heat transfer coefficient. Demonstrated by Maffulli [43], changes in the flow behaviour can lead to ±25% change in the observed HTC. The effect of flow unsteadiness, and sensor noise, is to introduce high frequency variance in the measured temperature and calculated heat flux. Seen in Figure 3.11, the T-q plot changes from a smooth line to a noisy plot. Best fit methods must then be used to extract the straight line corresponding to the time averaged HTC. The linearity of the data should be considered and cases that do not fit well may violate the constant HTC assumption.

Faisal Shaikh [41] presented a method to extract the time varying HTC using a best fit pivot about the adiabatic point. This method is better applied to the new integral post-processing, taking the pivot about (0,0). The method gives a good indication of the upper and lower bound of HTC. In cooling designs with mixed gases at different temperatures, the constant adiabatic point assumption likely becomes invalid. Large parts of the measured unsteadiness may be caused by the varying flow temperature.

Shaikh also presented an advanced frequency decomposition method which used multiple gauges in different spatial locations to identify common sources of unsteadiness. The assumption made is that common features in different locations are the result of bulk main-stream changes and spatially unique features are caused by local HTC variation. His analysis concluded that approximately 80% of unsteadiness is attributed to the local HTC.

The new integral method was tested independently on unsteady HTC and unsteady free-stream temperature. Both cases were shown previously in Figure 3.8 and the integral approach performed better than the legacy method. These two unsteadiness effects cannot be decoupled and the true source of unsteadiness remains unknown. Although better than the legacy approach, the integral method still suffers in the case of highly localised, large transient flow temperatures.

3.8 Summary

Analytic solutions for the 1D temperature and heat flux have been presented for both conduction and convection cases. Implemented via the Laplace domain, the full spatial-temporal solutions are more accessible and surface simplifications are not required. The solutions have been used to validate the impulse response method for high frequency transient analysis. Applied to the case of linear cascade testing, common misconceptions in the accuracy and suitable duration of Oldfield's unit step method have been highlighted. A new general purpose definition for the semi-infinite limit has been defined, extending application beyond a simple unit step case.

A new integral calculation method, using multiple impulse response filters, has been presented to solve the steady heat transfer coefficient in the case of bulk unsteady free-stream temperature. Employing properties of the $i^n erfc$ family, free-stream effects can be removed. This opens the door for new heater configurations and experimental rig designs provided due consideration is given to the constant HTC assumption. The integral method additionally offers the ability to handle subsurface data, enabling future use of embedded or multilayer probes.

The practical implications of real world convection experiments have been discussed, along with the effects these have on the measured data. Guidance on post-processing has been given, demonstrating how to mitigate these effects where possible. Thin films will continue to play an important role in experimental turbine analysis. Accurate measurement of the heat flux and heat transfer coefficient is vital to help calibrate simulation models for monitoring purposes. Where the assumptions of the impulse response hold, this method remains highly effective in the analysis of high frequency transient thermal data.

4

Multilayer 1D Heat Transfer Analysis

4.1 Introduction

Heat transfer systems often include laminates, cladding, insulation or assemblies with differing material properties. Traditional 1D impulse response methods assume a uniform, isotropic, semi-infinite material and do not accurately model the true behaviour of such systems. In this chapter, the errors caused by this assumption are evaluated and the required modifications for multilayer impulse response basis functions are outlined. Analytic and numeric solutions are presented to solve the laminate problem, required for evaluating insulated steam turbine casings, modern day thin films and future high temperature flux sensors for online monitoring.

4.2 1D Two-Layer Solutions

Doorly and Oldfield [46] investigated the behaviour of two-layer systems for different electrical insulators on a metal substrate. They showed that for long duration tests, when heat penetrates to the back substrate, a laminate construction has a significantly different behaviour to the single material case.

Oldfield published the analytic solutions in Equation 4.1 for a two-layer laminate to account for the different material properties of a simple kapton thin film gauge and metal test section. This dual-layer analytic solution has a more complicated basis function and many users choose to ignore this effect to simplify their analysis. The impact of this simplification is not well understood and this decision is often taken without validation or justification of the resulting error.

The Oldfield analytic solutions for an upper material ϕ_1 with thickness a and semi-infinite lower material ϕ_2 [46].

$$\phi_1(x,s) = \frac{\psi(0,s)}{k_1}(\alpha_1/s)^{\frac{1}{2}}\left(\frac{(1+\sigma)\ \exp[-(x-a)(s/\alpha_1)^{\frac{1}{2}}] + (1-\sigma)\ \exp[(x-a)(s/\alpha_1)^{\frac{1}{2}}]}{(1+\sigma)\ \exp[a(s/\alpha_1)^{\frac{1}{2}}] - (1-\sigma)\ \exp[-a(s/\alpha_1)^{\frac{1}{2}}]}\right)$$

$$\phi_2(x,s) = \frac{2\psi(0,s)}{k_1}(\alpha_1/s)^{\frac{1}{2}}\left(\frac{\exp[(a-x)(s/\alpha_2)^{\frac{1}{2}}]}{(1+\sigma)\ \exp[a(s/\alpha_1)^{\frac{1}{2}}] - (1-\sigma)\ \exp[-a(s/\alpha_1)^{\frac{1}{2}}]}\right) \tag{4.1}$$

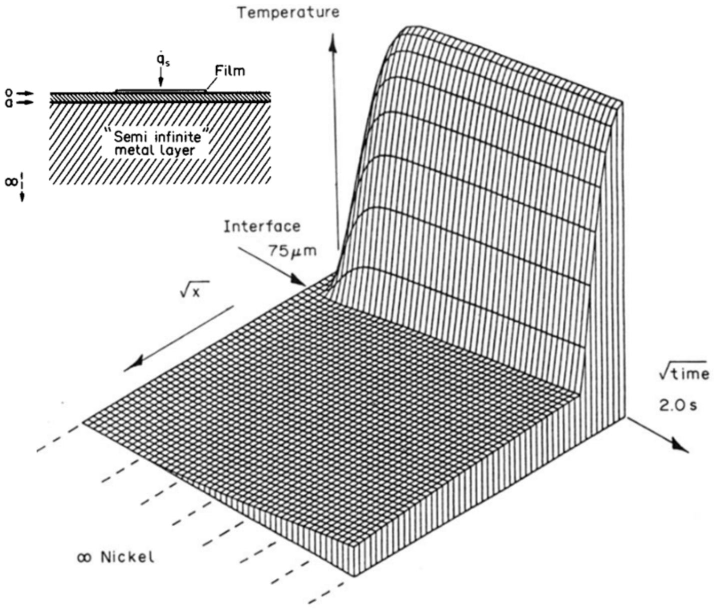


Figure 4.1: Analysis by Doorly and Oldfield of a two-layer system with a $75\mu m$ kapton thin film on a nickel substrate showing the distance-time temperature field for applied unit step surface heat flux [46].

When using thin film gauges, the gauge is adhered to the surface of the test piece. Even the simplest construction, gauge - adhesive - substrate, is a multilayer system and beyond the scope of Oldfield's single and two-layer solutions. The error introduced by a single layer simplification is evaluated in the following section and the general equation for the surface temperature of an n-layer laminate is defined for unit surface heat flux.

4.3 1D Multilayer Solutions

Thin film technology has advanced significantly since the work of Doorly in 1987. Collins[47], Ramm and Shaikh [41] introduced new manufacturing techniques using commercial flexible element PCBs. The upgraded manufacturing methods introduced interstitial or coverlay laminates in the thin film, which enable complex circuitry to be embedded in the PCB. Combined with high accuracy production tolerances, these new designs offered a higher density of thin films on the active surface. Along with the internal adhesive, main fixing adhesive and test geometry substrate, modern thin film analysis must always be considered a multilayer system.

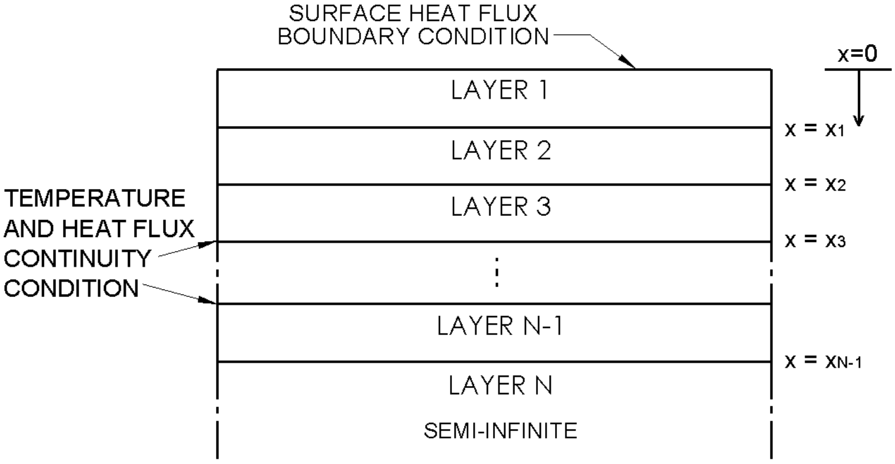


Figure 4.2: Section through a multilayer laminate system, e.g. a modern-day thin film gauge adhered to a semi-infinite substrate.

The n-layer laminate solutions are rather involved, especially in the case of high layer count. Given reluctance within the community to adopt the two-layer solutions, it is possible a more complex case will also see resistance. Two methods are therefore presented; the first is a complete analytic solution, propagating the boundary conditions through the laminate. This is non-trivial to implement and requires an adaptive code structure. The second is a significantly easier method, still with a high level of accuracy, capturing only the main features of the laminate effect. Conceptually identical to electromagnetic wave transmission and reflection, it can be routinely applied to single layer, two layer or complex laminate constructions.

4.3.1 Boundary Condition Method

The full laminate basis functions are derived in the same approach as the single layer case. Starting at the final layer, N, interface boundary conditions are propagated upwards through the laminate to the known top layer heat flux condition. The full derivation can be found in Appendix A and results in the following Laplace domain surface temperature solution $\phi(0, s)$.

$$\phi(0, s) = \frac{1}{k_1 \lambda_1 s} \left[1 + 2 \sum_{i=1}^{\infty} \eta_{1,2}^{i} \, \mathrm{e}^{-2 x_1 \lambda_1 i}\right] \tag{4.2}$$

with the following variables.

$$\lambda_n = \sqrt{s/\alpha_n}, \qquad \sigma_{m,n} = \frac{k_n \lambda_n}{k_m \lambda_m}, \qquad \gamma_{m,n} = \frac{1 - \sigma_{m,n}}{1 + \sigma_{m,n}}$$

$$\eta_{m,m+1}^{i} = \sum_{k=0}^{\infty} \left[\sum_{j=0}^{i} \binom{i}{j} \binom{i+k-1}{k} (-1)^k \, \gamma_{m,m+1}^{i-j+k} \, \eta_{m+1,m+2}^{j+k} \, \mathrm{e}^{-2(k+j)(x_{m+1}-x_m)\lambda_{m+1}} \right]$$

The above formulation of $\eta_{1,2}$ depends on the next laminate layer value of $\eta_{2,3}$. This variable must therefore be progressively substituted until the solution propagates downwards through the laminate. Each time this variable is replaced, a dual summation of the $\eta_{m+1,m+2}$ term in the next layer is required. This substitution is continued until $m > N - 2$ where the value of η depends only on γ and is fully defined by the material properties. This approach leads to a final solution with $1 + 2(N - 2)$ nested summations containing the material properties and thickness of each laminate layer.

Due to the variable number of nested summations for different layer counts, and the dependence of each summation on the index of the previous layer, dynamic nested for-loops are required to code this solution. This can be achieved by predefining a column matrix of each summation index i, j, k then iterating through the columns of this matrix at run time.

For a high number of materials in the laminate, the calculation time of the nested summation terms is significant. In such cases, the infinite series approximation may be truncated earlier, when the value of η falls below a defined negligible value or reaches a pre-set index. This cut-off index is material dependent and must be defined on a case by case basis. A typical value for negligible consideration is a summation term $< 10^{-6}$, which usually results from an integer cut-off index of $k = 5$.

An example index matrix for a four layer laminate with integer cut-off, $k = 5$, is shown below. The matrix entries correspond to the indices of the truncated terms only. The numerator terms are processed in full during each iteration. The outer summation of Equation 4.2 starts at index 1, defining the first column (1,0,0).

$$\begin{pmatrix} 1&1&1&1&1&1&1&1&1&1&1&1&1&1&\cdots&5&5&5&5&5&5&5&5&5&5&5&5&5&5 \\ 0&0&0&0&0&0&1&1&1&1&1&1&2&2&\cdots&3&3&4&4&4&4&4&4&5&5&5&5&5&5 \\ 0&1&2&3&4&5&0&1&2&3&4&5&0&1&\cdots&4&5&0&1&2&3&4&5&0&1&2&3&4&5 \end{pmatrix} \tag{4.3}$$

4.3.2 Transmission-Reflection Method

When handling more complex constructions, an alternative more user friendly method is preferred. In practice, many of the summation terms in the above solution can be considered negligible and only the significant terms need be considered. Similar to the transmission and reflection of electromagnetic waves at a material interface [48], heat penetration through a laminate can be solved in the same way; this similarity gives name to the method. The incident heat at the material interface is partially transmitted to the next material and partially reflected to the original material. This is seen as a change in the thermal gradient, affecting the heat flux through each layer. These split thermal profiles continue to propagate through the laminate, splitting again at each interface. Beyond the secondary reflections, the magnitudes can be considered negligible and a simplified approximate solution can be found. Solved at the surface, $x = 0$, the temperature profile is simply the sum of the incident function and any reflected terms seen at this location. The transmission-reflection process for a three layer laminate is outlined below in Figure 4.3.

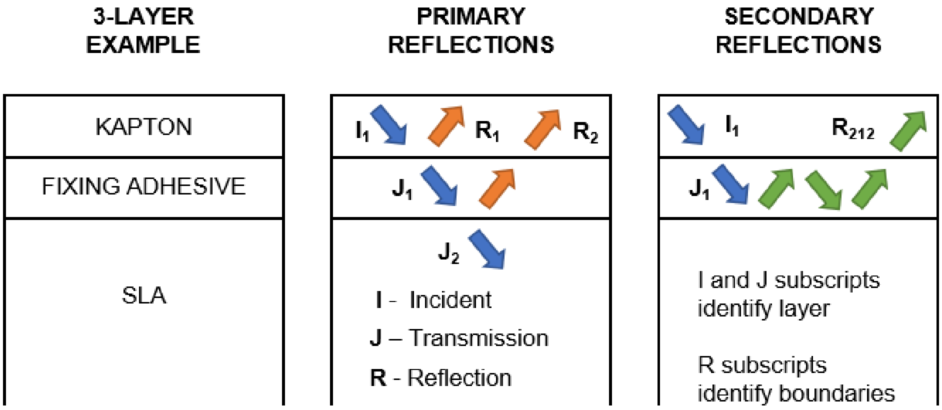


Figure 4.3: Transmission-Reflection method applied to a three layer laminate, showing the primary and secondary reflections.

Solving the unit step analytic solution adds unnecessary complication. Instead, any solution pair for temperature and heat flux may be found, then simply converted to the unit step solution afterwards if required. This is easily done using the impulse response; the filter $H(s)$ from the known heat flux solution to a unit step can be found, then applied to the corresponding known temperature. This method allows the easiest analytic solution pair to be used and further simplifies the approach. This is best implemented in the Laplace domain then inverted later to find the time domain form.

$$\phi_{unit\ step}(s) = H(s)\ .\ \phi_{known}(s)\ ; \qquad \text{where } H(s) = \frac{\psi_{unit\ step}(s)}{\psi_{known}(s)} \tag{4.4}$$

Any definition for the incident function, I_1, may be selected. The parabolic surface temperature for an isotropic material is recommended as the simplest form, and yields the easiest final temperature and heat flux solutions.

$$I_1(x,s) = \frac{1}{s^{3/2}}\ \mathrm{e}^{-x\sqrt{s/\alpha_1}} \tag{4.5}$$

At the first interface, $x = x_1$, the incident function is split according the transmission and reflection coefficients.

$$C_{Tij} = \frac{2n_i}{n_i + n_j} \ ; \quad C_{Rij} = \frac{n_i - n_j}{n_i + n_j} \ ; \quad \text{where } n_i = k_i/\sqrt{\alpha_i} \tag{4.6}$$

The transmitted part, J_1, progresses to the next material and is formed of the following components

- the transmission coefficient, C_{T12}
- the initial surface function, $1/s^{3/2}$
- the diffusion in the current layer, $exp(-[x - x_1]\sqrt{s/\alpha_2})$
- the diffusion from the previous layer, $exp(-x_1\sqrt{s/\alpha_1})$

$$J_1(x, s) = C_{T12}\ \frac{1}{s^{3/2}}\ \ \mathrm{e}^{-(x-x_1)\sqrt{s/\alpha_2}}\ \ \mathrm{e}^{-x_1\sqrt{s/\alpha_1}} \tag{4.7}$$

The reflected part, R_1, travels backwards in the first material and is formed of the following components

- the reflection coefficient, C_{R12}
- the initial surface function, $1/s^{3/2}$,
- the diffusion in this layer, $exp(-[2x_1 - x]\sqrt{s/\alpha_1})$

$$R_1(x, s) = C_{R12}\ \frac{1}{s^{3/2}}\ \ \mathrm{e}^{-(2x_1-x)\sqrt{s/\alpha_1}} \tag{4.8}$$

Note that at the first boundary interface, $x = x_1$, the following equality applies

$$I_1(x_1, s) + R_1(x_1, s) = J_1(x_1, s) \tag{4.9}$$

The transmitted part then progresses to the next material. Applying the reflection and transmission coefficients at each subsequent boundary, the remaining reflection terms from Fig.4.3 can easily be found.

$$R_2(x, s) = C_{T12}\, C_{R23}\, C_{T21}\, \frac{1}{s^{3/2}}\ \mathrm{e}^{-(2x_1-x)\sqrt{s/\alpha_1}}\ \mathrm{e}^{-2(x_2-x_1)\sqrt{s/\alpha_2}} \tag{4.10}$$

$$R_{212}(x, s) = C_{T12}\, C_{R23}\, C_{R21}\, C_{R23}\, C_{T21}\, \frac{1}{s^{3/2}}\ \mathrm{e}^{-(2x_1-x)\sqrt{s/\alpha_1}}\ \mathrm{e}^{-4(x_2-x_1)\sqrt{s/\alpha_2}} \tag{4.11}$$

The final surface temperature solution, $\phi(0, s)$, is given by the sum of all terms present at this location, $x = 0$.

$$\phi(0, s) = I_1(0, s) + R_1(0, s) + R_2(0, s) + R_{212}(0, s) \tag{4.12}$$

The heat flux, defined by $-k_1\, dT/dx$, can be found by spatial differentiation

$$\psi(0, s) = \frac{k_1}{\sqrt{\alpha_1}}\, \sqrt{s}\, \Big[I_1(0, s) - R_1(0, s) - R_2(0, s) - R_{212}(0, s) \Big] \tag{4.13}$$

The following Laplace transform inversions can be used to find the time domain solution for heat flux and temperature [37].

$$\begin{aligned} \frac{1}{s}\,\mathrm{e}^{-a\sqrt{s}} \quad &\Leftrightarrow \quad \mathrm{erfc}\left(\frac{a}{2\sqrt{t}}\right) \\ \frac{1}{s^{3/2}}\,\mathrm{e}^{-a\sqrt{s}} \quad &\Leftrightarrow \quad 2\sqrt{\frac{t}{\pi}}\,\exp\left(-\frac{a^2}{2t}\right) - a\,\mathrm{erfc}\left(\frac{a}{2\sqrt{t}}\right) \end{aligned} \tag{4.14}$$

The time domain form for both temperature and heat flux are therefore a sum of factored exponential and complimentary error functions. The factors are given by the product of transmission and reflection coefficients, the $erfc$ and exp terms are given by the sum of the $dx_i/\sqrt{\alpha_i}$ ratio in each layer passed through. The example three layer solution for surface heat flux and surface temperature are given below.

$$\begin{aligned} q(0,t) = \frac{k_1}{\sqrt{\alpha_1}} \Bigg[& \mathrm{erfc}\big(0\big) - C_{R12}\,\mathrm{erfc}\left(\frac{a}{2\sqrt{t}}\right) - \\ & C_{T12}\,C_{R23}\,C_{T21}\,\mathrm{erfc}\left(\frac{b}{2\sqrt{t}}\right) - \\ & C_{T12}\,C_{R23}\,C_{R21}\,C_{R23}\,C_{T21}\,\mathrm{erfc}\left(\frac{c}{2\sqrt{t}}\right) \Bigg] \end{aligned} \tag{4.15}$$

$$\begin{aligned} T(0,t) = 2\sqrt{\frac{t}{\pi}} + C_{R12} & \left[2\sqrt{\frac{t}{\pi}}\,\exp\left(-\frac{a^2}{2t}\right) - a\,\mathrm{erfc}\left(\frac{a}{2\sqrt{t}}\right)\right] + \\ C_{T12}\,C_{R23}\,C_{T21} & \left[2\sqrt{\frac{t}{\pi}}\,\exp\left(-\frac{b^2}{2t}\right) - b\,\mathrm{erfc}\left(\frac{b}{2\sqrt{t}}\right)\right] + \\ C_{T12}\,C_{R23}\,C_{R21}\,C_{R23}\,C_{T21} & \left[2\sqrt{\frac{t}{\pi}}\,\exp\left(-\frac{c^2}{2t}\right) - c\,\mathrm{erfc}\left(\frac{c}{2\sqrt{t}}\right)\right] \end{aligned}$$

$$\text{where } a = \frac{2x_1}{\sqrt{\alpha_1}}, \quad b = \frac{2x_1}{\sqrt{\alpha_1}} + \frac{2(x_2 - x_1)}{\sqrt{\alpha_2}}, \quad c = \frac{2x_1}{\sqrt{\alpha_1}} + \frac{4(x_2 - x_1)}{\sqrt{\alpha_2}} \tag{4.16}$$

For higher layer counts, internal reflections occur at all layers in both the upward and downward direction. To accumulate all secondary reflections, three nested loops are required in the analysis code.

- Loop one, define all first upward reflections, one at each material interface below the surface, $i = 1 : N - 1$.
- Loop two, define all first downward reflections, one at each layer above the current i^{th} layer, $j = 1 : i - 1$.
- Loop three, define all second upward reflections, one at each layer below the current j^{th} layer, $k = j + 1 : N - 1$.

The product of each transmission and reflection coefficient, along with the factors for the $erfc$ and exp terms, can be routinely collected within each loop. This method can use a static code structure and is substantially easier to implement than the full boundary condition solution. The transmission-reflection method is universal, allowing a laminate of any layer count to be automatically handled. The straightforward three loop implementation yields sufficient accuracy in this case however, if further accuracy is desired, users may add additional loop pairs to capture tertiary or higher order reflection terms.

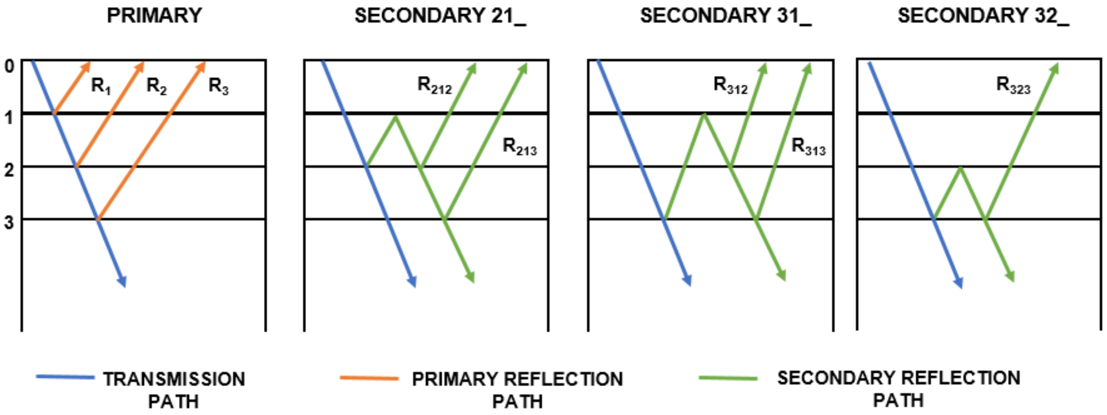


Figure 4.4: Reflection paths in a four layer laminate that are automatically handled by the standard three loop method.

Higher layer count laminates are handled in exactly the same way. A four layer laminate is shown in Figure 4.4, highlighting the captured reflection terms by the simple three loop model. Similar to the previous three layer example, the surface functions are given by the sum of terms present at $x = 0$.

$$\begin{aligned}\phi(0,s) = I_1(0,s) + R_1(0,s) + R_2(0,s) + R_3(0,s) + R_{212}(0,s) + \\ R_{213}(0,s) + R_{312}(0,s) + R_{313}(0,s) + R_{323}(0,s)\end{aligned} \tag{4.17}$$

$$\begin{aligned}\psi(0,s) = \frac{k_1}{\sqrt{\alpha_1}}\sqrt{s}\Big[I_1(0,s) - R_1(0,s) - R_2(0,s) - R_3(0,s) - R_{212}(0,s) - \\ R_{213}(0,s) - R_{312}(0,s) - R_{313}(0,s) - R_{323}(0,s)\Big]\end{aligned} \tag{4.18}$$

4.4 1D Multilayer Solution Evaluation

The analytic solutions using the transmission-reflection method have been evaluated for a single, two, three and five layer laminate with applied unit step surface heat flux $\psi(0,s) = 1/s$. The three and five layer cases examine the construction of a laboratory thin film test, where a polyamide kapton circuit is adhered to a 3D printed SLA test geometry. The material properties considered in this analysis are shown in Table 4.1 and Figure 4.5.

Material	Δx $[\mu m]$	α $[m^2/s]$	k $[W/mK]$	$k/\sqrt{\alpha}$
SLA	2500 ± 0.00	9.81×10^{-8}	0.176	561.9
Kapton	50.8 ± 1.27	7.76×10^{-8}	0.120	430.7
Coverlay	12.7 ± 0.3175	8.17×10^{-8}	0.120	419.9
Inner Adhesive	$12.7\ ^{+0.00}_{-2.00}$	10.9×10^{-8}	0.230	698.0
Fixing Adhesive	50.0 ± 5.00	8.21×10^{-8}	0.160	558.4

Table 4.1: Material properties of the typical construction layers used in a laboratory thin film test

The kapton and adhesive layers are significantly thinner than the SLA geometry, under traditional application of the impulse response method they would be ignored. Figure 4.6 shows the surface temperature function required to generate unit step heat flux under different combinations of these materials. Due to the lower thermal product of kapton and the fixing adhesive, a higher surface temperature is required to achieve unit step heat flux with these materials.

The impulse response filter is defined by a known basis function pair for temperature and heat flux. If the upper layers are ignored, an incorrect surface temperature function is used to define the filter. The error is thus directly built into the analysis filter and effects the accuracy of the post-processing only.

When applying the impulse response method, the physically measured surface temperature is post-processed with the response filter to find the heat flux. The impact of the embedded error is thus seen in the calculated flux value. Figure 4.6 shows the effect when the response filter is calculated using each of the different layer models. In all cases the calculated filter is applied to the five layer surface temperature, T_{5layer}. This solution best replicates the true geometry and thus best replicates the actual surface temperature that would be measured in a unit step heat flux test.

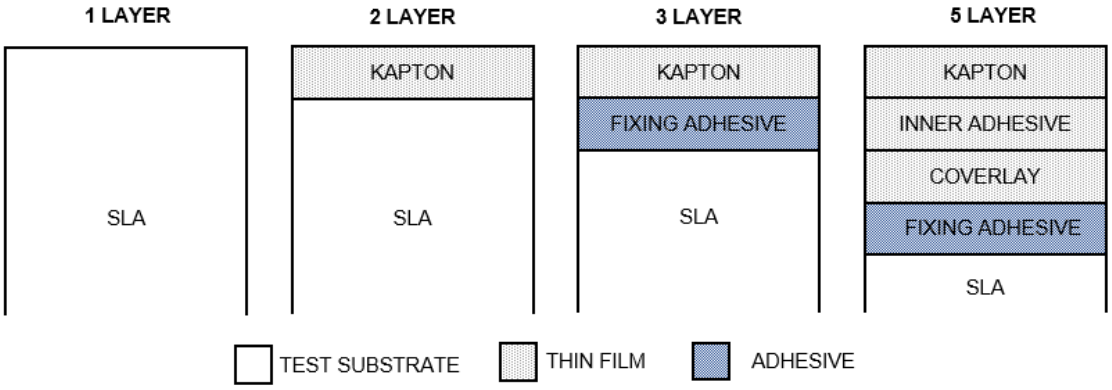


Figure 4.5: The four multilayer analysis cases, showing the laminate material construction for the different layer models

Seen in Figure 4.6, when using the simplified filters (h_1, h_2, h_3) the heat flux is overestimated, particularly during the early part of the analysis. This error reduces with time as the thermal profile penetrates past the additional layers and the back substrate material dominates. The value and duration of the large error region depends on the material properties and thermal penetration time of the additional laminate layers; this must be evaluated on a case by case basis.

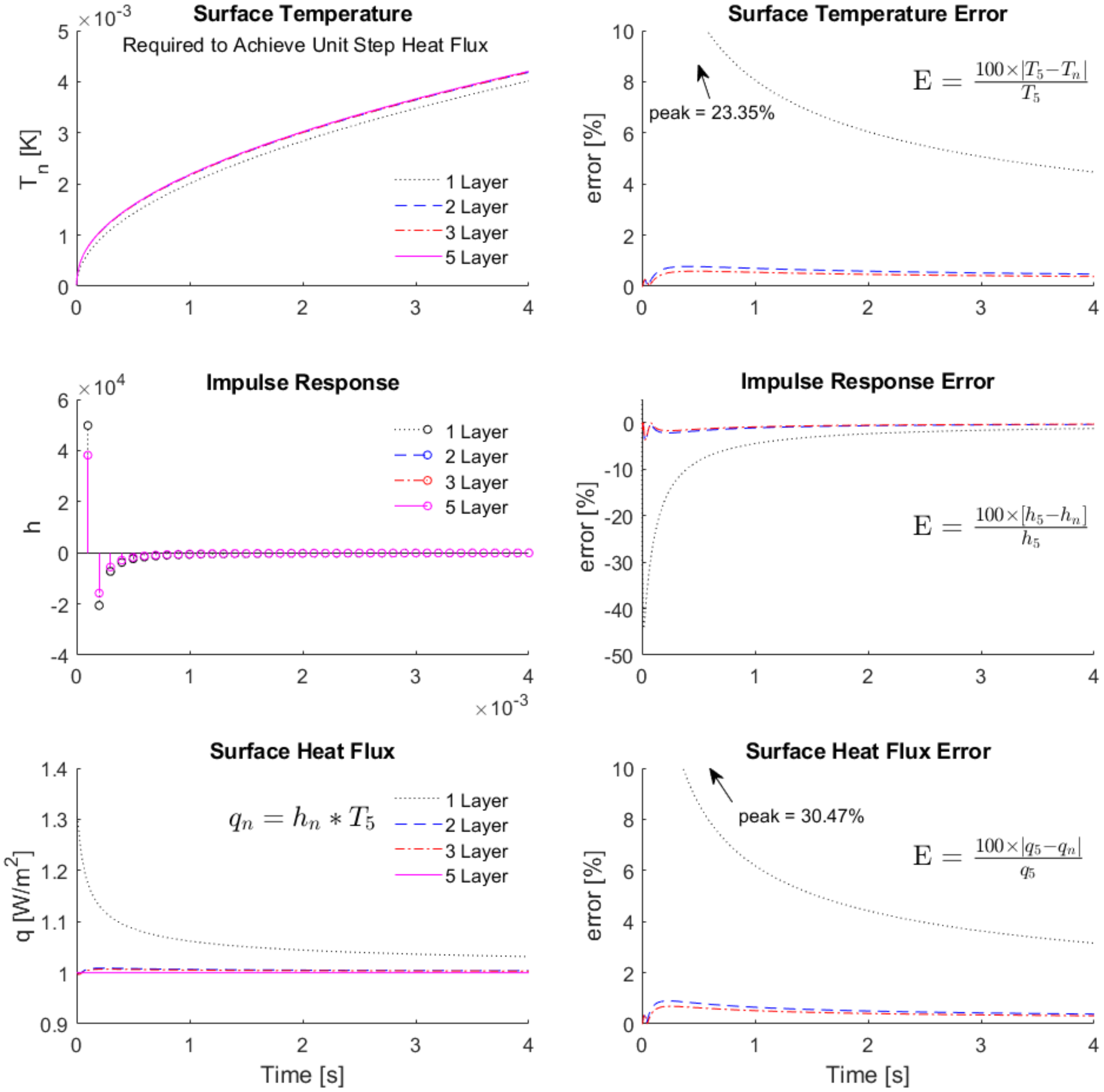


Figure 4.6: Surface temperature required for unit step heat flux, with the derived impulse response filter for the four different laminate layer cases and the calculated heat flux when applying each response function to T_5.

In this typical thin film construction, the impact of a single layer assumption is significant for the full duration of the test. Using the simplified impulse response leads to large errors in the calculated heat flux. Peak error in the heat flux calculation is 30.47%, reducing to 3.62%, with a time averaged value of 6.71%. Despite the thin nature of the additional layers, and approximately similar thermal properties, the error caused by simplifying the post-processing method is significant. When using the two or three layer impulse response, the error is notably better than the single layer case. In all cases where the test article has a laminate construction, with differing material properties, the impulse response should be derived from the multilayer system response to prevent large errors being inherited in the filter.

4.5 Manufacturing Assessment

The analysis above requires detail knowledge of the thermal properties and thickness of each layer. Accurate measurement of the thermal conductivity, especially in thin polyamide materials, is well known to have notable error bounds. Li [49] estimates the statistical uncertainty of TC-1000 thermal comparator measurement techniques to be in the region ±15% on films above $25.4\mu m$. Noting that the square root of the thermal conductivity is always referenced in the formulae, the full impact of this uncertainty is not seen in the heat flux calculation. The formulae include the thermal product of the material $k/\sqrt{\alpha} = \sqrt{\rho k c_p}$.

Measurement of the additional properties is significantly more accurate; density measurements to ASTM D-1505 are within ±0.05% [50] and specific heat capacity measurements using DSC methods are within ±3% [51]. Taking the maximum and minimum values of each property in the thermal product, the uncertainty in this value can be calculated as ±9.4%. Given the Laplace domain transfer function from temperature to flux is factored directly by the thermal product, this corresponds to the single layer material uncertainty in the impulse response calculation.

Modern-day thin films are manufactured as part of a flex circuitry panel, with several sensors cut from the same manufacturing run. The construction tolerances in Table 4.1 are considered. These are consistent across the panel and in cases where higher accuracy is required, a sacrificial sensor or dedicated coupon may be sectioned to measure the executed laminate thickness. The substrate layer is considered semi-infinite, so thickness tolerances of this layer may be ignored.

		MATERIAL TOLERANCE		
		min	nominal	max
THICKNESS TOLERANCE	min	-0.1009	-0.0005	0.0934
		0.0967	*0.0002*	*0.0898*
	nominal	-0.1004	0.0	0.0937
		0.0968	*0.0*	*0.0896*
	max	-0.1035	-0.0034	0.0900
		0.0982	*0.0016*	*0.0881*

Table 4.2: Peak Error and *Root Mean Square Error* in the calculated heat flux, caused by material property and layer thickness tolerances, compared to the true unit step response.

Table 4.2 shows the calculated bounds of the uncertainty caused by material and construction tolerances in the five layer laminate case. In each construction, the same surface temperature was applied to the laminate, corresponding to the heating profile for unit step heat flux in the nominal tolerance case, T_5 in Figure 4.6. The calculated heat flux for each combination of material and thickness tolerance was compared to the true unit step value. Table 4.2 shows the peak difference and *RMSE* in the calculated heat flux. Seen in the data, the impact of thickness tolerance is negligible, with peak error 0.34% and RMSE 0.16%, and could reliably be ignored in thin film testing.

Evaluation

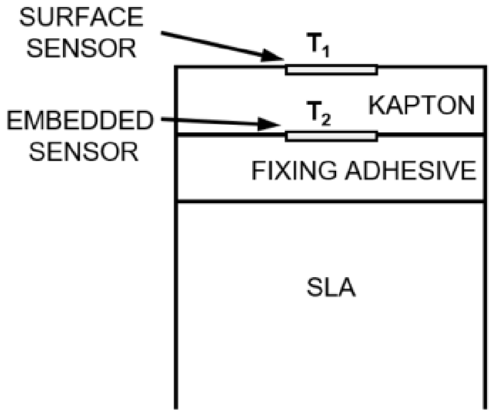
The material property effects dominate, with a peak discrepancy of 10.35% and RMSE 9.82%. This is slightly higher than the thermal product uncertainty, 9.4%, due to the additional effects in the transmission and reflection coefficients at the material interfaces. However, the thermal product uncertainty value gives a good indication of the behaviour in the full laminate.

These uncertainty values are comparable in magnitude to the time averaged error of the single layer assumption, 6.71%. However, it should be noted that even the worst case manufacturing uncertainty, 10.35%, is significantly less than the peak error, 30.47%, introduced by using the incorrect layer filter. Where possible the material properties should be closely controlled however, priority should be given to removing the known error source and using the correct multilayer response filter.

4.6 1D Embedded Solutions

The discussed examples focus on the calculation of surface heat flux from surface temperature. It should be noted that the impulse response can be taken between any two temperature or heat flux functions in the laminate. Using analytic solutions from different layers, the impulse response automatically infers both time delay and positional correction between the measurements. Subsurface embedded sensors can therefore be used to give direct output of the surface heat flux or temperature. In the case of embedded probes, the transmission - reflection method is equally suitable. The subsurface analytic solution may be found by collecting the relevant thermal profiles in that layer. The user need only correct the loop functions to sum the required terms and substitute the appropriate value of x for the spatial position of the measurement.

Seen in Section 3.5.2, the transfer function from the surface T_1 to a subsurface T_2 can easily be found, $filter(T_2, T_1, d)$. However, the reverse transfer function $filter(T_1, T_2, d)$ cannot be solved directly. Moving upwards through the material, the laplace domain transfer function is given by $exp\Big(+ x\sqrt{s/\alpha}\ \Big)$; as $s \to \infty$ this becomes infinite, resulting in a NaN filter. To solve this issue a carrier function $C(s)$ must be used. Applied to both T_1 and T_2, the laplace form of this function must tend to zero as $s \to \infty$, thereby forcing the filter to remain finite. Several carrier functions were tested, line 29.3.112 [37] proved best with additional k parameter to control the rate of decay. Values less than $1e^{-6}$ in T_1 or T_2 lead to errors in the $filter$ function and should be clipped before use.



$$C(s) = exp(k^2s^2)\ erfc(ks)$$
$$C(t) = \frac{1}{k\sqrt{\pi}}\ exp\Bigg(- \frac{t^2}{4k^2}\Bigg)$$

$$T_{1C} = fftfilt(T_1, C)$$
$$T_{2C} = fftfilt(T_2, C)$$

$$h_{12} = filter(T_2, T_1, d)$$
$$h_{21} = filter(T_{1C}, T_{2C}, d)$$

Figure 4.7: Embedded sensor schematic and calculation method for bi-directional transfer functions to compute surface values from subsurface measurements.

4.7 1D Numerical Crank-Nicolson Method

The impulse response method has limitations in the handling of time varying material properties and back surface boundary conditions. Direct numerical methods allow these limitations to be bridged and can offer comparable accuracy at the cost of computational effort. Several numerical approaches are available to solve the 1D heat equation however, some algorithms suffer with stability if the spatial and temporal discretisation is incorrect.

Described by Recktenwald [52], Crank and Nicolson introduced an algorithm with unconditional stability. The method uses a backward time difference weighted average of the discrete centre space calculation. It is implicit with second order truncation errors in both time $O(\Delta t^2)$ and space $O(\Delta x^2)$. Although computationally more expensive per time step, it is unconditionally stable and allows for larger time steps without divergence. This leads to an overall reduction in the number of steps required and lower computational cost. Applied to the 1D heat equation, the algorithm has the following form

$$\text{Backward time,} \quad \left.\frac{\partial T}{\partial t}\right|_{t_m,x_i} = \frac{T_i^m - T_i^{m-1}}{\Delta t} + O(\Delta t) \tag{4.19}$$

$$\text{Centre space,} \quad \left.\frac{\partial^2 T}{\partial x^2}\right|_{t_m,x_i} = \frac{T_{i-1}^m - 2T_i^m + T_{i+1}^m}{\Delta x^2} + O(\Delta x^2) \tag{4.20}$$

$$\text{Crank} - \text{Nicolson,} \quad \left.\frac{\partial T}{\partial t}\right|_{t_m,x_i} = \frac{\alpha}{2}\left(\left.\frac{\partial^2 T}{\partial x^2}\right|_{t_m,x_i} + \left.\frac{\partial^2 T}{\partial x^2}\right|_{t_{m-1},x_i}\right) \tag{4.21}$$

$$\frac{T_i^m - T_i^{m-1}}{\Delta t} = \frac{\alpha}{2}\left[\frac{T_{i-1}^m - 2T_i^m + T_{i+1}^m}{\Delta x^2} + \frac{T_{i-1}^{m-1} - 2T_i^{m-1} + T_{i+1}^{m-1}}{\Delta x^2}\right] \tag{4.22}$$

This can be rearranged to separate the two temporal steps: t_m and t_{m-1}

$$-\frac{\alpha}{2\Delta x^2}T_{i-1}^m + \Big(\frac{1}{\Delta t} + \frac{\alpha}{\Delta x^2}\Big)T_i^m - \frac{\alpha}{2\Delta x^2}T_{i+1}^m = \frac{\alpha}{2\Delta x^2}T_{i-1}^{m-1} + \Big(\frac{1}{\Delta t} - \frac{\alpha}{\Delta x^2}\Big)T_i^{m-1} + \frac{\alpha}{2\Delta x^2}T_{i+1}^{m-1} \tag{4.23}$$

This can be expressed more concisely in matrix form.

$$\begin{pmatrix} a_0 & b_0 & 0 & 0 & \cdots & 0 \\ c_1 & a_1 & b_1 & 0 & \cdots & 0 \\ 0 & c_2 & a_2 & b_2 & \cdots & 0 \\ 0 & 0 & c_3 & a_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & c_n & a_n \end{pmatrix} \begin{pmatrix} T_0^m \\ T_1^m \\ T_2^m \\ T_3^m \\ \vdots \\ T_n^m \end{pmatrix} = \begin{pmatrix} f_0 & -b_0 & 0 & 0 & \cdots & 0 \\ -c_1 & f_1 & -b_1 & 0 & \cdots & 0 \\ 0 & -c_2 & f_2 & -b_2 & \cdots & 0 \\ 0 & 0 & -c_3 & f_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -c_n & f_n \end{pmatrix} \begin{pmatrix} T_0^{m-1} \\ T_1^{m-1} \\ T_2^{m-1} \\ T_3^{m-1} \\ \vdots \\ T_n^{m-1} \end{pmatrix} \tag{4.24}$$

$$\text{where} : a_i = \frac{1}{\Delta t} + \frac{\alpha}{\Delta x^2},\ b_i = c_i = -\frac{\alpha}{2\Delta x^2},\ f_i = \frac{1}{\Delta t} - \frac{\alpha}{\Delta x^2}$$

The Crank-Nicolson method is commonly solved using a tri-diagonal matrix inversion on the left hand side, similar to the algorithm written by Holmes [53]. The formula is first rearranged to separate the two temporal steps and assembled to matrix form. The LHS matrix is then inverted using the Thomas algorithm or equivalent, which is computationally efficient with $O(n_x^2)$. MATLAB scripts to invert the LHS matrix are available via Mathworks Exchange [53] and have been implemented in a new object oriented numerical solver for a single material substrate.

Extending this method to a multilayer system, one must ensure that the heat flux is conserved at the interface boundaries. Hickson [54] presented a Taylor series expansion method for ensuring flux continuity at the laminate interfaces. Coefficients of nodes at locations $[-2\Delta x, -\Delta x, 0, \Delta x, 2\Delta x]$ are used, where Δx is the distance from the interface boundary. Adapted to the Crank-Nicolson scheme, this can be concisely written as a penta-diagonal matrix. To maximise sparsity and reduce the computational effort, the bulk of the solver follows the traditional Crank-Nicolson method, with only the interface nodes modified for cross-boundary heat flux continuity.

Heat flux continuity condition at the material interface:

$$\begin{aligned}\alpha_i \frac{\partial^2 T_j}{\partial x^2} = \frac{1}{6(k_i + k_{i+1})\Delta x^2} \Big[& \;[\;(2k_i + 3k_{i+1})\,\alpha_i - k_i\,\alpha_{i+1}]\,T_{j-2} \\ & + [-2\,(k_i + 3k_{i+1})\,\alpha_i + 4\,k_i\,\alpha_{i+1}]\,T_{j-1} \\ & + [\;4\,k_{i+1}\,\alpha_i - 2(3k_i + k_{i+1})\,\alpha_{i+1}]\,T_{j+1} \\ & + [\;-k_{i+1}\,\alpha_i + (3k_i + 2k_{i+1})\,\alpha_{i+1}]\,T_{j+2} \Big]\end{aligned} \tag{4.25}$$

Multilayer modified Crank-Nicolson scheme:

$$\begin{pmatrix} a_0 & b_0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ c_1 & a_1 & b_1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & c_2 & a_2 & b_2 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & e_3^* & c_3^* & a_3^* & b_3^* & d_3^* & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & c_4 & a_4 & b_4 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & c_5 & a_5 & b_5 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & c_6 & a_6 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & c_{n-1} & a_{n-1} & b_{n-1} \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & c_n & a_n \end{pmatrix} \begin{pmatrix} T_0^m \\ T_1^m \\ T_2^m \\ T_3^{*m} \\ T_4^m \\ T_5^m \\ T_6^m \\ \vdots \\ T_{n-1}^m \\ T_n^m \end{pmatrix} =$$

$$\begin{pmatrix} f_0 & -b_0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ -c_1 & f_1 & -b_1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & -c_2 & f_2 & -b_2 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & -e_3^* & -c_3^* & f_3^* & -b_3^* & -d_3^* & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & -c_4 & f_4 & -b_4 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & -c_5 & f_5 & -b_5 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -c_6 & f_6 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & -c_{n-1} & f_{n-1} & -b_{n-1} \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & -c_n & f_n \end{pmatrix} \begin{pmatrix} T_0^{m-1} \\ T_1^{m-1} \\ T_2^{m-1} \\ T_3^{*m-1} \\ T_4^{m-1} \\ T_5^{m-1} \\ T_6^{m-1} \\ \vdots \\ T_{n-1}^{m-1} \\ T_n^{m-1} \end{pmatrix} \tag{4.26}$$

where: * indicates a boundary interface and,

$$a_i = \frac{1}{\Delta t} + \frac{\alpha}{\Delta x^2}, \quad b_i = c_i = -\frac{\alpha}{2\Delta x^2}, \quad d_i = e_i = 0,$$

$$f_i = \frac{1}{\Delta t} - \frac{\alpha}{\Delta x^2}, \qquad a_i^* = f_i^* = \frac{1}{\Delta t},$$

$$b_i^* = \frac{-4k_{i+1}\alpha_i + 2(3k_i + k_{i+1})\alpha_{i+1}}{12(k_i + k_{i+1})\Delta x^2}, \quad c_i^* = \frac{2(k_i + 3k_{i+1})\alpha_i - 4k_i\alpha_{i+1}}{12(k_i + k_{i+1})\Delta x^2},$$

$$d_i^* = \frac{k_{i+1}\alpha_i - (3k_i + 2k_{i+1})\alpha_{i+1}}{12(k_i + k_{i+1})\Delta x^2}, \qquad e_i^* = \frac{-(2k_i + 3k_{i+1})\alpha_i + k_i\alpha_{i+1}}{12(k_i + k_{i+1})\Delta x^2},$$

Due to the sparse nature of both the left and right side Crank-Nicolson matrices, this equation can be stored in vector form extracting only the populated diagonal vectors $a,\ b,\ c,\ d,\ e$ and f along with the temperature vectors T^{m-1} and T^m. The penta-diagonal matrix can be solved efficiently using the algorithm in Askar and Karawia's publication [55], which is conceptually similar to the Thomas algorithm [56]. For constant material properties and time step, the method can be accelerated by pre-computing the algorithm factors $\sigma, \phi, \omega, \rho$ and ψ.

The algorithm applies row operations to the column vector of T^m. By stacking T^m vectors horizontally to form a matrix, this method can be efficiently parallelised with little impact on performance. Parallelisation requires consistent Crank-Nicolson matrix elements for all stacked temperatures. It is therefore most suitable for processing multiple datasets in the same system where material properties, Δt and Δx are identical. The row operations apply iterative subtraction to reduce the left side matrix to lower diagonal form. The RHS is then solved by standard matrix multiplication to give the vector ω, followed by forward substitution of the diagonal LHS.

$$\begin{pmatrix} 1 & 0 & 0 & 0 & \cdots \\ \sigma_1 & 1 & 0 & 0 & \cdots \\ \phi_2 & \sigma_2 & 1 & 0 & \cdots \\ 0 & \phi_3 & \sigma_3 & 1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} T_0^m \\ T_1^m \\ T_2^m \\ T_3^m \\ \vdots \end{pmatrix} = \begin{pmatrix} \omega_0^{m-1} \\ \omega_1^{m-1} \\ \omega_2^{m-1} \\ \omega_3^{m-1} \\ \vdots \end{pmatrix} \tag{4.27}$$

$$\phi_i = \frac{e_i}{\psi_i}, \qquad i = n, \ldots, 2$$

$$\sigma_i = \begin{cases} \dfrac{c_n}{\psi_n}, & i = n \\ \dfrac{c_i - \phi_{i+1}\rho_i}{\psi_i}, & i = n-1, \ldots, 1 \end{cases} \qquad \rho_i = \begin{cases} b_{n-1}, & i = n-1 \\ b_i - \sigma_{i+2}\ d_i, & i = n-2, \ldots, 0 \end{cases}$$

$$\omega_i = \begin{cases} \dfrac{y_n^m}{\psi_n}, & i = n \\ \dfrac{y_{n-1}^m - \omega_n\ \rho_{n-1}}{\psi_{n-1}}, & i = n-1 \\ \dfrac{y_i^m - \omega_{i+2}\ d_i - \omega_{i+1}\ \rho_i}{\psi_i}, & i = n-2, \ldots, 0 \end{cases} \qquad \psi_i = \begin{cases} a_n, & i = n \\ a_{n-1} - \sigma_n\ \rho_{n-1}, & i = n-1 \\ a_i - \phi_{i+2}\ d_i - \sigma_{i+1}\ \rho_i, & i = n-2, \ldots, 0 \end{cases}$$

4.8 1D Numerical Crank-Nicolson Validation

The penta-diagonal scheme has been validated against the uniform isotropic material cases for unit step heat flux and convection surface boundaries. It has also been tested on the multilayer analytic solutions used previously in this chapter.

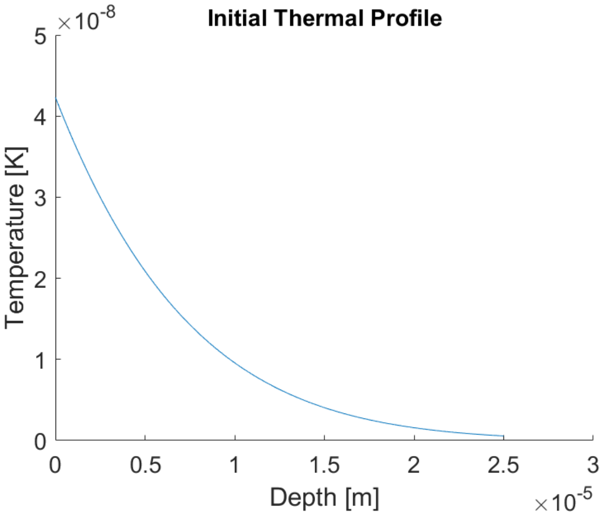


Figure 4.8: i^2erfc solution used for scaling the thermal profile after the first time step.

The spatial gradient from the solver temperature was used to extract the numerical heat flux at the surface. When modelling discontinuous flux boundary conditions, the performance can be improved by non-zero initialisation of the solver temperature. Using the analytic i^2erfc solution as the first time step, a complete thermal profile can be initialised. Scaling this profile to the first boundary temperature, this is equivalent to a linear temperature change over the first time step. This removes the discontinuity by establishing an initial thermal gradient throughout the full material. In doing so the performance is improved and the scheme converges more quickly. This approach has been used to evaluate all cases discussed below.

4.8.1 Single Material Validation

In the single material case, the full domain heat flux is within RMSE O(10^{-3}), this is comparable to the accuracy of the low and medium frequency impulse response methods. The numerical process suffers slightly at $t = 0$, where the unit step and convection temperature boundary conditions have high temporal gradients. Nevertheless, the performance is good throughout the full duration of the test and compares well with the known analytic result in both cases.

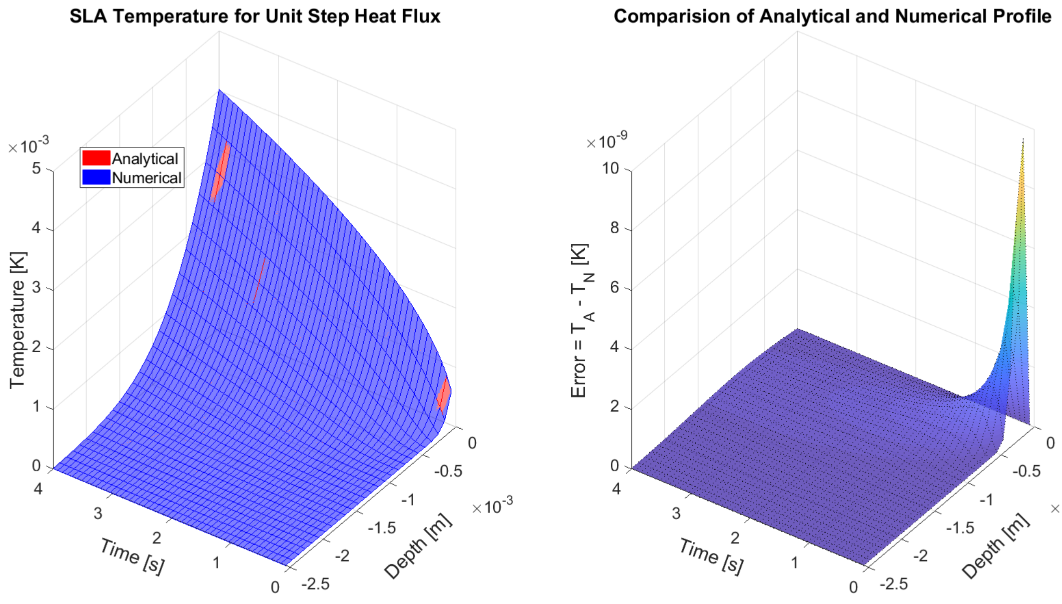


Figure 4.9: Temperature profile for SLA 7870 with applied unit surface heat flux and analysis of the Crank-Nicolson numerical error.

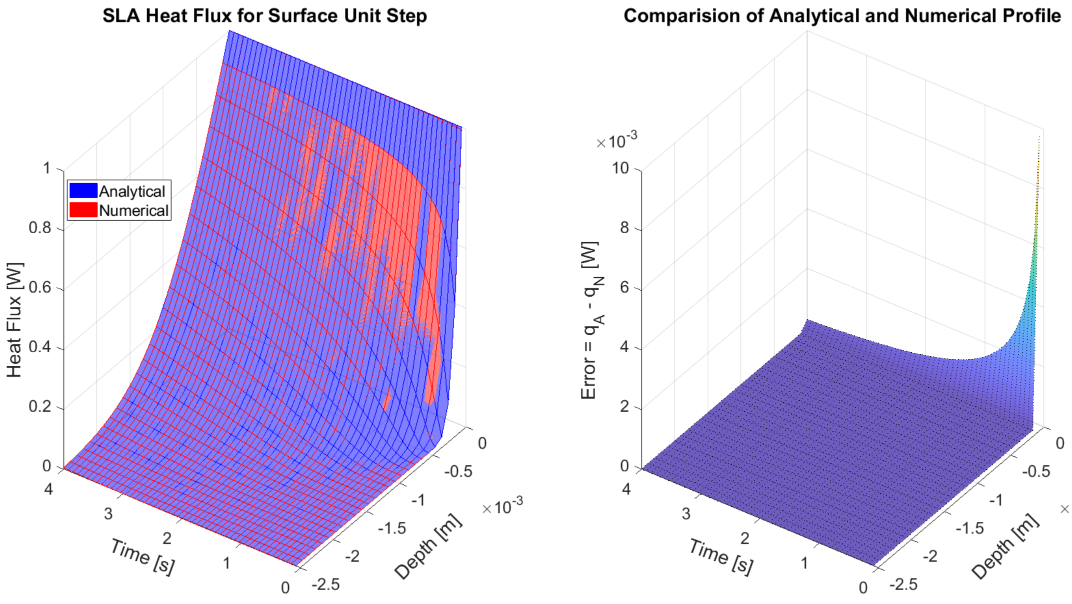


Figure 4.10: Heat flux profile for SLA 7870 with applied unit surface heat flux and analysis of the Crank-Nicolson numerical error.

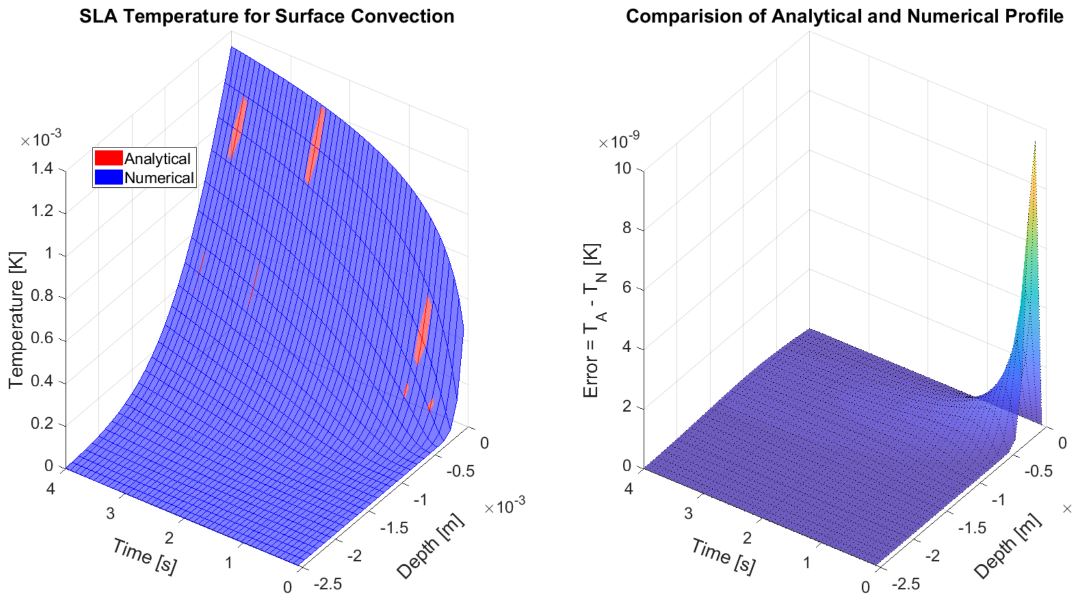


Figure 4.11: Temperature profile for SLA 7870 with applied surface convection, h=$\sqrt{\rho c k}$, and analysis of the Crank-Nicolson numerical error.

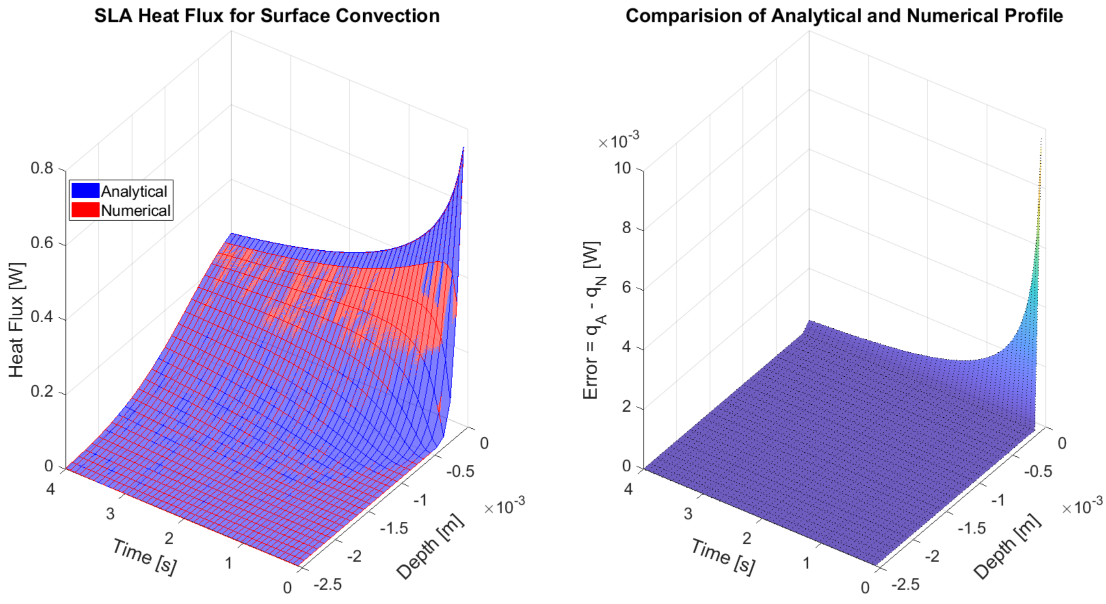


Figure 4.12: Heat Flux profile for SLA 7870 with applied surface convection, h=$\sqrt{\rho c k}$, and analysis of the Crank-Nicolson numerical error.

4.8.2 Laminate Validation

Figure 4.13 shows the multilayer performance of the numerical scheme along with a comparison to the impulse response results. The surface temperature for the full five layer laminate, T_5, was applied in each case. A model was built for each of the different laminates using the nominal layer sizes and material properties for the cases in Table 4.2 and Figure 4.5.

The temperature was specified as an upper boundary condition in the numerical simulation, using a semi-infinite back boundary. In all cases the thermal profile was initiated with the i^2erfc solution, scaled to the first time step surface temperature. The surface heat flux extracted from the model is then directly comparable to the calculation in Figure 4.6. The numerical heat flux compares well to the impulse response method in all cases. The error is largest at $t = 0$ when the thin layer transients dominate, all errors tend to zero as the time progresses and the final substrate layer defines the bulk transfer. A RMSE O(10^{-3}) between the two calculations confirms the suitability of the penta-diagonal scheme in the application of multilayer heat transfer analysis.

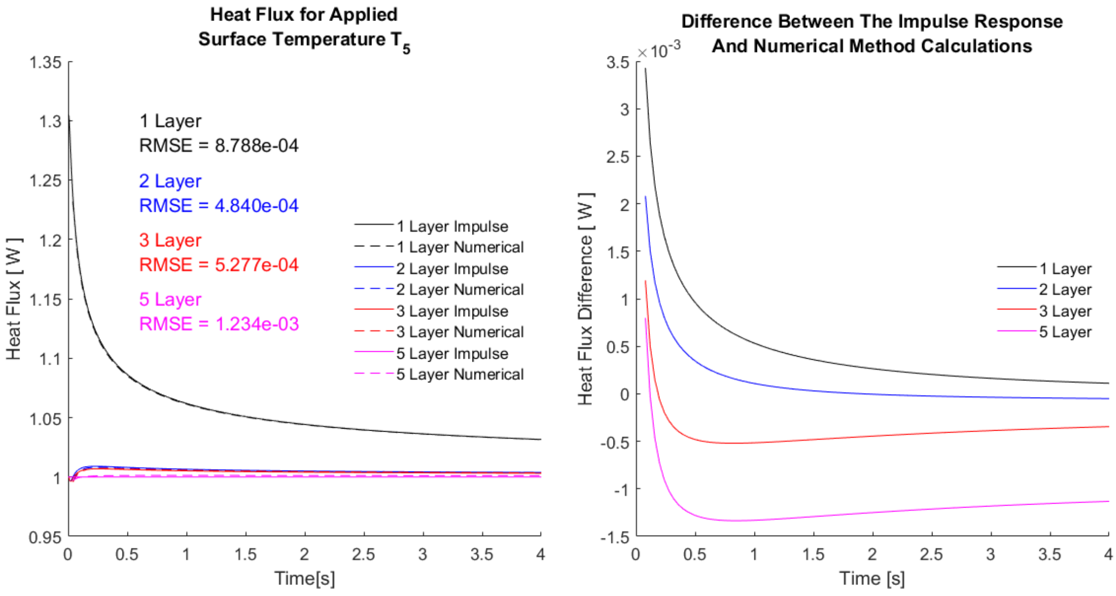


Figure 4.13: Multilayer laminate heat flux comparison of the numerical penta-diagonal scheme and the impulse response solutions.

4.9 Summary

Multilayer analytic solutions were validated and used to analyse the impact of a single layer assumption. This common simplification introduced a 30% peak error with a time averaged error of 6.7% over a four second test, corresponding to the permissible semi-infinite duration of the substrate in this case. These errors can be removed by replacing the impulse response basis functions with the correct multilayer solution. Two analytic methods have been presented and the second transmission-reflection method is easily implemented. Given the speed and simplicity of the impulse response method, in cases where the limitations of time invariance hold, this method should be preferred.

To support the calculation of time varying material properties, a numerical Crank-Nicolson method was also demonstrated for multilayer materials. This improved numerical scheme was validated against the analytic solutions and performed well across the test cases. Removing the semi-infinite and time invariant limitations, the numerical method offers a valid alternative for long duration testing or online monitoring.

Features and Capability	Single Layer Impulse	Multi-layer Impulse	Multilayer Numerical
Calculation ease and simplicity	✓	✗	✗
High speed frequency domain calculation	✓	✓	✗
Support for parallel processing of many sensors	✓	✓	✓
Support for advanced multilayer thin film gauges	✗	✓	✓
Support embedded sensor measurements	✗	✓	✓
Handle non-uniform material properties	✗	✗	✓
Handle back surface boundary conditions	✗	✗	✓
Not restricted by the semi-infinite test duration	✗	✗	✓
Variable signal length without padding	✗	✗	✓

Table 4.3: Features and capabilities of the different multilayer analysis methods.

5

Cylindrical 1D Heat Transfer Analysis

5.1 Introduction

In most practical heat transfer applications, the evaluated geometry does not only consist of planar surfaces. In such cases, the 1D Cartesian assumptions used to define the impulse response rarely hold. As a result, direct application of the tools discussed in the previous chapters leads to errors in the analysis. Particular importance is given to handling of freeform surfaces, which are especially prevalent in turbine components and casings. In the following chapter the impact of curvature is analysed and a new radial 1D method is presented to improve the accuracy of heat transfer analysis in non-planar systems.

5.2 Cylindrical 1D Heat Equation

When analysing the effects of curvature, the radial heat flow in a cylindrical system can be used. The 1D assumptions are then valid but subject to additional criteria. Firstly, the radial heat flow must be significantly greater than the axial and circumferential heat flow.

$$\text{Radial assumption} : \frac{\partial^2 T}{\partial r^2} >> \frac{\partial^2 T}{\partial \theta^2} \; ; \; \frac{\partial^2 T}{\partial z^2} \tag{5.1}$$

Secondly, when evaluating heat flux into the cylinder, semi-infinite behaviour cannot be assumed as $r \to 0$. Similar to heating a flat plate from multiple sides, which forces the 1D semi-infinite depth to be half the plate thickness; the semi-infinite duration in a cylinder is limited by the penetration depth to the centre and thus limited by the minimum radius of curvature. If these two assumptions are valid, along with the other assumptions required by impulse response analysis (isotropic, constant uniform material properties, linear time invariance, zero initial state), the heat equation in a cylindrical coordinate system can be used to define curvature basis functions.

$$\frac{\partial T}{\partial t} = \alpha\left(\frac{\partial^2 T}{\partial r^2} + \frac{1}{r}\frac{\partial T}{\partial r}\right) \tag{5.2}$$

Taking the Laplace transform with zero initial conditions, Equation 5.2 can be rearranged to give a second order modified Bessel differential equation [57].

$$\frac{d^2\phi(r,s)}{dr^2} + \frac{1}{r}\frac{d\phi(r,s)}{dr}\Big) - \frac{s}{\alpha}\phi(r,s) = 0 \tag{5.3}$$

This has the well-known general solution given by the modified Bessel functions.

$$\phi(r,s) = A(s)\ I_0\left(r\sqrt{\frac{s}{\alpha}}\right) + B(s)\ K_0\left(r\sqrt{\frac{s}{\alpha}}\right) \tag{5.4}$$

Two separate conditions must be considered.

1. Heat flux radially inwards, to a solid cylinder, bounded by the region $0 < r < R$.
2. Heat flux radially outwards, to a semi-infinite substrate with a hole, bounded by the region $R < r < \infty$.

In Case 1, the solution must remain finite as $r \to 0$ and the solution is governed by the modified Bessel function of the first kind $I(s)$. In Case 2, the solution must remain finite as $r \to \infty$ and the solution is defined by the modified Bessel function of the second kind $K(s)$.

$$\phi(r,s) = \begin{cases} A(s)\ I_0\left(r\sqrt{\frac{s}{\alpha}}\right) & 0 < r \le R \\ B(s)\ K_0\left(r\sqrt{\frac{s}{\alpha}}\right) & R \le r < \infty \end{cases} \tag{5.5}$$

5.3 Cylindrical Unit Step Heat Flux

Following the same methodology as the 1D planar case, the time domain form can be found by inverting the Laplace solution. Carslaw and Jaeger [57] presented the cylindrical solutions for unit step surface heat flux. Applying the boundary condition $q(0,s) = 1/s = -k\ d\phi/dr$ and using the following properties of the modified Bessel functions.

$$\frac{dI_0(z)}{dz} = I_1(z) \tag{5.6}$$

$$\frac{dK_0(z)}{dz} = -K_1(z) \tag{5.7}$$

The Laplace domain solutions for unit step heat flux, applied to a region bounded by a cylinder of radius R, are given by

$$\phi(r,s) = \begin{cases} \frac{-1}{s^{3/2}} \frac{1}{\sqrt{\rho c k}} \frac{I_0\left(r\sqrt{\frac{s}{\alpha}}\right)}{I_1\left(R\sqrt{\frac{s}{\alpha}}\right)} & r < R \\ \\ \frac{1}{s^{3/2}} \frac{1}{\sqrt{\rho c k}} \frac{K_0\left(r\sqrt{\frac{s}{\alpha}}\right)}{K_1\left(R\sqrt{\frac{s}{\alpha}}\right)} & r > R \end{cases} \tag{5.8}$$

Inverting these solutions, the time domain form can be found. Case 1, radially inward heat flux to a solid cylinder, bounded by the region of radius $0 < r < R$ [57].

$$T(r,t) = -\frac{R}{k}\left[\frac{2\alpha t}{R^2} + \frac{r^2}{2R^2} - \frac{1}{4} - 2\sum_{s=1}^{\infty} \mathrm{e}^{-\alpha\gamma_s^2 t/R^2} \frac{J_0(r\gamma_s/R)}{\gamma_s^2 J_0(\gamma_s)}\right] \tag{5.9}$$

$$\text{where, } \gamma_\mathrm{s} \text{ are the positive roots of } \mathrm{J}_1(\gamma) = 0$$

Case 2, radially outward heat flux, to a semi-infinite substrate with a hole, bounded by the region of radius $R < r < \infty$ [57].

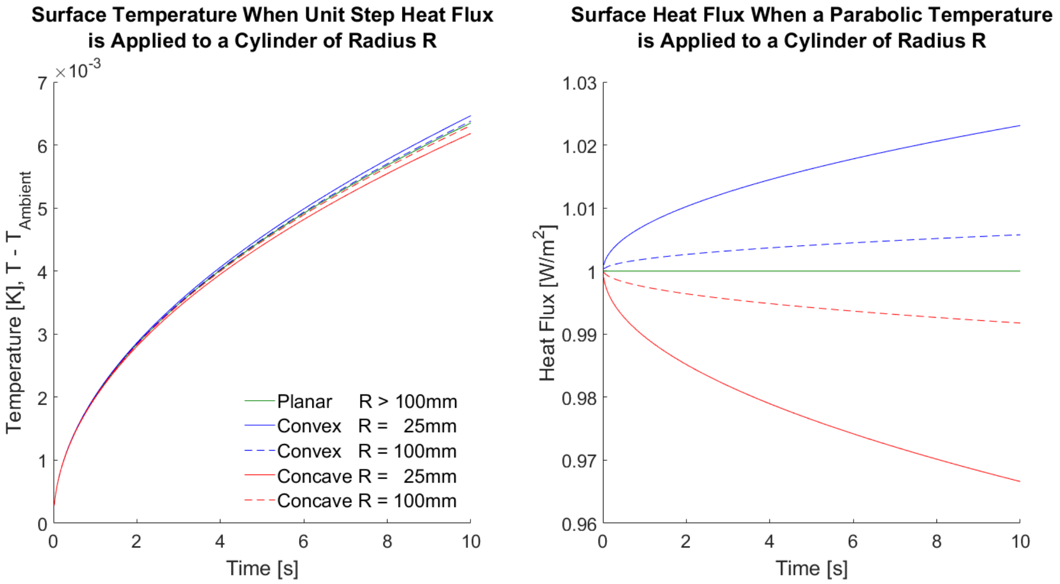
$$T(r,t) = \frac{2}{k}\left(\frac{\alpha R t}{r}\right)^{\frac{1}{2}}\left[ierfc\left(\frac{r-R}{2(\alpha t)^{\frac{1}{2}}}\right) - \frac{(3r+R)(\alpha t)^{\frac{1}{2}}}{4Rr} i^2 erfc\left(\frac{r-R}{2(\alpha t)^{\frac{1}{2}}}\right) + ...\right] \tag{5.10}$$

The solution of the second case uses the asymptotic expansion of the modified Bessel function of the second kind $K_n(r,s)$. This expansion is valid for small values of $\alpha t/R^2$ only. Carslaw and Jaeger recommend a typical limit of $\alpha t/R^2 < 0.02$. In cases where this limit is exceeded, a large time approximation is available via Equation 5.11 and has an error term $O(R^2/\alpha t)$. In the application of 3D printed SLA, for test durations of approximately 20 seconds, the small value limit is not exceeded and the above expanded form in Equation 5.10 should be used.

$$T(r,t) = \frac{R}{2k} ln\Big(\frac{4\alpha t}{\gamma r^2}\Big) + O\Big(\frac{R^2}{\alpha t}\Big) \tag{5.11}$$

where, $ln(\gamma) = 0.57722...$ is the Euler constant and $\alpha t/R^2 > 0.02$

Figure 5.1 shows the impact of the radius of curvature on the surface temperature required for unit step heat flux. In SLA, surfaces with radii greater than 100mm can be considered planar and there is negligible modification to the surface temperature function. The radial threshold is material dependent and should be evaluated on a case by case basis. For radii below the threshold value, the curvature effects should not be ignored and the system behaves notably different to a planar 1D case.


Surface Temperature When Unit Step Heat Flux is Applied to a Cylinder of Radius R

Surface Heat Flux When a Parabolic Temperature is Applied to a Cylinder of Radius R

Figure 5.1: Effect of curvature on the surface temperature and heat flux for 1D unit step solutions in a cylindrical SLA substrate.

In 2008 Martin Oldfield, Oxford Thermofluids Institute, investigated the effect of curvature and implemented basic MATLAB code to evaluate the surface temperature of the convex case only, Equation 5.9. This code is available internally at the OTI.

Buttsworth and Jones [58] analysed the effect of radial conduction on both convex and concave geometries. They demonstrated a HTC correction method to allow the flat plate solutions to approximate cases with curvature. These approximations were tested against the analytic solutions by Carslaw and Jaeger presented earlier, Equation 5.9 and 5.10. The errors were shown to vary depending on the application, with concave results being the most accurate. The error also depended on the Biot number and was generally within ±1% for a test duration to the semi-infinite limit.

Use of the Carslaw and Jaeger analytic results directly in the impulse response removes the need for a flat plate HTC correction in curved cases. This direct method is preferred and evaluated in the following sections.

5.4 Surface Curvature Calculation

3D CAD geometry is increasingly common in engineering practice. CAD software allows complex freeform surfaces to be built, often via spline or mesh specification of the surface. These models are widely used for numerical simulation or direct manufacture via 3D printing. The rise of additive manufacture has led to the development of neutral format surface topology files. Stereolithography files (.STL) are among the most common. These files segment the surface into the many faceted 2D shapes. STL files use triangulation to capture surface curvature and can be exported from most CAD or meshing software.

Rusinkiewicz [59] and Shabat [60] presented an algorithm to calculate the curvature at each vertex of an STL file. The algorithm takes a vector of face and vertex data defined in the STL, parses the normals, then returns the principal radii of curvature at each vertex. Using default settings, some CAD packages do not output unified normal files. This means that adjacent surfaces may have normals that point in opposing directions and are defined by how the user first constructed them. In such cases, the normals must first be unified so that they all aligned outwards from the geometry. Exporting an STL from a meshing package e.g. ICEM, this unification step is usually done by default. It is therefore recommended that mesh STL files be used rather than native CAD.

Johnson [61] provided MATLAB code to read standard ASCII or binary STL files, combining this with the principal radius algorithm, the surfaces of any CAD model can be evaluated easily. The effective radius of curvature can be found at any location by taking the mean of the vertex principal values.

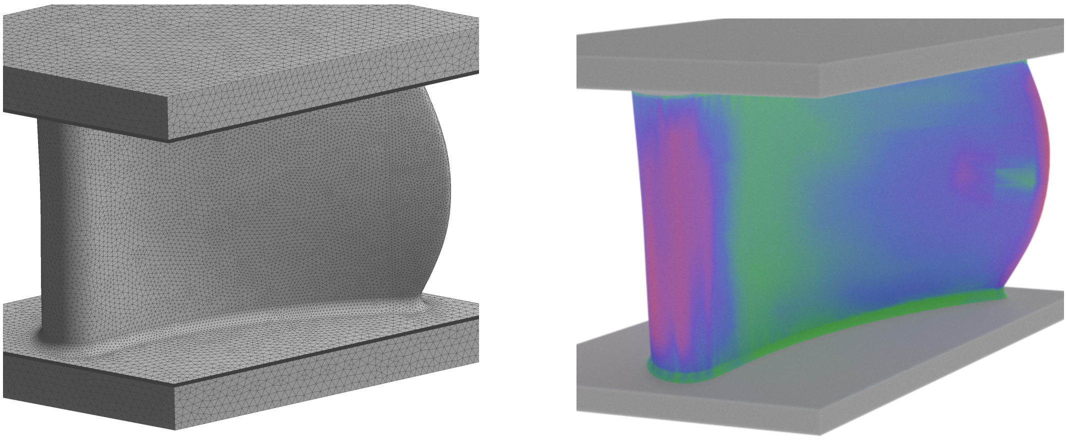


Figure 5.2: ICEM mesh of a 3D linearised turbine nozzle guide vane geometry, showing the curvature map of the main blade surfaces

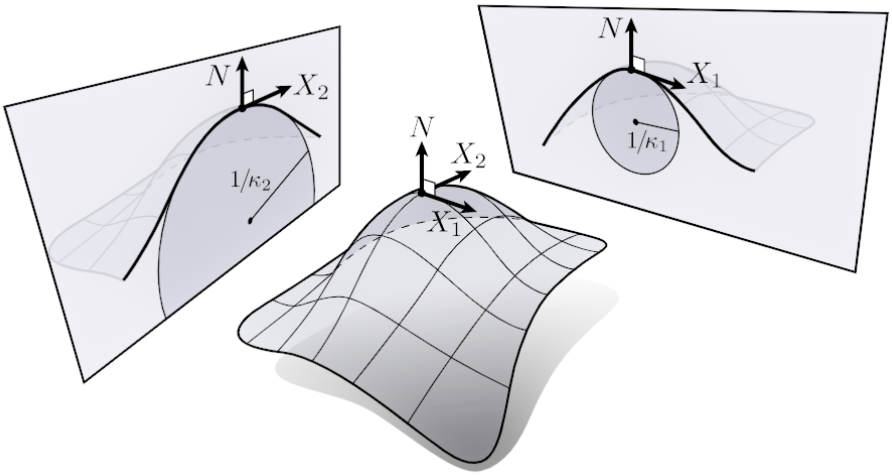


Figure 5.3: Principal radius of curvature used to define the equivalent cylindrical system, $\kappa_m = \frac{1}{2}(\kappa_1 + \kappa_2)$ [62]

5.5 Radial 1D Analysis Method

Knowledge of the surface curvature can be used to improve the 1D impulse response analysis method. Taking the CAD geometry of the test article, the principal surface curvature at each location can be found. Assuming the heat flow is normal to the surface, and that each location can be individually modelled by an equivalent cylindrical system, a unique impulse response can be calculated for each measurement position. Depending on the number of sensors installed, this may be a time consuming process but is only required once per test campaign.

Due to the unique nature of each location filter, curvature correction in this way prevents parallel processing of thermo-lithography data. Thermal camera data is often analysed pixel-wise assuming the same 1D behaviour at each pixel allowing all to be simultaneously post-processed. Curvature compensations now requires that each pixel be considered unique, having its own specific filter. This computational cost affects every test run and is likely not viable in many applications. In such cases, a trade off between accuracy and computation effort must be accepted. A defined range of filters may be applied for differing radii and pixels clustered by the most appropriate curvature.

Alternatively, rather than generate a unique response filter at each location, $F_{T_r q_r}$, one can generate the impulse response between a curved and planar system, then handle the flux calculation in an equivalent planar space. This requires three filters: two for the system transfer, F_{rx} and F_{xr}, then one for the planar heat flux calculation, $F_{T_x q_x}$. Applying F_{rx} gives the temperature that would have been measured had the test taken place on a planar geometry.

Converting the temperatures to and from a 1D planar equivalent may seem an unnecessary additional step. However, this approach does have benefits compared to the direct heat flux solution. Once in planar form, all of the features and methods discussed in the previous chapters (integral HTC method and multi-layer correction) can now be used directly, thus extending their capability to handle complex freeform geometry. The final data can either be transformed back to the cylindrical system using F_{xr} or simply left in the planar form for run-to-run comparison.

$$F_{rx} = filter\Bigg(T(x,t), T(r,t), \delta\Bigg) \tag{5.12}$$

$$q(x,t) = F_{Tq} * F_{rx} * T(r,t) \tag{5.13}$$

Although the radial method provides a significant improvement on traditional 1D planar analysis, the approach still assumes heat flux in the surface normal direction. Applications that produce high axial or circumferential temperature distributions are not well suited to this method. Similarly, geometries with regions of very small radii, such as the trailing edge of turbine blades, do not satisfy the curvature semi-infinite criteria and cannot use this approach. These cases are not considered radial-1D and should be modelled using the 3D numerical techniques discussed later in this thesis.

When handling sharp geometric features such as edges or corners, Jiang et al. [63] presented an efficient superposition method to process the differing heat penetration directions. This method is effective at handling corner geometry and is recommended if these features exist. The STL surface curvature calculation method can be used to identify regions of very high curvature and automatically tag these as edges. These regions should first be isolated for special handling before applying the radial or planar methods.

5.6 Radial 1D Validation

The radial procedure was validated by comparison to a 3D transient thermal numerical analysis completed in ANSYS. The test article chosen uses a linearised turbine Nozzle Guide Vane (NGV). This geometry contains all features of interest: planar faces, variable curvature, convex and concave surfaces, small radii and regions that violate the semi-infinite criteria.

A uniform step change in surface heat flux ($10^4 W/m^2$) was applied to all main passage faces and the time-varying temperature profile recorded for several monitor locations with differing curvature. The test was simulated for 10 seconds, corresponding to the semi-infinite duration of the geometry endwall. A surface STL file was extracted and used to calculate the effective curvature at each vertex. The radial 1D analytic temperatures, Equations 5.9 and 5.10, were compared point-wise to the monitor results. The trailing edge region of the geometry was removed from the analysis because it has a radius of curvature too low to be considered semi-infinite.

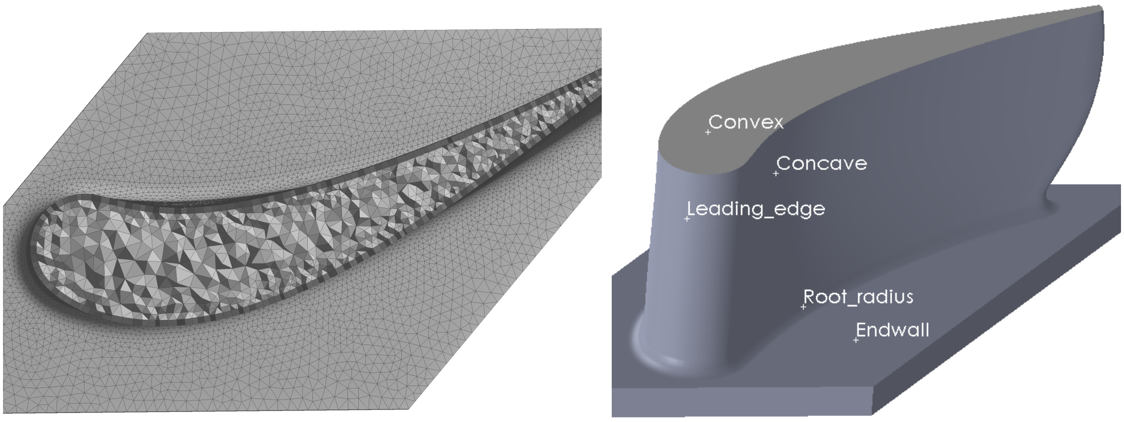


Figure 5.4: Monitor point locations used in the ANSYS simulation and a section through the NGV ICEM mesh, showing the inflation layers and tetrahedral core

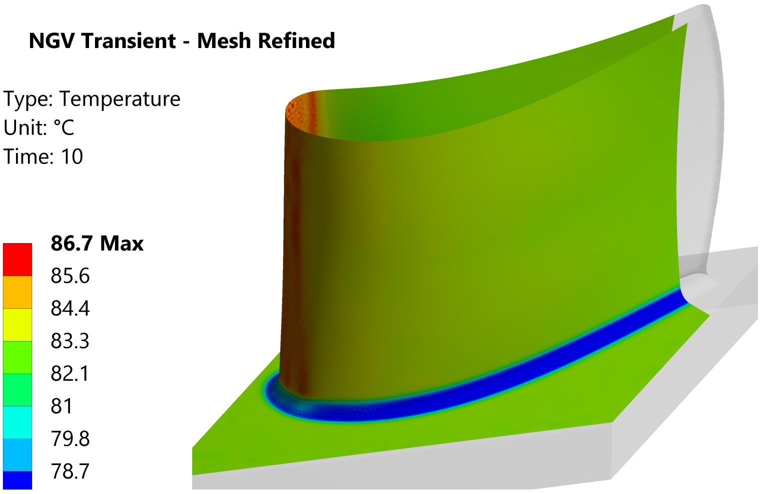


Figure 5.5: ANSYS solution of the main blade passage surface temperatures at the end of the 10 second simulation with applied $10^4 W/m^2$ surface heat flux and initial temperature 20^oC.

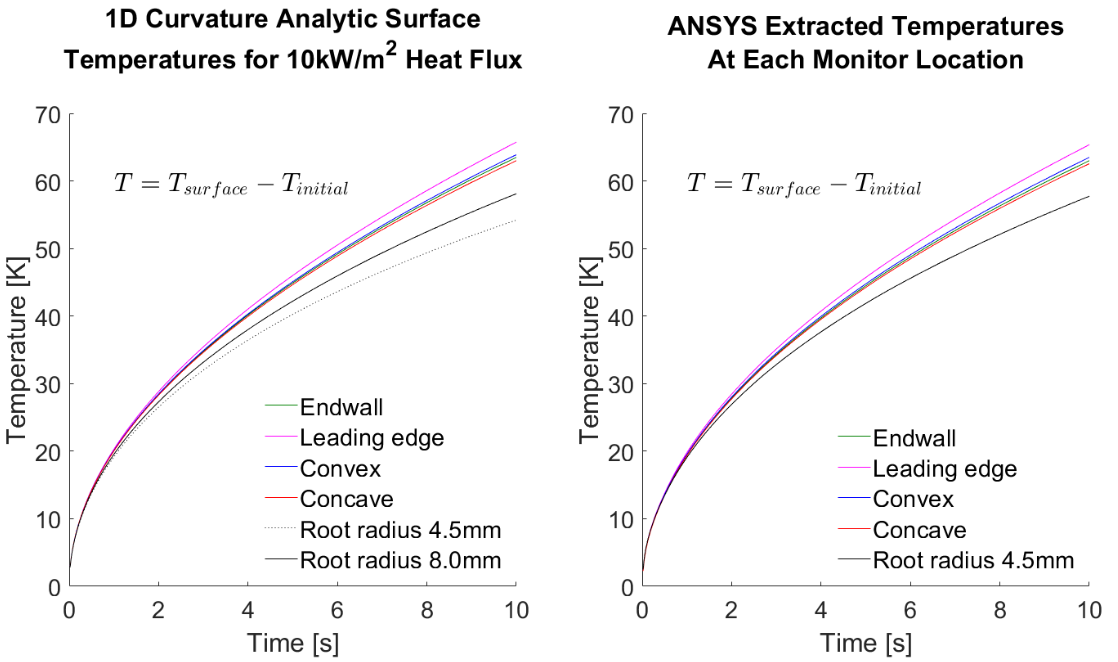
The ANSYS simulation further demonstrates that the convex regions: the blade leading edge and suction surface, heat up faster than a planar case and the concave regions: the pressure surface and blade root radii, heat up notably slower. This is due to the convergent or divergent radial heat penetration, either concentrating or diffusing the heating effect.

Figures 5.6 and 5.7 compare the temperatures in the ANSYS simulation and the analytic solutions. The error between the ANSYS surface monitor points and the analytic temperature is less than 1K for the curvature solution. Significantly higher errors are seen when using the planar 1D approximation. Only the blade root radii, where the curvature is most extreme, is any significant error seen in the curvature solution. This is caused by subsurface diffusion of heat from the adjacent regions, thereby reducing the accuracy of the 1D radial assumption. Diffusion from the endwall and the blade reduces the effective volume into which the heat is dispersed. The surface thus heats up faster than the 1D cylindrical prediction. In these cases of extreme curvature, it is required to modify the 1D cylindrical calculation. This should be done by applying an equivalent radius in the formula rather than the true value. In this application, selecting an equivalent radius of 8.0mm showed excellent agreement with the numerical simulation.

The largest temperature difference is seen immediately, at the start of the simulation. This behaviour was previously seen in Section 4.8.1, on the 1D numerical solver. The numerical error is largest when thermal gradients are first being established in the spatial simulation. The difference between the ANSYS and analytic solutions stabilises, showing a near constant offset in the latter part of the simulation. This offset is most likely caused by the discrete domain in the ANSYS simulation, and the use of a subsurface point to set the constant heat flux boundary condition.

Figures 5.8 and 5.9 compare the heat flux in the ANSYS and analytic cases. The impulse response function has been calculated for the 1D planar and 1D curvature solutions, then applied to the extracted ANSYS temperatures. Evaluating the 1D planar case, the error in traditional impulse response assumptions can be evaluated. The high curvature regions are the most effected, with large discrepancies for the root radius (11%) and leading edge (4%). The analysis also shows notable error (1.5%) across large parts of the concave blade surface.

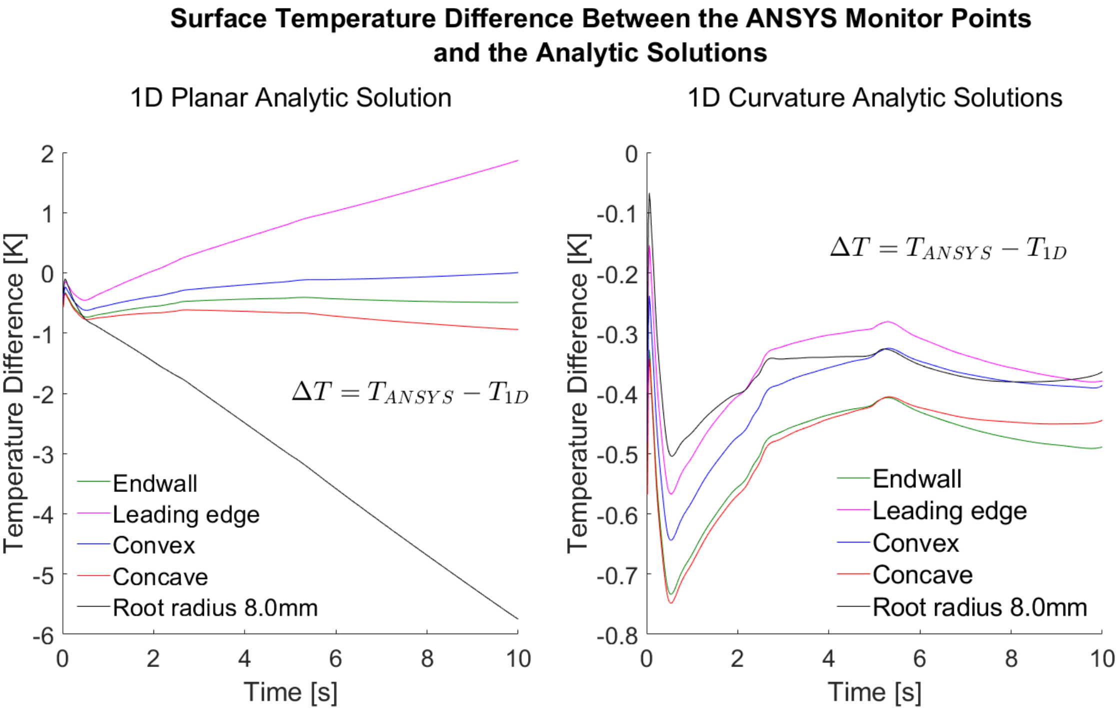
The error casued by the planar assumption increases with time and curvature effects must be considered when operating long duration tests. The curvature corrected solutions are notably more accurate, being uniformly within 1% once the initial transients subside. The peak error is likely caused by the inaccuracies in the ANSYS simulation temperature as the thermal gradient is first established. The impulse response convolution is affected by the full time history, so the initial transients also affect the latter part of the analysis. Despite this, the curvature solutions demonstrate a significant improvement in the accuracy when compared to the traditional planar assumption of the impulse response.



1D Curvature Analytic Surface Temperatures for 10kW/m^2 Heat Flux

ANSYS Extracted Temperatures At Each Monitor Location

Figure 5.6: Transient surface temperatures for the 1D analytic curvature solutions and monitor points in the 3D NGV ANSYS simulation for unit step surface heat flux $10^4 W/m^2$. The analytic endwall case is equivalent to the 1D planar assumption.



Surface Temperature Difference Between the ANSYS Monitor Points and the Analytic Solutions

1D Planar Analytic Solution

1D Curvature Analytic Solutions

Figure 5.7: Comparison between the extracted ANSYS monitor point temperatures and the analytic solutions for both the 1D planar and 1D curvature corrected cases.

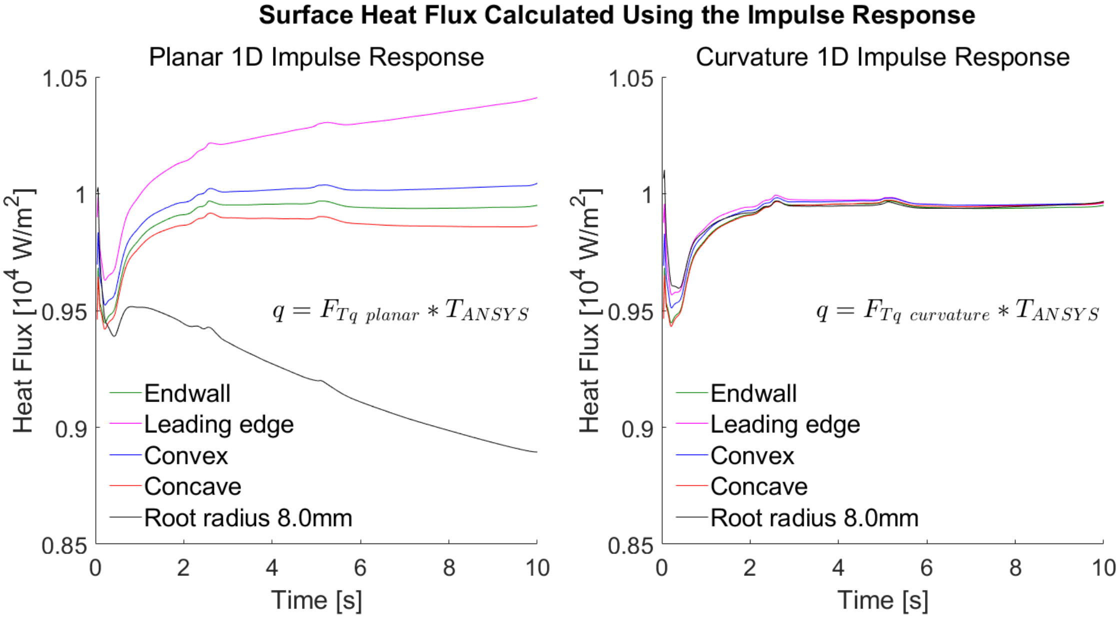


Figure 5.8: Calculated surface heat flux when applying the impulse response to the extracted ANSYS monitor point surface temperatures.

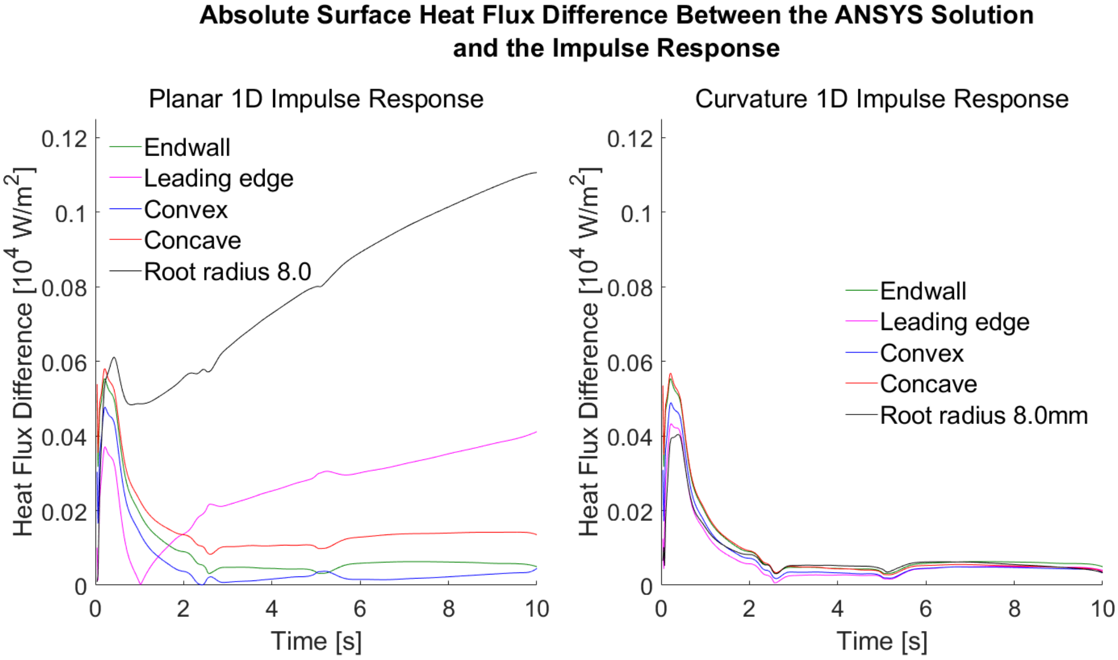


Figure 5.9: Comparison between the impulse response calculated surface heat flux and the $10^4 W/m^2$ ANSYS boundary condition, showing the improved accuracy of the curvature corrected solutions.

5.7 Radial 1D Crank-Nicolson Method

The 1D Crank-Nicolson method can be modified for use in a cylindrical system. Mori and Romão [64], and Duda [65], presented a polar implementation of the scheme for a single material. The bulk of the multilayer solver can be updated to follow this solution which requires a modification to the b and c vectors. The interfaces for heat flux continuity can be approximated by using the existing relation from the planar 1D case. This simplification has little impact on the result, particularly if the nodal count in the simulation is high. The following modifications are required to the Crank-Nicolson method to allow use in a cylindrical system. The resulting penta-diagonal matrix can be solved using the algorithm described in the previous chapter without modification.

Single material cylindrical Crank-Nicolson solution.

$$\left.\frac{\partial T}{\partial t}\right|_{t_m,r_i} = \frac{\alpha}{2}\left(\left.\frac{\partial^2 T}{\partial r^2}\right|_{t_m,r_i} + \frac{1}{r}\left.\frac{\partial T}{\partial r}\right|_{t_m,r_i} + \left.\frac{\partial^2 T}{\partial r^2}\right|_{t_{m-1},r_i} + \frac{1}{r}\left.\frac{\partial T}{\partial r}\right|_{t_{m-1},r_i}\right) \tag{5.14}$$

$$\begin{aligned}\frac{T_i^m - T_i^{m-1}}{\Delta t} = \frac{\alpha}{2\Delta r^2}\Bigg[&T_{i-1}^m - 2T_i^m + T_{i+1}^m + \frac{\Delta r}{2r}\left(T_{i+1}^m - T_{i-1}^m\right) + \\ &T_{i-1}^{m-1} - 2T_i^{m-1} + T_{i+1}^{m-1} + \frac{\Delta r}{2r}\left(T_{i+1}^{m-1} - T_{i-1}^{m-1}\right)\Bigg]\end{aligned} \tag{5.15}$$

This can be rearranged to separate the two temporal steps: t_m and t_{m-1}. These solutions are very similar to the 1D planar case, with only one additional $\pm\alpha/4r\Delta r$ term in the spatial $i+1$ and $i-1$ positions.

$$\begin{aligned}&\left(\frac{\alpha}{4r\Delta r} - \frac{\alpha}{2\Delta r^2}\right)T_{i-1}^m + \left(\frac{1}{\Delta t} + \frac{\alpha}{\Delta r^2}\right)T_i^m - \left(\frac{\alpha}{4r\Delta r} + \frac{\alpha}{2\Delta r^2}\right)T_{i+1}^m = \\ &\left(\frac{\alpha}{2\Delta r^2} - \frac{\alpha}{4r\Delta r}\right)T_{i-1}^m + \left(\frac{1}{\Delta t} - \frac{\alpha}{\Delta r^2}\right)T_i^m + \left(\frac{\alpha}{2\Delta r^2} + \frac{\alpha}{4r\Delta r}\right)T_{i+1}^m\end{aligned} \tag{5.16}$$

This can be combined with the material interface factors and expressed more concisely in matrix form.

$$
\begin{pmatrix}
a_0 & b_0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\
c_1 & a_1 & b_1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\
0 & c_2 & a_2 & b_2 & 0 & 0 & 0 & \cdots & 0 & 0 \\
0 & e_3^* & c_3^* & a_3^* & b_3^* & d_3^* & 0 & \cdots & 0 & 0 \\
0 & 0 & 0 & c_4 & a_4 & b_4 & 0 & \cdots & 0 & 0 \\
0 & 0 & 0 & 0 & c_5 & a_5 & b_5 & \cdots & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & c_6 & a_6 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & c_{n-1} & a_{n-1} & b_{n-1} \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & c_n & a_n
\end{pmatrix}
\begin{pmatrix}
T_0^m \\ T_1^m \\ T_2^m \\ T_3^{*m} \\ T_4^m \\ T_5^m \\ T_6^m \\ \vdots \\ T_{n-1}^m \\ T_n^m
\end{pmatrix}
=
\begin{pmatrix}
f_0 & -b_0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\
-c_1 & f_1 & -b_1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\
0 & -c_2 & f_2 & -b_2 & 0 & 0 & 0 & \cdots & 0 & 0 \\
0 & -e_3^* & -c_3^* & f_3^* & -b_3^* & -d_3^* & 0 & \cdots & 0 & 0 \\
0 & 0 & 0 & -c_4 & f_4 & -b_4 & 0 & \cdots & 0 & 0 \\
0 & 0 & 0 & 0 & -c_5 & f_5 & -b_5 & \cdots & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -c_6 & f_6 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & -c_{n-1} & f_{n-1} & -b_{n-1} \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & -c_n & f_n
\end{pmatrix}
\begin{pmatrix}
T_0^{m-1} \\ T_1^{m-1} \\ T_2^{m-1} \\ T_3^{*m-1} \\ T_4^{m-1} \\ T_5^{m-1} \\ T_6^{m-1} \\ \vdots \\ T_{n-1}^{m-1} \\ T_n^{m-1}
\end{pmatrix}
\tag{5.17}
$$

where: * indicates a boundary interface and,

$$
a_i = \frac{1}{\Delta t} + \frac{\alpha}{\Delta r^2}, \quad b_i = -\frac{\alpha}{2\Delta r^2} - \frac{\alpha}{4r\Delta r}, \quad c_i = -\frac{\alpha}{2\Delta r^2} + \frac{\alpha}{4r\Delta r},
$$

$$
d_i = e_i = 0, \qquad f_i = \frac{1}{\Delta t} - \frac{\alpha}{\Delta r^2}, \qquad a_i^* = f_i^* = \frac{1}{\Delta t},
$$

$$
b_i^* = \frac{-4k_{i+1}\alpha_i + 2(3k_i + k_{i+1})\alpha_{i+1}}{12(k_i + k_{i+1})\Delta r^2}, \quad c_i^* = \frac{2(k_i + 3k_{i+1})\alpha_i - 4k_i\alpha_{i+1}}{12(k_i + k_{i+1})\Delta r^2},
$$

$$
d_i^* = \frac{k_{i+1}\alpha_i - (3k_i + 2k_{i+1})\alpha_{i+1}}{12(k_i + k_{i+1})\Delta r^2}, \qquad e_i^* = \frac{-(2k_i + 3k_{i+1})\alpha_i + k_i\alpha_{i+1}}{12(k_i + k_{i+1})\Delta r^2},
$$

The modified curvature matrix now includes the location specific value r. Similar to the modified impulse response, this prevents a fully parallelised implementation when analysing data from differing locations on the test article. The above solution can be applied to both convex and concave cases by setting the respective values of r in the domain. Similar to the 1D planar case, the numerical Crank-Nicolson method offers a solution beyond the semi-infinite duration if a back boundary condition is known. Included here for completeness, preference should be given to the impulse response if the required assumptions hold.

5.8 Summary

Many engineering applications use curved or freeform geometry and approximating these by planar surfaces leads to significant errors in heat transfer analysis. Analytic solutions for a 1D radial system have been investigated and applied in the case of heating a turbine nozzle guide vane. These solutions were then compared to a full 3D ANSYS numerical simulation. The low order radial 1D method showed comparable performance across most of the geometry, suffering only in localised very high curvature regions on the blade root.

The errors associated with 1D planar assumptions were evaluated, demonstrating that where possible curvature corrected methods should be used. A novel Cylindrical to Cartesian impulse response was presented, allowing the upgraded methods from the previous chapters to be applied on complex freeform geometry. The required cylindrical modifications to the multilayer 1D Crank-Nicolson scheme were discussed, extending the range of suitable applications for this method.

Freeform surfaces are common in turbine geometry, especially on bladed components. The tools in this chapter allow a qualitative assessment of the 1D planar assumption and provide improved methods to handle these geometries where required. In the case of fitted, curved or flexible heat flux sensors for monitoring applications, it is recommended that the appropriate radial model be applied.

6

Neural Network Thermal Prediction

6.1 Introduction

The impulse response method from preceding chapters is widely used in heat transfer analysis. The method relies on a unique variable transfer from a single input to a single output and the full time history must be known. This approach is therefore best suited to short duration laboratory analysis. Alternative methods are therefore required for the continuous live simulation of real-world power turbines. Monitoring applications require the ability to handle multiple inputs and simultaneously predict the value of multiple outputs. It is also necessary to have finite signal lengths so that the entire history of the turbine operation is not required. A machine learning approach is investigated for predicting the turbine thermal profile under transient hot-start conditions.

6.2 Neural Networks

Machine learning covers a broad range of algorithms that can self-model the behaviour of a system. These techniques are generally classified into two main groups: supervised and unsupervised. In supervised learning, the model is shown a selection of inputs and corresponding outputs and through modification of its internal weights can derive a mapping from one to the other. In unsupervised learning, the algorithm is supplied with only the input data and given freedom to define the output classification. Supervised algorithms are widely used across a range of industries to understand or cluster behaviour. The rise of *Big Data*, *Digitalisation* and *Online Computing* have greatly supported the development of machine learning algorithms [66].

Neural networks form a sub group of machine learning algorithms. When used with large data sets they are very powerful in their ability to classify or model a system. A neural network mimics the behaviour in the human brain, where neurons transfer signals to connected neurons depending on their activation. They connect inputs to outputs via a collection of layers with mathematical operators termed neurons. Each neuron is built from a bias, a weight and an activation function that when triggered transfers data to the next layer in the network.

6.2.1 Fully Connected Neural Networks

Fully connected networks are the most common network type, in this architecture all neurons in a layer are connected to every other neuron in the preceding and subsequent layers. The intermediate layers between the input and output are referred to as hidden layers. The number of hidden layers, and number of neurons within each layer, controls the complexity of the network and must be carefully chosen to adequately resolve the system. Selection of these hyper-parameters often relies on user experience and definition is unique to the posed problem.

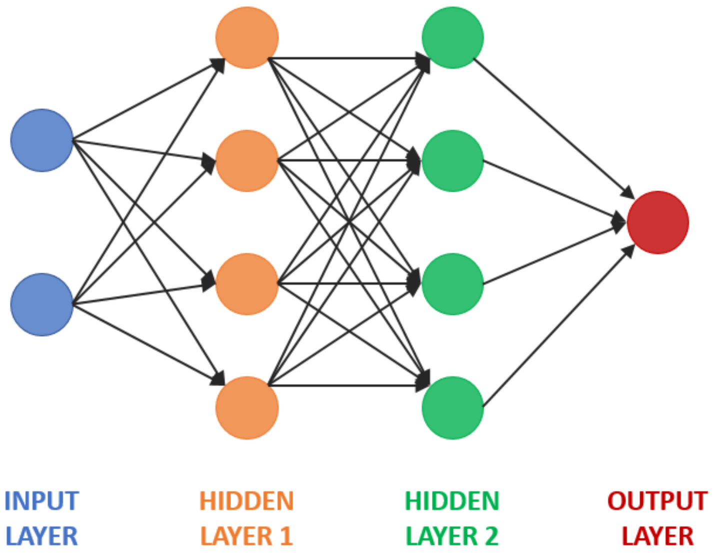


Figure 6.1: Diagram of a fully connected neural network, with two hidden layers, showing the data flow from input to output.

6.2.2 Recurrent Neural Networks

Fully connected networks perform well on classification tasks but can struggle with time series data. Recurrent Neural Networks (RNN) have been developed as a time series upgrade to give better performance when working with temporal information [67]. Unlike the standard fully connected layer, a recurrent layer stores information about its own previous state and can adjust the current output value based on historical data. An RNNs ability to consider previous values in the current mapping makes the recurrent network better suited to time series handling and they are heavily used in data forecasting and trend analysis.

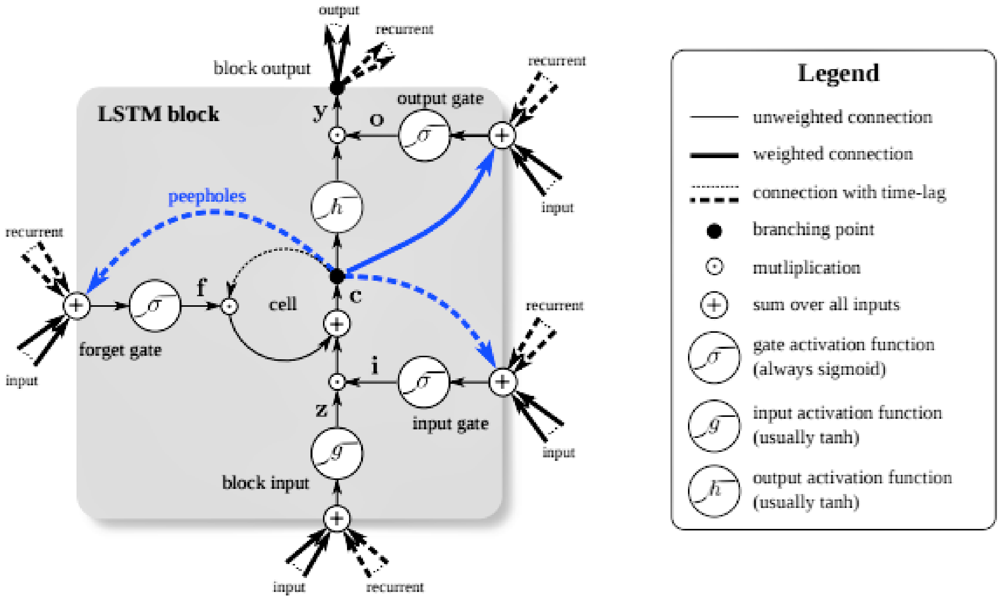


Figure 6.2: Diagram of a Long Short Term Memory block, showing the advanced construction of this neuron type, with additional recurrent inputs and history length forget input to learn time based features [67].

Long Short Term Memory (LSTM) blocks are a sub class of RNN architecture and have been proven across many different applications of time series data processing. The blocks consist of an additional gate alongside the input and output that allows the cell to decide when to allow previous states to be considered. During training, the LSTM block can self-define the required memory allocation to accurately parse the data. They therefore excel in processing time series where the periodicity or time lag between important features is unknown [67]. Conceptually similar to the impulse response, these methods define an output from a weighted average of a time series input. Alongside these weights, non-linear activation functions may be used to achieve a more comprehensive model.

6.2.3 Activation Functions

Activation functions are used on the output of each neuron to determine the value to be passed to the next connection. These functions generally look to map values within the range [-1,1] or [0,1], depending on whether the neuron should fire or not. An activation value close to zero stops the neuron affecting subsequent calculations and in effect turns this neuron off for the given input. This controlled activation can allow several routes through the network and allow the model to cope with a variety of conditions. Many different activation functions are available and must be selected by the user when constructing the model. The most common functions are shown in Figure 6.3; this thesis has focused on the use of tanh, sigmoid and rectified linear (ReLu) to allow the full range of activation values to be achieved. Note all activation functions are differentiable which is required for the gradient descent training methods.

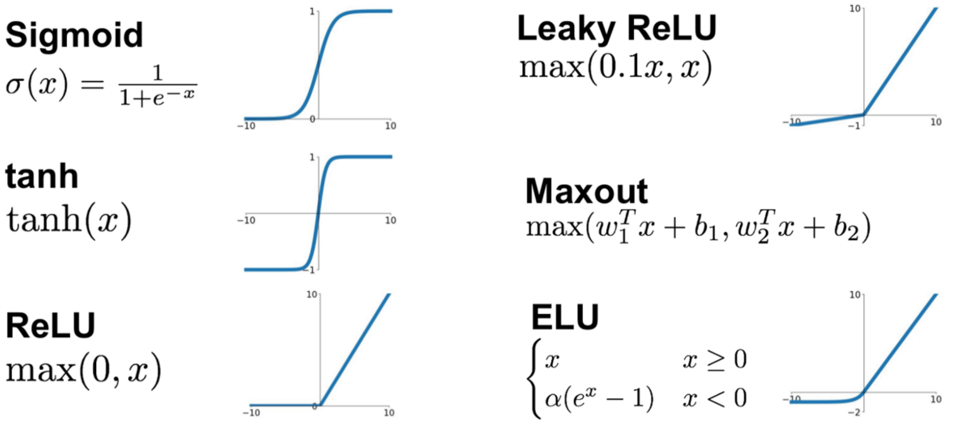


Figure 6.3: Common activation functions used in neural network models [68].

6.2.4 Training and Optimisation

Neural networks are trained using an optimisation process of forward and backward propagation, adjusting the neuron bias and weight values on each pass to minimise the overall cost function. Stochastic Gradient Descent (SGD) methods are commonly employed to optimise the network. Several alternative algorithms have been developed to further improve the learning rate. Adam is one of the more popular first-order gradient-based methods and introduces momentum to speed convergence and prevent trapping in local minima. The Adam optimiser has been the preferred selection for this thesis, it is computationally efficient and well suited to large data sets [69].

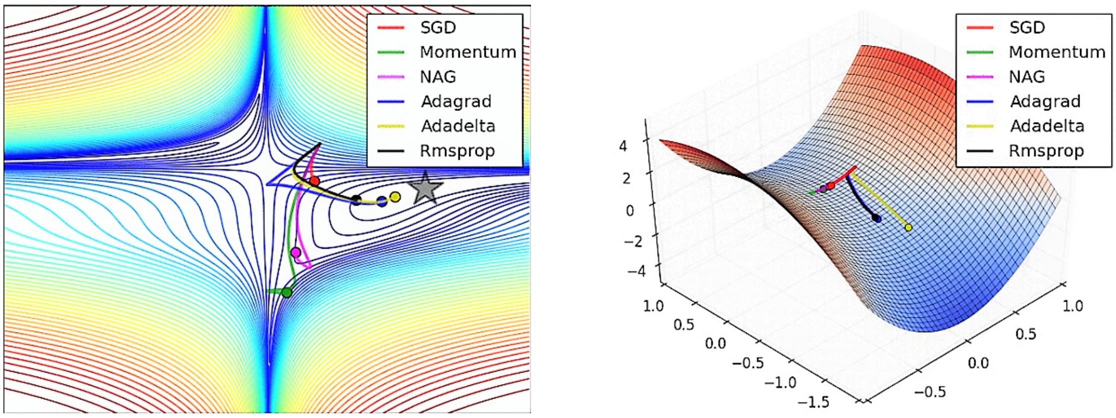


Figure 6.4: Comparison of optimisation algorithms commonly used in neural networks [70].

The performance of a network is measured by the loss function and overall loss value. During training, one pass of the forward and backward propagation is called an epoch. Depending on the range of available data and the complexity of the network architecture, training times can vary from a few epochs to several thousand. Selection of the correct cost function is essential to successful training. Where required, customised loss functions can be written. In the case of classification, a one-hot SoftMax loss function is usually applied. For numerical data, the Mean Square Error (MSE) loss function is generally preferred.

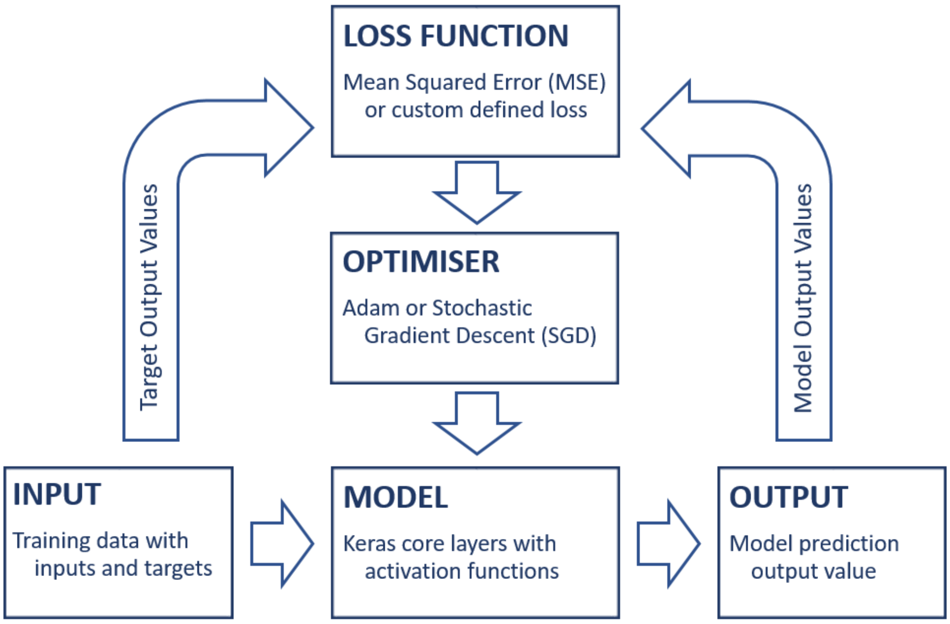


Figure 6.5: Schematic of data flow during one training epoch of a neural network.

When training a network, selection of the training data and number of epochs is important for overall performance. If the model is trained using too little data or with too many epochs, the model becomes over-fit and too specific to the training data. To avoid this, it is good practice to split the available data into a training set and validation set [66]. The model weights are adjusted after each training run but overall performance is determined on the validation set which has not previously been seen by the model. This forces the model to remain generalised and therefore better able to handle a wider input data range. Typically, the validation set consists of 10-20% of the data depending on the available data size.

Neuron dropout is regularly employed to prevent over-fitting of the model. During a training epoch, a percentage of the neurons in the network will be removed, or *dropped out*. This forces the algorithm to rely on a wider range of neurons in the architecture and each subsequent epoch cannot simply reinforce a single route. This gives rise to a network with a large range of active neurons which is better able to represent a variable input. Dropout is a hyper-parameter and is often specified as a percentage, being the probability that a neuron is dropped during an epoch [66].

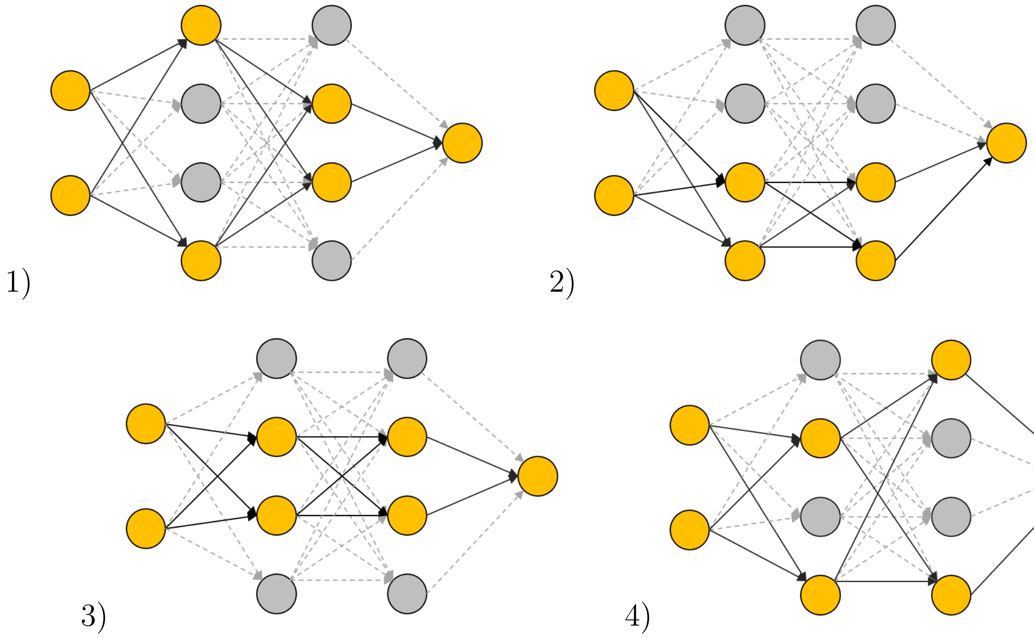


Figure 6.6: Schematic of data flow during training, showing active and dropped out neurons for four possible cases, with a 50% drop out probability in a fully connected feed forward network.

The use of neural network techniques requires the training and calibration of the network layer parameters. The supervised learning method necessitates a large volume of data spanning different operating conditions and covering the entire envelope of the operating regimes. Such methods are therefore best suited to the development of monitoring systems for existing turbines installed in the field. In these applications, the large data sets required for training either exist or can be collected from the legacy installations. In the case of new turbine designs, artificial training sets may be generated by extracting simulation data and combining this with prototype testbed measurements.

6.3 Improving Performance

The most common methods for improving performance are boosting and bagging [66]. In boosting, networks are chained in series and the output of one network fed to the input of another. This method allows for systematic error correction, with each boosting model learning the error in the preceding model. Bagging uses a combination of smaller sub-networks to make higher level decisions. A typical example is face recognition, where several smaller models are trained to identify each facial feature: eyes, nose, mouth, ears. The final classification is a grouped output of all networks. Boosting has been applied to allow efficient training of several smaller models.

Transfer learning is becoming more common as the availability of pre-existing networks increases. A pre-trained commercial network, e.g. Oxford VGG [71], is employed and only the final hidden layer retrained for the specific application. This gives the user access to a much higher accuracy network without the extensive training and data requirements to develop such a model from scratch. The method should allow for faster development of future models and a boosting layer has been included in the model construction.

Data standardisation is commonly used. Most activation functions operate in the range [-1,1] or [0,1] and using data within this range is advantageous. Standardising the inputs and outputs prevents very large or small weights in the network and helps ensure meaningful gradient descent updates. It is common to standardise the data to a zero mean and unit variance. Preference in this thesis is given to the SciKitLearn MinMaxScaler [72], Equation 6.1, which allows an explicit feature range to be given $[X_{min}, X_{max}]$.

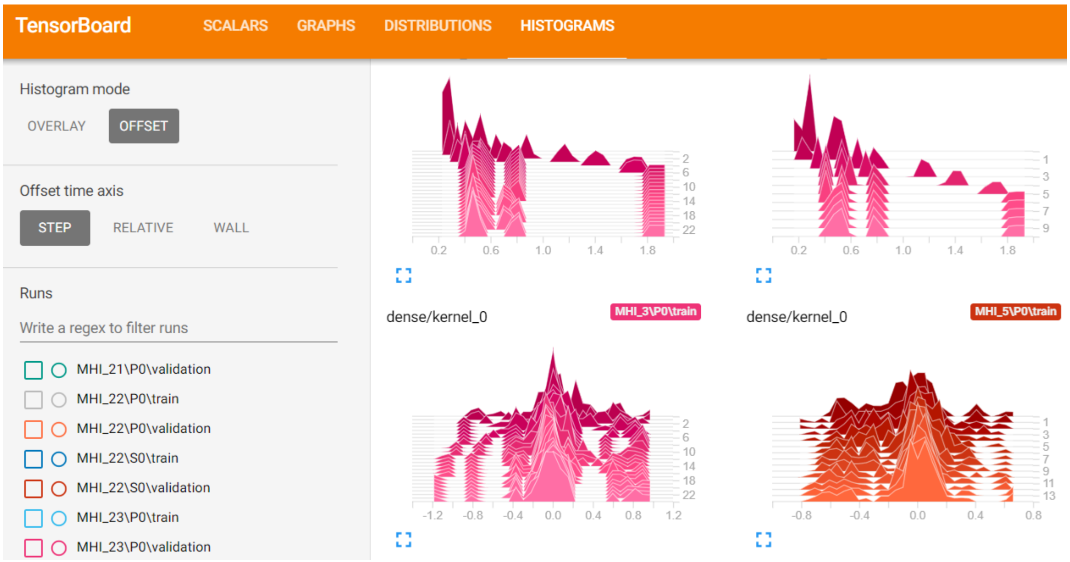
$$\begin{aligned} X_{std} &= (X - X.min(axis = 0))/(X.max(axis = 0) - X.min(axis = 0)) \\ X_{scaled} &= X_{std} * (max - min) + min \end{aligned} \tag{6.1}$$

6.4 TensorFlow and TensorBoard

There are a number of commercial and open source software packages available for the application of machine learning. MATLAB has a wide range of tools implemented in its Machine Learning Toolbox but requires a commercial license and is restricted to a higher-level interface. TensorFlow is an end-to-end open source software library developed by Google and distributed under the Apache 2.0 license [73]. Integrated with Python, it can be run on Linux or Windows and supports both CPU and GPU processing. The user gains low level access with full control over network customisation and hyper-parameters. Several transparent, open source, high level libraries have been built on top of TensorFlow. Keras and TFLearn are two of the more popular APIs. Keras has been selected as the preferred application due to the wider range of implemented features and the ability to seamlessly interface with core functional TensorFlow network methods.

Deploying TensorFlow on a GPU provides a significant advantage in the training performance. GPUs can reduce training time by more than 50% compared to a CPU application [74]. The improvement is dependent on the number of cores in the graphics card and the CUDA computing capability. Nvidia cards present the best local installation option and have been used exclusively in this thesis. Further performance enhancement could be achieved by moving to cloud services but this step was not necessary within this application.

TensorFlow comes with graphic visualisation tool TensorBoard. This allows the neural network graph to be inspected and provides a visual output of the metrics and weights during training. This addition to TensorFlow makes debugging under-performing models easier and allows several models to be loaded in parallel allowing performance comparison [73]. A example of the Tensorboard output showing a comparison of the weights in two models during training is shown in Figure 6.7.



Histogram mode

OVERLAY OFFSET

Offset time axis

STEP RELATIVE WALL

Runs

Write a regex to filter runs

☐ ○ MHI_21\P0\validation
☐ ○ MHI_22\P0\train
☐ ○ MHI_22\P0\validation
☐ ○ MHI_22\S0\train
☐ ○ MHI_22\S0\validation
☐ ○ MHI_23\P0\train
☐ ○ MHI_23\P0\validation

Figure 6.7: TensorBoard graphical interface to visualise the model parameters and training data.

6.5 Training Data

In order to test the suitability of a neural network prediction method, Mitsubishi Heavy Industries (MHI) provided an example data set containing multiple hot-starts conducted over a one month period. The data has been used to test a variety of network architectures to find a suitable structure and methodology for implementing a thermal prediction tool. The aim was to track the temperature critical positions across the high pressure and intermediate pressure turbine, by using only readily available site measurements. In total 70 locations across the turbine were selected, split across several axial cross-sections of the machine.

The raw data provided by MHI was collated from a series of tests conducted in 2014. Split across many different Microsoft Excel documents and internal worksheets, several hundred sensor measurements were provided covering all aspects of the turbine and plant facility. Sampled every minute for 25 days, 36k samples were provided for each sensor location. To protect the intellectual property of MHI, the raw data is not provided in this thesis and where sensitive has been normalised as shown in Figure 6.9 and Figure 6.10.

6.5.1 Fault Identification

Several examples of sensor faults exist within the data and the initial task required data cleaning and validation before training. Sensor faults are common in practical applications and have significant effect on the accuracy of online monitoring systems. Several research groups are investigating this problem and tools already exist for the live detection of sensor failure or misleading data [75] [76]. It is therefore assumed that input data will be correct and faults were not used to train the network.

In order to clean the MHI training data, an automatic data fault and correction algorithm has been written. Identified faults are interpolated to return a more accurate representation of the true sensor measurement. In many cases, the data acquisition system did not return a defined fault code (-156). Many of the faults needed to be identified by gradient or value steps in the data. A first derivative Savitzky-Golay filter has been used, along with percentile outlier detection. The filter applies a smoothing to the data and is robust for noisy input signals, the method fits a low order polynomial using a linear least squares method by finding a set of convolution coefficients [77].

After finding the localised smooth gradient, a median deviation sliding window is applied, comparing individual data points to the average gradient in the window. Points with gradients outside the accepted percentile bounds (1.5%, 98.5%) are removed as outliers. This method allows detection of both local sensor instability and large scale sensor drop out. Implemented with Python Pandas dataframe, the data cleansing code allows the automatic collection and cleaning of multiple sensors from many Excel source files.

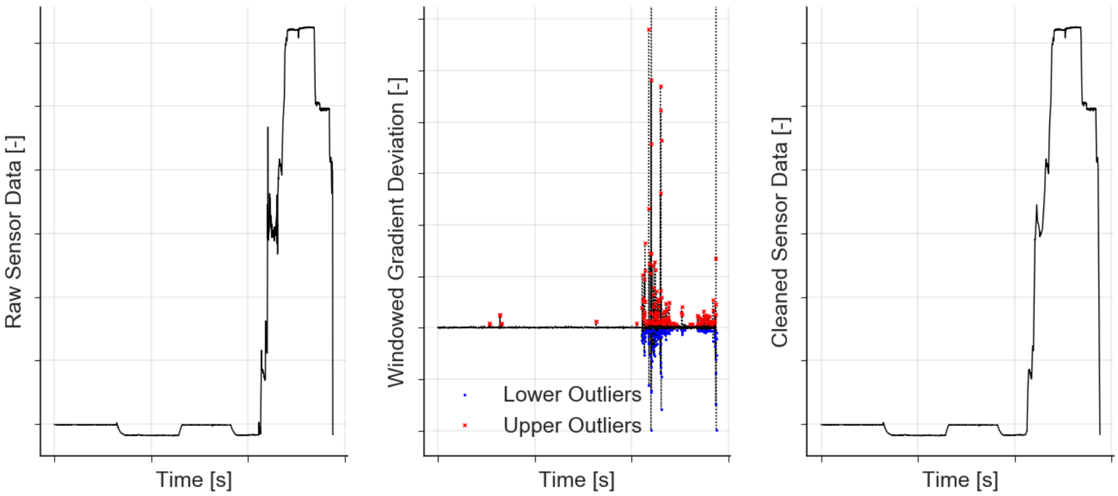


Figure 6.8: Comparison of the input data for one channel before and after the Savitzky-Golay filtering, showing the percentile outliers highlighted for correction in the first derivative median deviation method.

Figure 6.8 shows data for one sensor before and after cleaning. The method efficiently removes outliers whilst being able to retain localised high frequency features and large step changes. The degree of retained unsteadiness can be adjusted by closing the bounds on the accepted gradient percentile range. However, many of these features are likely of interest so a wide bound has been selected. A Gaussian smoothing filter, over a small window of 11 samples, was added to reduce the amplitude of the sensor noise but retain the core fluctuations. Training with data that includes noise and some high frequency components is beneficial as this forces the network to find a more robust solution. The network must apply weights across the full input range to avoid over sensitivity to a single noisy input. Data augmentation techniques are common in neural network training and often simulated noise is added to achieve a similar effect.

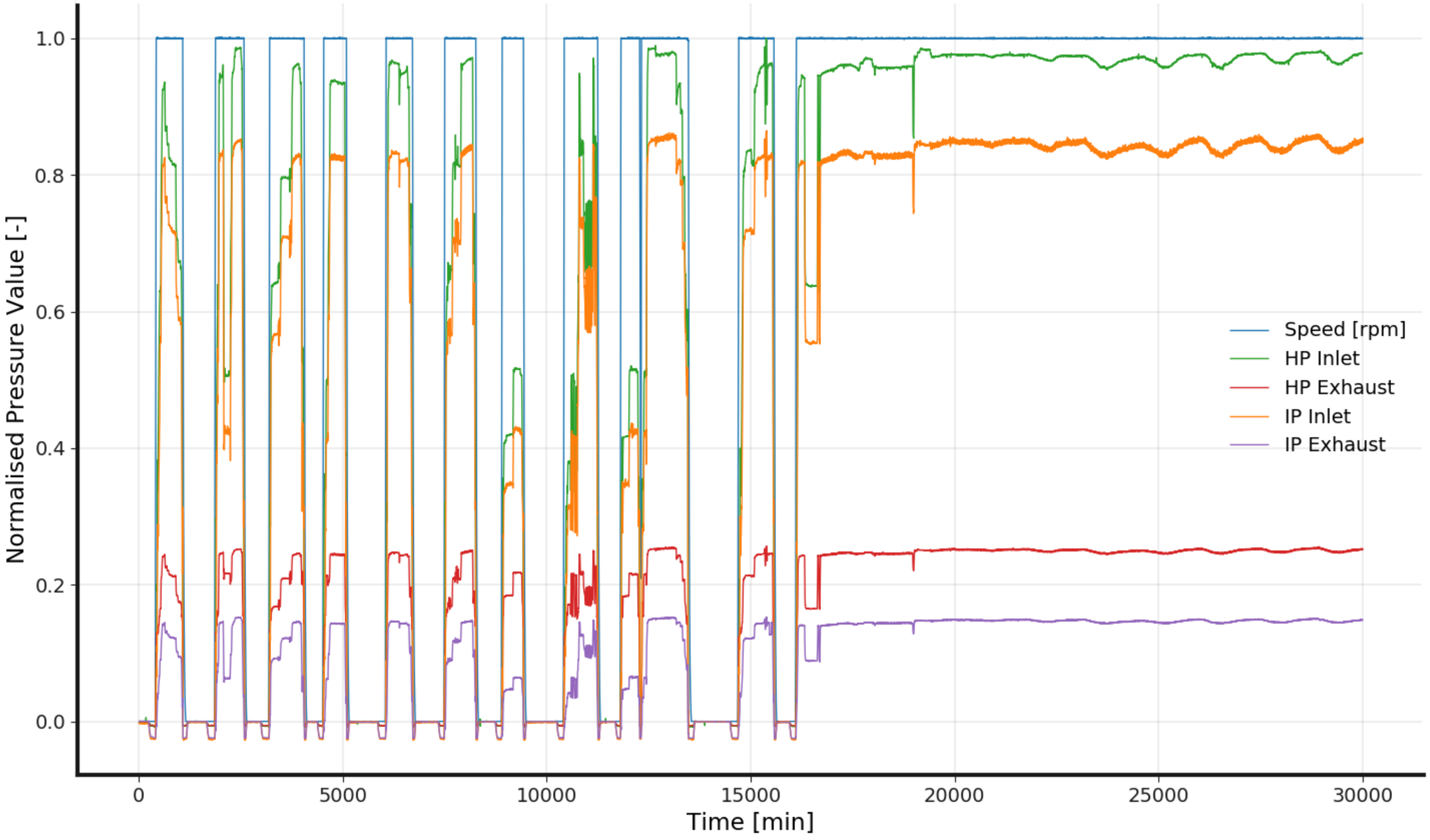


Figure 6.9: Plot of the normalised pressures, showing the data used for model development with large scale transient repeated hot-starts and small scale transient operation.

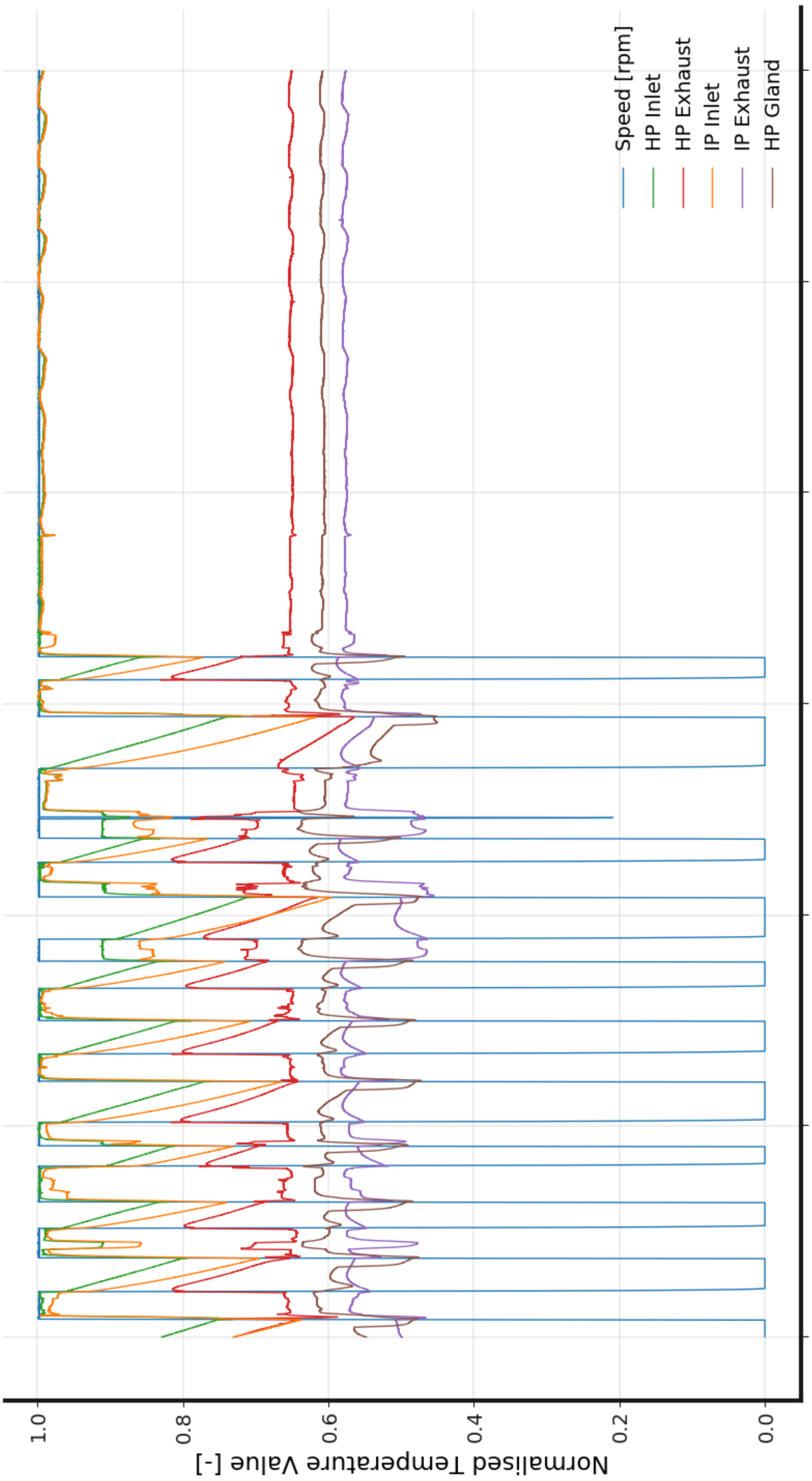


Figure 6.10: Plot of the normalised steam temperatures, showing the data used for model development with large scale transient repeated hot-starts and small scale transient operation.

6.5.2 Feature Selection

Forty seven input sensors were extracted from the raw data and identified as suitable candidates for the neural network. In order to optimise the network and have the best possible performance, i.e. minimum training and calculation time, it is desirable to reduce the inputs to the minimum requirement. A feature selection analysis was completed using the cross-correlation between each input and each data set value. Inputs that were shown to have high correlation with the target temperature were prioritised. Input values that had strong correlation to another input were discarded to prune the data set and reduce the model size. Figure 6.11 shows the correlation process, a red green colour-map has been applied to visually display high (green) and low (red) degrees of correlation. Large blocks of green can be seen and show sensors with a high level of redundancy for use in this application. One example is the inboard and outboard bearing temperatures which are largely similar in normal operation and track each other closely.

The maximum, minimum and percentile correlations were considered for each input; this screening enabled the sensor list to be reduced to 21 required measurements. All of the inputs selected are readily available on site and do not require invasive measurements or complex instrumentation. Metal temperature data for the inlet flange and exhausts has been included, this is routinely collected on prototype machines and could easily be added to site installations where not already measured. The additional six measurements also form part of the total 70 tracking locations so the number of unresolved targets is reduced to 64.

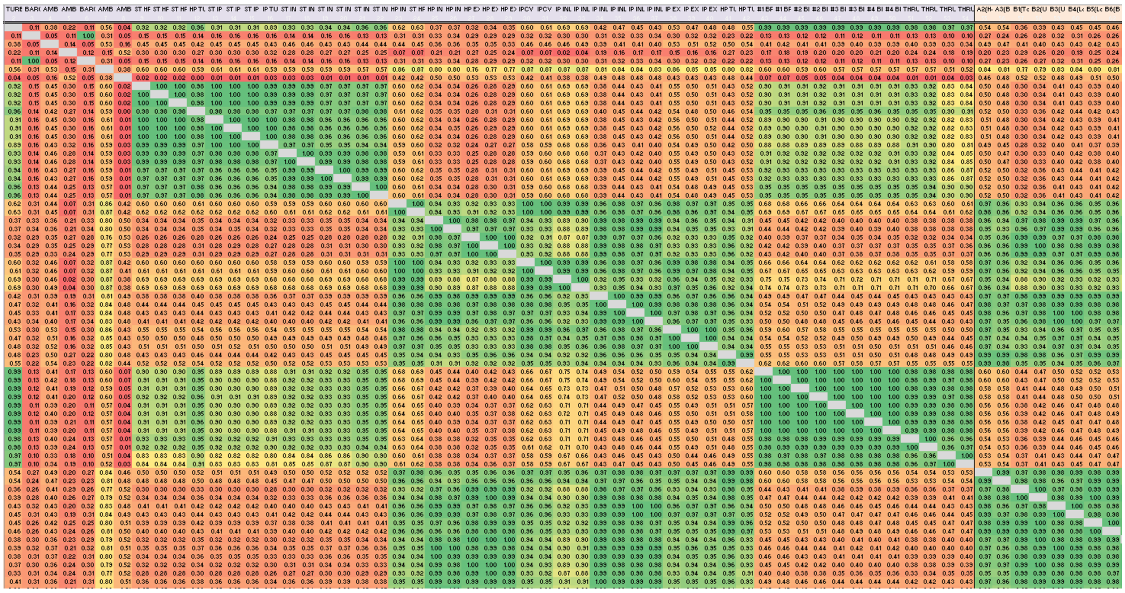


Figure 6.11: Microsoft Excel cross-correlation calculation, comparing all available inputs and outputs, colour coded in a red-green colourmap with green showing a high degree of correlation.

The 21 selected model inputs:

- Ambient temperature
- Turbine speed
- HP pressure, inlet
- HP pressure, exhaust
- HP pressure, air
- HP pressure, condenser vacuum
- HP temperature, inlet steam
- HP temperature, exhaust steam
- HP temperature, gland steam
- HP temperature, inlet flange
- HP temperature, exhaust upper metal
- HP temperature, exhaust lower metal
- IP pressure, inlet
- IP pressure, exhaust
- IP temperature, inlet steam
- IP temperature, exhaust steam
- IP temperature, inlet upper metal
- IP temperature, inlet lower metal
- IP temperature, exhaust upper metal
- IP temperature, exhaust lower metal
- Bearing metal temperature

6.5.3 Principal Component Analysis

It is common in statistical analysis to further reduce the input dimension by applying Principal Component Analysis (PCA) [78]. PCA decomposes the data into a minimum spanning vector set, using weighted components of each input. By taking the N most important vectors, one can find a reduced set of features that span the majority of the data range. PCA was applied to both the 21 inputs and 70 outputs. The explained variance for each feature was analysed using a cumulative Pareto method, Figure 6.12. The minimum criteria was defined at 99.5% explained variance, which resulted in 8 input features and 4 output features.

The low number of output features suggests large parts of the turbine thermal profile follow a very similar pattern and the bulk behaviour of the machine can be greatly simplified. It was therefore decided to add a preliminary predictor tool to first learn the underlying low order output PCA feature. This predicted value was then fed alongside the input PCA to give a backbone to the final output prediction. Due to the low input and output count, this additional PCA model has little effect on the final training or calculation time.

6.5.4 Standardisation

Data standardisation was applied to all input and output data using the SciKitLearn MinMaxScaler, with a feature range [0.05:0.95]. This was applied to each input independently before computing the PCA, this ensured uniform dependence on each input. The resulting features were not additionally scaled and the primary components of the PCA features in the range [1.5:3.5] were accepted.

To improve ease of use, the scaler parameters were extracted to define a preprocessing and post-processing layer within the network. These layers convert the raw data from absolute values to the standardised range of the model, then reverse the scaling on the output. This ensures all standardisation is contained within the network and the correct data range is always used. The pre and post layers were built with TensorFlow functional methods, defining custom layers in a Keras model.

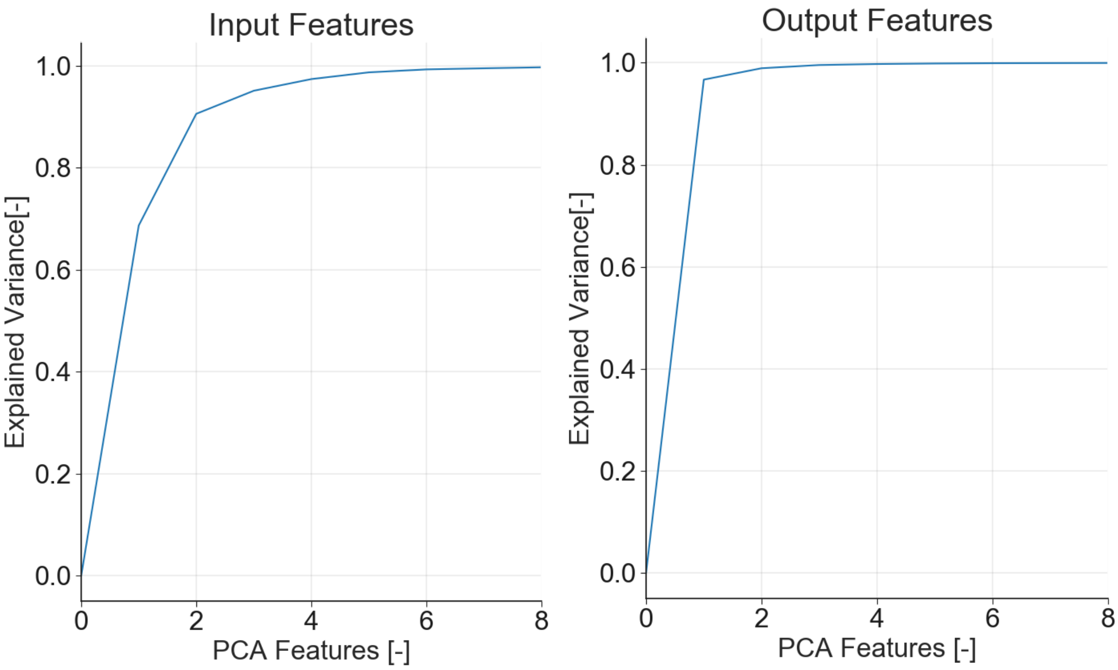


Figure 6.12: Cumulative Pareto analysis showing the impact of the number of features on the percentage variance explained for the input and output data sets.

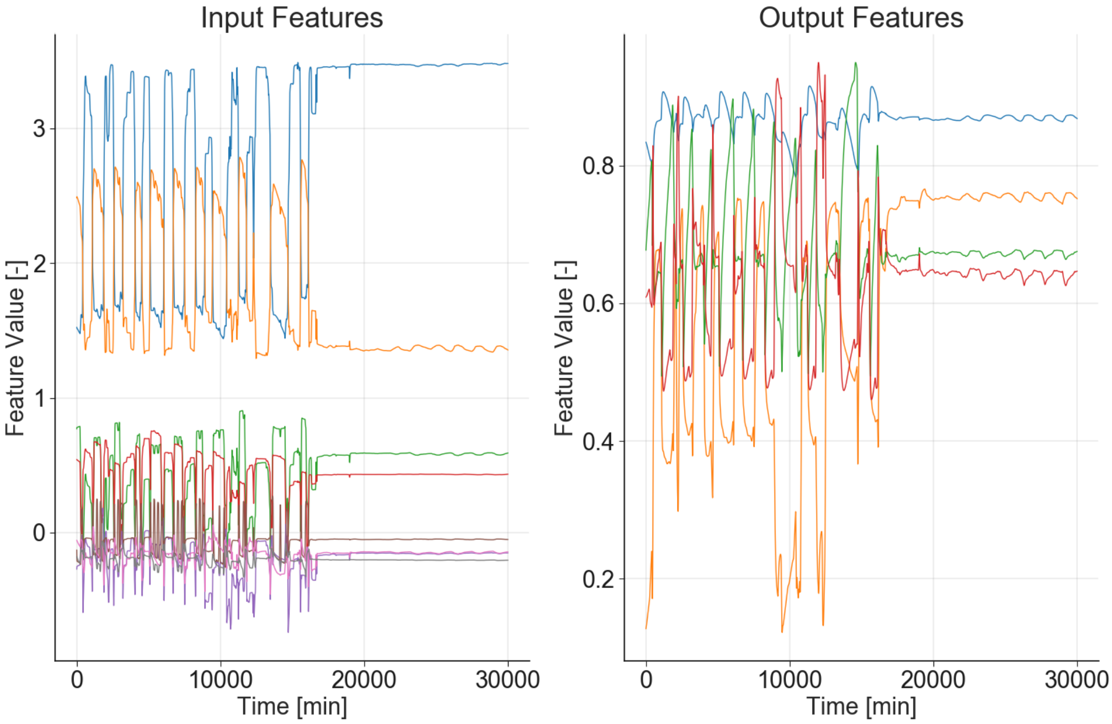


Figure 6.13: The input and output principal component features used in the training of the neural network model.

6.6 Model Architecture

Although the LSTM network is capable of self learning its own required history, there is performance benefit in using the minimum possible history length. This reduces the volume of data recalled from disk during runtime and training. Several different history lengths of LSTM network were trialled, restricting the history to 32, 64, 96, 128 and 256 data samples per input. There was a significant improvement in the network performance up to 128 samples, a little over two hours of data, beyond this limit the additional data did not yield better accuracy. A LSTM history length of 128 samples was selected for the model construction.

6.6.1 Submodel Structure

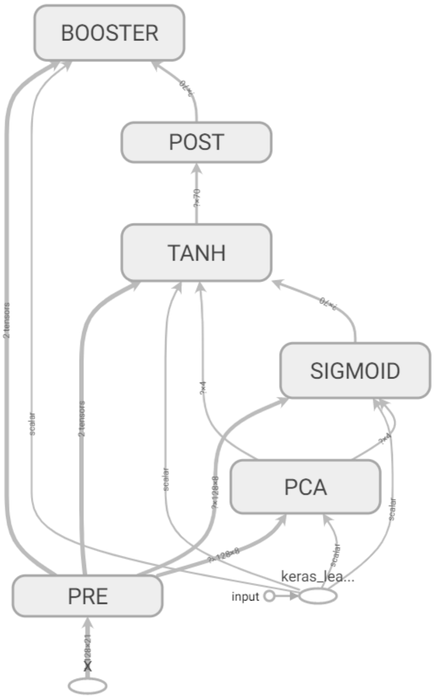


Figure 6.14: Top level schematic of the neural network model structure and data flow, showing the submodels used in the network construction.

The main core of the model uses a predictor-corrector construction. An initial prediction model sets the approximate value, then applies a second model adjusts the output and improve accuracy. The predictor was applied using sigmoid activation to map the standardised value within [0,1]. The corrector used tanh activation to allow bi-directional adjustment. An additional scalar was applied to the corrector to focus the activation function and allow correction within ±5%. Both predictor and corrector use a LSTM input with single output layer, an additional dense hidden layer was trialled but negligible improvement was seen. Several different neuron counts were tested [8,16,32,64,128] with 32 offering the best compromise between accuracy and training effort. In the case of higher node count, the number of parameters in the full model exceeded the number of training samples, this is bad practice and leads to over-fitting. Higher nodal count models may be of benefit in future projects if additional data is made available.

A single booster was added after the post-processing layer to further improve performance. This layer targets output data channels that have low accuracy. Trained on the model error, the booster learns where the network under-performs and compensate the output. Different booster architectures were tested, a single LSTM layer with dense output showed the best performance. The booster layer also provides support for transfer learning. By retraining only the final layer, the model may be tuned to individual machines without excessive computational cost.

6.6.2 Optimisers and Filters

Multiple Adam optimisers were used to support the different training needs of each submodel. Learning rates varied from $1e^{-3}$ to $1e^{-5}$, with 32 to 256 epochs used. In all cases the Mean Square Error (MSE) loss function was applied, using the reduced mean across all channels. The data was split into a training and validation set using a 15% allocation for validation. A random integer state was used to split the data so that validation points were taken from across the full time history. The SKLearn test train split feature was used to allow consistent random selections for repeatability. Dropout was applied on all layers to reduce network dependence on individual neurons. Each submodel was built and trained independently, where necessary using data from the preceding submodel. The final network was constructed using a Keras model class, defining each submodel as a class method to build the combined output. A Gaussian filter was applied to each channel to smooth the output. The filter window is unique to each channel and was selected to give the best reduction in absolute error. The filter is applied using SciPy libraries but could be incorporated in the TensorFlow environment if desired.

6.7 Results and Analysis

The network was trained on a GeForce Nvidia 940MX graphics card with CUDA compute capability 5.0. Training for all submodels was completed in 4 hours. Model save and load scripts have been written that allow the trained model to be reloaded in 6.7 seconds. The first pass of the full architecture is completed in 1.1 seconds however the majority of this time is overhead caused by the transfer of the model to the GPU. Once loaded and run the first time, the model was able to run all 30k test cases in 17.8 seconds, predicting all 70 output variables simultaneously. The speed of this model makes it well suited to real-time applications, once coupled with a fast acquisition data system it could be applied in a live monitoring application.

The prediction accuracy is good across all data channels. The MSE for each channel over the full data set is less than 2.2K. Comparisons between the model output and sensor value are shown for the three worst channels in Figure 6.15. These channels have the maximum single peak error. The trend in all cases is very good and the majority of large unsteady events are well resolved. In the worst performing channel, located close to the gland seal inlet, the peak instantaneous error is less than 6.8K. The main cause of this error is the sudden transients in the data set, effecting a large change in the reference temperature before the model has time to react.

Importantly, the results show a high level of stability in these peak error cases. The prediction always recovers and the error is reduced shortly after the transient event. This ability to handle large scale transients without divergence makes the LSTM neural network superior to a traditional numerical scheme. Despite a dependence on its own prior state, the limited time history prevents errors perpetuating though the simulation, enabling the predicted temperature to recover.

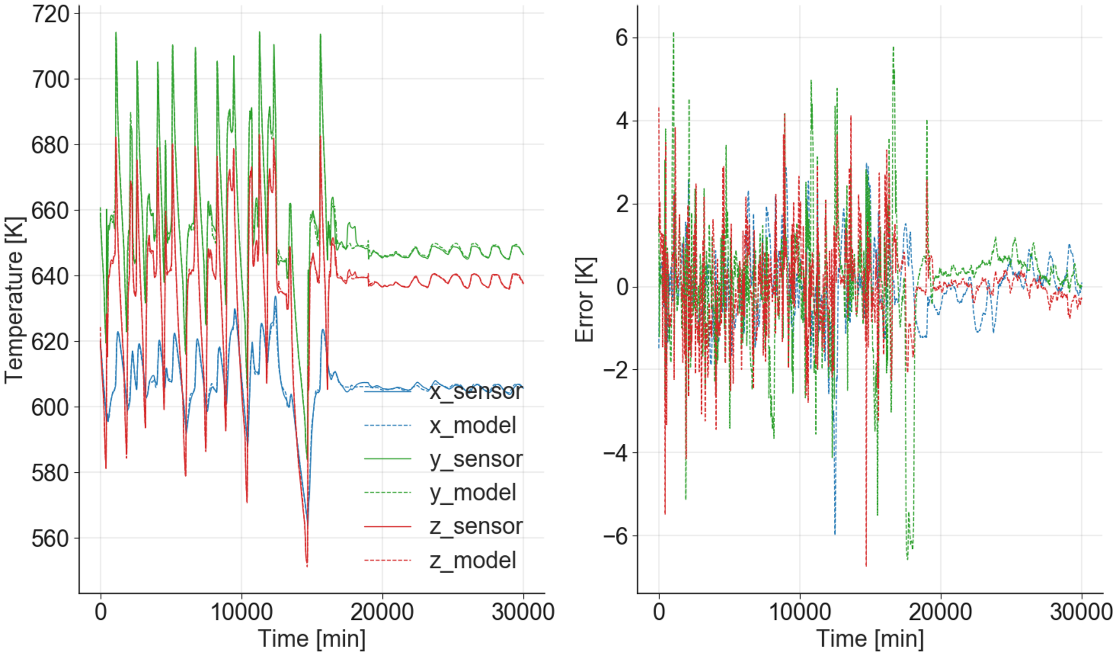


Figure 6.15: A comparison of the worst three channels in the model output with their correct sensor value, showing the worst case error between the model and reference data.

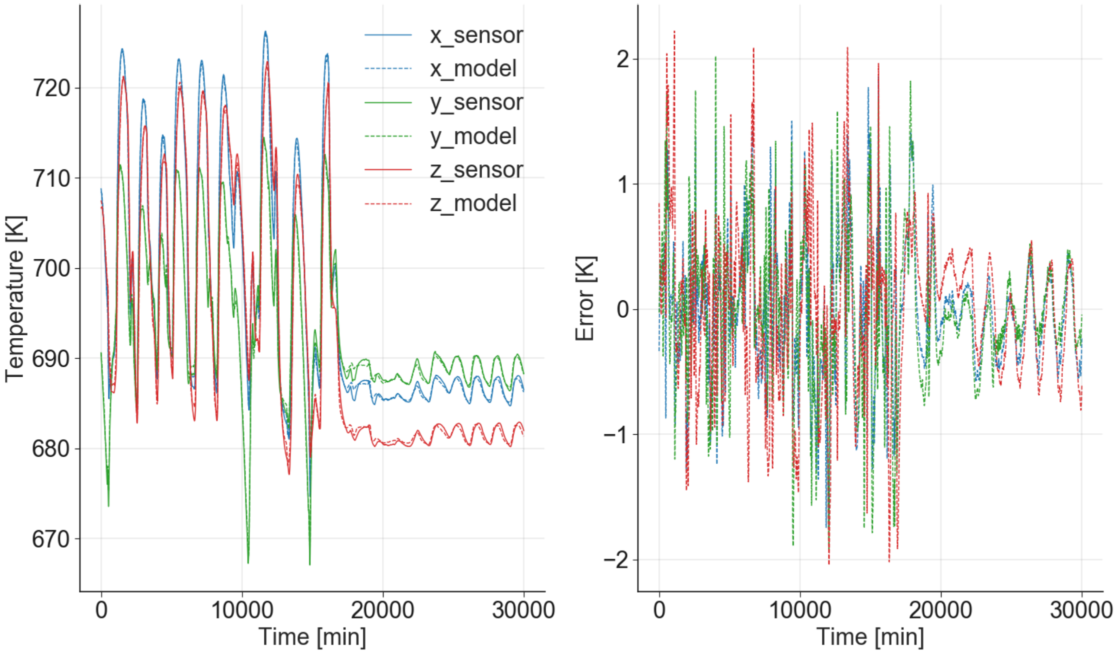


Figure 6.16: A comparison of the best three channels in the model output with their correct sensor value, showing the error between the model and reference data.

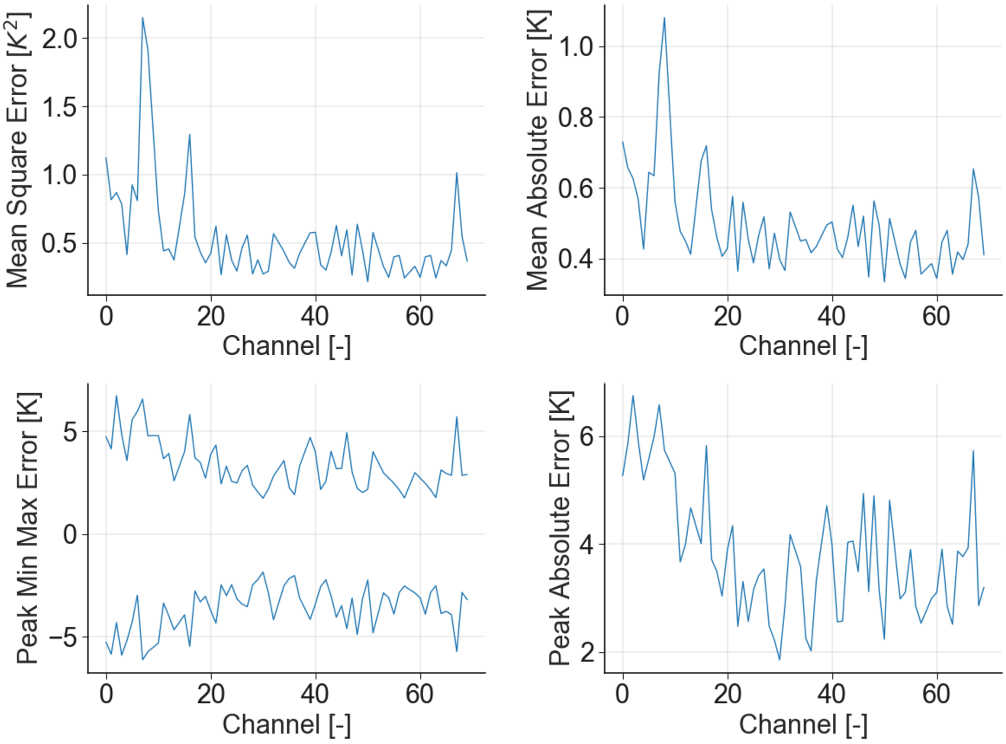


Figure 6.17: Full range of error metrics presented for each of the 70 output channels, taken over the full time history of both the training and validation data.

Their resistance to divergence, and dependence on measured values only, makes LSTM networks a good candidate for simulation anchoring. Traditional numerical methods are prone to divergence because each iteration builds on the error from the previous step. This is not true of LSTM methods and each temperature prediction is independent, based only on a time history of the measured boundary condition. Running in parallel with a traditional numerical scheme, the LSTM could be used as a point-wise thermal constraint, updating the numerical code with an anchoring value. These points in the numerical domain would not suffer from large scale divergence and could be used to hold the surrounding locations within a stable bound. Furthermore, if large regions of the model's outer surface could be defined by LSTM networks, a bounded fully-constrained model may be possible. The combined neural-numerical concept, using both neural networks and a traditional numerical scheme, is explored further in Chapter 7 Hybrid Fidelity Geometry Representation, Meshing and Data Transfer.

6.8 Limitations

The model performs very well across the full data set for hot-start transient analysis. In order to assess the bounds of usability, and test how well the model transitions from unseen to known cases, the model was applied to a data set for a cold-start into normal hot operation. A turbine model in the field will be exposed to many different operating conditions in full-load, part-load, start-up and shutdown. This cold-start test case is just one of many possible scenarios that may be encountered.

The same data pre-processing was applied to find the principal component features in this case. Seen in Figure 6.18, both the input and output features differ significantly from the training set in Figure 6.13. Both feature sets have regions significantly below those seen during the training exercise; as expected, this leads to large initial errors in the model prediction.

Figure 6.19 compares the worst three output channels in the test case. The initial large error is caused by the layer parameters, specifically the inclusion of bias values in the network. Seen in Figure 6.20 and Figure 6.21, the model layers have a uniform kernel weight distribution but a bi-modal bias distribution. The hot-start training data is all high, so positive bias has been learned to centre on this upper value and allow the weights to adjust for local unsteadiness. The cold region data all sits below the bias and therefore the predicted output is vastly overestimated. The analysis demonstrates that neural networks can be poor when used outside the bounds of their training data. Due to the non-physical calculation in the model, these error are often substantially worse than a traditional physics based numerical solver. The user must therefore take additional care to ensure the application is suitable.

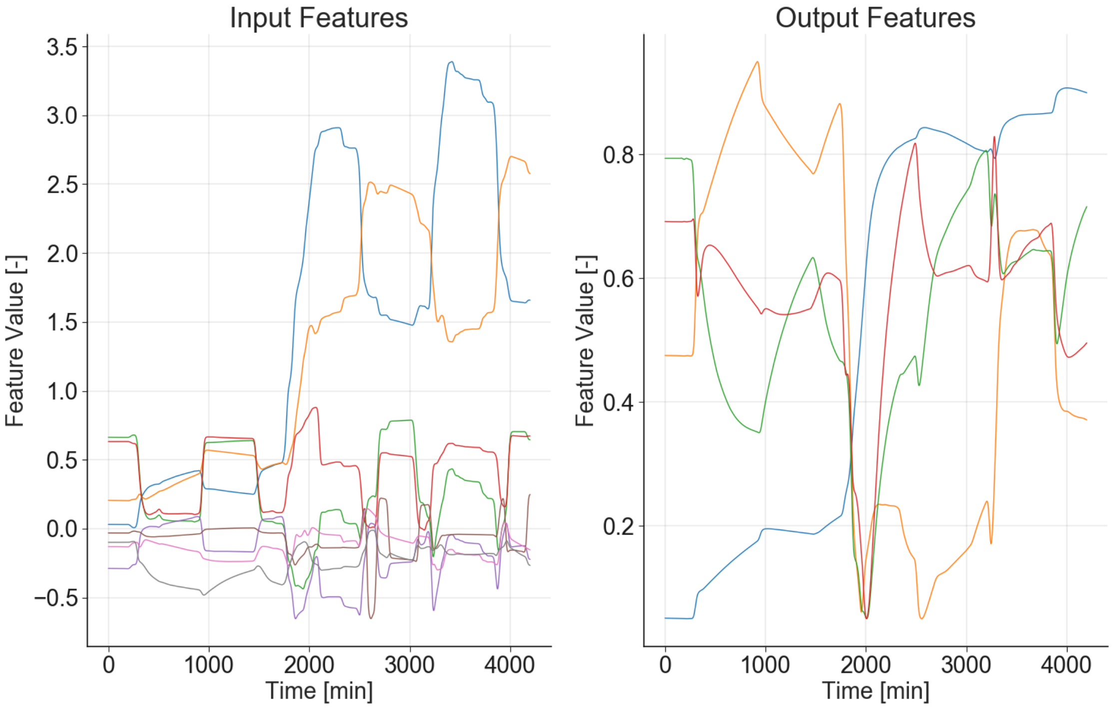


Figure 6.18: The input and output principal component features for a cold-start leading into transient hot-cycle operation.

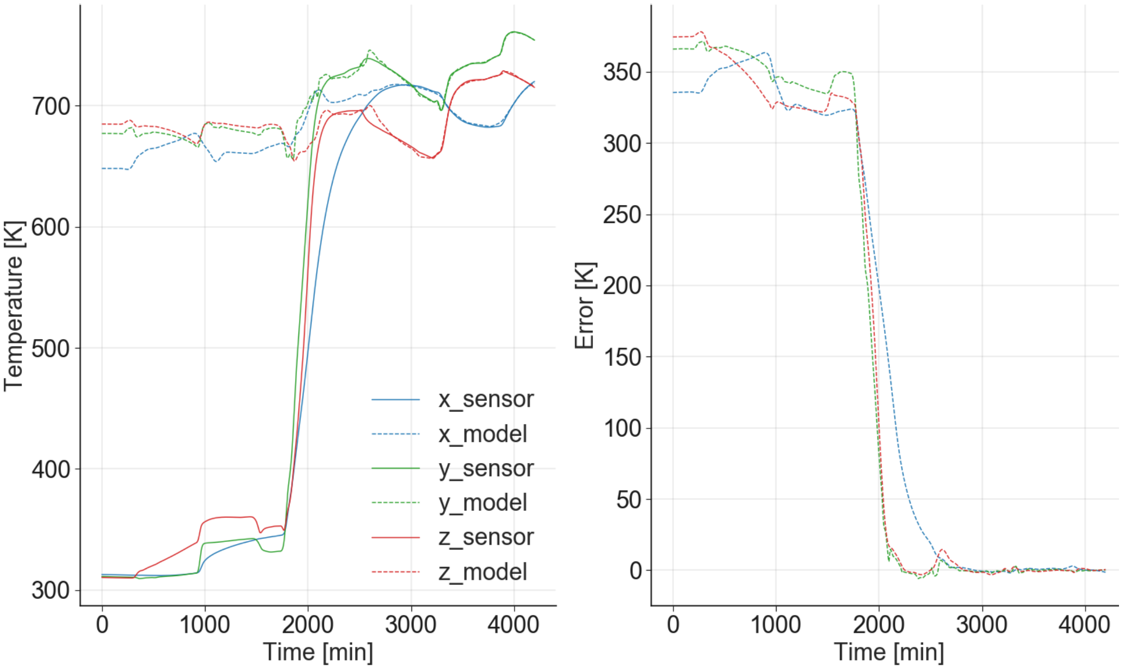


Figure 6.19: A comparison of the worst three channels in the model output to their correct sensor value, showing the worst case error between the model and cold-start data.

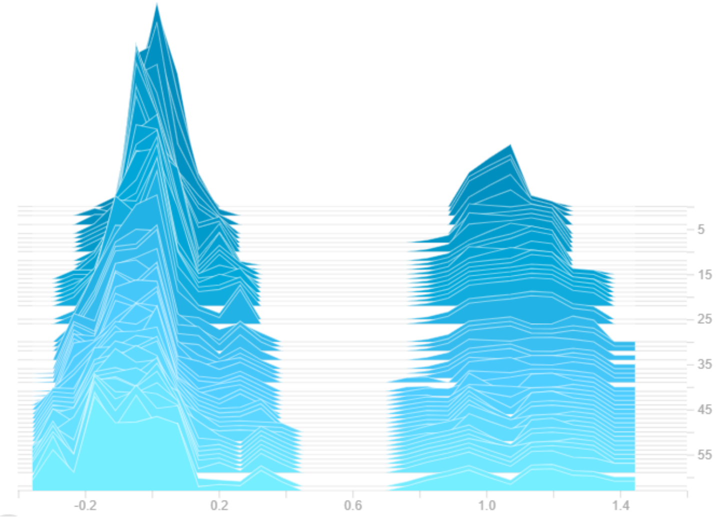


Figure 6.20: TensorBoard histogram bias values in the LSTM layer of the PCA calculation submodel, showing the evolution during training from back to front and bimodal bias distribution.

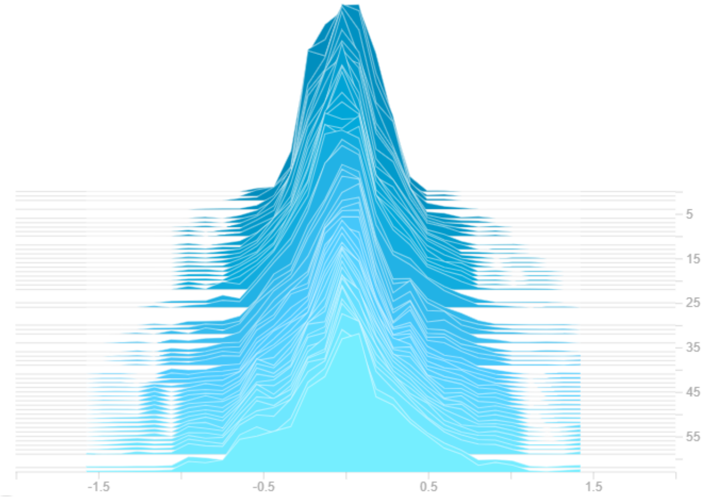


Figure 6.21: TensorBoard histogram kernel values in the LSTM layer of the PCA calculation submodel, showing the evolution during training from back to front with a uniform weight distribution.

Although it is clear that the hot-start trained model is unable to handle the early data whilst the turbine is cold. The large error rapidly drops off as the turbine approaches the design temperature. This highlights the key advantage of neural models over a traditional spatial numerical scheme. The stronger dependence on the measured boundary, rather than the previous internal state, allows the model to recover quickly once in a region of confidence. From this point onwards the neural network performs well and the low error is sustained. This highlights the potential of using these models as boundary conditions, but also demonstrates the need for a confidence check on the model output. Neural models can be additionally trained to predict if the inputs are within the range of its training experience. This confidence factor, ranging from zero to one, may then be used to assess the validity of the output or transition between models that target different operating cases.

Handling the full range of operating conditions in one model is likely not viable. The behaviour during start-up, shutdown, full-load, part-load, vacuum destruction, gland pre-warming, natural and forced convective cooling is notably different. Similar to Ibrahem [35], an ensemble of models could be used to track the full transient behaviour, switching to the most suitable in any given operating mode. This switching could be automated with a classification network, simultaneously allowing the confidence factor to be defined for the model output.

Insufficient data was available across the full operating range to test the ensemble theory however, the architecture for the hot-start model has been developed and proven. It is expected that direct application of this model structure would be equally successful in predicting the other operating cases. The ensemble could be trained using either one-hot encoded selection so that the best model is used at each time step, or using a SoftMax activation - resulting in a direct weighted average of all submodel outputs.

6.9 Summary

Steam turbine operating data, provided by Mitsubishi Heavy Industries, has been used to develop and train a neural network capable of thermal profile prediction. An automatic data cleansing algorithm has been written, allowing fast processing of the faults in raw sensor data. A feature extraction analysis has been completed, reducing the required input to 21 readily available sensors in a typical site installation. Principal Component Analysis further reduced the feature count, compressing the data to 8 input and 4 output values.

A variety of model architectures, layer counts, neuron counts, training cycles and optimisers were tested. The final model used a four step construction with PCA prediction, output prediction, output correction and accuracy booster. The two-step sensor calculation with sigmoid and tanh activations proved effective in tracking local instability. Long Short Term Memory blocks were used throughout and showed a high level of accuracy without multiple appended hidden layers.

Analysis of a cold-start test case demonstrated the model limitations when outside the trained region. Rapid recovery of the model accuracy was seen as the temperatures approached the full-load design condition. More data is required to train an ensemble of models to predict the temperatures during operating under different conditions. The performance was very good across all hot-start validation cases; the peak absolute error is less than 6.8K across the entire 25 day operating window. Most importantly, the model showed a high level of stability. The truncated history of the LSTM method ensured the model did not suffer from divergence.

Where a full spatial profile is required, the networks could be used to form anchoring points in the numerical simulation, reducing divergence over a longer run. This strategy requires coupling a fast-acting thermal numerical solver and is investigated further in the following chapters.

7

Hybrid Fidelity Geometry Representation, Meshing and Data Transfer

7.1 Introduction

Research projects tend to have either a numerical or experimental focus. Where both areas are investigated, this is traditionally to validate or confirm the other. The two parts are treated separately, accepting their respective limitations.

Pure experimental research gives high accuracy real-time data and is especially valuable when analysing complex systems with variable flow behaviour. However, measurement access is limited and the resulting information density is generally low. Construction, commission and testing of highly instrumented experimental facilities is also expensive.

Numerical research is more cost effective and allows freedom to explore a wider range of geometries and test cases. Full domain information can be found, including regions of the machine inaccessible to measurements. However, the absolute accuracy is limited in complex cases and fully transient simulations may take significant time to compute.

Hybrid methods look to incorporate both strategies simultaneously into a single analysis system. The combination aims to bridge the limitations in data resolution, speed and accuracy seen when handling each method independently. Central to the success of hybrid methods is the mapping of data between congruent systems of varying fidelity. Methods to allow data transfer between such systems and provide the functions necessary for 3D hybrid applications are outlined in the following chapter.

7.2 Multi-fidelity Analysis

In the case of analysing a steam turbine; a physical machine can be run through a number of test cycles, collecting process information such as operating flow rates, temperatures and pressures. Seen in the previous chapter, this gives high accuracy data but low resolution of the overall domain.

Moving on a step in fidelity, a thermal model can be constructed to calculate the low order spatial behaviour of the system. A higher resolution is achieved and the model's simplicity allows fast simulation. However, the simplified method is prone to divergence and accuracy is limited.

At the higher resolution scale, decoupled CFD and FEA analysis can be used to further increase the fidelity. Although better able to model complex flows inside the machine, this improvement comes at the cost of computation time. Thus fewer process options or designs can be evaluated within the same time-frame.

A final step in fidelity can be achieved by utilising full conjugate thermal simulations. Additional complexities in meshing, computation, set-up and simulation duration allow even fewer design variations on a given project.

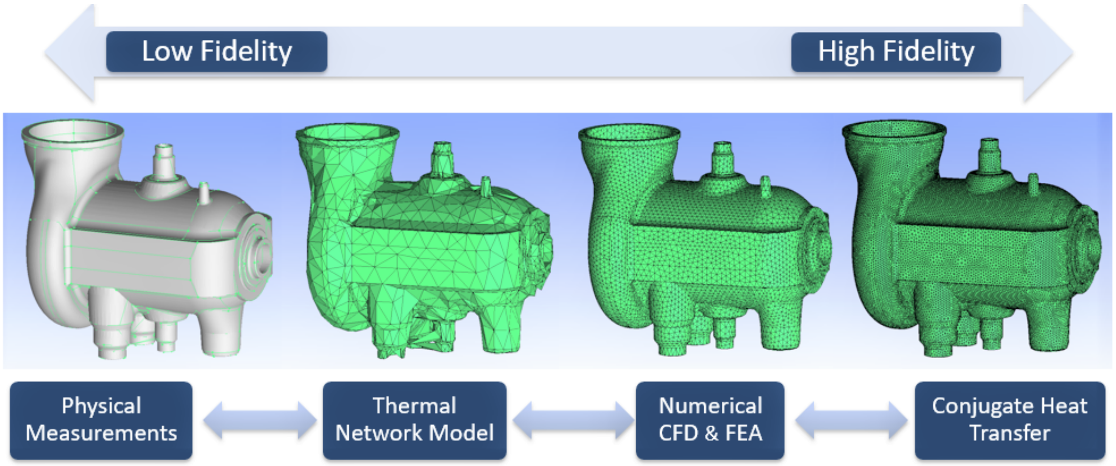


Figure 7.1: Available fidelity range of analyses in the example case of a High Pressure (HP) / Intermediate Pressure (IP) steam turbine.

The full range of fidelities describe the same physical system and information can be shared to improve the quality or speed of the current model. The ongoing digitalisation in industry will see an increase in remote monitoring and optimisation systems. These are fuelling the need for data and many power equipment manufacturers are starting to implement these technologies [36]. Online numerical analysis with low order models will soon become the norm, operating alongside traditional safeguard monitoring. Often described as *Digital Twins*, these numerical codes attempt to replicate the physical system and identify a key application of hybrid multi-fidelity analysis [79].

7.3 Geometry and Mesh Construction

Four geometries form the test cases; two are based on Mitsubishi Heavy Industries (MHI) geometries, one from the Oxford Thermofluids Institute (OTI) and one from the Stanford University Computer Graphics Laboratory.

1. MHI Steam Turbine - data from this machine was used for neural network thermal prediction in the previous chapter. MHI provided 2D section drawings of the main regions of the turbine geometry. These have been modified in SolidWorks and used to reconstruct a representative 3D geometry, providing a realistic test case whilst protecting MHI design philosophy.

2. MHI Nozzle Guide Vane (NGV) - the NGV geometry was provided by Dr Faisal Shaikh, formerly of the OTI, who linearised the design and tested the geometry in the MHI cascade.

3. OTI Casing Rig - this heated cylinder test rig replicates mixed flow conditions in a turbine casing. The 3D data and mesh were provided by Oguzhan Murat who designed the new facility.

4. Stanford Bunny - available from the Stanford Graphics repository, this geometry is widely used as a benchmark for surface mesh manipulation.

Meshing of the geometries was completed in ICEM. The MHI turbine geometry was resolved with a coarse unstructured tetrahedral mesh, suitable for thermal network model use. The high density OTI mesh, provided by Murat, uses a structured multi-zone mesh with fine boundary layer resolution suitable for CFD simulation. The low density OTI mesh and NGV mesh use the same coarse definition as the turbine test case. The different mesh geometries and densities have been used to validate the cross fidelity data mapping methods.

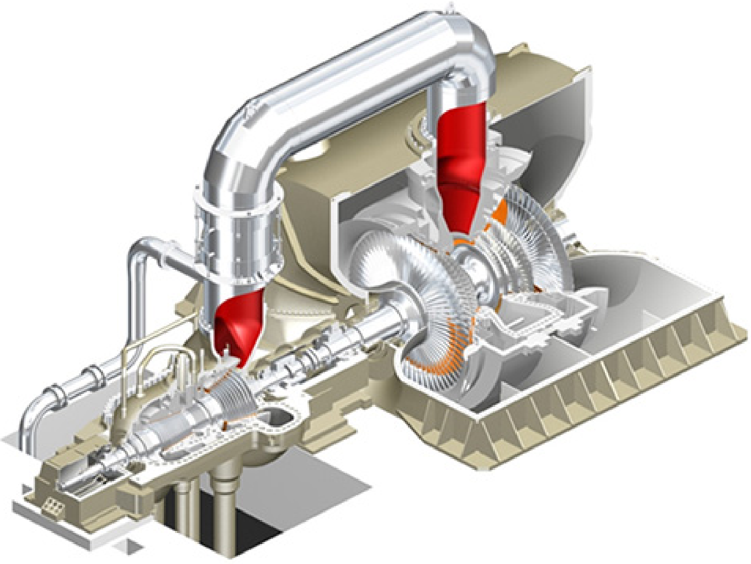


Figure 7.2: Mitsubishi Power two casing turbine, High Pressure (HP)/Intermediate Pressure (IP) - Low Pressure (LP) for 312 MW, Gas Turbine Combined Cycle plant applications.

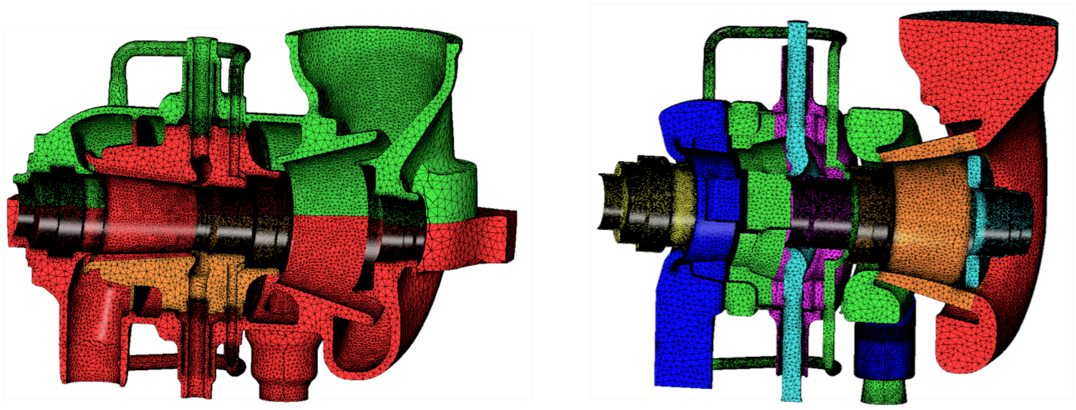


Figure 7.3: Reconstructed 3D models of the solid and fluid domains of the HP/IP region of the steam turbine, showing the coarse mesh definition for thermal network model use.

7.4 Low Density Data Mapping

Low density mapping techniques allow sparse data to be mapped to a higher resolution to define regional or surface boundary conditions. These methods are most effective when the expected behaviour is known. If the system can be considered cylindrical axisymmetric, these criteria can be employed and the overall accuracy of the fit improved. In the following section mapping options are evaluated and a new method for optimised probe placement is introduced.

7.4.1 Spatial Interpolation

A gridded interpolation function is fit to the data and used to evaluate the fit variable at the desired location. Once calculated, the function return is relatively fast and large regions can be evaluated without significant cost. The method is best suited for low dimensional data in 1D or 2D. Although viable in higher dimensions, the volume of data required to achieve an accurate fit increases, along with the computational cost to compute and return the fit.

In the case of analysing rotational equipment, transferring to axisymmetric cylindrical coordinates can simplify the method. A 2D axis-theta fit is often more effective than a 3D x-y-z equivalent. SciPy 2D interpolation methods rely on an underlying Delaunay triangulation procedure. The method generates triangles between all data points such that no point is within the boundary of an element, aiming to maximise the minimum angle of the triangle elements [80]. Default triangulation can be poor and leads to erroneous data. The user should evaluate the sparse triangulation before employing the spatial interpolation.

Figure 7.4 demonstrates applying this method to the target thermocouple locations analysed in the previous chapter. In this application, the impact of poor triangulation is clearly seen. The sparse measurement locations cannot be used alone and must first be supplemented with additional data. Interpolated pseudo-thermocouples can be added at specific locations to improve the fit, taking their value from a user specified interpolation of the nearby physical data.

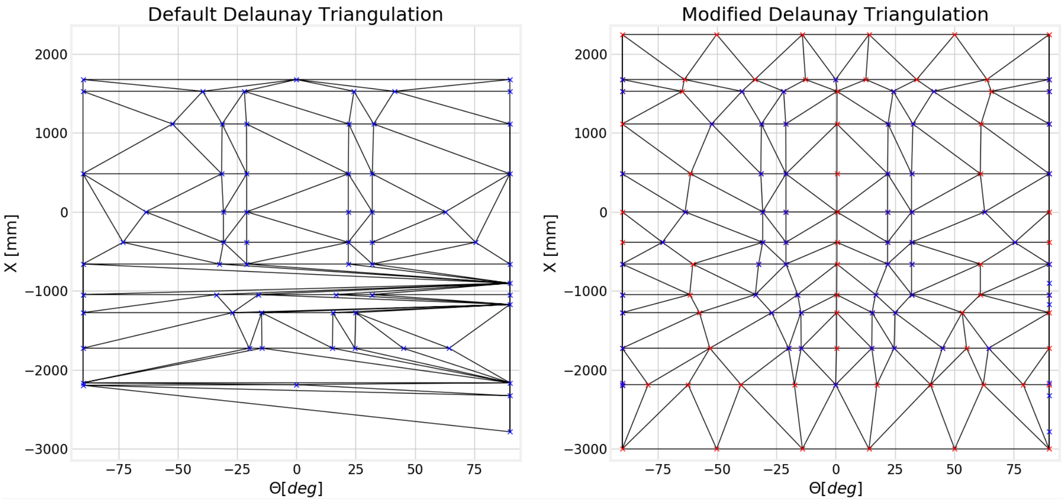


Figure 7.4: Comparison of the default and modified Delaunay triangulation calculations for the MHI thermocouple data in the axis (X) and circumferential (θ) plane.

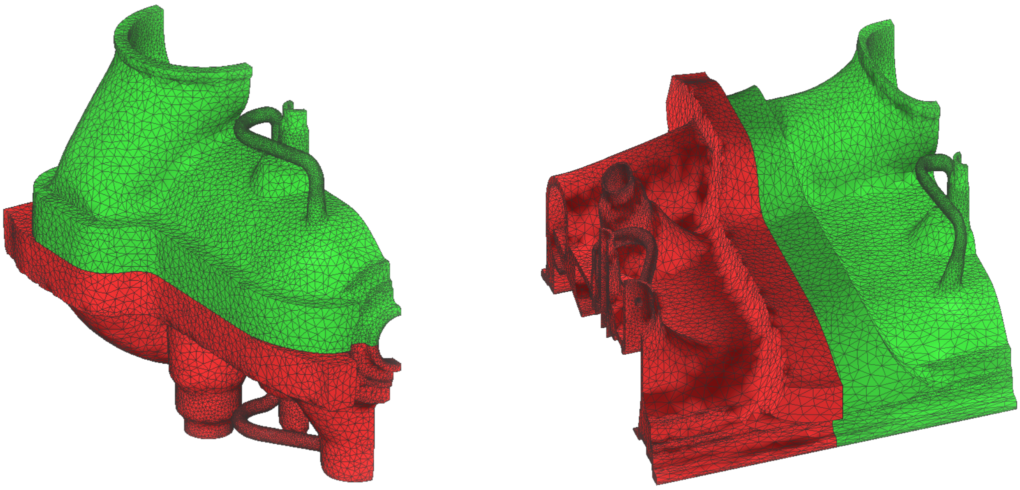


Figure 7.5: Outer casing of the MHI turbine geometry, unwrapped circumferentially to show the $X - \theta$ surface on which the triangulation is applied.

The outer casing has many non-circumferential features, e.g. the nozzles and balance lines, which need special handling. This can be achieved by slicing the surface with defined geometric constraints. Each region can then be assigned its own mapping function and the results blended to achieve a smooth temperature transition.

Figure 7.6 shows the regions that were assigned unique data maps in red, with the blending regions to the main cylindrical body shown in blue. The process allows complex 3D thermal profiles to be recreated by employing multiple low order 2D maps with simple blending. Significant user input is required during the initial specification. However, once defined, the method allows fast processing of surface boundary conditions from sparse thermocouple measurements. Examples of the resulting surface temperatures are shown in Figure 7.7 and can be used as boundary conditions for thermal network models.

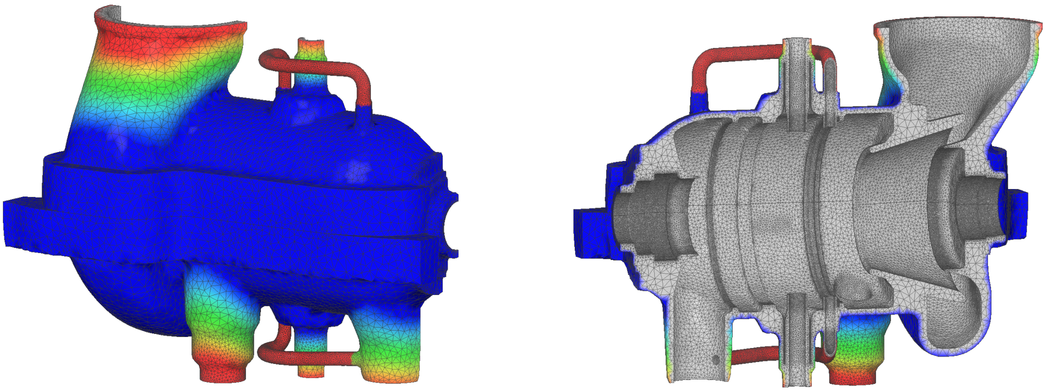


Figure 7.6: Boundary condition blending regions, showing the external mapped surface with special treatment of the nozzles, balance lines and masked internal region.

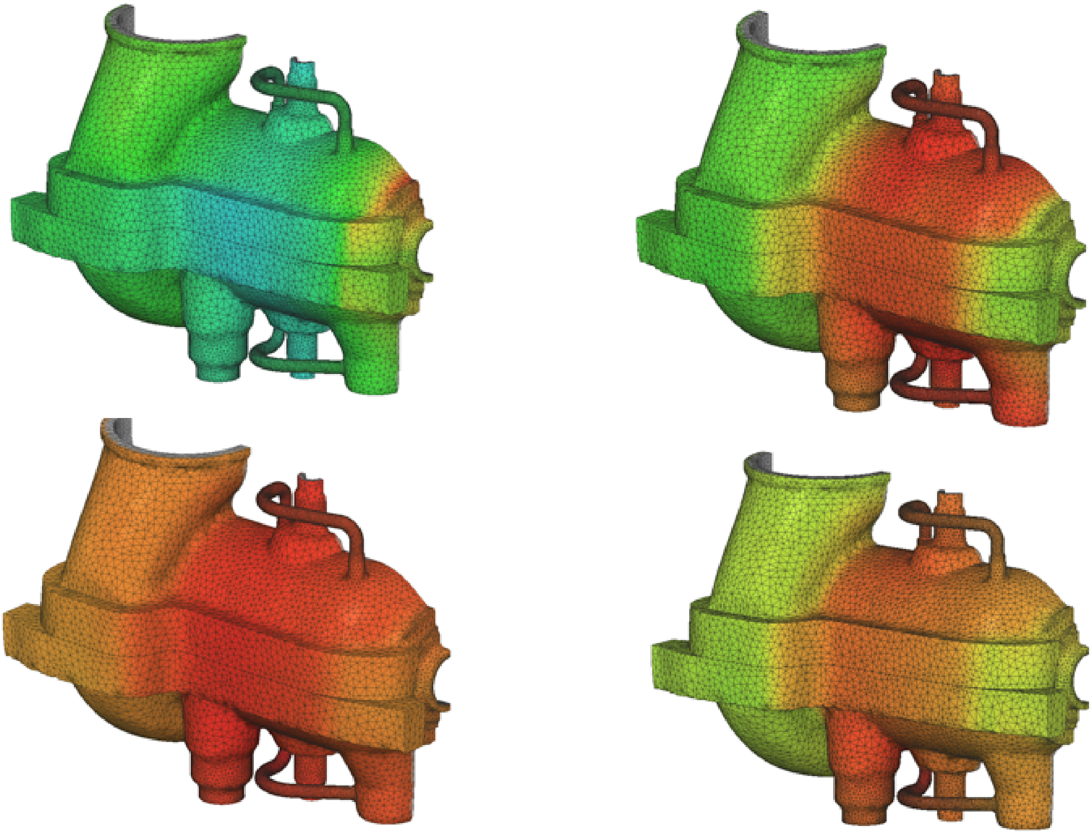


Figure 7.7: Example cases of the resulting mapped surface boundary temperature at four different conditions: gland heating, start-up, design point operation and shutdown.

A low order thermal network solver, developed later in this thesis, was used to simulate the outer surface temperature of the MHI turbine. Figure 7.8 compares the two surface temperatures. The two solutions are broadly similar, but with notable discrepancy on the balance line and nozzles. The Delaunay method only had access to thermocouples on the casing shell and nozzles, requiring a linear blend between the two to approximate the balance line. This method has missed a localised heat sink on the return line and therefore over predicts the temperature of this region. Following the comparison of the two analyses, it is recommended that a balance line probe be installed to directly track the temperature of this casing feature.

In addition, the Delaunay map for the HP inlet nozzles extends downwards past the shoulder of the nozzle sleeve. This extended map smooths the thermal gradient on the nozzle, making the temperature appear cooler and more uniform. A notably stronger thermal gradient is seen in the numerical result. The use of multiple analyses has exposed these issues in the Delaunay interpolation. It is recommended that adjustments be made to the nozzle maps and the additional probe installed before utilising this boundary type.

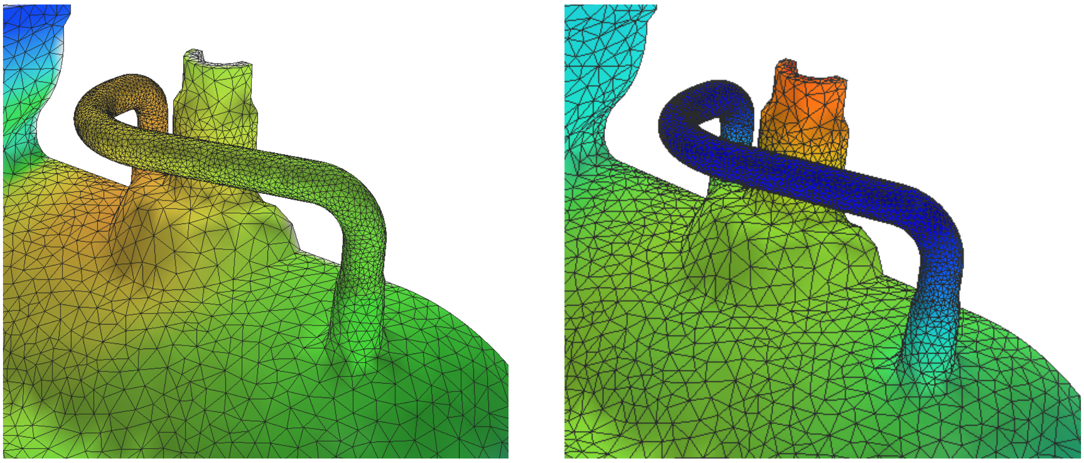


Figure 7.8: Comparison of the balance line temperature for the Delaunay interpolation (left) and thermal network numerical solution (right).

7.4.2 Neural Network Autoencoder

Autoencoders have grown in popularity and are routinely applied in image analysis. They are commonly used for dimensionality reduction, using unsupervised methods to discover a reduced *latent space* representation of the data. The network structure creates a bottleneck at its centre before returning to the full dimensionality at the output. The network is trained to recreate the input having passed through the compressed state, effectively encoding itself whilst simultaneously solving the decoder for reconstruction [81]. Similar in concept to Principal Component Analysis in the previous chapter, this tool can be used to find a reduced data set which sufficiently approximates the system.

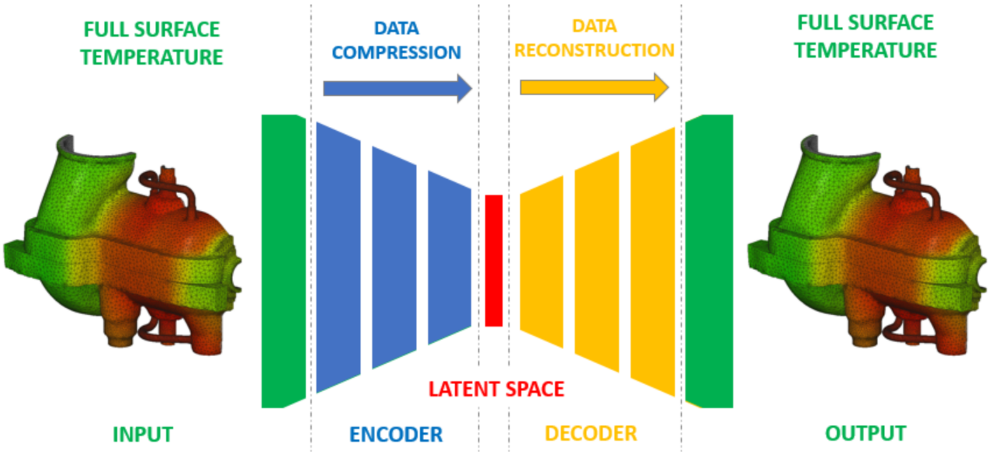


Figure 7.9: Autoencoder method showing data compression and reconstruction from the learned latent space representation.

A new methodology is presented, adapting the concept of an autoencoder to transfer data between fidelity states. Applying the method to both the high fidelity mesh data and low fidelity measurement data, additional constraints can be introduced on the loss function. Forcing the two models to minimise the difference between their respective centre layers, they are constrained to find a common latent space representation. This common layer can then be used to map across the fidelity range, by simply exchanging the combination of encoder and decoder used. In practice, small differences in the respective latent spaces will exist, so a dedicated decoder may be preferred to compensate any error in a given transfer function.

The method was applied to analysing the MHI turbine geometry using data from the previous Delaunay interpolation. The low fidelity input took 70 sparse thermocouple inputs. The high fidelity data used the Delaunay temperature for the 12,757 nodes on the casing exterior surface. The latent space method removes the additional Delaunay tasks: making pseudo-thermocouples, splitting cylindrical regions and blending different map functions. Figure 7.11 shows the results and confirms suitable reconstruction accuracy, with time averaged error below 1K.

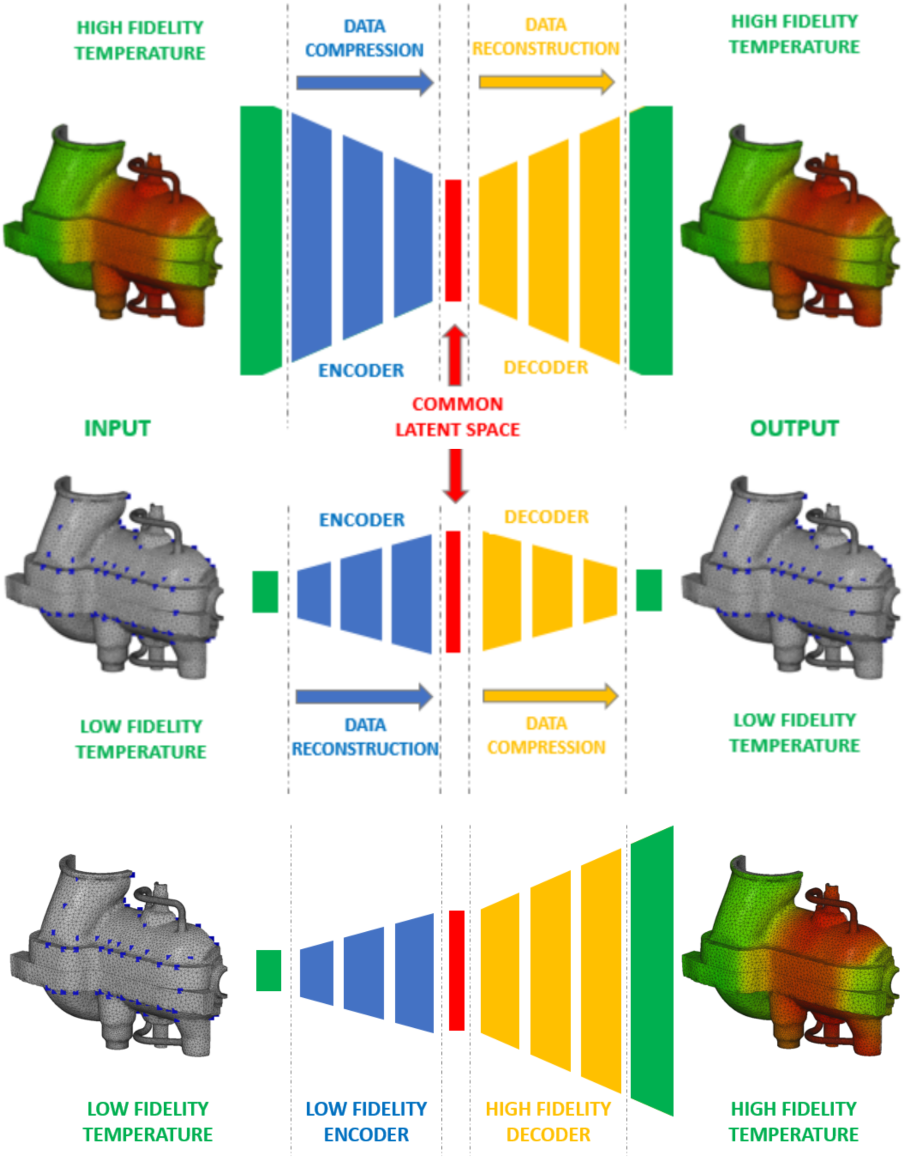


Figure 7.10: High and low fidelity autoencoders are independently trained to learn a common latent space representation. Low to high fidelity data mapping is then achieved by exchanging encoder/decoder combination and retraining.

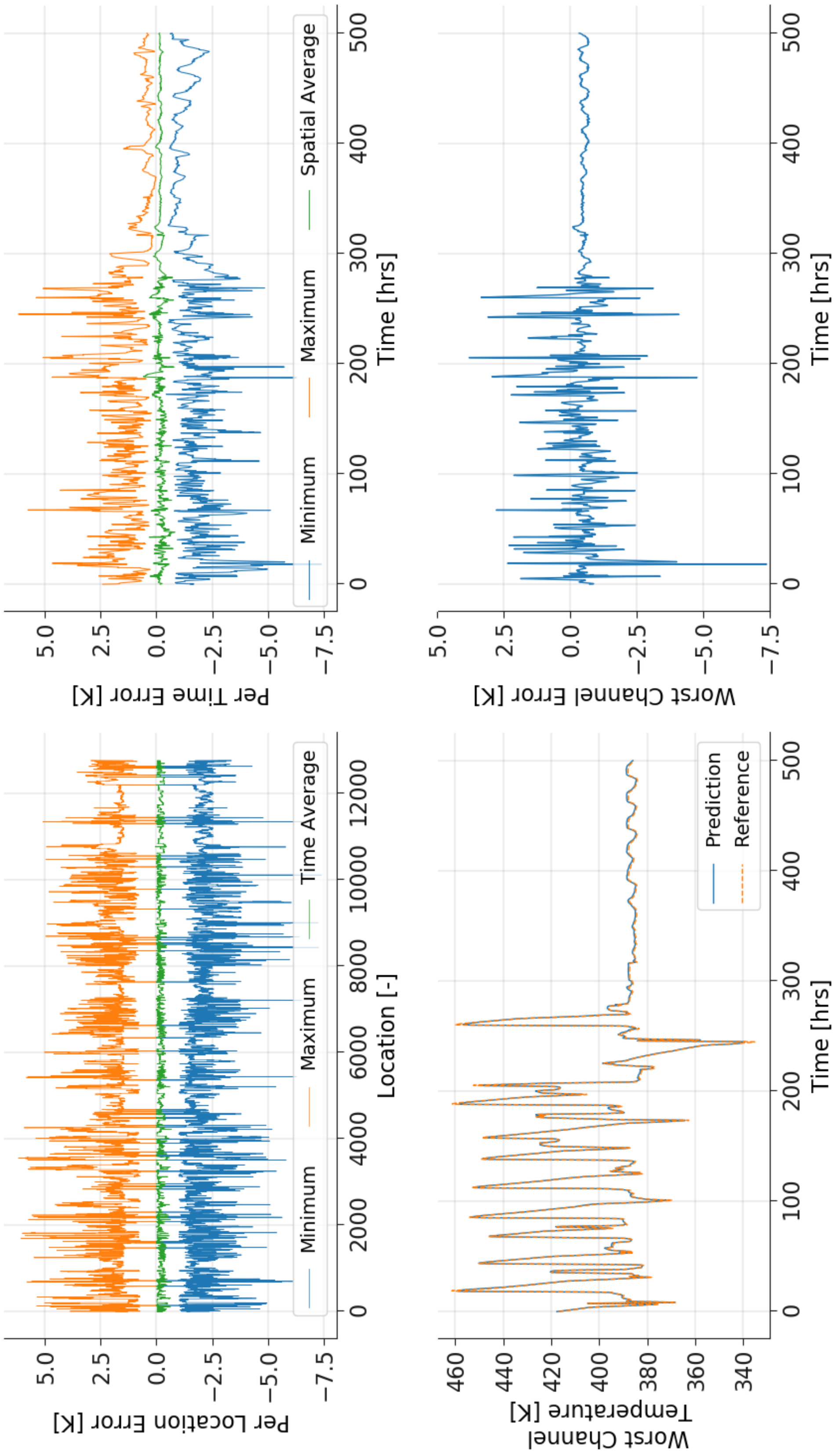


Figure 7.11: Performance of the default autoencoder method, using the existing thermocouple locations on the MHI turbine. Accuracy across the 500 hours of data is good, with peak error in any one surface location of -7.38K, and worst location time-average absolute error 0.65K.

7.4.3 Measurement Probe Optimisation

Building a new hybrid test facility, it is known that the user intends to map low density measurements at the start of the campaign. Data measurement in a high speed test facility can cost several hundred pounds per channel and reducing the required measurements is desirable. One should therefore look to minimise the number of probes, whilst optimising their location for the best data. Similar concepts have been proposed based on a Kriged Kalman Filter for optimised probe placement [82].

A new approach is presented as an extension of the encoder-decoder method. The network is given the full domain information at the input and additionally tasked to select its preferred measurement positions. The user can introduce additional constraints, such as machining or access restrictions, by simply masking the available locations. A dynamic selection architecture is presented for a predefined number of measurement points, top-k methods are used to clip the data and select the preferred inputs. An optional boolean mask is supported, allowing the user to screen the available locations.

The method was tested on the same MHI turbine data, requesting only 32 inputs. It was observed that the dynamic mask preferred to cluster probes close together, spanning the largest thermal gradients on the surface. Tightly clustered solutions are likely infeasible in practice due to space and machining requirements. A boolean mask input was required to force the model to space the probes apart. This restriction and resulting selection can be seen in Figure 7.13. Although mathematically optimum, these probe placements still appear counter intuitive at first glance. If desired, a more structured placement can be achieved by further restricting the user defined boolean mask.

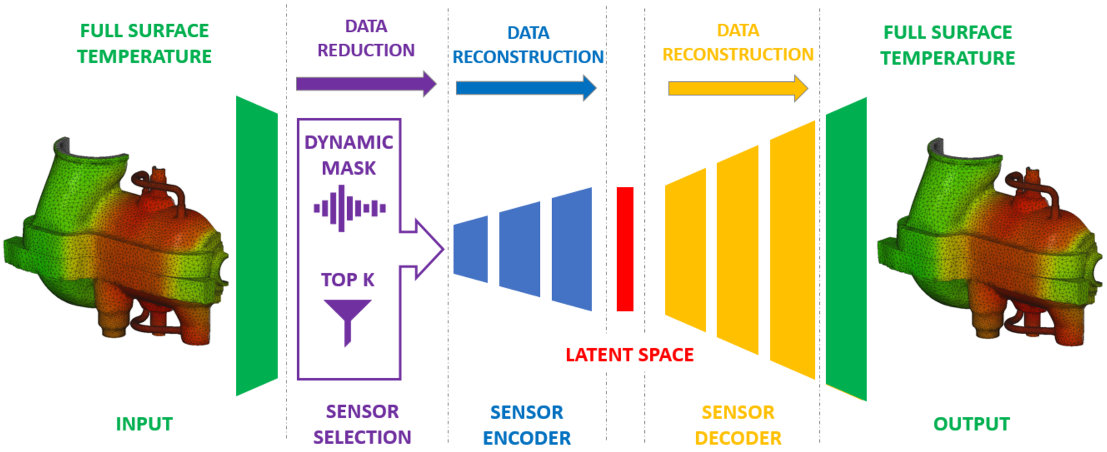


Figure 7.12: The low fidelity definition is replaced by the full fidelity with a dynamic sampling mask. The input selection is optimised to best reconstruct the output.

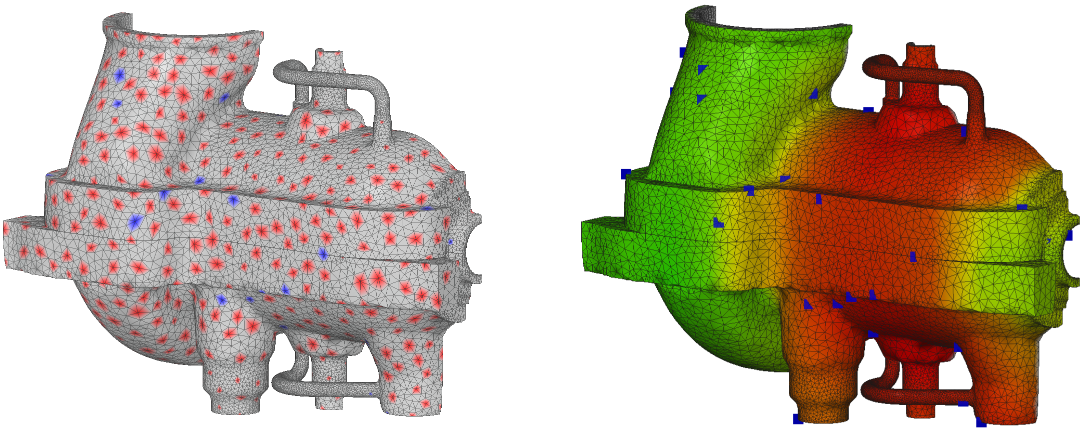


Figure 7.13: Boolean mask of the available points for selection (red) with the final selected positions (blue), showing their location relative to the thermal gradients in the machine.

The performance of the reduced model is shown in Figure 7.14. The model allowed full surface reconstruction within 9.70K, showing comparable performance to the 7.38K error in the initial test. Requiring only half of the measurement inputs, the automated selection method offers benefits in speed, data storage and cost. However, by placing a higher dependence on any one individual input, the reduced selection model is likely more prone to errors from instrumentation faults. This was not tested but may be evaluated in future using simulated faults on the data inputs.

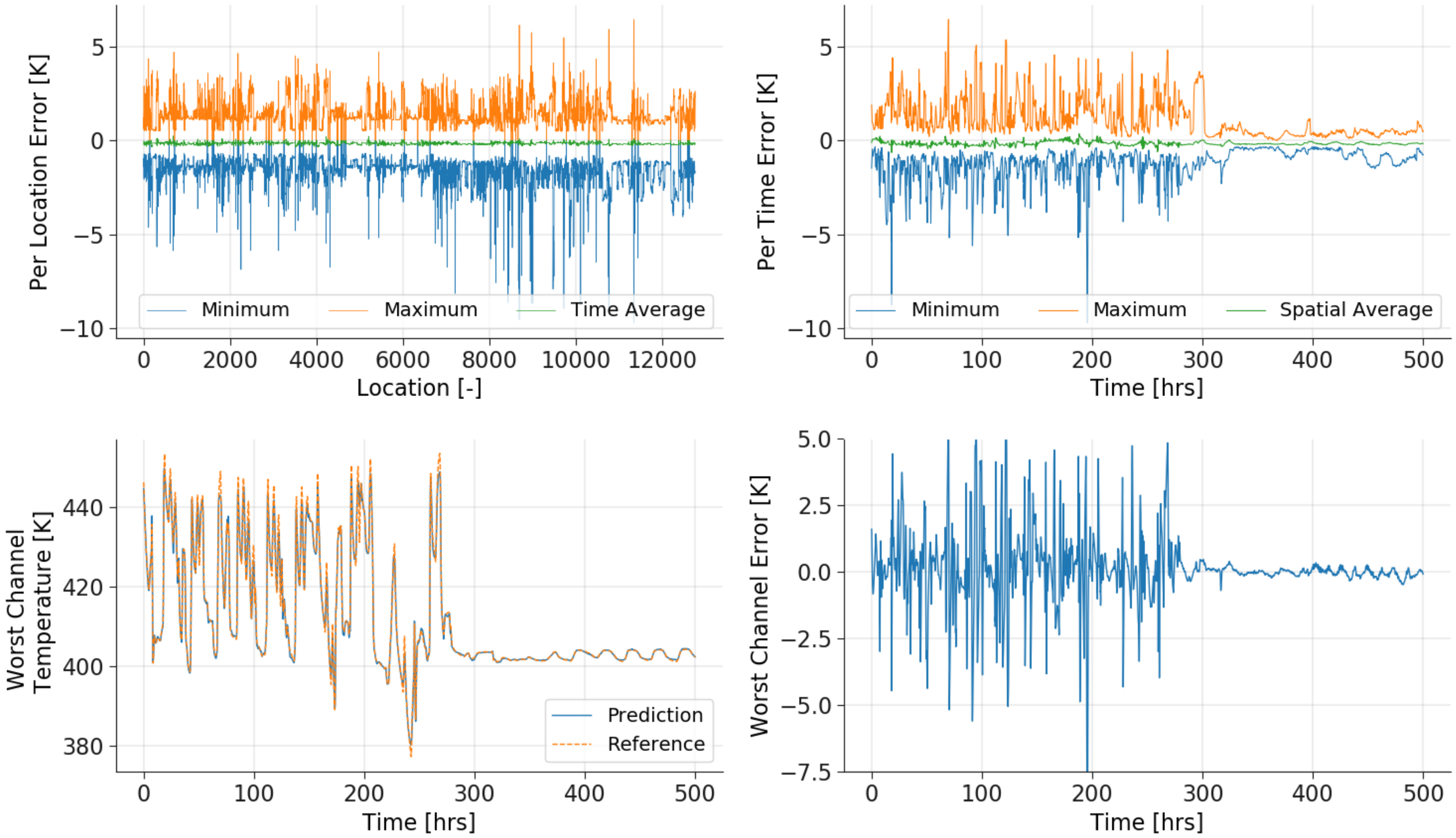


Figure 7.14: Performance of the optimised autoencoder method using only 32 selected locations. Accuracy across the 500 hours is comparable to the initial 65 thermocouple model, with peak error in any one location of -9.70K, and worst location time average absolute error 0.69K.

7.5 Medium Density Data Mapping

Fine spacing thin films and thermal camera observations are becoming more common in heat transfer analysis of experimental facilities. These measurement types produce medium density data that provide either partial or full definition of an entire surface. This higher fidelity information allows additional statistical and mapping methods to be used for high resolution reconstruction.

7.5.1 Kriging Method

Kriging is a statistical method that allows the value at a desired location to be computed by weighted fractions of known reference data in the surrounding neighbourhood. The method is well suited to spatial inference cases where a single variable is desired from a complex field [83]. The method assumes stationarity - that the behaviour is the same at all locations in the domain, and isotropy - that the behaviour is equal in all directions. Taking constant material properties, these assumptions are acceptable for temperature estimation within a single component.

Several variations of the Kriging method exist, with the most common being Simple, Ordinary and Universal. Simple uses a constant user defined mean which in practice is often unknown and limits the application. Ordinary uses a constant unknown mean which is solved alongside the model, and Universal uses a regression mean that varies with the spatial parameters.

Kriging models are based on a spatial variogram, evaluating the covariance of reference points and the distance between them. This prevents the interpolation becoming skewed if several reference points are closely positioned. The method is poorly suited to extrapolation and should not be used to evaluate a variable outside the vicinity of the known neighbourhood. This makes it best suited for boundary surface or internal body interpolation [83].

The main advantage of the Kriging solution is that it does not require known connectivity of the reference data. This makes it valuable in the case of handling multiple fidelity systems where congruent connections may not exist. The main disadvantage is that the method assumes a continuous field and therefore suffers in the case of handling U-shaped geometry where a spatial disconnects exist.

The Universal Kriging method, via PyKrige [84], was tested in the case of interpolating the boundary surface of the MHI turbine. A sparse data set of the surface temperature was extracted by point-wise sampling. The density was progressively reduced to assess the effect of data sparsity. In each case, the sampling was chosen to maximise the spatial distance between sample locations. Points were permitted on all external surfaces including the main shell, nozzles and balance lines. A single Kriging variogram was applied to the entire outer surface, evaluating the error between the reference and reconstructed thermal profile. Figure 7.15 shows two of the sampling cases used for 128 and 512 points. In each image the locations of the activated points are highlighted red.

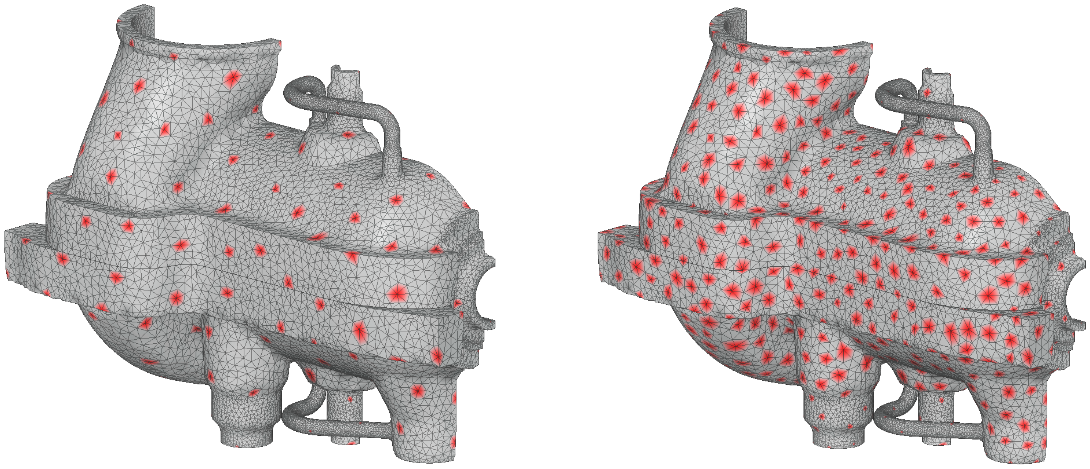


Figure 7.15: Surface sampling points for two density options, 128 and 512 points, used as the input reference locations for the Kriging model.

Figure 7.16 compares the model performance using different density inputs. The histogram shows the node count in each error band for the total 12,757 nodes on the exterior surface of the casing. The mean and 3σ values are shown for each density, identifying the error bounds covering 99.73% of the surface. The 512 density model performed best, able to reconstruct the surface within 6.89K. This is comparable performance to the autoencoder method but requires several times more input data. The absolute peak errors and 3σ values of the low density Kriging models are very large. Errors in excess of $20K$ are seen and these are not well suited for this application. The Kriging method is only viable for very localised predictions or high density data samples.

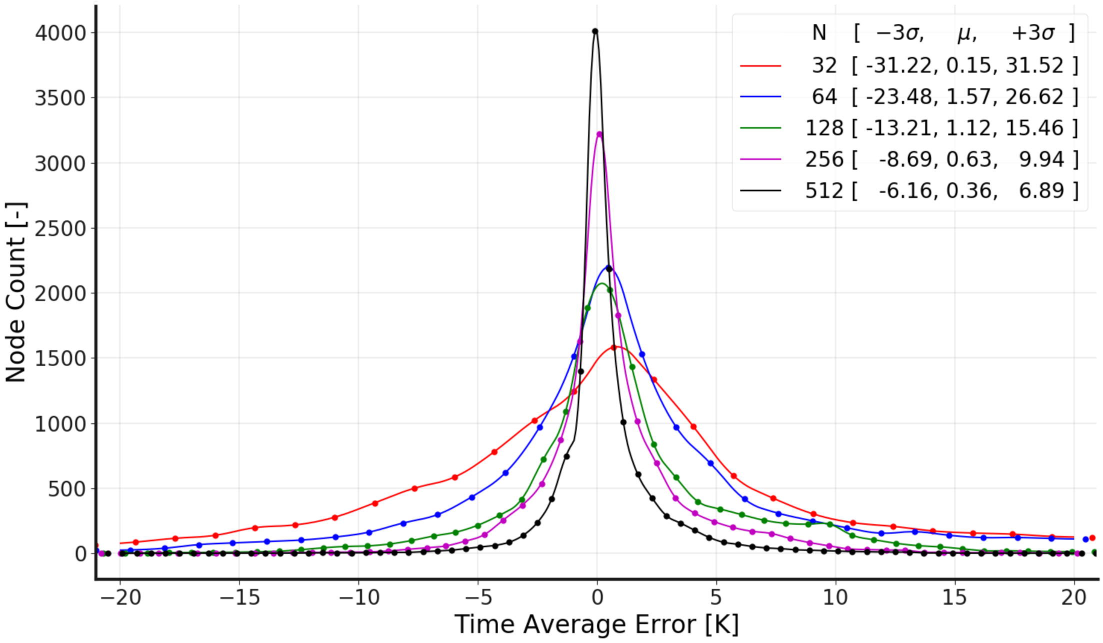


Figure 7.16: Histogram comparison of the time averaged surface temperature error when using different density inputs to the Kriging model, showing the mean μ and 3σ values of the error.

The regions of large thermal gradient between the high and intermediate pressure sections of the machine suffered most. These errors appear high but are caused by a small displacement in the gradient's position. This is best seen in Figure 7.17 which shows the error greater than 2K. Large regions of the surface are shown grey because they fall under the 2K limit. The errors are highlighted in the transitions between the casing shell and the flange where the gradient is misaligned to the reference value.

In addition, areas around the nozzles and balance lines showed the largest error. These locations are weighted highly in the spatial variogram due to their proximity to several large U-shaped and overhanging geometries. The errors are caused by the disconnect in these regions and incorrect assumption of a continuous field. For this application, the results confirm the requirement to use a medium density Kriging variogram that spatially isolates disconnected regions. This method is later used in Chapter 8 for inverse blocking calculations in the thermal network model software.

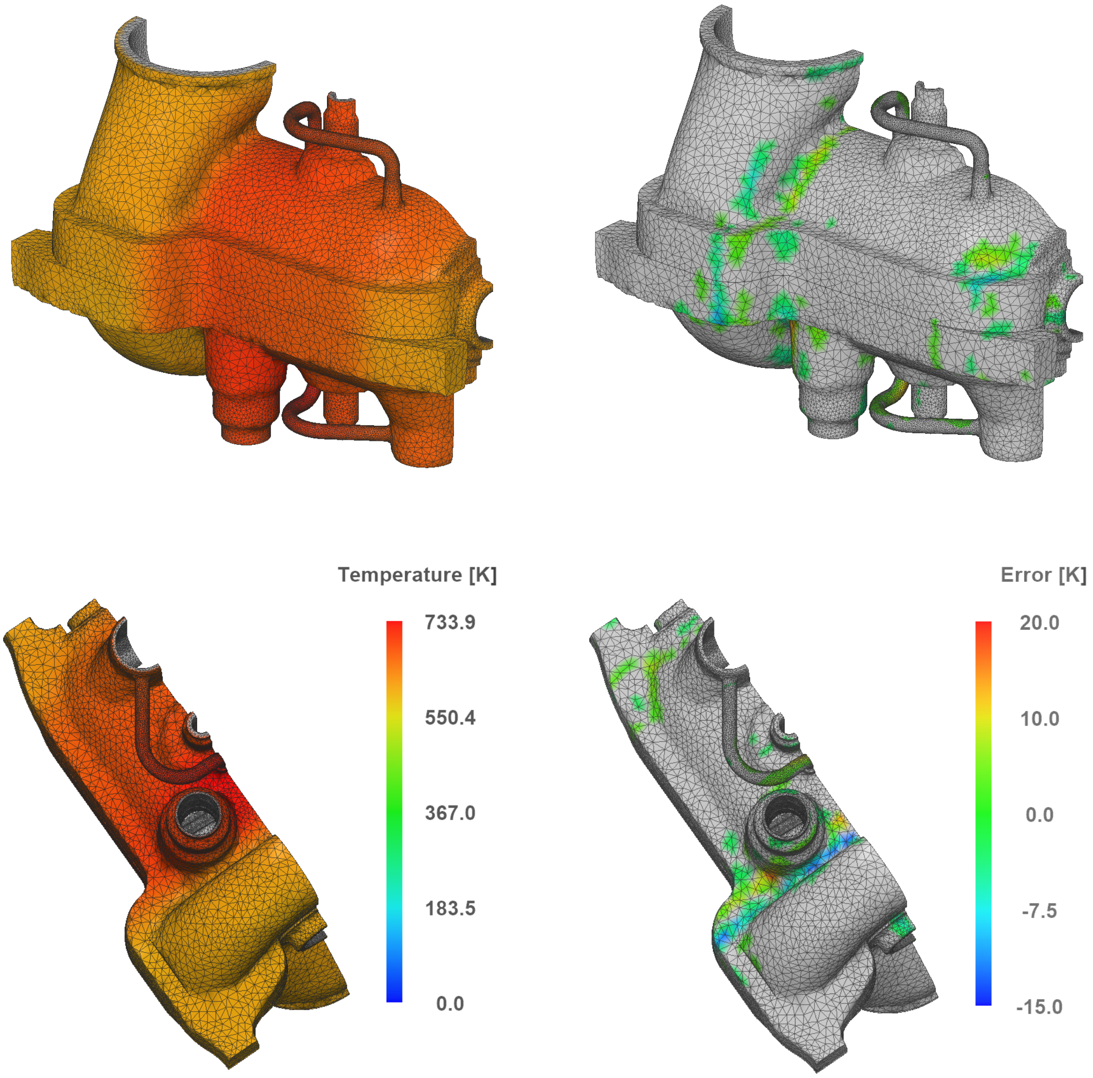


Figure 7.17: 3D surface plots of a single time step, showing the Kriging calculated thermal profile using the 512 input density model and the associated error values greater than 2K.

7.5.2 Thermal Camera Mapping

Thermal imaging cameras are increasingly used as a non-invasive method to measure component surface temperature in an experimental environment. These cameras provide a high resolution temperature image of the surface but are limited by line of sight access, often obstructed if analysing shrouded components. Many users choose to analyse thermal image data in the native 2D, thus neglecting curvature and other geometric effects. Presented below is a novel 3D mapping process that allows these data to be used for boundary conditions in hybrid model simulations.

Traditional post-process methods are applied pixel-wise to the image assuming the geometry surface to be planar one dimensional. The 2D pixel data is split to a time series of 1D data sets then recombined to show a final result. Direct comparison between runs, or with other measurement data, requires the user to manually register the image to account for misalignment. Using computer vision techniques developed by the Oxford Robotics Institute [85], the imaging camera matrix can be defined. Knowledge of the CAD model and perspective projection matrix can then be used to register the image to 3D. The mapped data can then be used for improved quasi-1D analysis or direct 3D simulation.

Figure 7.18: FLIR thermal camera used in linear cascade tests at the OTI.

A perspective projection camera is represented by a 3x4 matrix. The eipolar coordinates between two views can be represented by the fundamental matrix P, which combines the camera calibration, rotation and translation matrices. Computing P from a set of known scene and image locations reduces to a vector minimisation problem of two equations.

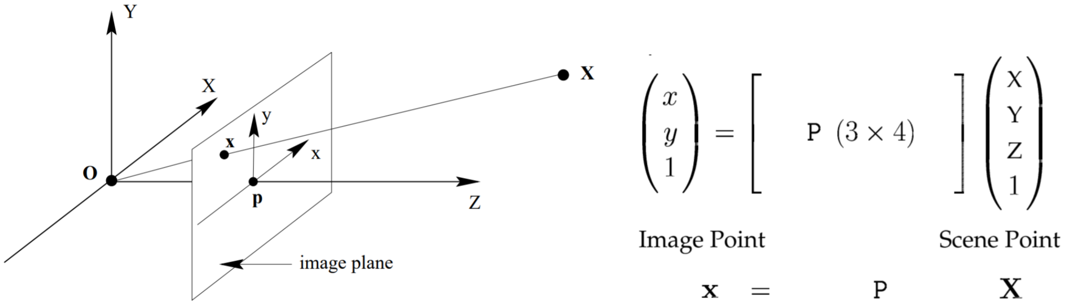


Figure 7.19: Imaging plane, showing matrix P that maps world coordinates to the image.

$$\mathbf{x_i} = P\,\mathbf{X_i} \qquad x_i = \frac{p_{11}X_i + p_{12}Y_i + p_{13}Z_i + p_{14}}{p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}} \qquad y_i = \frac{p_{21}X_i + p_{22}Y_i + p_{23}Z_i + p_{24}}{p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}}$$

This can be rearranged to give linear equations in the matrix elements of P

$$x_i\Big(p_{11}X_i + p_{12}Y_i + p_{13}Z_i + p_{14}\Big) = p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}$$

$$y_i\Big(p_{21}X_i + p_{22}Y_i + p_{23}Z_i + p_{24}\Big) = p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}$$

Then concatenated to generate a $2n$ x 12 matrix, A, such that $A\ \mathbf{p} = 0$.

$$\begin{pmatrix} X & Y & Z & 1 & 0 & 0 & 0 & 0 & -xX & -xY & -xZ & -x \\ 0 & 0 & 0 & 0 & X & Y & Z & 1 & -yX & -yY & -yZ & -y \end{pmatrix} \mathbf{p} = 0$$

$$\text{where } \mathbf{p} = (p_{11}, p_{12}, p_{13}, p_{14}, p_{21}, p_{22}, p_{23}, p_{24}, p_{31}, p_{32}, p_{33}, p_{34})^T$$

This must be solved using a linear least squares solution that minimises $|A\ \mathbf{p}|$, taking the smallest eigenvector from the singular value decomposition of A.

$$\min_p \sum_i \Big(\big(x_i, y_i\big) - P\big(X_i, Y_i, Z_i\big)\Big)^2 \tag{7.1}$$

Once the camera matrix P is known, the image data can be transferred to the 3D geometry. A surface point cloud of the 3D model is first extracted, then mapped to the image using matrix P. The corresponding spatial and data values can then be found. This process is easiest done using a surface stereolithography file (STL) of the CAD geometry. Exporting the file from a meshing software package, e.g. ICEM, provides full control on the density and coverage of the sampling point cloud. The STL method has the added advantage that the previously discussed curvature correction methods can easily be applied.

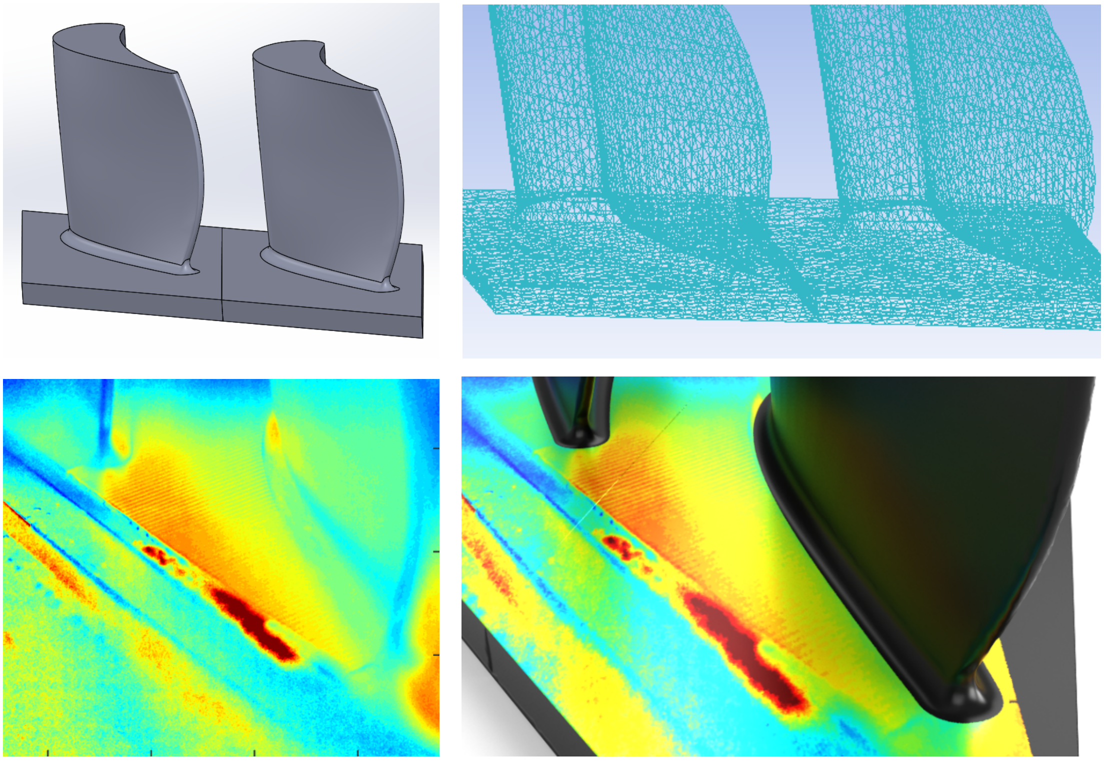


Figure 7.20: CAD geometry of a linearised nozzle guide vane and STL surface mesh exported from ICEM for point cloud control, with a native 2D thermal image from a FLIR camera [41] and the mapped 3D registered Blender data of the endwall temperature.

Open source rendering software Blender [86] can be used to generate custom 3D visualisations. The application allows specification of the camera, material, geometric properties and nodal data. A template Blender material and camera file has been created and a Python 3.6 script written to automate processing of the temperature mapped STL file. Registering the data to the 3D has other advantages besides allowing curvature correction methods and direct support for 3D simulations.

- Exact camera alignment is not critical and registering corrects for positional variation between runs. Static views can be created in the visualisation software, ensuring a consistent output across a test campaign.

- Multiple cameras can all be mapped to a single 3D model, allowing several data sets to be efficiently merged for full coverage of a vane surface.

- HD video or images can be created, controls within the software provide a higher level of functionality for improving image quality. For example, data can be clipped by surface, thereby enabling only the geometric features of interest to be displayed.

7.6 High Density Data Mapping

Mapping high density data to a lower resolution mesh is equally important in multi-fidelity analysis. Contrary to the above methods, the main challenge is not the accuracy of the fit but the speed of the sampling calculation. Transferring data from a fine CFD mesh of many million elements to a coarse network model of a few hundred thousand requires an efficient transfer algorithm.

7.6.1 One-to-One Decimated Mesh

The simplest solution is to achieve a one-to-one mapping between the two meshes. This can be done by mesh decimation, directly inferring a coarse mesh from the fine. Several tools exist, for example Blender mesh decimate [86], that allow a mesh to be simplified by removing nodes then repairing facet connections. By rejecting smoothing operations, the retained nodes are left in their original position, producing a direct data mapping between the two meshes.

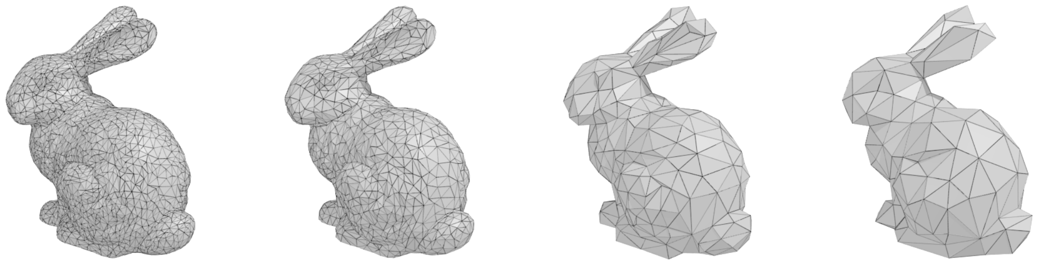


Figure 7.21: Example mesh decimation in Blender using the Stanford bunny geometry. Demonstrating the disadvantage of this method in the poor resolution of high curvature regions near the ears and feet.

Although the final data transfer is simple and efficient, the resulting coarse mesh is often less than ideal. Decimated meshes without smoothing can cause large skew in the mesh elements, impacting the accuracy of the low fidelity solver. Smoothing the mesh after decimation adjusts the nodal positions and loses the direct mapping between the locations. Small node offsets may be tolerated depending on the use case if data gradients are low.

This method is best suited to simple geometries, with large planar regions and low curvature, where the mapped field data has small gradients. The method is limited to nodal data only, with cell and face centres being updated in the decimate process. In the case of mapping between two turbine meshes, with complex cavity geometry and large regions of variable curvature, this approach is not well suited and a bespoke coarse mesh is preferred.

7.6.2 Exhaustive Nearest Neighbour

The nearest neighbour method is the easiest to implement for mapping two congruent meshes with their own independent elements. The lower resolution mesh searches in the dense mesh for the closest reference and extracts the field variable from this point. Either node, face or cell centred data can be used offering more flexibility than mesh decimation.

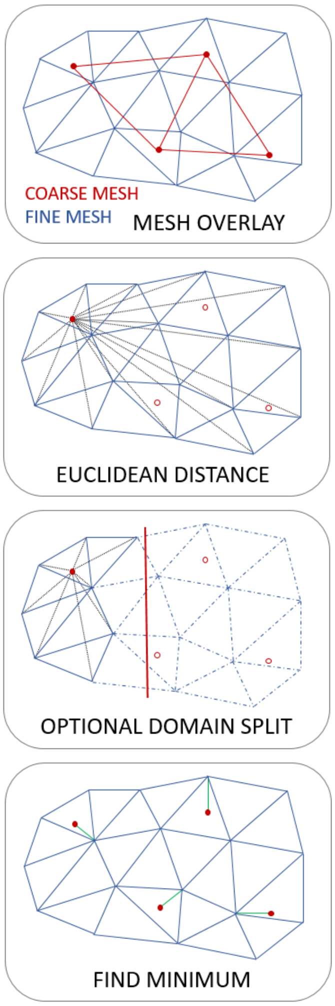


Figure 7.22: Schematic of the exhaustive euclidean nearest neighbour search.

Several algorithms exist for nearest neighbour searches but these tend to focus on finding the k-nearest [87]. In this application, only the first nearest point is required and the connectivity is not important. This simplifies the algorithm, which can be reduced to finding only the neighbouring point with minimum euclidean distance. An exhaustive search of all points in the domain is O(n_1 x n_2), where n_x represents the number of points in each mesh.

For large meshes, exhaustive searching can have excessive computation cost. In these cases the domain can be split and the search cost reduced by handling zones independently. How to split the domain, such that no point is isolated from its paring counterpart, can be non-trivial for complex geometries. Either the process should be repeated for different domain splits, or sectioning should be defined by the user. The preference for a single pass automated process gave rise to the hash table method discussed below.

7.6.3 Hash Table Nearest Neighbour

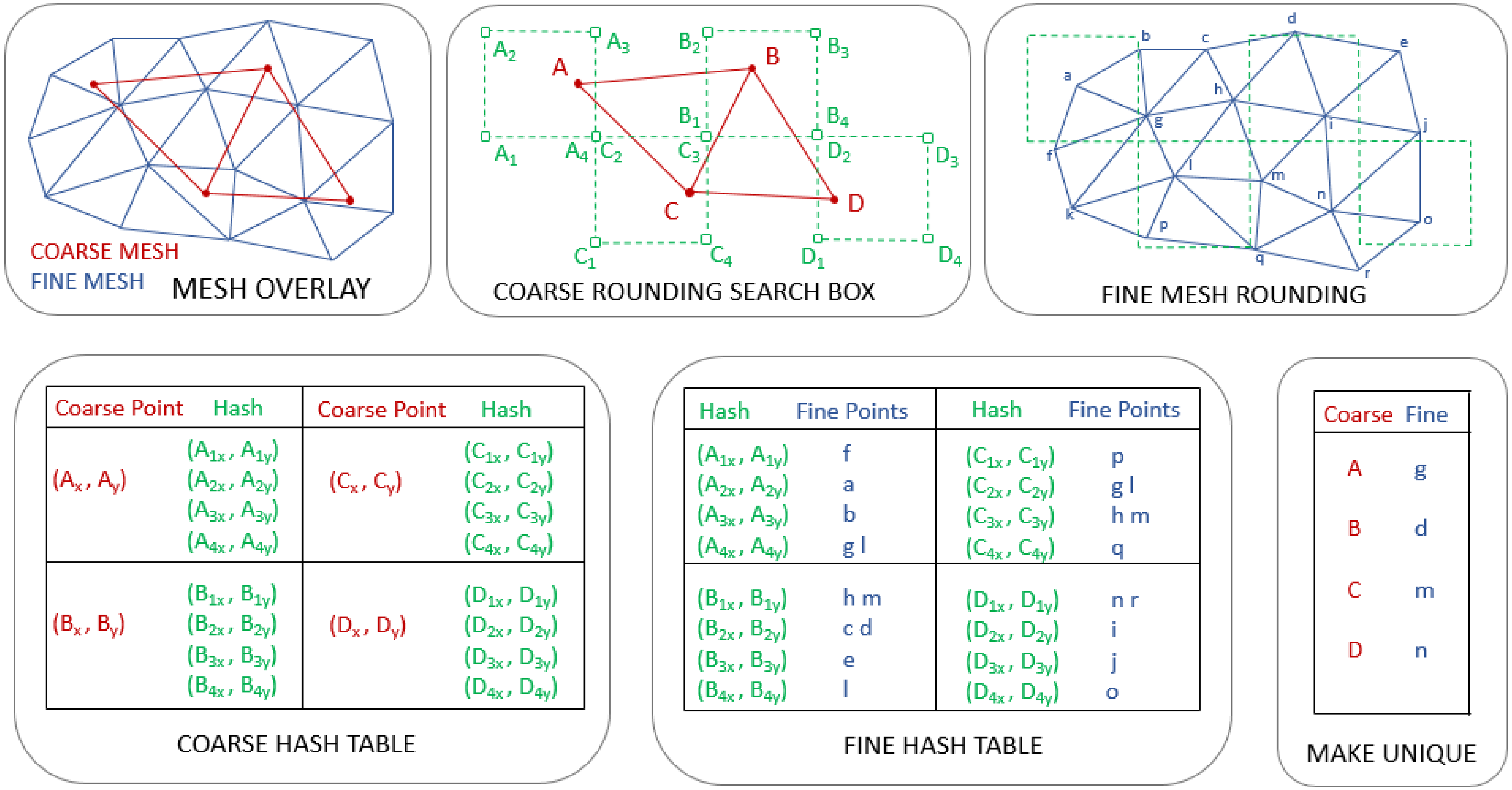
Hashing is a method to create a fixed size indexed table from a larger data set. Values are passed through a hash function that generates the appropriate index or hash code for the table. The data is then stored in the table under the corresponding hash. This method is widely used in big data handling and can be easily implemented with Python dictionaries.

Rather than map one mesh to another directly, a common hash table can first be generated, then both mesh independently mapped to the same table. Hashing each mesh and zone can be handled separately, greatly reducing the computational effort compared to an exhaustive search.

Hash values or dictionary keys can take many different forms and numeric, tuple and string values are all permitted in Python. A novel 3D spatial hash method is presented, using a rounded tuple of the coarse mesh coordinates to build a common spatial hash. Adjusting the decimal place rounding, the size of the collected region can be tuned and the search volume defined. This method efficiently builds a localised search region around each coarse node, greatly accelerating the nearest neighbour search.

The algorithm was tested on the OTI geometry to sample CFD data to the face centres of a coarse mesh. The face sampling was used to extract flow velocities, allowing an approximate flow field to be sampled. This technique is useful for calculating thermal network resistance values and is discussed further in the following chapter. The search can be extended to include the face bounding nodes if a more accurate area averaged value is preferred.

The fine CFD mesh had 8,145,601 cells and the coarse mesh had 621,161 faces. The hash table method took an average 351.9 seconds to compute all data mapping over five runs. In comparison, the full domain exhaustive nearest neighbour search would require in excess of 870 hours, based on extrapolation from an average 505.5 seconds to compute the first 100 faces.



Coarse Point	Hash	Coarse Point	Hash
(A_x, A_y)	(A_{1x}, A_{1y}) (A_{2x}, A_{2y}) (A_{3x}, A_{3y}) (A_{4x}, A_{4y})	(C_x, C_y)	(C_{1x}, C_{1y}) (C_{2x}, C_{2y}) (C_{3x}, C_{3y}) (C_{4x}, C_{4y})
(B_x, B_y)	(B_{1x}, B_{1y}) (B_{2x}, B_{2y}) (B_{3x}, B_{3y}) (B_{4x}, B_{4y})	(D_x, D_y)	(D_{1x}, D_{1y}) (D_{2x}, D_{2y}) (D_{3x}, D_{3y}) (D_{4x}, D_{4y})

Hash	Fine Points	Hash	Fine Points
(A_{1x}, A_{1y})	f	(C_{1x}, C_{1y})	p
(A_{2x}, A_{2y})	a	(C_{2x}, C_{2y})	g l
(A_{3x}, A_{3y})	b	(C_{3x}, C_{3y})	h m
(A_{4x}, A_{4y})	g l	(C_{4x}, C_{4y})	q
(B_{1x}, B_{1y})	h m	(D_{1x}, D_{1y})	n r
(B_{2x}, B_{2y})	c d	(D_{2x}, D_{2y})	i
(B_{3x}, B_{3y})	e	(D_{3x}, D_{3y})	j
(B_{4x}, B_{4y})	l	(D_{4x}, D_{4y})	o

Coarse	Fine
A	g
B	d
C	m
D	n

Figure 7.23: Schematic of the hash table method, showing the resulting two hash tables with grouping for the nearest neighbour search and the final unique solution for the nearest node.

The method has the following steps.

1. Set the scale and decimal place rounding for the hashing. Each ordinate may have its own rounding and this can be applied element wise if desired.
2. Round all points in the coarse mesh. Points can refer to cell, face or node centres and may be in Cartesian or Cylindrical coordinates.
3. Calculate the region that rounds to zero for the given rounding scale. With equal rounding in each dimension, in 2D this is a square and 3D a cube centred at (0,0,0).
4. Extend the coarse point list to include all possible rounded values from the previous step. This is done by adding the offset rounding region to each reference point from step 2.
5. Take a tuple of the extended coarse point list and create two hash tables with an entry for each tuple.
6. Add the coarse mesh node, face or cell index corresponding to each hash entry in the first table.
7. Apply the same spatial rounding to the fine mesh and take a tuple of the these values.
8. Add the corresponding node, face or cell index of the fine mesh to the hash entry in the second table.
9. Loop through both tables, collecting a list of shared values between the coarse and fine meshes. The size of the rounding region may be adjusted to minimise the number of fine indices collected by each coarse point.
10. Apply a simple nearest neighbour search to each paired collection, returning the closest fine mesh point for a unique mapping.

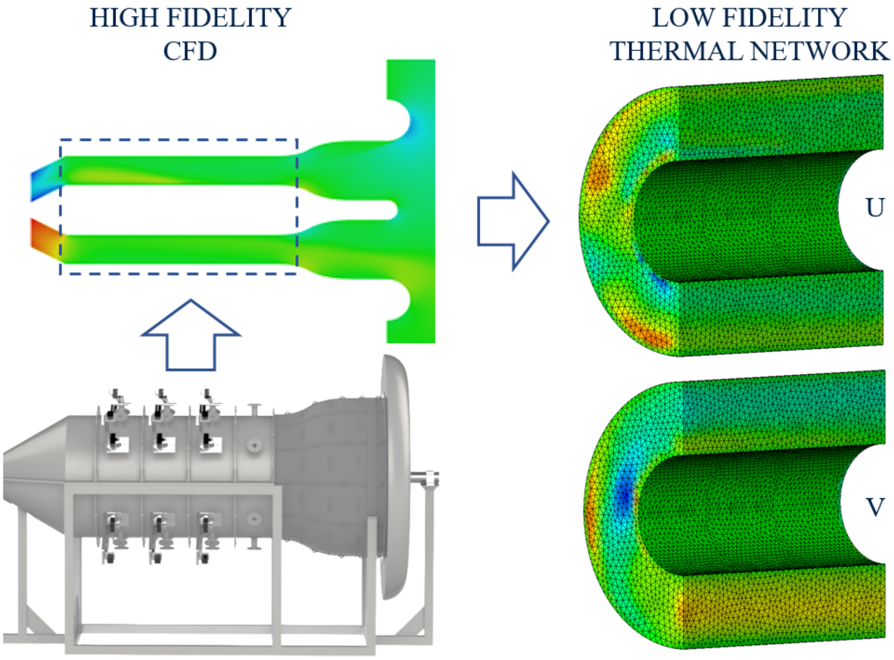


Figure 7.24: Sampling of high fidelity CFD data to a lower resolution sub domain of the mesh test section, using the face centred hash table nearest neighbour method to extract U and V velocity components.

7.6.4 Barycentric Coordinates

When the highest density mesh is still relatively coarse, significant differences may exist across a cell. In such cases, extracting only the nearest value introduces errors in the sampled data and this should additionally be interpolated within the cell. The hash table method should first be used to identify the cell paring and nodes for the interpolation.

Barycentric coordinates are commonly used to define the internal region of a bounded triangular facet or tetrahedral volume. Data is assigned to the defining nodes and internal points found using a weighted sum of these values. This method is widely used in 3D graphics for rendering in vertex fragment shaders [88].

The additional tasks to evaluate all nodal data and weightings make this method more complex than the previously discussed options. However, these tasks need only be done on the first iteration. Once defined, the barycentric weights between two meshes can be stored for routine future calculation.

The method can be used to map data in either direction. Low to high fidelity is prone to over smoothing, but has the benefit that large scale gradients can be retained. The method results in a sparse matrix of size $n_1 \times n_2$ with four populated elements on each row, corresponding to the weight value for each bounding tetrahedral node. Memory limits may cause issues for larger meshes, in such cases the matrices should be stored as single precision 32-bit values.

An internal point splits a triangle into three smaller triangles and splits a tetrahedron into four smaller tetrahedra. These divided regions define the weight applied to each node. The node weight is the area or volume factor of the opposing region. The sum of barycentric coordinates is always 1. Figure 7.25 shows the calculation for a triangular facet and tetrahedral cell [89].

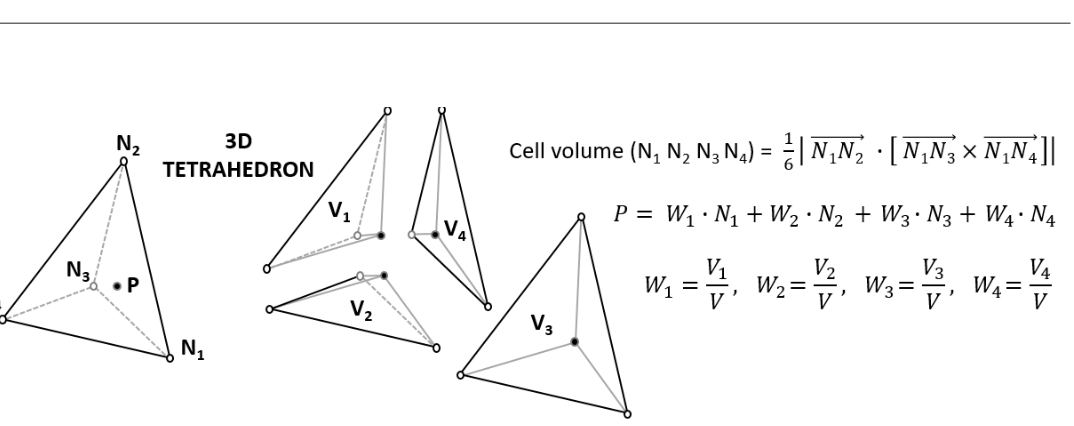


Figure 7.25: Barycentric coordinate method for a 2D triangle facet and 3D tetra cell.

7.7 Summary

Hybrid fidelity methods combine data from a range of analyses to bridge the limitations in any particular one. Central to the success is the ability to map data between congruent systems of varying fidelity. Several methods to support data transfer across the fidelity range have been discussed.

Spatial interpolation methods, based on Delaunay triangulation, were tested in the case of mapping thermocouple data to a turbine surface. In sparse data cases, additional pseudo inputs are required and the user must manually control the triangulation process. Complex surface and temperature profiles were reconstructed using axisymmetric blending of multiple simple 2D methods. Autoencoders were demonstrated in a novel low to high fidelity mapping and an automated probe placement tool. Able to handle both spatial and time varying data, autoencoder methods are well suited to boundary condition calculation on complex turbine geometries.

Geometric mapping for infra-red cameras and Kriging interpolation was tested. The 3D mapped IR method offers several advantages compared to traditional 2D data handling and is now in use at the OTI. The Kriging method was shown to be viable on a complex turbine casing. Areas of high thermal gradient with surrounding geometries were shown to suffer and a medium density variogram with isolated disconnected regions is required. A Kriging solution is used in the following chapter for internal reconstruction of lump mass thermal networks.

High density sampling methods were investigated and a new spatial hash table method proposed. The method demonstrated a significant speed improvement compared to a classic exhaustive search in sampling CFD data. Easy to implement, the method offers a fast solution for domain blocking and is used later in this thesis to automate lump mass calculations.

Barycentric coordinates are essential for handling large spatial variations across a mesh. They offer bi-directional interpolation across the fidelity range and should be used when mapping data between two meshes with high internal gradients.

8

Thermal Network Model Software, Development and Structure

8.1 Introduction

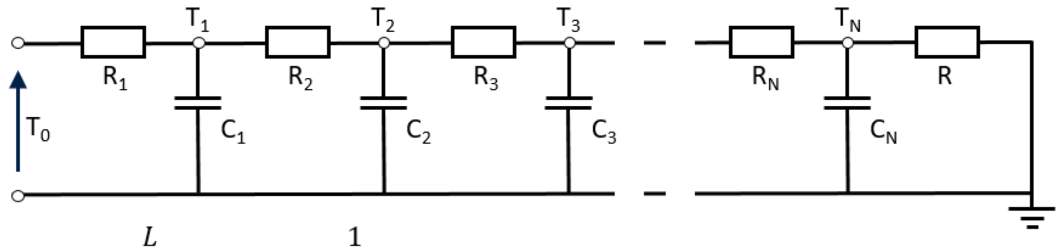
It is common in engineering applications to use a simplified Thermal Network Model (TNM) instead of a more complex full conjugate heat transfer analysis. The method uses an electrical equivalence model incorporating the thermal resistance and thermal capacitance of the materials. These models can be solved quickly using Kirchhoff electrical laws and allow fast, approximate simulation of heat flow in the domain.

Although fast to run, models of complex geometry often take many weeks to build and calibrate. Construction of a full turbine TNM, with mixed fluid and solid domains, may take several months to complete. These models require significant user involvement to define resistance, capacitance and nodal connectivity.

A novel automated TNM construction software has been developed, allowing thermal network models to be rapidly built from CAD geometry. The calculated networks have been validated against analytic solutions and used to construct a full machine model of the Mitsubishi Heavy Industries turbine. The required algorithms, structure and underlying code for automated thermal network modelling are outlined in the following chapter.

8.2 Simple Thermal Network Models

In the case of simple 1D analysis, a Cauer RC ladder can be used to simulate the flow of heat. After applying the Kirchhoff laws to sum heat flow at each node to zero, the calculation matrix can be found. The matrix is identical to the forward difference 1D scheme and gives the same coefficients as the right hand side of the 1D Crank-Nicolson model from Chapter 4 Multilayer 1D Heat Transfer Analysis.



$$R_c = \frac{L}{kA} \qquad R_v = \frac{1}{hA}$$

$$C = \rho V C_p \qquad R_a = \frac{1}{\dot{m} C_p}$$

R_c – Conduction resistance
R_v – Convection resistance
R_a – Advection resistance
C – Thermal capacitance

T – Temperature [K]
ρ – Density [kg/m^3]
k – Thermal conductivity [W/m K]
C_p – Specific heat capacity [J/kg K]
ṁ – Mass flow rate [kg/s]
L – Length [m]
A – Area [m^2]
V – Volume [m^3]

Figure 8.1: 1D electrical equivalent Cauer RC ladder, outlining resistance and capacitance values for different heat transfer mechanisms.

Murray [90] demonstrated the effectiveness of a simplified 2D model in the analysis of effusion cooling systems for HP turbine blades. In the case of 3D analysis, the matrix build is slightly more complex. The capacitance, volume and resistance length must be extracted from the geometry. Traditional TNM are built manually and the user must infer these geometric values from drawings, Milan and Gebremedhin [91] presented a concept to generate a thermal network model from a three-dimensional tetrahedral mesh. Using the mesh geometric features and connectivity, the network can be modelled quickly without extensive user involvement.

This concept has been adapted to focus on a cell-centred approach, applying a thermal resistance to each face and a single capacitance to the tetrahedral volume. The new approach results in two series resistances that connect the centroids of adjacent cells. In order to accommodate multi-zone meshes, with mixed material types, these two resistances remain separate. Designation of left and right face resistance follows the ANSYS Fluent mesh file structure, defining left as the side to which the positive face normal points.

Both modes of direct connectivity heat transfer, conduction and convection, are initially permitted to happen in parallel on every face. A calculation mask is subsequently applied to each heat transfer mode, either permitting or blocking each circuit path. This approach allows for consistent vector and matrix sizes across the full domain, whilst allowing each face to be individually controlled for the heat transfer mechanism. Radiation is included but is handled separately and discussed further in Section 8.3.3.

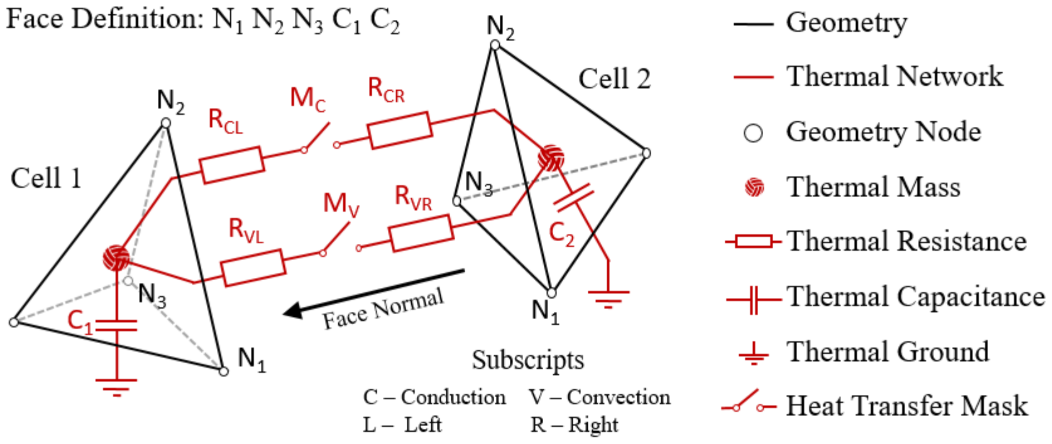


Figure 8.2: Thermal resistance and heat transfer paths across a common face (N_1, N_2, N_3) between two connected tetrahedral cells.

8.3 Thermal Network Modeller Software

The Thermal Network Modeller Software (TNMS) codes are split into four sections: Mesh, Pre, Solve and Post. This structure is conceptually similar to the ANSYS Workbench data flow.

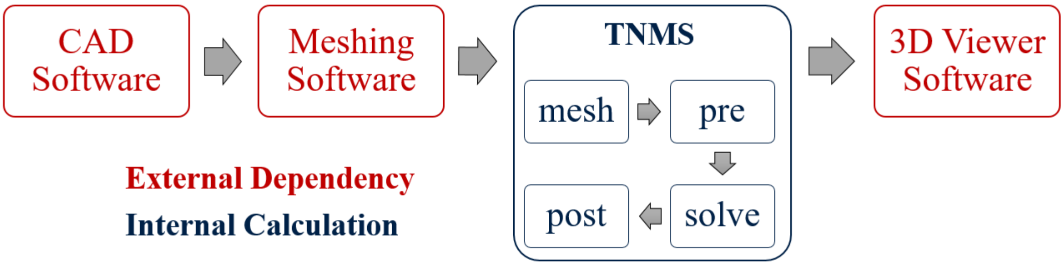


Figure 8.3: The workflow and software requirements when using TNMS, showing the internal structure and external dependencies.

8.3.1 TNMS Overview

Mesh handles all the geometric calculations, element connectivity and boundary condition settings. It reads a Fluent .msh file and outputs a Python pickled TNMS mesh file. Full mesh connectivity is retained and the process supports multi-zone and multi-material meshes, with a range of common boundary types required for heat transfer analysis.

Pre calculates all thermal resistances and capacitances for the different heat transfer mechanisms within the domain. These values, along with the connectivity, are used to construct the static solver matrices. In order to accelerate the calculation, the user has the option to introduce blocks into the mesh. If requested, the user specifies the zone and centroids of the blocks and these regions of the mesh are merged. Pre automates this lump mass process and calculates the associated thermal properties and connectivity of the blocked structure.

Solve runs the simulation matrices for the defined boundary inputs and time step. Implemented with TensorFlow sparse matrices for GPU acceleration, it is directly compatible with neural network structures. Future functionality will allow these features to be merged. Embedded neural networks could be used for boundary conditions, anchor node calculations or parameter optimisation. The user has the option to run the solver either with or without radiation.

Post evaluates the solver output and allows calculation of face temperatures, heat flow and heat flux not required in the solver codes. This helps accelerate Solve by removing redundant calculations. Post additionally allows blocked geometry to be reverted to mesh geometry, spatially interpolating the data using a localised Kriging method for inverse blocking.

To support the code development and help visualise the mesh, blocking and results; a custom 3D viewer has been written in Python OpenGL. Designed specifically for this application, the viewer offers a unique code debugging option, not available in a commercial package. The user interface is limited but could be extended if additional functionality is required. The long-term need will primarily be data visualisation, in which case an export to a more established software such as ParaView may be preferred.

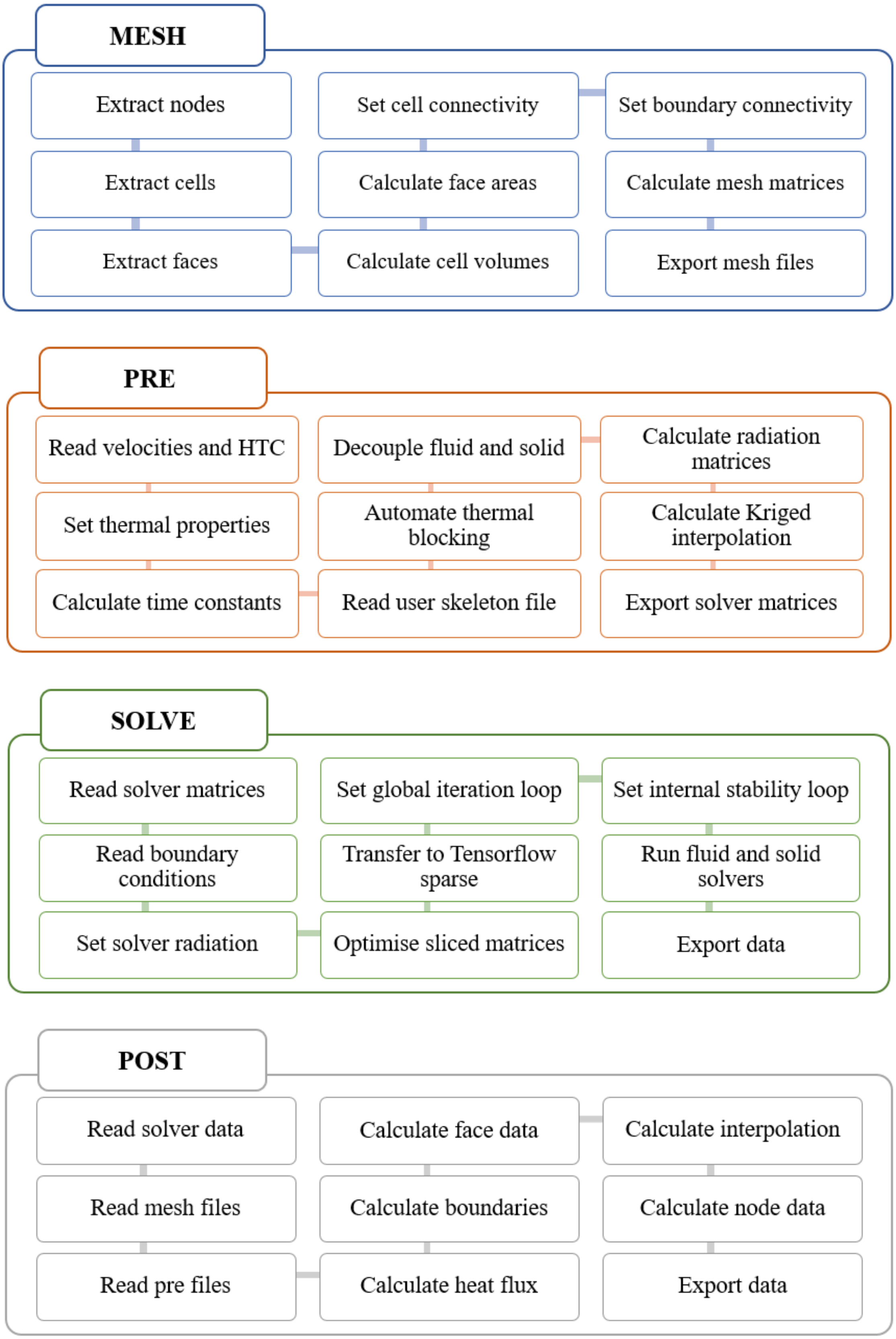


Figure 8.4: Work flow of TNMS, showing the internal calculation procedure.

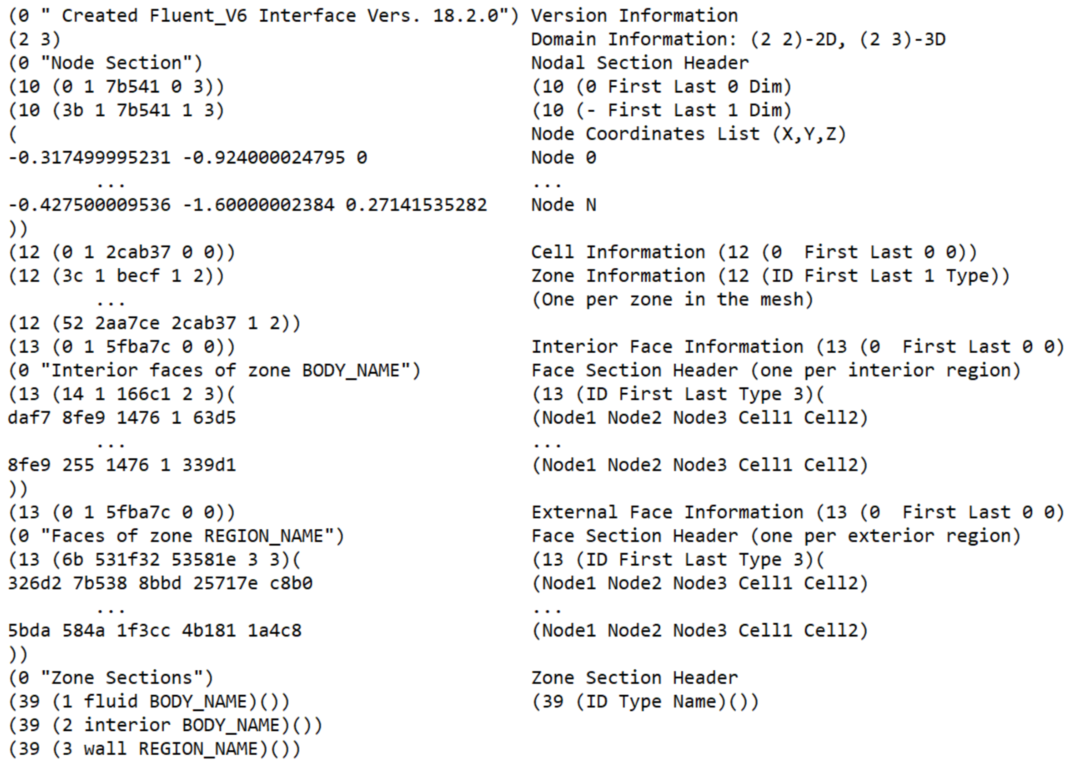
8.3.2 TNMS Mesh

The current implementation is limited to ANSYS Fluent multi-zone unstructured tetrahedral meshes only. This format is widely supported by many different meshing packages and is one of the more robust and easy to build formats when handling complex geometry. Future work may look to expand the range of supported mesh formats. TNMS analysis is by nature intended to be a low order approximate method and therefore less emphasis is placed on having a heavily refined mesh.

The code to read and extract the mesh file data is based on the Python library *meshio*, developed by Nico Schlömer at MIT [92]. This library enables the translation from one mesh type to another and will likely support future upgrades to handle a wider selection of mesh formats. The current version of this library is not designed to handle multi-zone meshes and it has been modified for this application to allow conjugate heat transfer analysis.

Included in the upgrade is the additional functionality to infer boundary conditions from the names and ID of the mesh zones. This allows the user to define fixed or variable: temperature, heat flux and convection boundary conditions, simply by specifying this in the name of the mesh zone. This name can either be set in the meshing software directly using ICEM, or manually defined in a text editor after exporting the .msh file.

The ANSYS Fluent mesh format is shown in Figure 8.5. When defining each face, it also provides additional information about the orientation and connectivity across that face. The order of the nodes in the triangular face gives the face normal, following the right hand vector cross product rule. The order of the cells listed for each face gives the connectivity and direction of the face connecting bond. The normal direction of each face is important, especially for those on a shared boundary, as this controls the visibility of the face during the render and 3D visualisation.



```
(0 " Created Fluent_V6 Interface Vers. 18.2.0") Version Information
(2 3)                                         Domain Information: (2 2)-2D, (2 3)-3D
(0 "Node Section")                            Nodal Section Header
(10 (0 1 7b541 0 3))                          (10 (0 First Last 0 Dim)
(10 (3b 1 7b541 1 3)                          (10 (- First Last 1 Dim)
(                                             Node Coordinates List (X,Y,Z)
-0.317499995231 -0.924000024795 0             Node 0
        ...                                   ...
-0.427500009536 -1.60000002384 0.27141535282  Node N
))
(12 (0 1 2cab37 0 0))                         Cell Information (12 (0  First Last 0 0))
(12 (3c 1 becf 1 2))                          Zone Information (12 (ID First Last 1 Type))
        ...                                   (One per zone in the mesh)
(12 (52 2aa7ce 2cab37 1 2))
(13 (0 1 5fba7c 0 0))                         Interior Face Information (13 (0  First Last 0 0)
(0 "Interior faces of zone BODY_NAME")        Face Section Header (one per interior region)
(13 (14 1 166c1 2 3)(                         (13 (ID First Last Type 3)(
daf7 8fe9 1476 1 63d5                         (Node1 Node2 Node3 Cell1 Cell2)
        ...                                   ...
8fe9 255 1476 1 339d1                         (Node1 Node2 Node3 Cell1 Cell2)
))
(13 (0 1 5fba7c 0 0))                         External Face Information (13 (0  First Last 0 0)
(0 "Faces of zone REGION_NAME")               Face Section Header (one per exterior region)
(13 (6b 531f32 53581e 3 3)(                   (13 (ID First Last Type 3)(
326d2 7b538 8bbd 25717e c8b0                  (Node1 Node2 Node3 Cell1 Cell2)
        ...                                   ...
5bda 584a 1f3cc 4b181 1a4c8                   (Node1 Node2 Node3 Cell1 Cell2)
))
(0 "Zone Sections")                           Zone Section Header
(39 (1 fluid BODY_NAME)())                    (39 (ID Type Name)())
(39 (2 interior BODY_NAME)())
(39 (3 wall REGION_NAME)())
```

Figure 8.5: Example structure of the *Fluent_V6* mesh file, with annotations describing the hexadecimal content of each line.

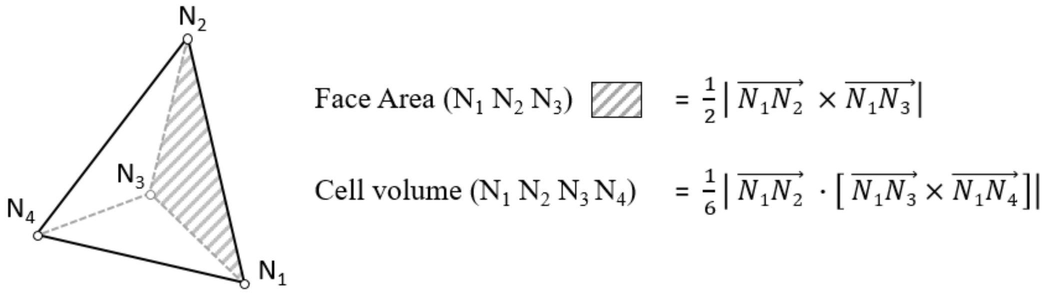


Figure 8.6: Face area and cell volume calculation for a tetrahedral mesh element.

After reading the mesh, the face areas and cell volumes for all tetrahedra are calculated. The connectivity is extracted from the face data and used to define the cell connections, connection type and permitted modes of heat transfer. Boundary conditions are read from the zone information at the foot of the mesh file. Alongside the standard Fluent boundary ID, a unique numeric ID is assigned to track each condition. This allows simplified handling when using a large number of common boundary types.

8.3.3 TNMS Pre

Over-conductivity and Ensemble Models

Where possible, all matrix and coefficient calculations are moved from Solve into Pre. This allows the complex set-up tasks to be run once and reduce memory and calculation time in the solver. Material properties, flow parameters and geometric features are assumed constant throughout the simulation. In cases where the material properties are heavily temperature dependent, this behaviour could be handled using an over-conductivity solution, factoring the temperature difference to achieve equivalent heat flow. This method is not implemented, but would allow the main matrices to remain static whilst compensating for variable properties.

$$Q = \frac{kA}{L}\Delta T \;= \frac{A}{L}(kT_1 - kT_2) \;= \frac{A}{L}(f_1 k_0 T_1 - f_2 k_0 T_2) \;= \frac{k_0 A}{L}(f_1 T_1 - f_2 T_2)$$

Alternatively, an ensemble of multiple static models may be built and the weighted average result calculated. The final blocked TNM model is not memory intensive, so this is the preferred approach when handling large transients in flow regime or material parameters.

Lump Mass Blocking

In order to accelerate the simulation, optional blocking is made available and regions of the mesh are merged together to form a lumped mass model. The thermal properties of the lump mass are calculated from the collected mesh cells. The connectivity is transferred from the high density mesh to the low order model. This process combines rows of the solver matrices, thereby reducing the number of calculations required at runtime.

The blocking concept is most easily demonstrated in 1D on a Cauer ladder, Figure 8.7. This linear form is easily converted to a lump mass by switching the positions of the resistors and capacitors. This results in series resistors and parallel capacitors. Both electrical configurations can be simplified by the sum of their components.

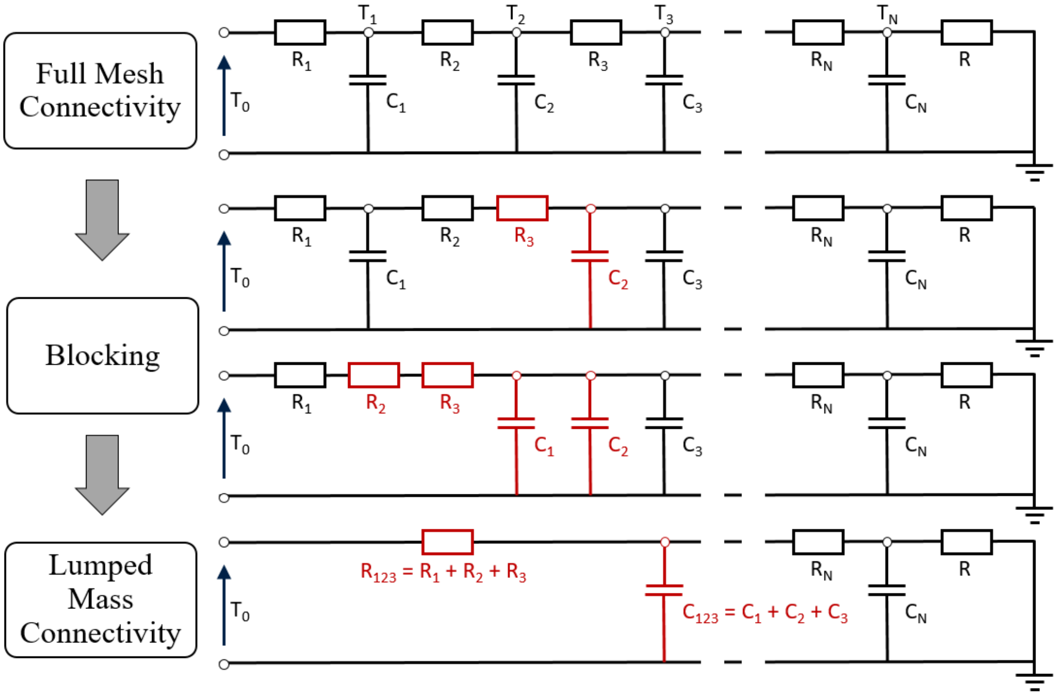


Figure 8.7: Lump mass blocking example on a 1D Cauer ladder, showing the process of exchanging the positions of the thermal resistances and capacitances, highlighting the changes on each row in red.

The blocking process is more complex in 3D but can be automated using a growth algorithm similar to Dijkstra minimum path [93]. Blocking is calculated on a zone by zone basis and is achieved by switching the positions of the resistor and capacitor components. Fluid blocks behave slightly differently in that only mass flow across the boundary is considered. The user can choose to have a mixed strategy within the model, blocking some zones and retaining the mesh in others. The user defines a skeleton file which contains the target centroid for each block they wish to create. The algorithm grows outwards from these targets, forming new interfaces, lumped regions and connections.

As the lumped mass grows, the next adjoining cell may have multiple faces attached to the mass. In this case the parallel resistance is converted to a series equivalent before advancing the lumped mass front and the resistance connection. For robustness the blocking process is split into four parts.

1. Spatial Hash Blocking

   The first step applies an approximate split to the domain using the hash table method on the rounded coordinates. This process efficiently groups cells surrounding the targets but can generate duplicate assignments. These duplicates are removed, retaining only the closest target point. This initial step is very fast but is not sufficiently robust; it can lead to sections of the mesh becoming disconnected in undercut or local U-shaped geometry.

2. Connectivity Check

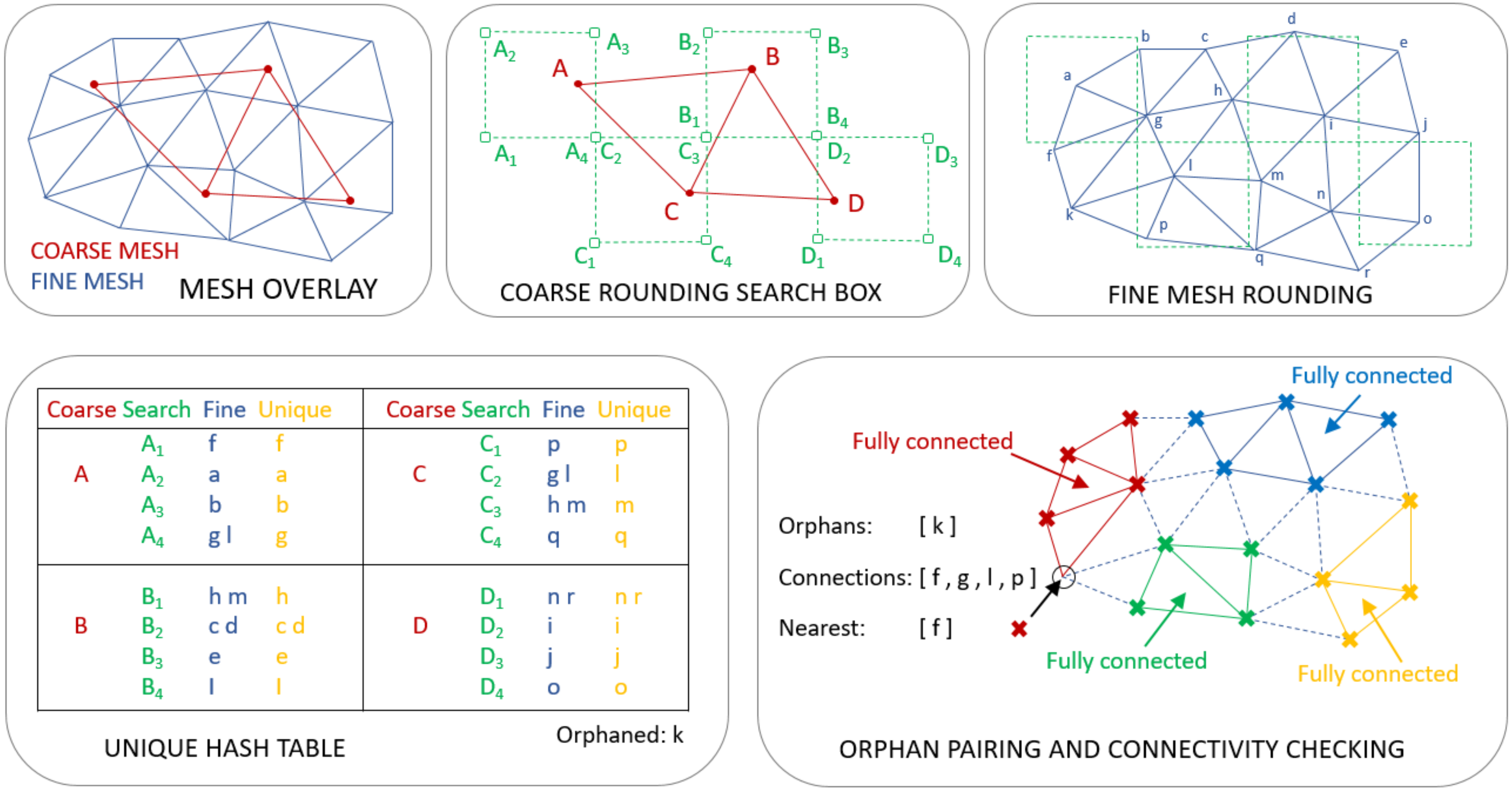
   The second process applies a Dijkstra-like growth algorithm to ensure connectivity. Starting from the block centroid, immediate connected cells are then joined to the lump mass, followed by their respective connections until no further cells in the hash can be attached. Any cell that remains unattached is flagged as disconnected and designated an orphan cell. Applying this step after the initial hashing restricts the growth algorithm to local regions only, improving performance and allowing parallel calculation.

3. Orphan Pairing

   Step three runs the orphan collection code, finding the most suitable block for each of the disconnected cells, ensuring continuous connection back to the lump mass centroid. Following this step, all of the cells have been allocated to the closest block requested by the user input file.

4. Connectivity Calculation

   The final process repeats the growth algorithm from step two. With orphans removed, all cells can be collected and the algorithm should run to completion. During this second pass the volume, thermal capacitance, thermal resistance and connectivity are iteratively transferred to the lump mass. Similar to the Cauer ladder example, the resistor and capacitor positions are exchanged then amalgamated to advance the lumped mass front.



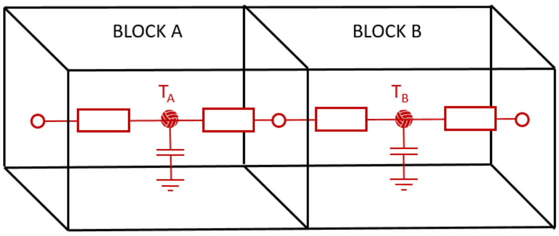
Coarse	Search	Fine	Unique	Coarse	Search	Fine	Unique
A	A_1	f	f	C	C_1	p	p
	A_2	a	a		C_2	g l	l
	A_3	b	b		C_3	h m	m
	A_4	g l	g		C_4	q	q
B	B_1	h m	h	D	D_1	n r	n r
	B_2	c d	c d		D_2	i	i
	B_3	e	e		D_3	j	j
	B_4	l	l		D_4	o	o

Figure 8.8: Overview of the hash table blocking method, showing an example for two basic meshes with one orphan pairing on node k.

The optional blocking can form three types of interface. Where possible these interfaces are simplified, reducing connectivity within the domain and maximising the sparsity of the final solver matrices. The three types are shown below.

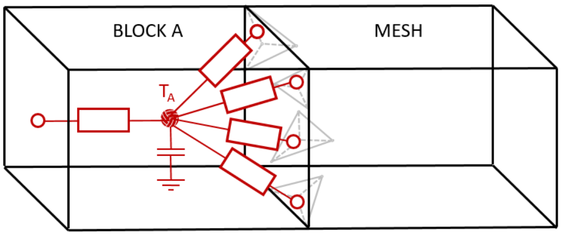
1. Block to block

   In this case the thermal resistance between the two blocks can be further simplified; this is done using a parallel resistor calculation for each of the paths between the two block centroids.



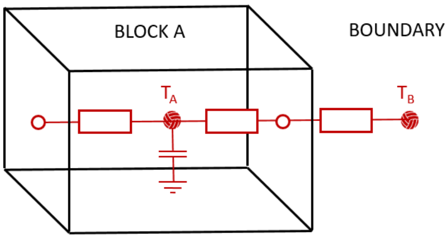
2. Block to mesh

   The lumped mass connects to many individual cells in the adjoining zone. All resistances are kept and only the mesh connectivity updated to reference the new lump mass.



3. Block to boundary

   All resistances connected to the boundary are replaced by a single parallel equivalent between the lumped mass and the boundary.



Boundary Conditions

Boundary conditions are handled by the addition of *arithmetic cells*. These are non-physical cells that are added to the mesh and carry either the temperature or heat flux specified by the user. The face connectivity of the domain is modified to pair the boundary elements with their associated arithmetic cell.

Supported boundary types are

- Fluid inlet (constant mass flow, transient temperature)
- Fluid outlet (constant mass flow, transient temperature)
- Temperature (constant or transient)
- Heat flux (constant or transient)
- Convection (constant transfer coefficient, constant or transient temperature)

Radiation Models

In order to retain the calculation speed, radiation is only considered across an internal fluid zone where heat transfer by this mechanism cannot reasonably be neglected. The radiation model can only be activated when the internal fluid region is a lumped mass zone with blocking.

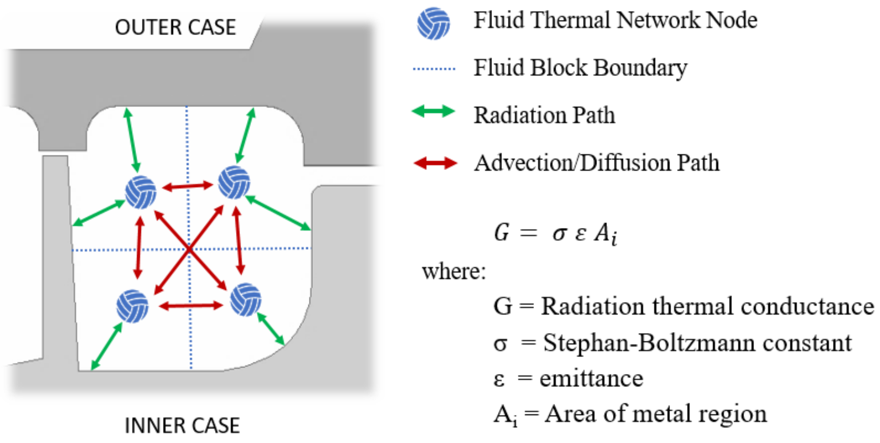
When internal fluid blocks are defined, patches are created that contain all mesh faces where one fluid block attaches to a metal wall. During blocking, the algorithm automatically returns these surface patches. Radiation is only calculated between these metal patches and from each patch to the immediately adjoining fluid block. This selection criteria greatly reduces the number of radiation heat paths, but retains the critical connections across the cavity and fluid wall.

The metal patches are further restricted. They are limited to find radiation paths only when they share a common fluid region and sit on a different body. This filter automatically prevents radiation through the machine between two visibly obstructed regions. It also prevents radiation within the same body, where temperature differences are less significant.

Two modes of radiation are considered: Surface to Fluid (S2F) between a solid patch and its paired fluid block and Surface to Surface (S2S) between two solid patches. Simple grey body calculations are used throughout, absorption and scattering as radiation passes through the cavity is not considered. To account for the fast changing fluid temperature over the solid time step, the time averaged temperatures from the previous step is used in the calculation.

Surface to Fluid Radiation (S2F)

MHI advised that a significant portion of the heat transfer from the cavity steam to the metal casing is in the form of radiation and cannot be ignored. A radiation model is therefore required to handle this mode of heat transfer. The S2F model uses a simple grey body calculation between the adjoining fluid block and metal patch. Other than the local interaction between the fluid block and the wall, no other fluid radiation is considered. In reality heat is absorbed, scattered and emitted by every part of the fluid cavity. By ignoring this effect the mesh connectivity is significantly simplified and the number of calculations required at runtime greatly reduced. This assumption means the basic S2F model has a notably localised wall effect compared to the true behaviour. This is partially compensated by advection and diffusion in the cavity, especially considering the faster fluid time stepping.



$$G = \sigma \, \varepsilon \, A_i$$

where:

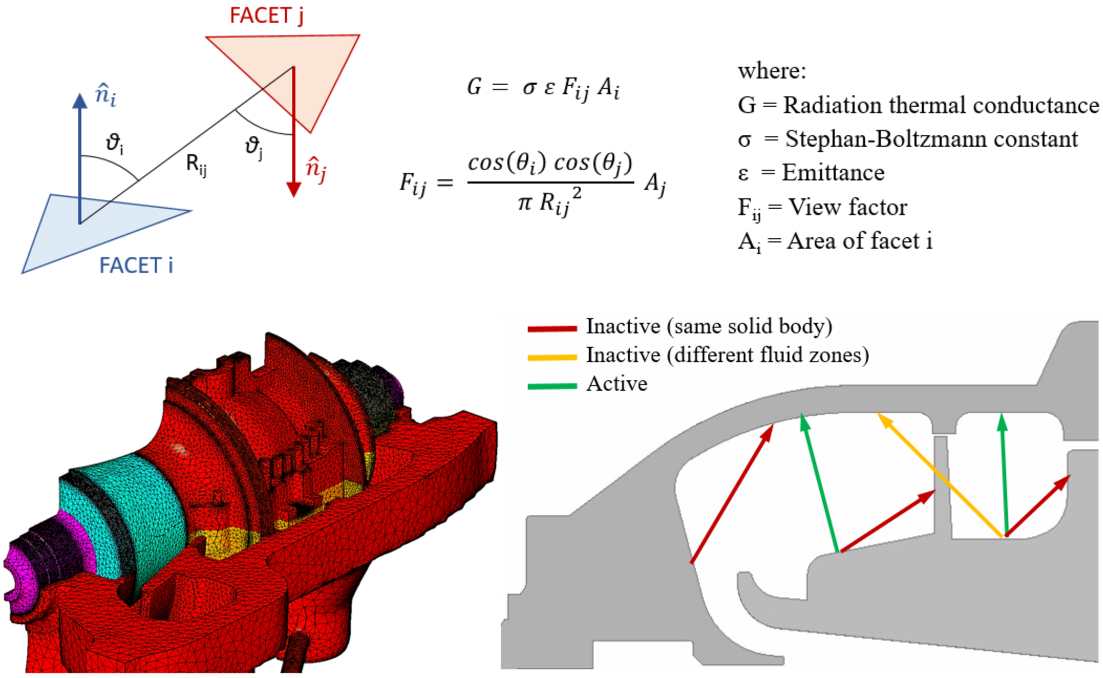
G = Radiation thermal conductance
σ = Stephan-Boltzmann constant
ε = emittance
A_i = Area of metal region

Figure 8.9: Details of the S2F radiation model showing the active radiation paths between adjoining fluid blocks and metal zones.

Surface to Surface Radiation (S2S)

The S2S method uses an area averaged view factor between all triangular facets in each metal patch. The view factor is calculated once during TNMS Pre and stored in the solver matrix. Many thermal network model solvers opt to linearise the radiation term by first pre-computing a temperature dependant radiation resistance [94]. Given the intent of this solver is to run a low order approximation with a low nodal count, it is anticipated that the computational cost to calculate T^4 is acceptable and therefore the radiation term can be calculated directly.

Full transmission is assumed and radiation passes through the cavity without absorption or scattering. Ray tracing and obstruction detection are not included. However, these tools are not required if the user selectively cuts and names the fluid regions. The geometric shape factor automatically accounts for backward facing or non-visible surfaces, setting these view factors to zero. Patches on the same body do not interact and radiation is only considered from one zone to another.



$$G = \sigma \, \varepsilon \, F_{ij} \, A_i$$

$$F_{ij} = \frac{cos(\theta_i)\, cos(\theta_j)}{\pi \, R_{ij}{}^2} \, A_j$$

where:
G = Radiation thermal conductance
σ = Stephan-Boltzmann constant
ε = Emittance
F_{ij} = View factor
A_i = Area of facet i

Figure 8.10: Details of the S2S radiation model showing the view factor definition, active and inactive radiation paths.

Time Constant

The time constant of each zone is calculated using an RC circuit equivalent method in each cell. The capacitance is taken at the first percentile of the minimum value in each domain, to remove over-dependence on very small elements. The resistance is taken as the full parallel resistance into a cell. The calculated time constant is set to an integer multiple or factor of 1 second to allow accurate control over the simulation duration.

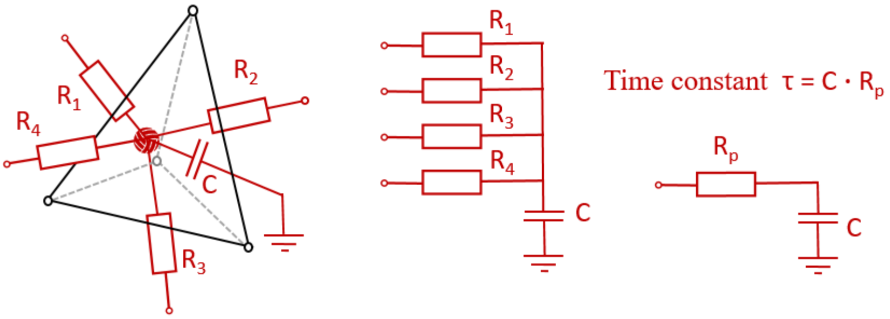


Figure 8.11: Cell time constant calculation using an equivalent parallel resistance RC circuit analogy.

To support time constant matching between the fluid and solid; the fluid domain time constant is set to a log_2 multiple of the solid domain. This allows repeated self multiplication of the fluid matrix to construct an equivalent solid time step matrix. This method propagates the sparse elements in the fluid matrix until it saturates and becomes fully populated. Each multiplication beyond saturation offers a time saving in the fluid calculation compared to iterative use at runtime. Note however, this method is highly memory intensive and must only be applied in the case of low order models. It is only suitable here due to the blocking methods, assumptions of constant fluid properties and constant temperature boundary conditions in the decoupled fluid domain. An example calculation for a 2^4 time constant factor is shown below to build the equivalent fluid matrix, M_{16}. In practice, factors of 2^{10} to 2^{15} can be expected.

$$\tau_{Solid} = 2^4 \times \tau_{Fluid}$$

$$\begin{aligned} M_2 &= M_F \times M_F \\ M_4 &= M_2 \times M_2 \\ M_8 &= M_4 \times M_4 \\ M_{16} &= M_8 \times M_8 \end{aligned} \tag{8.1}$$

Matrix Calculation

The solver matrix calculation is split into several task specific matrices which are then multiplied to give the final solver. The solid and fluid domains are decoupled and handled sequentially to allow for different time stepping in each domain. These component matrices simplify the build and are assigned letters F, G, E, C and D.

F (Face delta) - this computes the temperature difference between adjacent cells across the face. The matrix is composed of ±1 only and is fully defined by the mesh connectivity. The delta is calculated $cell_1 - cell_2$ and the matrix is sparse with shape $N_{faces} \times N_{cells}$.

G (Conductance) - this defines the thermal conductance of each face and includes conduction, convection and advection values. The matrix is assumed static and the thermal properties and flow regimes are fixed at build time. Connections removed during blocking are assigned zero conductance. It is diagonal and has shape $N_{faces} \times N_{faces}$.

E (Summation) - this computes the heat flow summation into each cell of the mesh, including those from user defined boundary nodes. It is composed of ±1 only and is fully defined by the mesh connectivity. The heat flow into each cell is computed and the sign of the elements is set by the orientation of the face delta in [F]. It has shape $N_{cells} \times N_{faces}$ and is sparse.

C (Capacitance) - this defines the thermal capacitance of each cell and includes a factor of the domain time step. Multiplying the net heat flow into each cell, it calculates the temperature change over the time step. It is also assumed static, with the material properties and time step fixed during the build. Cells removed in blocking are assigned zero capacitance. It has shape $N_{cells} \times N_{cells}$ and is diagonal.

D (Diagonal) - this is a diagonal matrix of ones, with shape $N_{cells} \times N_{cells}$, used to sum the initial conditions to the cell delta values. Ones are populated for the active cells only and merged block cells are ignored.

The separate solid and fluid matrices are formed using the multiplication below, resulting in sparse square matrix of shape $N_{cells} \times N_{cells}$. Due to the blocking strategy, many of the original mesh cells and faces are inactive. By zeroing elements in [G],[C] and [D], these are naturally ignored in the sparse matmul implementation and the matrices retain the full dense mesh sizing N_{cells}.

$$M = D + C \cdot E \cdot G \cdot F \tag{8.2}$$

The solid domain wall is assumed isothermal during fluid stepping. The heat flow through the wall is calculated during the decoupled fluid simulation and stored. This additional vector collects all heat transferred across the boundary, later applying the time and area averaged value to the solid domain. The stability of this combination of boundary conditions was analysed by Giles [19] and recommended as a viable solution to the loosely coupled problem. Combined with the additional radiation vector for active faces, T^4, this gives the full definition of the solver matrices in Equations 8.3 and 8.4.

$$\text{Fluid domain} \quad \begin{pmatrix} T_F \\ q_F \\ T_0^{*4} \end{pmatrix} = \begin{pmatrix} M_F & 0 & S_F \\ Q_F & 1 & R_F \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} T_0 \\ 0 \\ T_0^{*4} \end{pmatrix} \tag{8.3}$$

$$\text{Solid domain} \quad \begin{pmatrix} T_1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} M_S & Q_S & S_S \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} T_F \\ q_F \\ T_0^{*4} \end{pmatrix} \tag{8.4}$$

where,

T_i – Temperature at step i
T_i^* – Time average temperature from the previous step
q_F – Time average heat flow aross the decoupled boundary
M_F – Fluid advection/convection matrix
M_S – Solid conduction matrix
S_F – Fluid S2F radiation matrix
S_S – Solid S2S radiation matrix
R_F – Decoupled boundary radiation tracker
Q_F – Decoupled boundary convection tracker
Q_S – Decoupled boundary heat flow summation

8.3.4 TNMS Solve

Solve loads the static matrices for the fluid and solid domains and executes the simulation. The matrices are transferred to TensorFlow and allow optional GPU computation. In the case of highly simplified thermal network models, or short duration simulations, the calculation requirements are comparatively low and the overhead of data transfer to/from the GPU can dominate. All non-essential calculations, not required to advance the solution, are ignored and handled later in Post. Time stamped data can be exported if a specific simulation snapshot is required to be analysed. The user defined boundary conditions must be given as inputs to the associated arithmetic cells.

8.3.5 TNMS Post

Post loads the exported data from Solve and computes all requested analysis of the snapshot information. Heat flux, thermal gradients and internal temperatures can be extracted. A localised Kriging interpolation method is implemented to recover the spatial information inside the blocks, returning an approximate interpolated solution for the initial dense mesh.

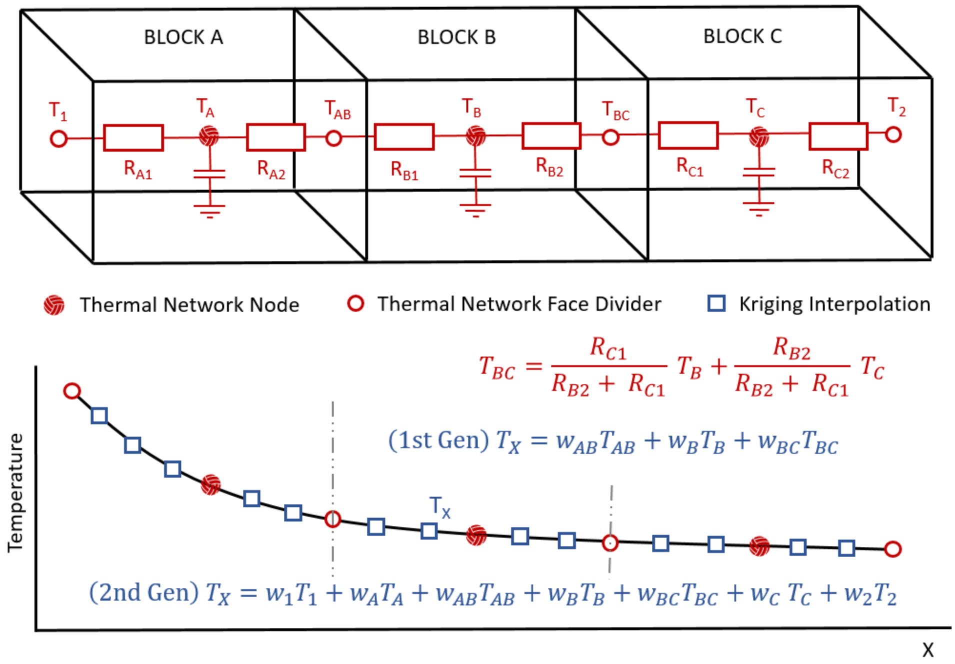


Figure 8.12: Localised Kriging interpolation showing the calculation for T_x in block B using the 1st and 2nd generation localised connectivity and Kriging weights w_x.

The block temperatures, block connectivity and thermal resistances are used in a potential divider to recover the block boundary temperatures. These boundary values are then used to interpolate the data inside each block. To speed the calculation, and prevent errors from U-shaped geometry, Kriging weights are limited to the surrounding neighbourhood only. This neighbouring region is defined either by the first or second generation of block connectivity and is set by the user. Figure 8.12 shows the effect of the connectivity limit, highlighting the difference in the calculation result. Second generation connectivity gives a wider range of data values, it better resolves the spatial gradients at the added cost of extra terms in the interpolation matrix.

8.3.6 TNMS Viewer

The 3D viewer is designed primarily as a debugging tool, allowing the developer to customise the displayed data beyond typical support in commercial packages. The interface is minimal and it is intended to be used at code level. The user can add functions to visualise anything in TNMS, allowing them to inspect issues in connectivity, blocking, cell collections or element ID. This tool is invaluable for validating and debugging the codes, particularly TNMS Pre, where mesh connectivity is altered.

The viewer combines Python pyglet with an OpenGL vertex fragment shader [88] [95]. The TNMS mesh data is first cleaned to extract only the visible nodes and faces on the boundary of each zone. The user may also define cross-sections to be viewed. Any faces identified in this section are also extracted and this minimal data set is transferred to a Vertex Buffer Object (VBO) on the graphics card.

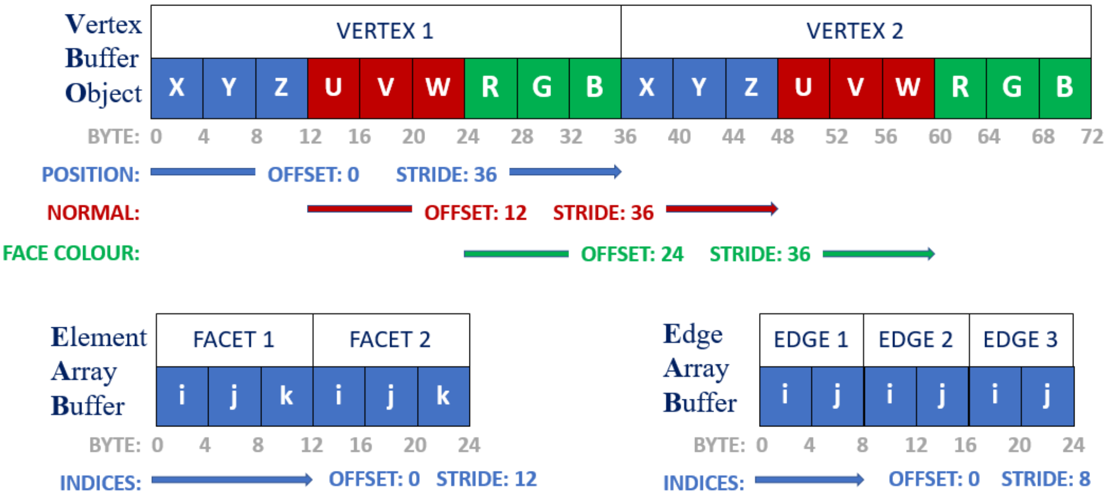


Figure 8.13: Description of the Vertex Buffer Object and Element Array Buffers used in the 3D TNMS visualisation tool.

The VBO also holds data regarding the node colour and normal direction, important for defining visibility and interaction with the shader light. A separate Element Array Buffer (EAB) is created for the faces and edges, defining the connectivity of nodes. A unique VBO and EAB is generated for each zone and cross-section, allowing these individual parts to be easily hidden or shown.

The VBO and EAB are created once and saved on the first use of the viewer only. These are reloaded for repeat visualisations of the same TNMS geometry and only the colour data changed for new results. The current implementation parses the data for each zone into a separate static sub buffer. However, the colour and data buffers could be implemented dynamically to support real-time visualisation should this feature be required in a future update. A simplified user interface is provided, allowing basic mouse and keyboard inputs to control different viewing options and data visualisation.

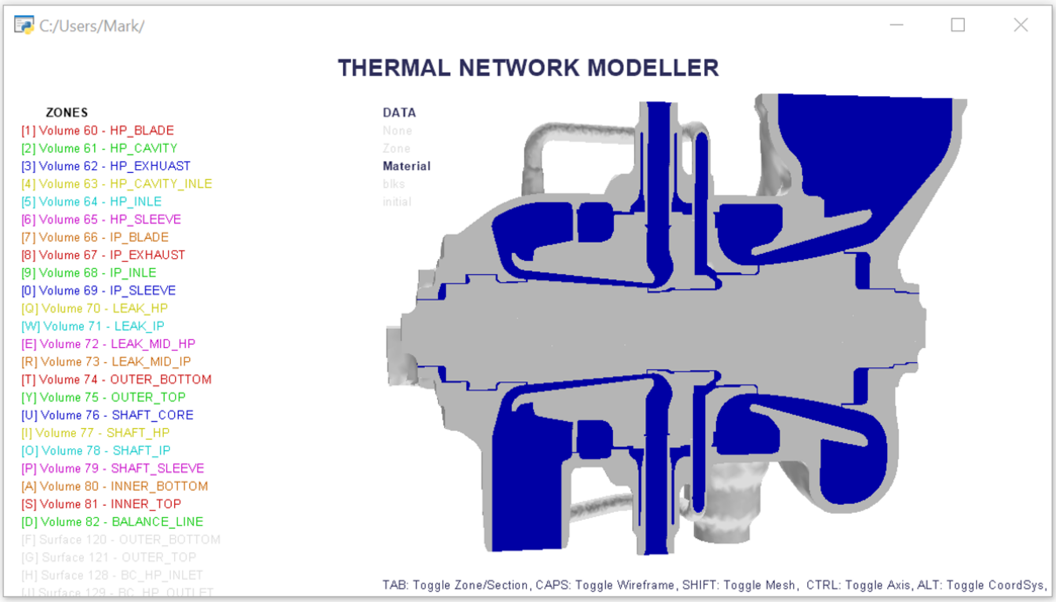


Figure 8.14: Screenshot of the TNMS viewer, showing the mesh zone information in the left hand panel and 3D view of the MHI turbine solid and fluid materials.

8.4 Summary

New software for the fast construction of complex thermal network models has been developed. Using a novel automated blocking method a multi-zone, multi-material, tetrahedral mesh can be routinely converted to a thermal network. The software automates the computation of the electrical equivalence components and outputs the static solver matrices with or without radiation.

Supporting both mesh and lump mass calculation modes, the thermal model can be tailored for accuracy or speed. When applying lump mass methods, the domain is automatically merged using a Dijkstra-like growth algorithm, computing the thermal properties and connectivity.

Several features are embedded to support the use of multi-fidelity methods. Functionality is available to: import data from high fidelity simulations, apply measurement anchoring nodes, supply boundary information from neural networks and recover data density using Kriging interpolation.

The new software is validated in the following chapter, comparing performance against analytic and known reference cases.

9

Thermal Network Model Software, Validation and Future Application

9.1 Introduction

In the following chapter the Thermal Network Model Software is validated and applications for future use and development are explored. Focus is given to the possibilities in integrating all parts of this thesis to construct a single hybrid analysis system.

The software is first validated against known test cases and confirms the model's suitability for fast, approximate thermal simulation of complex geometries. The potential applications are then explored alongside the opportunities for future development. Preliminary application studies, conducted by Masters students under my supervision at the Oxford Thermofluids Institute, are used to demonstrate the effectiveness of the software as a design and monitoring tool.

9.2 Thermal Network Validation

The Thermal Network Model Software tools were validated against three test cases.

- Quasi-1D heating on the end wall of a solid steel bar.
- Transient convection heating on a simplified model of the high pressure region of a steam turbine outer casing.
- A conjugate heat transfer study of the MHI steam turbine geometry.

9.2.1 Quasi 1D Case

A solid bar test case with uniform material properties, one end wall boundary condition and all other walls adiabatic was used. Two different boundary types were tested, constant heat flux and constant free-stream temperature convection. These were compared to the 1D analytic solutions discussed at the beginning of this thesis.

A tetrahedral bar mesh was created in ICEM and exported for use in TNMS. The solver was tested in both full mesh and blocking mode to compare the accuracy and speed of the two solutions. 225 blocks were used, 9 in each axial section with 25 sections along the bar axis. A simulated 20 minutes of heating was applied in each case, taking 74.7s to solve in mesh mode and 5.29s for the blocked solution. These tests were conducted on a HP Pavilion laptop with two i7-7500U 2.70GHz cores, 8GB RAM and a Nvidia GeForce 940MX graphics card.

The results for both the heat flux and convection boundary cases confirm the suitability of TNMS in this application. Figures 9.1 and 9.2 show the performance of the heat flux test case for the mesh and blocked calculation mode. Both suffer slightly at the boundary condition face (x=0), this is likely due to skew in the mesh and blocks, causing the heat flux to be locally misaligned.

The Kriging-like interpolation recovered the full thermal profile but introduced partial steps into the solution. Seen in Figure 9.2, a small flat is visible at the interface between each axial section. This is caused by the localised method, restricting dependence to the immediate neighbourhood only. Allowing additional Kriging references helps smooth the profile but introduces many additional terms in the interpolation matrix. This extra memory cost is likely not viable in more complex cases. The stepping error can be improved by repeating the transfer between face-centred and node-centred data. This transformation acts to locally smooth the data in the domain. These matrices are calculated in TNMS Mesh and are made available in Post for smoothing.

Future work should look to compensate the effects of mesh skew and expand the capability to support more mesh types. This mostly affects the TNMS Mesh codes and updated methods for area, volume and connectivity would be required. The Cell and Face subclasses used to store and calculate mesh data would also require updating. 3D Viewer updates would be necessary, splitting tetrahedral and hexahedral elements to unique element sub buffers to support the differing offsets and strides during rendering.

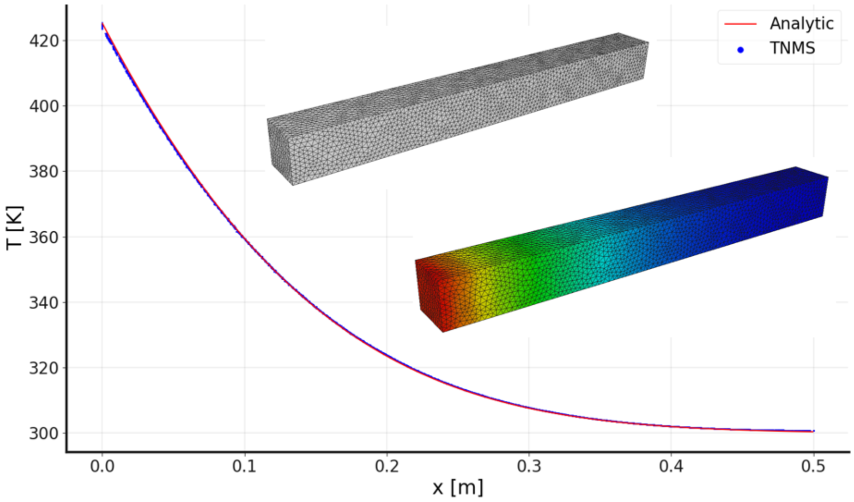


Figure 9.1: Comparison of the 1D analytic temperature profile and TNMS simulated temperature for a constant heat flux boundary, showing the mesh and final 3D solution.

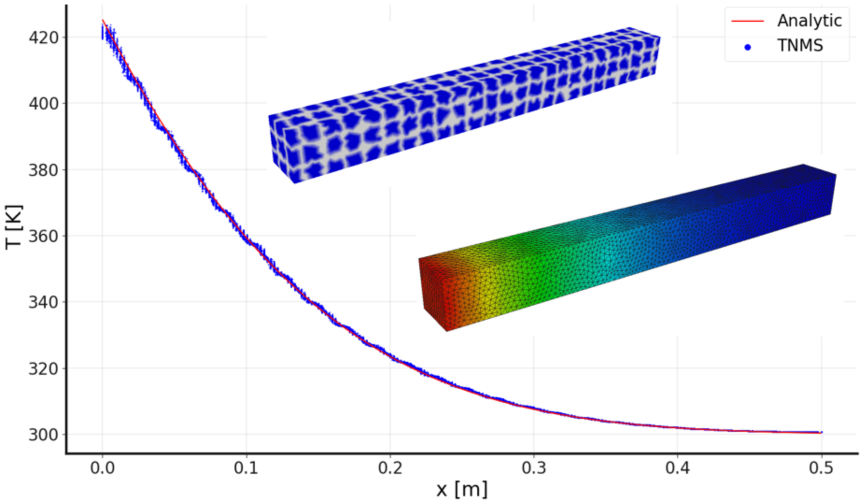


Figure 9.2: Comparison of the 1D analytic temperature profile and TNMS simulated temperature for a constant heat flux boundary, showing the domain blocking and final 3D solution.

9.2.2 Simplified HP Turbine Case

To support the work conducted by final year undergraduate students at the OTI, a simplified outer casing model of the high pressure region of a steam turbine was created. This design reduced the complexity of modelling but included all relevant features for outer case simulation: variable thickness, multiple nozzles, branch reinforcement and thick section flanges.

The simplified casing was used as a benchmark test for transient convection analysis, simulating a 1hr heating period from ambient conditions. A constant internal steam temperature ($600^oC, 873.15K$) and constant heat transfer coefficient ($100W/m^2K$) were applied to all internal surfaces. A constant ambient air temperature ($30^oC, 303.15K$) and constant heat transfer coefficient ($10W/m^2K$) were applied to all external faces. TNMS was run in full mesh mode on a medium density mesh with 156,291 tetrahedral cells. The final temperature profile was then compared to an equivalent simulation conducted with ANSYS Transient Thermal on a higher density mesh. Figure 9.3 compares the results of the two solutions.

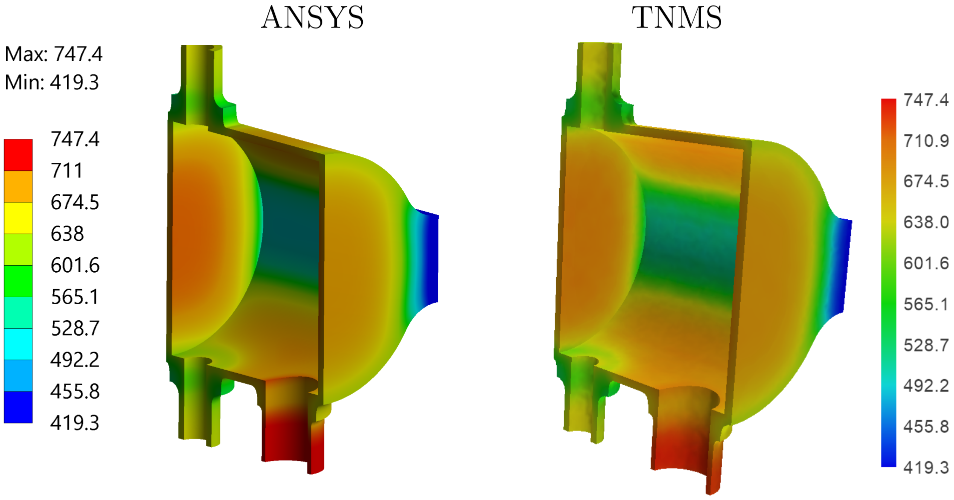


Figure 9.3: Comparison of the final Kelvin temperature distribution, after 1hr of heating with internal and external convection boundaries, solved in ANSYS Thermal and TNMS.

The TNMS maximum and minimum temperatures (743.5K, 418.4K) were within ±4K of the reference ANSYS solution. Most importantly, the thermal profile compares very well and TNMS sufficiently captures the gradient changes across the varying thickness of the casing features. The full TNMS simulation was run in 71.2 seconds on the HP Pavilion laptop. The simplified nature of TNMS makes it well suited for fast, approximate, thermal simulations run on low level hardware. The 2GB laptop graphics card was comfortably able to handle the simulation. Upgraded hardware would likely yield further performance gains, substituting a graphics card with more CUDA cores and higher compute capability.

Figure 9.4 plots the internal energy of the two simulations over the 1hr transient run. The two results correlate very well, showing the same energy development in both software. The difference between them has a near constant gradient in time, this is most likely caused by the coarse mesh definition used in TNMS, affecting the surface area and net heat flow. The final TNMS solution has a slightly lower overall internal energy, confirming the lower temperatures seen in Figure 9.3.

The TNMS energy conservation residual, equal to the difference between net heat into the domain and net change in internal energy, was evaluated at all time steps. Figure 9.5 plots the two values and the resulting difference. In 32-bit single precision calculation mode, an energy residual less than 0.75kJ was achieved, equivalent to less than 0.05% of the net heat flow per time step. The single precision solver uses float32 values in the simulation matrix for improved calculation speed. This has been used for all data and figures in the validation, demonstrating suitable accuracy in this calculation mode. If necessary, further improvement in accuracy can be achieved by switching to double precision. Operating in 64-bit mode, the TNMS energy residual is reduced by several orders of magnitude to $1.5e^{-11}kJ$. This however comes at the cost of 2.5 times the simulation duration.

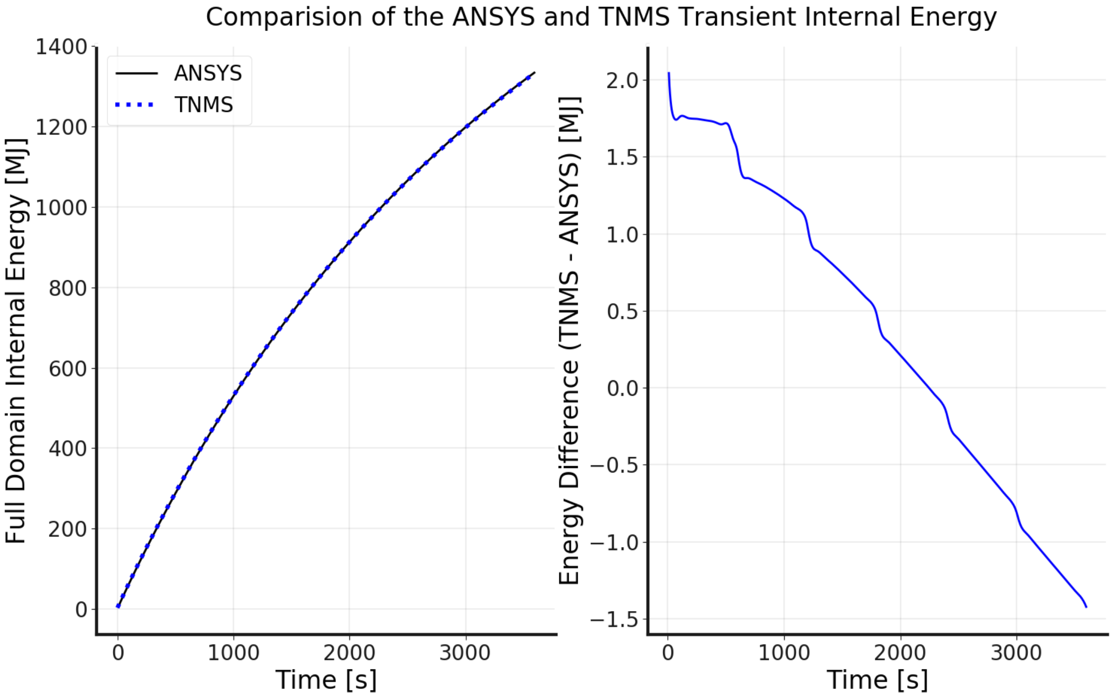


Figure 9.4: Comparison of the internal energy development over the duration of the simulation in both ANSYS and TNMS.

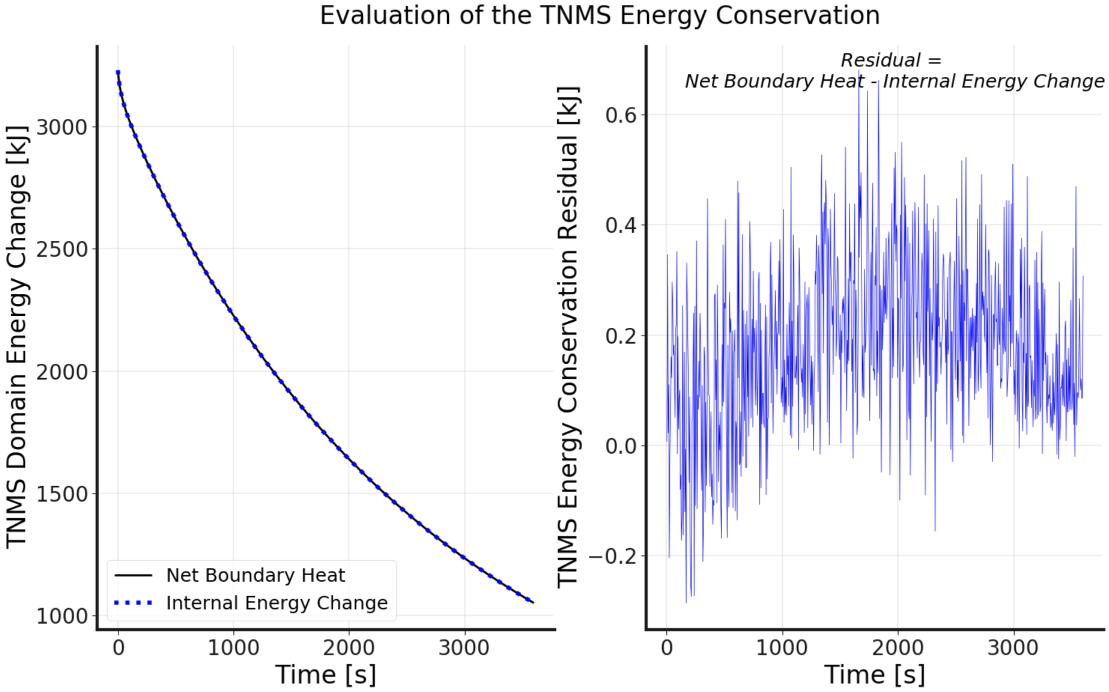


Figure 9.5: Internal energy conservation in TNMS, evaluating the net heat transfer across the domain boundary and change in internal energy.

9.2.3 MHI HP/IP Turbine Case

The full MHI HP/IP turbine comprising: rotor, inner case, outer case, main fluid path, fluid cavities and glands, was used to test the Thermal Network Model Software. The challenging geometry was robustly handled and the software successfully automated the blocking of the requested zones. The exported solver matrices, suitable for simulation with or without radiation, were stable and the software suitably automated the low order model construction of the full turbine geometry. A full model can now be built from CAD within days, compared to several weeks of work when manually constructing the thermal network.

Automated blocking was used in the rotor and fluid domains with the full mesh retained in the casing parts. Boundary conditions for inlet, exhaust and gland seal temperatures were taken from the MHI test data provided for the neural network monitoring development. Flow conditions were read from an MHI steady state simulation of the machine. The internal HTC and velocity were assigned using the TNMS user inputs which allow the user to manually define these by zone. Fluid properties were taken from superheated steam values corresponding to the MHI operating data and were independently assigned for each zone in the HP and IP regions. The casing external surface used an atmospheric convection boundary with HTC $0.5 - 4W/m^2K$ to account for cladding.

TNMS Mesh and Pre took approximately one hour to compute. The solid time constant of the mesh was 0.033 seconds, requiring 30 internal stability steps for each global 1 second time advance. 1380 blocks were introduced into the domain, which reduced the fluid time constant factor to 2^8. The fluid 2^8 multi-step matrix took 557.2 seconds to compute, saturating at 6.4 million elements. A one hour conjugate simulation of the full MHI turbine without radiation completed in 49 minutes 16 seconds on the HP laptop. Gains in performance for a real-time solution with radiation could be found by blocking the casing parts or moving to a dedicated GPU with higher compute capability and internal memory.

Temperature

Figure 9.6 compares the point-wise results in the TNMS simulation with the MHI thermocouple data. In each case, the temperature of the TNMS node closest to the recorded position of the thermocouple was extracted. The figure shows notable errors at section A, and near the IP inlet cavity at sections H and I. The simplified CAD did not include the full gland labyrinth or the secondary balance line. These geometric approximations have likely affected the calculation. Future projects should look to better resolve these features. Small offsets between a thermocouple's true position and the extracted monitor point, combined with the strong thermal gradients near sections H and I, may partially explain the larger errors in these regions.

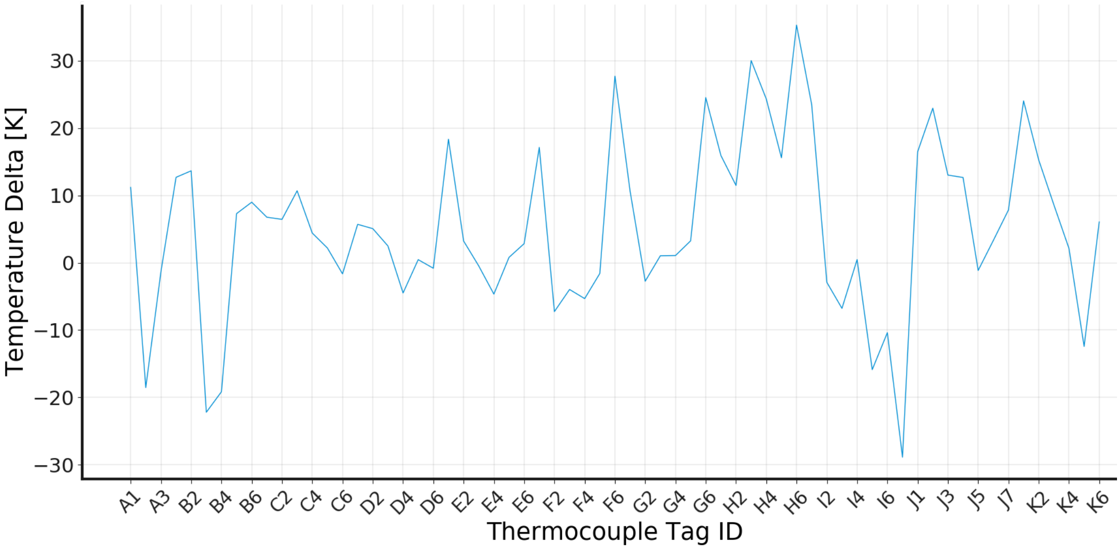


Figure 9.6: Comparison of the TNMS model predicted temperature and the measured thermocouple temperature during normal operation.

Overall the temperature prediction is reasonable, showing the full machine outer surface within ±30K. However, the accuracy is significantly lower than the LSTM and autoencoder predictions discussed previously in Sections 6.7 and 7.4.2. This highlights the potential of a hybrid methodology. The autoencoder boundary had a surface accuracy of ±7.5K. Replacing the physical HTC boundary definition with the data driven boundary condition will improve the accuracy. The boundary neural networks run an order of magnitude faster than the thermal network so there is minimal cost to the overall simulation duration. The boundary can easily be coupled by simply substituting the relevant node temperatures after each global time step.

Energy Conservation

In order to evaluate the multi-step fluid solver and decoupled solid-fluid boundary, an energy conservation analysis has been performed on the various regions of the domain. At the decoupled boundary, the time averaged heat flow to the fluid is stored during the multi-step fluid solver run. This heat flow is then applied as a boundary condition to the single step solid solver. The overall time step is the same for both solvers, but due to the multi-stepping fluid, the individual time constant of the two solvers is not the same. Therefore, to reliably evaluate the continuity at each global time step, the total heat transfer across the boundary has been tracked independently in the two solvers.

Due to the blocking in the fluid zone, a single fluid thermal network node may connect to many thousand solid domain mesh cells. In this case, the time averaged heat flow across the boundary is additionally area averaged to the solid domain. This extra step, in 32-bit single precision, is a potential source of error and supports the need to independently track the total heat transfer from both the fluid and solid side.

Figure 9.7 plots the energy conservation in the different regions of the simulation. Energy residuals are calculated for the individual fluid and solid solvers, the full combined domain and the difference between the fluid and solid side decoupled boundary trackers. The plots show the residual error in internal energy for the respective regions during each one second global time step. Data is shown for the first 900 time steps, spanning the first auto save period of the TNMS simulation. Both the solid and fluid domain show reliable energy conservation; with the residual several orders of magnitude less than the total energy of the domain. There is a slightly higher residual in the fluid domain, likely caused by the multi-stepping nature of the solver.

The decoupled boundary heat transfer compares well between the separate solid and fluid trackers on either side of the boundary. The residual is comparable in magnitude to the fluid solver which is most likely the cause of the small discrepancy. The full domain residual, comparing the net heat transfer over all external boundaries and the net change in internal energy of the entire domain, further confirms the suitability of the low order model. The following calculations were used to define the energy residuals in Figure 9.7.

$$\textit{Solid Domain Residual}, \qquad R_S = Q_S \cdot \tau_S - Q_{BS} \cdot \tau_S - \Delta U_S \tag{9.1}$$

$$\textit{Fluid Domain Residual}, \qquad R_F = \sum_{n=1}^{N_f} Q_{F_n} \cdot \tau_F + \sum_{n=1}^{N_f} Q_{BF_n} \cdot \tau_F - \Delta U_F \tag{9.2}$$

$$\textit{Full Domain Residual}, \quad R_{S+F} = Q_S \cdot \tau_S + \sum_{n=1}^{N_f} Q_{F_n} \cdot \tau_F - \Delta U_S - \Delta U_F \tag{9.3}$$

$$\textit{Decoupled Boundary Residual}, \qquad R_B = \sum_{n=1}^{N_f} Q_{BF_n} \cdot \tau_F - Q_{BS} \cdot \tau_S \tag{9.4}$$

where,

τ_S – Solid solver time constant
τ_F – Fluid solver time constant
N_f – Number of fluid solver steps per solid solver step, $\tau_\mathrm{S}/\tau_\mathrm{F}$
ΔU_S – Change in internal energy, in the solid domain, per global time step
ΔU_F – Change in internal energy, in the fluid domain, per global time step
Q_S – Net heat flow across the solid domain external boundary
Q_F – Net heat flow across the fluid domain external boundary
Q_{BS} – Area averaged heat flow on the internal decoupled boundary (solid side)
Q_{BF} – Time averaged heat flow on the internal decoupled boundary (fluid side)

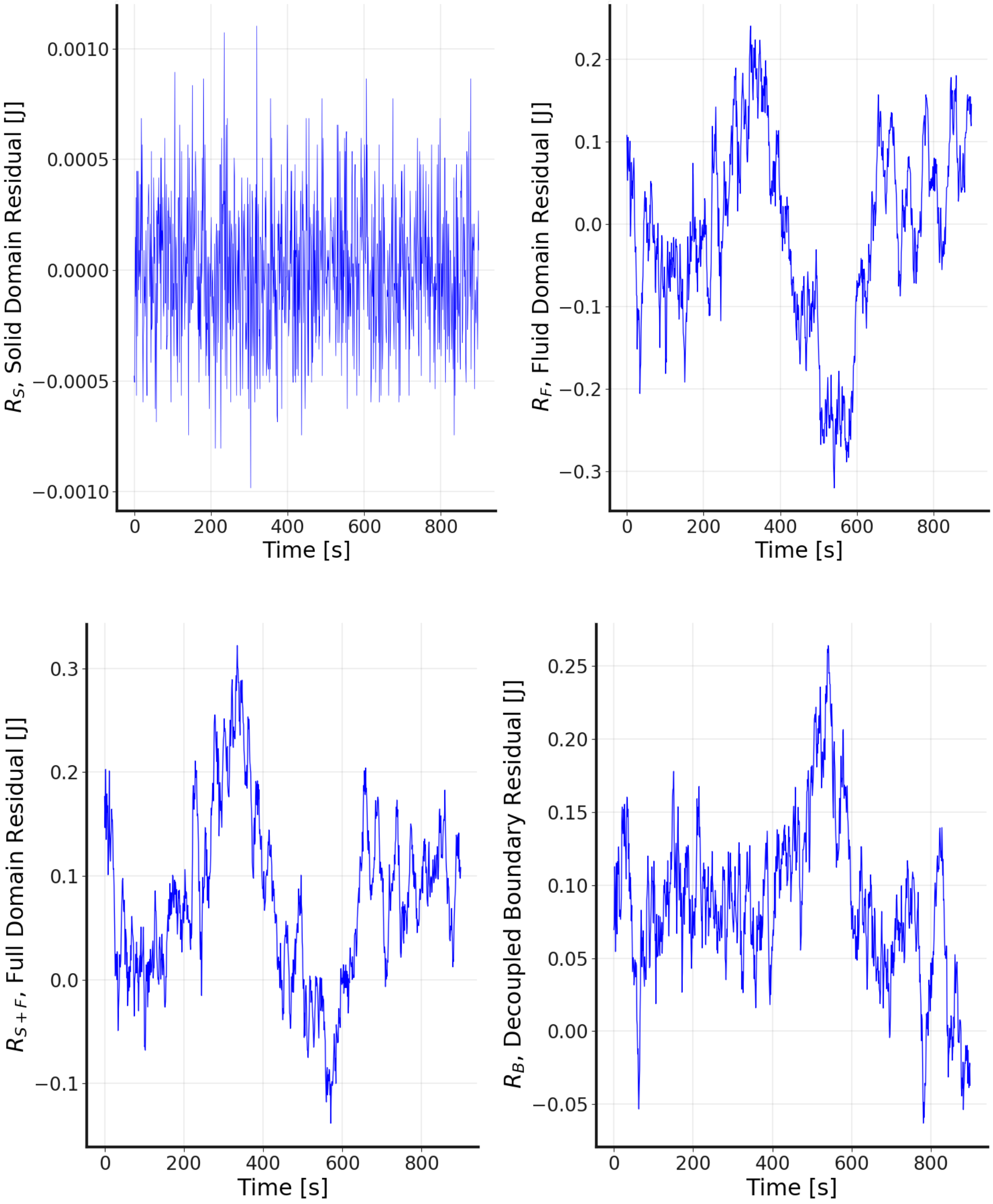


Figure 9.7: Internal energy conservation in TNMS, evaluating the net heat transfer across the sub-domain boundaries and respective changes in internal energy over the first 900 time steps.

The main limitation remains the specification of the fluid domain, primarily the fixed HTC and mass flow rate. Current methods dictate a user defined input or CFD data, calculating an approximate value from the local y*, u*, Prandtl number and Jayatilleke equation parameter [96]. This dependence on CFD data partially undermines the speed gain on offer from the low order model and a fast fluid replacement should be investigated.

The electrical equivalence permits bi-directional heat transfer. In the case of fluid nodes, this does allow heat to propagate upstream. This is most notable in the leakage paths and balance lines at the centre of the machine. Future implementation should look to add diode equivalents to control this effect. Upstream thermal propagation can however be advantageous close to flow outlets, where a known temperature is measured.

Figures 9.8 to 9.12 show the process and successful application of TNMS in modelling the full MHI turbine. The software was able to fully automate the blocking procedure and solve the associated thermal network connectivity, lumped mass and block thermal properties. Figures 9.13 to 9.15 show the final temperature result after applying Kriging interpolation in the different regions of the machine.

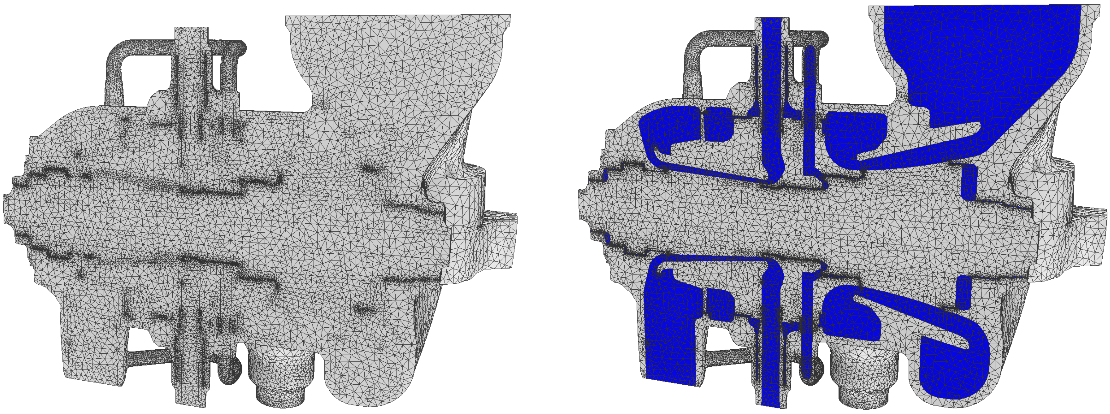


Figure 9.8: Fluent mesh of the full MHI turbine including casing cavities and gland leakage paths, highlighting the solid and fluid regions to be decoupled.

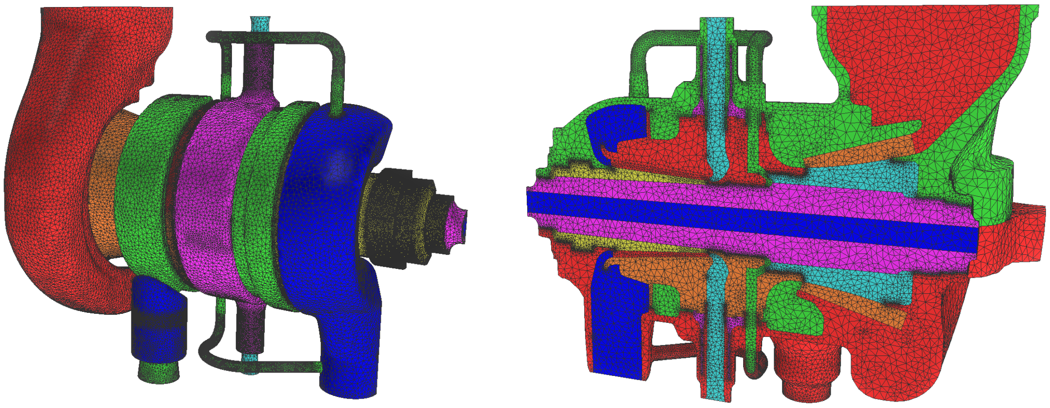


Figure 9.9: Different zones of the mesh that each have unique material and thermal properties, showing the splits in the fluid domain to control radiation connectivity.

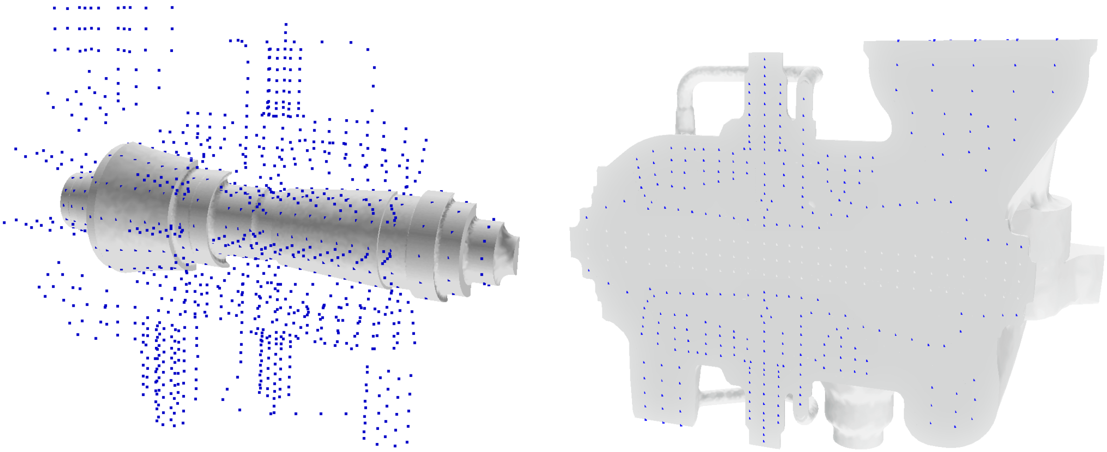


Figure 9.10: User defined blocking skeleton point data, assigned per zone, dictating the regions of the mesh to be blocked and target centroid for each lumped mass.

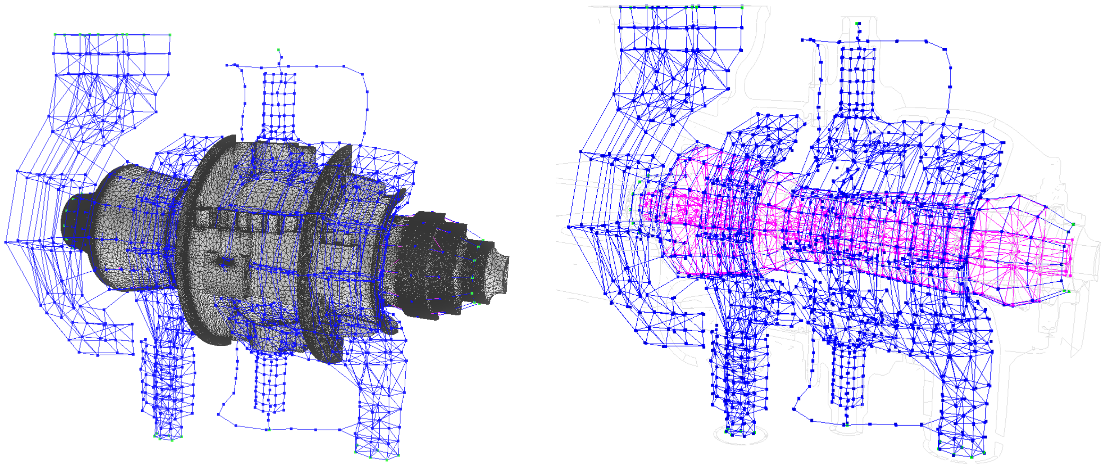


Figure 9.11: Automated thermal network blocked connectivity, solved via the Dijkstra-like growth algorithm, showing fluid regions in blue and solid regions in red.

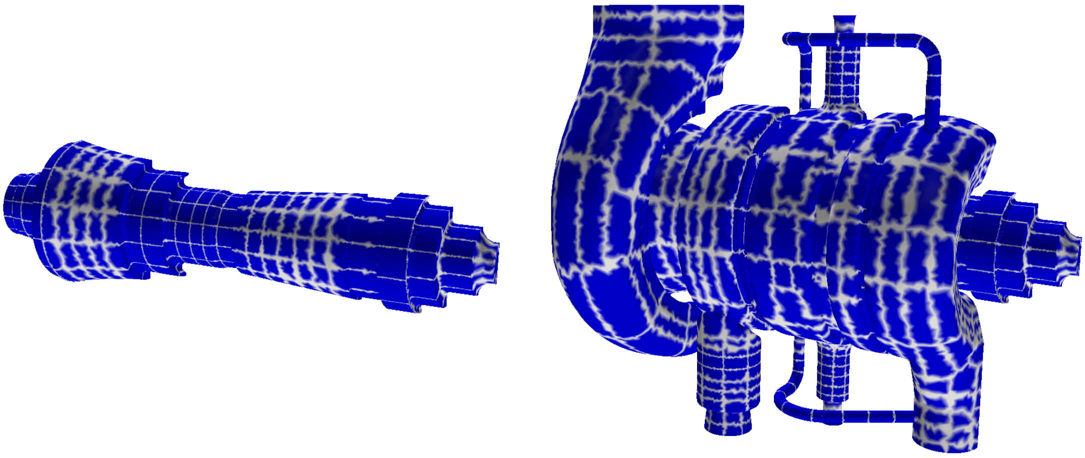


Figure 9.12: Result of the blocking algorithm, showing the thermal lumped mass regions collected in the fluid domain and shaft geometry.

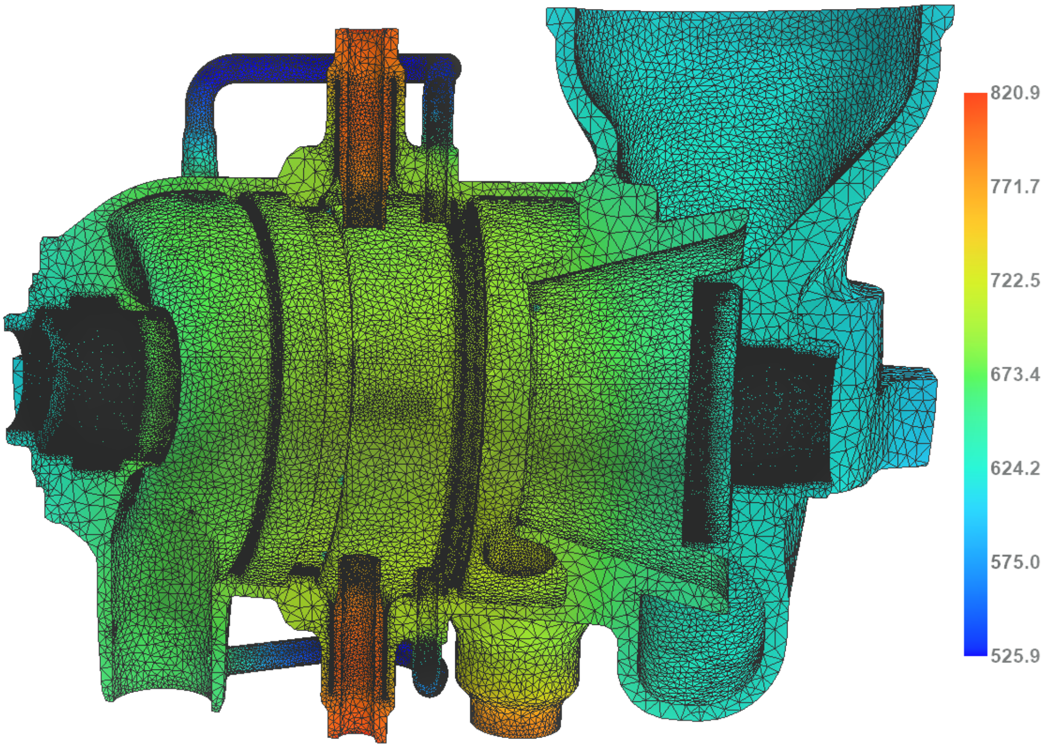


Figure 9.13: Temperature of the outer case at the selected operating point, viewed from the axisymmetric split face.

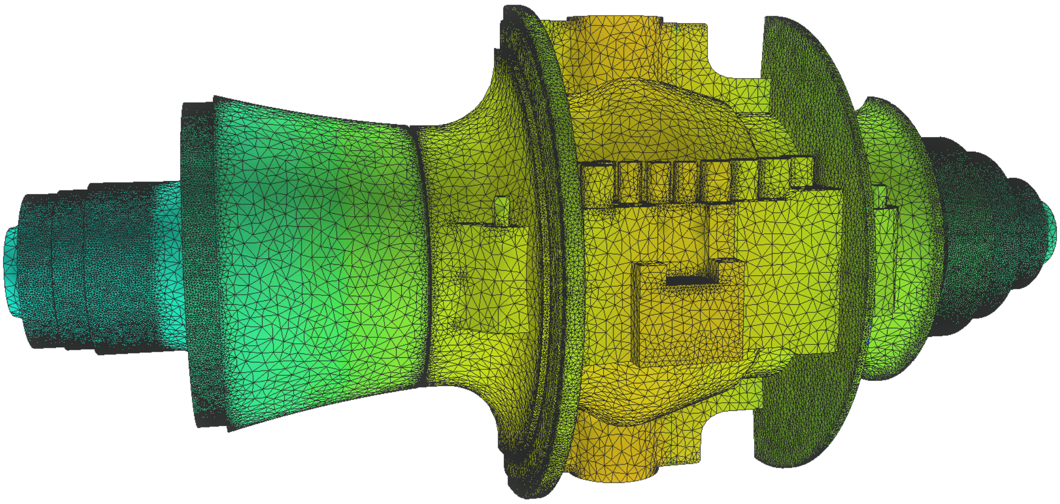


Figure 9.14: Temperature of the inner case and shaft at the selected operating point, showing the complex casing features handled by TNMS.

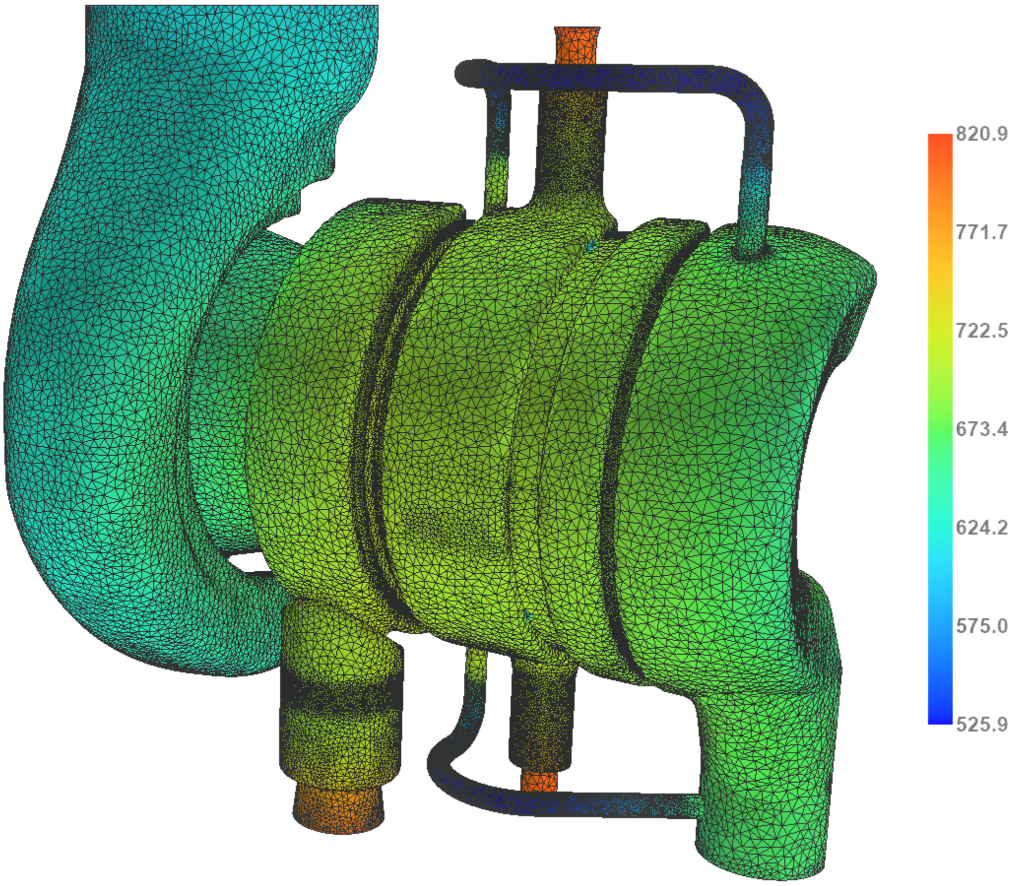


Figure 9.15: Temperature of the fluid domain at the selected operating point, including the nozzle sleeves and casing cavities.

9.3 Future Applications

Potential applications for TNMS were tested, including two projects on active and passive control by final year undergraduate students under my supervision at the OTI. These preliminary studies and other opportunities are briefly discussed below.

9.3.1 Latent Space Solver

The autoencoder and LSTM features from Chapters 6 and 7 both identified a suitable thermal latent space. This low order data stores sufficient information to approximately recreate the full external profile. If a solver can be found that time-marches successive steps of the latent space, a faster low order method may be possible. Likely opposition to such a solver are the full black-box nature; not only using a non-physical calculation but also relying on an unsupervised latent space. However, calibrated to a single machine, such a solver may offer significant speed benefits for online monitoring.

9.3.2 Reinforcement Process Optimisation

Reinforcement learning algorithms control applications are growing in popularity. Demonstrated on complex multi-variable decision tasks; these networks are able to learn optimum control mechanics by gaining experience in the environment [97]. Reinforcement agents have traditionally been demonstrated on game architectures, but engineering simulations are an equally valid application [98]. The main obstruction is the longer duration of typical engineering simulations. The low order simplified matrices of TNMS allow many approximate simulations to be run in parallel, opening the possibility of using these exploration methods.

In the case of turbine start-up optimisation, freedom could be given to specify boundary pressures and temperatures, with the aim to achieve design operating conditions. Rewards could be assigned based on lifetime-consumption and cost attributed to transition time.

A basic Monte-Carlo reinforcement learning method was written and tested on a 2D cart-pole control task using a modification of OpenAI [99]. This case only has two inputs and is much easier than a turbine control case, but has similarly defined constraints, goals and rewards. The reinforcement method was shown to work well when capturing the environment state as a snapshot of both position and velocity.

Implementing this reinforcement method on a turbine start-up task is left for future evaluation. Application to turbine control would likely require both absolute and rate of change of the turbine condition to be monitored. The approach should be viable given a sufficiently high event count and fast-acting simulation. The more advanced control requirements with multiple pressures, temperatures and speed inputs may necessitate more complex actor-critic [100] or policy based models [101] [102].

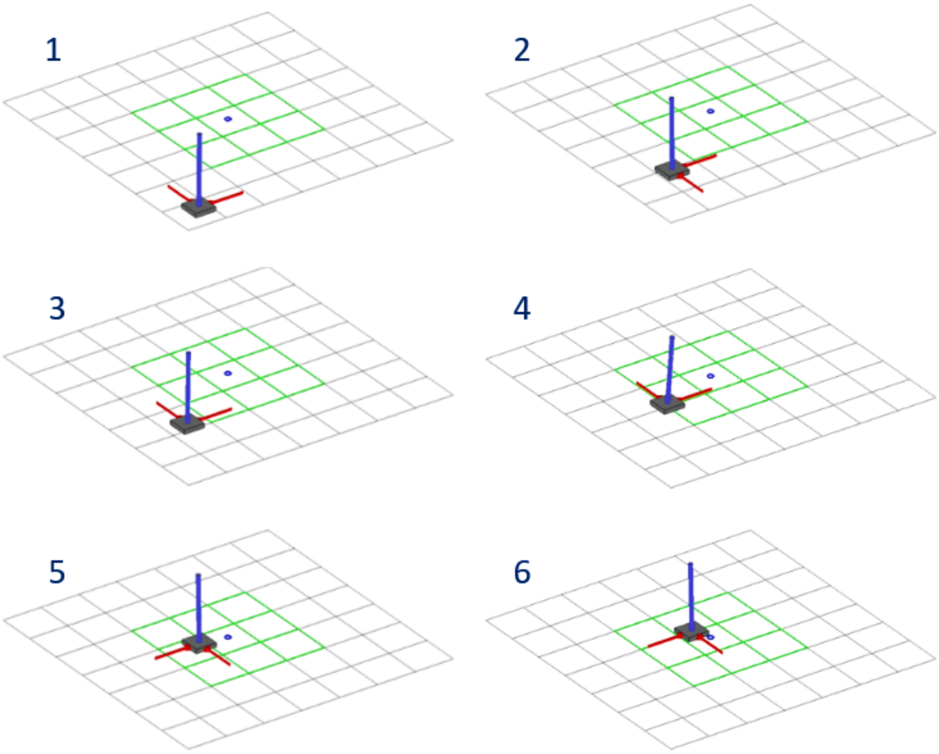


Figure 9.16: Six time-stamp images from the 3D cart-pole reinforcement optimisation task, showing the learned control for x-y force inputs (red), to achieve optimum position near the centre of the grid (green), whilst ensuring the pole (blue) remains balanced within pre-set angle limits (15^o).

9.3.3 Passive Control, Thermal Cladding

Passive control looks to add thermal bridges, or cladding, to shape the thermal profile as the turbine cools, ensuring more uniform clearances. This reduces the requirement for pre-warming and reduces the overall time between demand and availability.

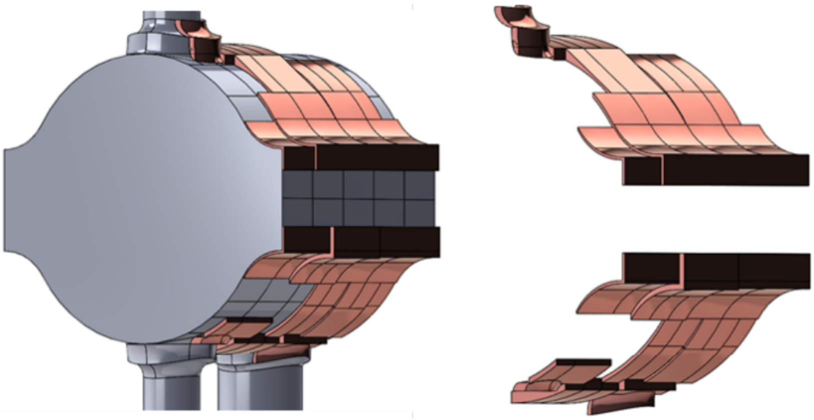


Figure 9.17: 3D model of the simplified HP turbine casing and final copper cladding design for optimum reduction of thermal stress during start-up [103].

Final year Undergraduate, Barley [103], used the TNMS codes to design a shaped copper cladding to reduce the thermal gradients during start-up. A custom thickness cladding was added using variable resistors to model the copper thermal resistance. The resistors were optimised to reduce the peak thermal gradients during a transient start-up then the associated copper thickness extracted. Using the low order TNMS, many designs were evaluated quickly and manually optimised. The final design was then evaluated in ANSYS Thermal. The accumulated thermal gradient over the full start-up process was reduced by 12-32% across the different ordinate directions.

Future upgrades should look to use embedded optimisers in the thermal network model. Replacing the manual variable resistors with Tensorflow variables, coupled gradient descent tools could be used to adjust the copper resistances automatically. Adding a custom loss function to minimise the thermal gradient, the training exercise could solve the cladding shape directly. Automating the analysis would allow this to be done at mesh resolution, rather than the simplified block structure used in Barley's project.

9.3.4 Active Control, Targeted Heating

Active control looks to shape the thermal profile using active heating and cooling. Started by Barley and continued by Umavannan [104]; the TNMS codes were used to optimise heater pad placement and input power, with the aim to reduce the thermal gradients during start-up. Both students chose to place pads manually, adjusting the location and power based on data from previous simulations.

Similar to the suggested improvement above, embedded optimisers in the thermal calculation could be used to solve a per facet ideal heat flux. This optimised case could then be used to back-solve a heater distribution at higher resolution. Both Barley and Umavannan validated their final design against an ANSYS thermal calculation, both showing good agreement with the expected thermal profile from TNMS. Figure 9.18 shows the result from Umavannan's work, identifying the possible reduction of thermal gradients under different heater configurations. The test conditions used the first 100 hours of the MHI operating data from Chapter 6, with analysis starting at time 37hr.

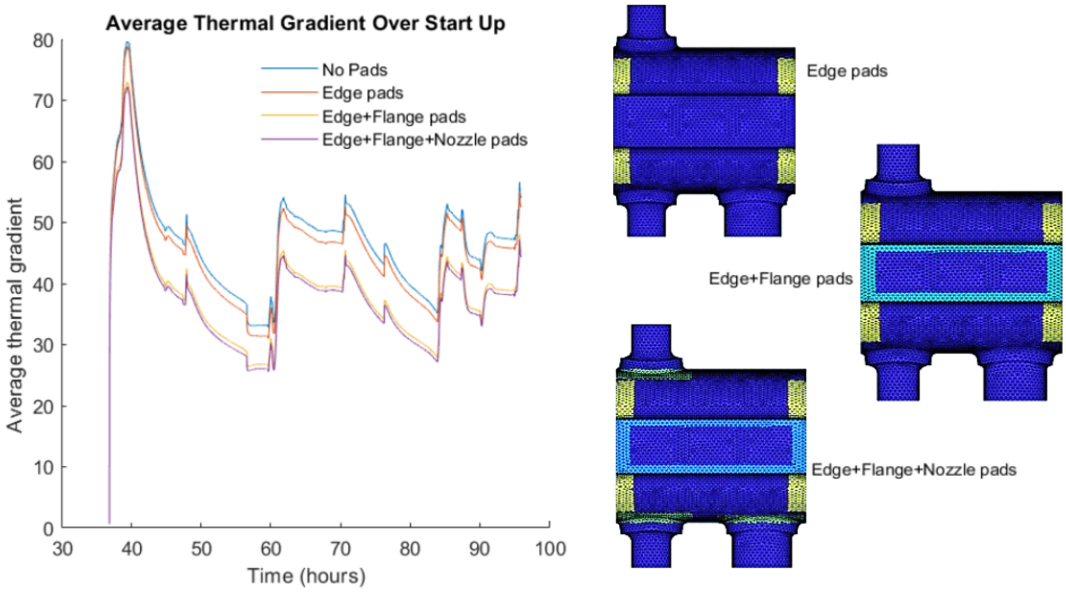


Figure 9.18: The average thermal gradient (dT/dx) in the simplified casing model during a cold and hot start operating period (37 - 96hr), comparing the effectiveness of different heater pad designs and placement.

9.3.5 HTC Solution by Optimisation

Umavannan extend the unknown resistor concept to look at the solution of unknown HTC on the inside of a turbine casing. Using known boundary conditions for inlet and exhaust temperature, a model was run and compared to specified thermocouple data on the casing exterior. A HTC shape function was manually defined with TNMS to set the thermal resistance on the inside of the casing. The shape function was then adjusted to minimise the error between the model and measured external probes.

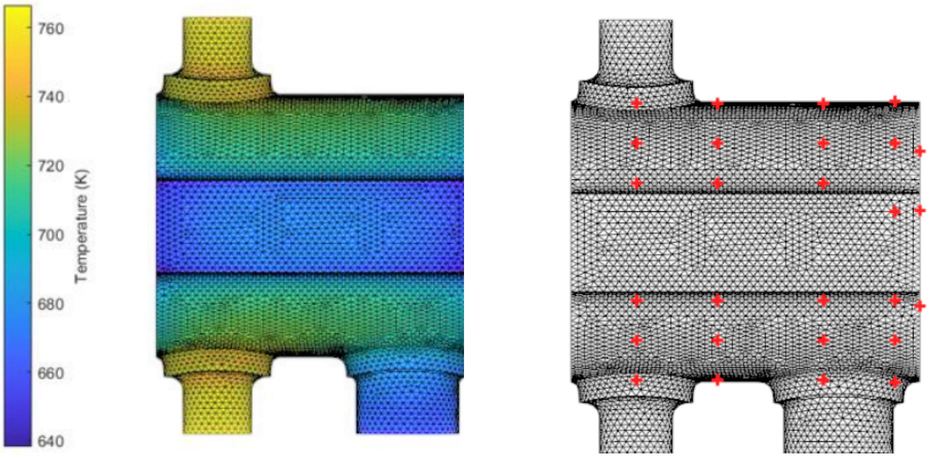


Figure 9.19: The external temperature distribution from TNMS, showing the point-wise thermocouple locations used to tune the model output to measured data values.

Umavannan used the method to identify that three different shape functions were required to accurately model the changing HTC caused by the differing flow conditions during start-up. The internal HTC was defined as a function of axial and circumferential position on the casing inner cylindrical surface, one example shape function is shown in Figure 9.20. The process additionally allowed him to identify the ideal transition between the shape functions.

Unknowns for both internal temperature and HTC cannot be decoupled, so an approximation was required to set the internal steam temperature. This was achieved by taking a location specific weighted average of the known inlet and exhaust steam temperatures. Future upgrades should look to use a more accurate fluid model or possibly import data from CFD to better resolve the internal steam temperature.

This application highlighted the strength of a hybrid methodology. By combining measurements and simulations with TNMS optimisation, unknown HTC parameters in inaccessible locations have been estimated. The ability to run many simulations quickly, using low order vector algebra with defined gradients, is the key to the success of this approach. Expanding this concept to tackle more complex geometry that better replicates a full HP/IP steam turbine would certainly be of interest.

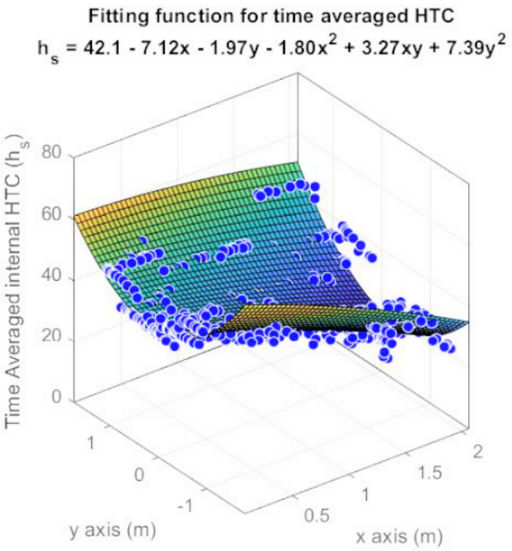


Figure 9.20: Internal HTC shape function for the casing interior surface in the axial (x) and circumferential (y) directions.

9.3.6 Thermal-Neural Modelling

Recurrent neural network nodes, with connecting weights, behave the same as a thermal capacitor and resistor network. Using a recurrent neural model would allow an equally fast thermal simulation but with the added ability to easily include unknowns in the domain. Setting a known target temperature or heat flux at a specified location, the embedded gradient descent optimisers could be forced to adjust the recurrent nodes to solve the full network. Better still, thermal resistance and capacitance could be specified for known regions of the domain and defined untrainable. This would lock large regions of the network, forcing the gradient descent updates to affect only specified unknowns.

This focused optimisation replaces the classic black-box use of AI tools, which is the largest criticism and restriction to their use. Such a model could be used to solve unknown HTC values, unknown mass flow rates or equivalent lump thermal properties. Careful specification of the network would allow meaningful weight values to be extracted, returning useful engineering information. The approach requires custom functions with defined back propagation gradients; support is already available in TensorFlow to write these and this functionality has been used to define the internal stability loop in TNMS Solve.

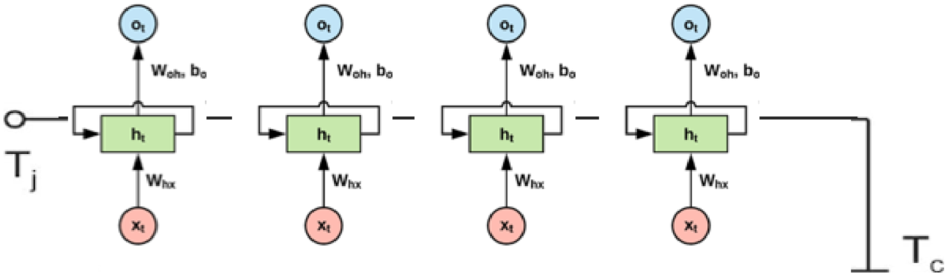


Figure 9.21: Thermal-neural model of a Cauer ladder showing the recurrent weights (h), connectivity weights (w) and bias (b) required to mimic conventional thermal resistance and capacitance.

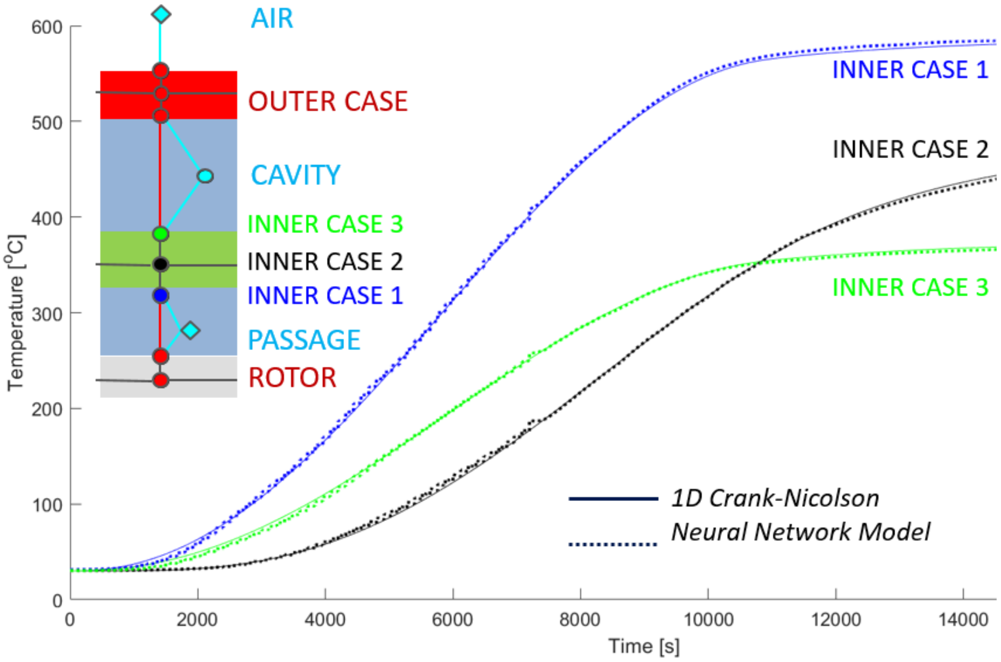


Figure 9.22: Performance of an 11 node 1D thermal-neural model trained to match the boundary and internal temperature of the inner case during a 5hr start-up regime.

The concept was briefly tested on a simplified 1D section through a turbine rotor, passage, cavity and casings. The Crank-Nicolson method for multilayer materials was modified to allow internal convection boundaries. This solver was used to generate a series of observed temperatures during start-up at the boundary and mid-point of each region in the domain. A 1D recurrent network was then trained to match the thermal behaviour and given full freedom to define material properties and thermal resistances.

The behaviour of 4,464 nodes in the 1D Crank-Nicolson algorithm was replicated by 11 thermal-neural LSTM blocks in the network. Further work is required to assess the suitability of this method. The highly reduced thermal-neural model likely has limited accuracy beyond the scope of the observed training case.

9.4 Summary

The Thermal Network Model Software has been validated for three cases, testing the different boundary conditions and simulation modes on offer. The accuracy has proven to be suitable whilst allowing fast simulation of heat transfer. The advantage of the hybrid methodology, to combine data across the fidelity range, has been demonstrated for the complex case of a full HP/IP turbine. This multi-fidelity approach offers a viable solution for real-world monitoring of flexible operation turbine. Future work is required to further accelerate the software using optimised Tensorflow graphs and dedicated high performance hardware.

Several further projects that branch from this thesis have been discussed, along with preliminary results and development opportunities. Combining the hybrid tools with integrated optimisers in TNMS offers exciting prospects in thermal system analysis. Neural network integration for latent space solvers, fast fluid models or process optimisation all demonstrate the wider range of possibilities for the hybrid concept. These challenging projects identify areas for new research, critical to the continued development of flexible operation power turbine.

10

Conclusion

10.1 Thesis Summary

The global market shift towards renewable energy places demanding requirements on power turbines. New load-levelling applications dictate high flexibility, fast start-up and unlimited load changes. This thesis set out to evaluate fast, approximate, transient heat analysis methods, essential for the development and support of power turbines in this application. New methodologies, along with improvements to existing standard practices, have been presented and validated for both 1D and 3D thermal analysis. Hybrid methods that allow data combination from a wide fidelity range have been shown to bridge the limitations in traditional research. A combined model using measurement data, neural network tools, CFD data and thermal network modelling was used to successfully demonstrate the concept. Several publications from this work are planned and have been outlined in Appendix B.

10.2 1D Impulse Response

Widely used in the heat transfer community, the impulse response remains one of the most effective fast transient analysis tools for 1D data. Assumptions in the method's derivation are often overlooked and the true accuracy poorly understood. Full domain analytic solutions have been presented which clarify the use, accuracy and limitations of the method. An improved post-processing methodology was demonstrated, allowing the calculation of surface HTC analysis in transient flows. Future high temperature thin films will provide direct thermal data for coupling in low order simulations. Empirical laboratory analysis of HTC will continue to be highly valuable in defining simulation parameters for turbine thermal models.

10.3 1D Multilayer and Curvature

Geometric effects are often overlooked when analysing laboratory test articles. Improved numerical methods and analytic solutions were presented for handling multilayer and curved geometries. Better able to resolve the true behaviour, these new methods offer improved accuracy in heat transfer analysis. Fast 1D numerical methods are a prerequisite for future high temperature heat flux gauges to support online monitoring. The numerical route removes the semi-infinite limit, allowing thin films to used in extended operation.

10.4 Neural Networks for Thermal Prediction

Long Short Term Memory blocks were demonstrated for online thermal prediction from standard plant monitoring measurements. Using a truncated time history, these tools offer the support for long duration simulation without divergence. A dependence on large data sets restricts their immediate use. However, the ongoing industry transition to data collection and monitoring will soon facilitate these techniques. An autoencoder was shown to reliably compress data to a recoverable latent space. This reduced representation offers opportunities in fidelity transfer, high speed latent solvers and optimum probe placement.

10.5 Thermal Network Modelling

Electrical equivalence network models have been shown to offer suitable approximation in the analysis of complex geometries. A new software was written to automate thermal network construction, allowing optional blocking by zone. Assumptions in static thermal properties and flow parameters allow upfront definition of the solver matrices, further accelerating the calculation. Support for a range of boundary conditions and directly couple measurement data. Options to approximately simulate internal radiation are also included. Further work is needed to better resolve the fluid domain and optimise the hardware requirements for online simulation.

10.6 Hybrid Systems for Flexible Operation

The transition to flexible operation necessitates a wider operating window and off-peak operation. Exhaustive CHT to evaluate all possible operating conditions is becoming inviable. Data driven solutions, using live measurements coupled with online simulations, will prove essential to support flexible operation. Taken independently, the tools and methods analysed in this thesis are insufficient to accurately model the transient behaviour of a full turbine in flexible operation. Taken collectively, using all tools in a single hybrid model, a full machine can be resolved.

This shift from upfront physical analysis to live black-box data is concerning for many Original Equipment Manufacturers (OEM). Hybrid methods must therefore focus on white-box solutions to retain some physical understanding. Future systems need to be considered as a whole, with data from all stages of the design cycle used. Traditionally isolated processes must now feed the same hybrid system. Research, development, design, prototyping, build, installation and monitoring should no longer be considered independent tasks.

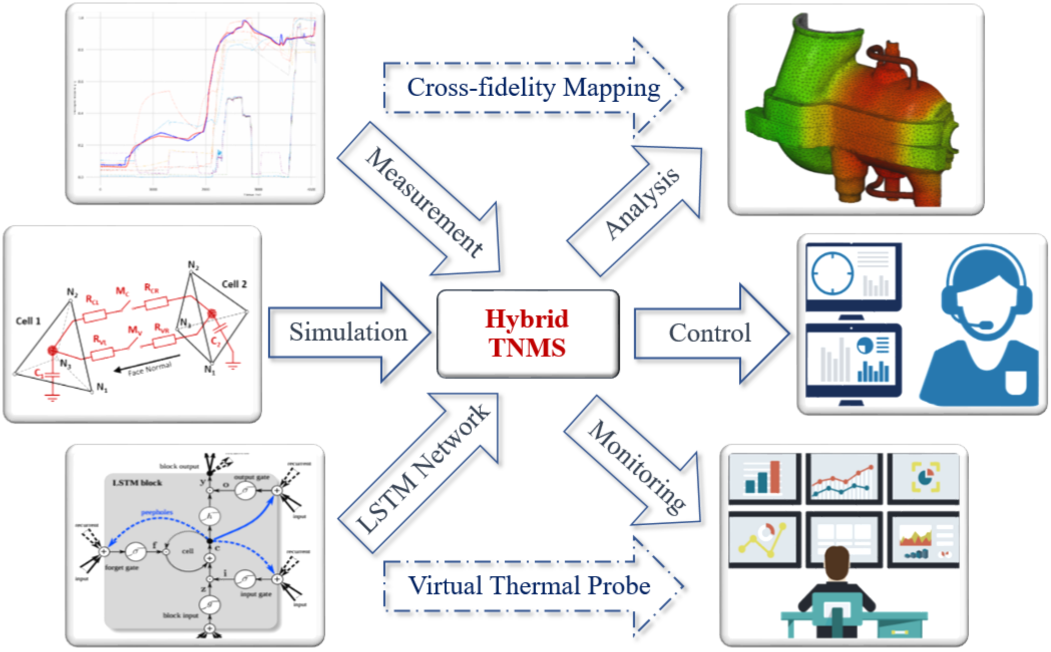


Figure 10.1: Overview of the hybrid system, showing the range of features that can be integrated with fast-acting thermal network models.

10.7 Final Remarks

The transition towards a renewable energy future is important and has a large impact on the role of power turbines. Focus must now be given to availability, supporting faster and repeated load changes. Modelling flexibility for load-levelling operation necessitates a new mindset in heat transfer analysis. Hybrid methodologies are required, supporting a wider portfolio of analysis techniques. Numerous research opportunities exist in further developing the hybrid concept for this challenging application.

Appendices

A

Multilayer Impulse Response Basis Function Derivation

The full laminate basis functions are derived in the same approach as the single layer case. Starting at the final layer, N, interface boundary conditions are propagated upwards through the laminate to the known top layer heat flux condition. The underlying assumption is that the back substrate is semi-infinite and has the same Laplace domain solution as the single layer case discussed in Section 4.3.1.

$$\phi_N(x,s) = A_N(s)\ \mathrm{e}^{-x\sqrt{s/\alpha}} \tag{A.1}$$

In the laminate case, the surface function $A_N(s)$ is taken from the preceding layer, $N-1$, with the following interface boundary conditions

1. Temperature continuity: $\phi_{n-1}(x_{n-1},s) = \phi_n(x_{n-1},s)$
2. Heat flux continuity: $\psi_{n-1}(x_{n-1},s) = \psi_n(x_{n-1},s)$

Introducing the variable substitutions:

- $A_n = A_n(s),\ B_n = B_n(s)$
- $\sqrt{s/\alpha_n} = \lambda_n$

Temperature continuity at $x = x_{N-1}$ requires:

$$A_{N-1}\ \mathrm{e}^{-x_{N-1}\lambda_{N-1}} + B_{N-1}\ \mathrm{e}^{x_{N-1}\lambda_{N-1}} = A_N\ \mathrm{e}^{-x_{N-1}\lambda_N} \tag{A.2}$$

Heat flux continuity at $x = x_{N-1}$ requires:

$$k_{N-1}\lambda_{N-1}A_{N-1}\ \mathrm{e}^{-x_{N-1}\lambda_{N-1}} - k_{N-1}\lambda_{N-1}B_{N-1}\ \mathrm{e}^{x_{N-1}\lambda_{N-1}} = k_N\lambda_N A_N\ \mathrm{e}^{-x_{N-1}\lambda_N} \tag{A.3}$$

Introducing the variable $\sigma_{m,n}$:

$$\sigma_{m,n} = \frac{k_n\lambda_n}{k_m\lambda_m} \tag{A.4}$$

Then multiplying A.2 by σ and subtracting A.3, the following relation is found

$$(1-\sigma_{N-1,N})A_{N-1}\ \mathrm{e}^{-x_{N-1}\lambda_{N-1}} = (1+\sigma_{N-1,N})B_{N-1}\ \mathrm{e}^{x_{N-1}\lambda_{N-1}} \tag{A.5}$$

Introducing the variable $\gamma_{m,n}$:

$$\gamma_{m,n} = \frac{1-\sigma_{m,n}}{1+\sigma_{m,n}} \tag{A.6}$$

B_{N-1} can be found in terms of A_{N-1}

$$B_{N-1} = A_{N-1}\ \mathrm{e}^{-2x_{N-1}\lambda_{N-1}}\ \ \gamma_{N-1,N} \tag{A.7}$$

Moving up one layer in the laminate stack, the temperature and heat flux conservation at this interface requires:

$$\mathrm{T}: \quad A_{N-2}\ \mathrm{e}^{-x_{N-2}\lambda_{N-2}} + B_{N-2}\ \mathrm{e}^{x_{N-2}\lambda_{N-2}} = A_{N-1}\ \mathrm{e}^{-x_{N-2}\lambda_{N-1}} + B_{N-1}\ \mathrm{e}^{x_{N-2}\lambda_{N-1}} \tag{A.8}$$

$$\begin{aligned}\mathrm{F}: \quad & k_{N-2}\lambda_{N-2}A_{N-2}\ \mathrm{e}^{-x_{N-2}\lambda_{N-2}} - k_{N-2}\lambda_{N-2}B_{N-2}\ \mathrm{e}^{x_{N-2}\lambda_{N-2}} = \\ & k_{N-1}\lambda_{N-1}A_{N-1}\ \mathrm{e}^{-x_{N-2}\lambda_{N-1}} - k_{N-1}\lambda_{N-1}B_{N-1}\ \mathrm{e}^{x_{N-2}\lambda_{N-1}}\end{aligned} \tag{A.9}$$

Substituting variables σ, λ and B_{N-1}, then multiplying A.8 by σ, the two continuity equations above become A.10 and A.11. Noting that x_n is the spatial position of the back surface of layer n, $(x_{N-1} - x_{N-2})$ defines the thickness of layer $N-1$.

$$\begin{aligned}\sigma_{N-2,N-1}\ A_{N-2}\ \mathrm{e}^{-x_{N-2}\lambda_{N-2}} + \sigma_{N-2,N-1}\ B_{N-2}\ \mathrm{e}^{x_{N-2}\lambda_{N-2}} = \\ \sigma_{N-2,N-1}\ A_{N-1}\ \mathrm{e}^{-x_{N-2}\lambda_{N-1}}\Big[1 + \gamma_{N-1,N}\ \mathrm{e}^{-2(x_{N-1}-x_{N-2})\lambda_{N-1}}\Big]\end{aligned} \tag{A.10}$$

$$\begin{aligned}A_{N-2}\ \mathrm{e}^{-x_{N-2}\lambda_{N-2}} - B_{N-2}\ \mathrm{e}^{x_{N-2}\lambda_{N-2}} = \\ \sigma_{N-2,N-1}\ A_{N-1}\ \mathrm{e}^{-x_{N-2}\lambda_{N-1}}\Big[1 - \gamma_{N-1,N}\ \mathrm{e}^{-2(x_{N-1}-x_{N-2})\lambda_{N-1}}\Big]\end{aligned} \tag{A.11}$$

Multiplying the temperature equation A.10 by the RHS [ ] term in the flux equation and multiplying the flux equation A.11 by the RHS [ ] term in the temperature equation. The RHS of the above two equations becomes equal and A.10 can be equated to A.11

$$\begin{aligned}\sigma_{N-2,N-1}\ \Big[1 - \gamma_{N-1,N}\ \mathrm{e}^{-2(x_{N-1}-x_{N-2})\lambda_{N-1}}\Big]\ \Big[A_{N-2}\ \mathrm{e}^{-x_{N-2}\lambda_{N-2}} +\ B_{N-2}\ \mathrm{e}^{x_{N-2}\lambda_{N-2}}\Big] = \\ \Big[1 + \gamma_{N-1,N}\ \mathrm{e}^{-2(x_{N-1}-x_{N-2})\lambda_{N-1}}\Big]\ \Big[A_{N-2}\ \mathrm{e}^{-x_{N-2}\lambda_{N-2}} -\ B_{N-2}\ \mathrm{e}^{x_{N-2}\lambda_{N-2}}\Big]\end{aligned} \tag{A.12}$$

This can be rearranged to find B_{N-2} in terms of A_{N-2}

$$B_{N-2} = A_{N-2}\ \mathrm{e}^{-2x_{N-2}\lambda_{N-2}}\frac{(1-\sigma_{N-2,N-1}) + (1+\sigma_{N-2,N-1})\ \gamma_{N-1,N}\ \mathrm{e}^{2x_{N-2}\lambda_{N-2}}}{(1+\sigma_{N-2,N-1}) + (1-\sigma_{N-2,N-1})\ \gamma_{N-1,N}\ \mathrm{e}^{2x_{N-2}\lambda_{N-2}}} \tag{A.13}$$

This equation has the same form as A.7 and can be written

$$B_{N-2} = A_{N-2}\ \mathrm{e}^{-2x_{N-2}\lambda_{N-2}}\quad \eta_{N-2,N-1} \tag{A.14}$$

$$where: \eta_{N-2,N-1} = \frac{\gamma_{N-2,N-1} + \gamma_{N-2,N-1}\ \mathrm{e}^{-2(x_{N-1}-x_{N-2})\lambda_{N-1}}}{1 + \gamma_{N-1,N}\ \gamma_{N-1,N}\ \mathrm{e}^{-2(x_{N-1}-x_{N-2})\lambda_{N-1}}}$$

The process from A.10 to A.14 can be repeated for all additional layers in the laminate stack. For each layer above $N-2$, the relation $\eta_{m,m+1}$ is based on the value of η in the below layer and γ of the current layer.

$$\eta_{m,m+1} = \frac{\gamma_{m,m+1} + \eta_{m+1,m+2}\ \mathrm{e}^{-2(x_{m+1}-x_m)\lambda_{m+1}}}{1 + \gamma_{m,m+1}\ \eta_{m+1,m+2}\ \mathrm{e}^{-2(x_{m+1}-x_m)\lambda_{m+1}}}\ ; \quad 0 < m < N-2 \tag{A.15}$$

Once this process has propagated to the top surface of the material stack, the surface temperature and flux boundary conditions can be applied.

$$\phi(0,s) = A_1 + B_1 \tag{A.16}$$

$$\psi(0,s) = k_1\lambda_1 A_1 - k_1\lambda_1 B_1 \tag{A.17}$$

Substituting B_1 from A.14 into the above surface conditions gives

$$A_1 = \frac{\psi(0,s)}{k_1\lambda_1}\left[\frac{1}{1-\eta_{1,2}\ \mathrm{e}^{-2x_1\lambda_1}}\right]$$

$$B_1 = \frac{\psi(0,s)}{k_1\lambda_1}\left[\frac{\eta_{1,2}\ \mathrm{e}^{-2x_1\lambda_1}}{1-\eta_{1,2}\ \mathrm{e}^{-2x_1\lambda_1}}\right] \tag{A.18}$$

$$\phi(0,s) = \frac{\psi(0,s)}{k_1\lambda_1}\left[\frac{1+\eta_{1,2}\ \mathrm{e}^{-2x_1\lambda_1}}{1-\eta_{1,2}\ \mathrm{e}^{-2x_1\lambda_1}}\right]$$

$\phi(0,s)$ can be rearranged into the following form

$$\phi(0,s) = \frac{\psi(0,s)}{k_1\lambda_1}\left[1+\frac{2\eta_{1,2}\ \mathrm{e}^{-2x_1\lambda_1}}{1-\eta_{1,2}\ \mathrm{e}^{-2x_1\lambda_1}}\right] \tag{A.19}$$

To solve for unit step heat flux, $\psi(0,s) = 1/s$, the denominator in this expression must be expanded as a power series

$$\frac{1}{1-z} = \sum_{i=0}^{\infty} z^i \ ; \ |z| < 1 \tag{A.20}$$

This gives the final Laplace domain surface temperature solution $\phi(0,s)$

$$\phi(0,s) = \frac{1}{k_1 \lambda_1 s} \left[1 + 2 \sum_{i=1}^{\infty} \eta_{1,2}^i \ \mathrm{e}^{-2x_1\lambda_1 i}\right] \tag{A.21}$$

The variable $\eta^i_{m,m+1}$ from A.15 is given by

$$\eta^i_{m,m+1} = \left[\frac{\gamma_{m,m+1} + \eta_{m+1,m+2} \ \mathrm{e}^{-2(x_{m+1}-x_m)\lambda_{m+1}}}{1 + \gamma_{m,m+1} \ \eta_{m+1,m+2} \ \mathrm{e}^{-2(x_{m+1}-x_m)\lambda_{m+1}}}\right]^i \quad ; \quad 0 < m < N-2 \tag{A.22}$$

Expanding the numerator and denominator as a binomial series, η can be expressed as a product of summations.

$$\begin{aligned} \eta^i_{m,m+1} = & \left[\sum_{j=0}^{i} \begin{pmatrix} i \\ j \end{pmatrix} \gamma^{i-j}_{m,m+1} \ \eta^j_{m+1,m+2} \ \mathrm{e}^{-2j(x_{m+1}-x_m)\lambda_{m+1}}\right] \times \\ & \left[\sum_{k=0}^{\infty} \begin{pmatrix} i+k-1 \\ k \end{pmatrix} (-1)^k \ \gamma^k_{m,m+1} \ \eta^k_{m+1,m+2} \ \mathrm{e}^{-2k(x_{m+1}-x_m)\lambda_{m+1}}\right] \end{aligned} \tag{A.23}$$

The above product can be combined to give an infinite sum of a finite series. For progressive terms in the infinite summation, the exponential term tends to zero. The infinite series can therefore be accurately approximated by a truncated finite series.

$$\eta^i_{m,m+1} = \sum_{k=0}^{\infty} \left[\sum_{j=0}^{i} \begin{pmatrix} i \\ j \end{pmatrix} \begin{pmatrix} i+k-1 \\ k \end{pmatrix} (-1)^k \ \gamma^{i-j+k}_{m,m+1} \ \eta^{j+k}_{m+1,m+2} \ \mathrm{e}^{-2(k+j)(x_{m+1}-x_m)\lambda_{m+1}}\right] \tag{A.24}$$

B

Planned Publications

International Journal of Heat and Mass Transfer: Accuracy of the 1D Impulse Response Heat Transfer Method

Derivation of the temporal-spatial heat transfer solutions for 1D conduction and convection. Description of the 1D convection impulse response basis functions. Validation of the impulse response method and accuracy, outlining the new method for semi-infinite criterion calculation.

International Journal of Heat and Mass Transfer: Subsurface 1D Impulse Response Heat Transfer

Overview of the new integral impulse response method for transient bulk flow experiments, demonstrating the calculation procedure for both surface and subsurface embedded measurements. Validation of the method with one example experimental case.

International Journal of Heat and Mass Transfer: 1D Multilayer Heat Transfer

Multilayer transmission-reflection method, with derivation and nested loop solution code. Definition of the 1D Crank-Nicolson method with interface flux conservation and a comparison of the numerical and impulse response calculations.

International Journal of Heat and Mass Transfer: Cylindrical 1D Impulse Response Heat Transfer

Definition of the 1D cylindrical solutions for convex and concave geometry with an explanation of the effect on surface temperature under uniform heating. Introduction to the concept of effective 1D radius with a comparison to a 3D ANSYS Thermal case. Guidance on the 3D camera mapping and Delaunay spatial interpolation procedure for curvature correction.

ASME Power: Long Short Term Memory Neural Network Thermal Prediction

Present the LSTM neural network methods with split model architecture for prediction and correction. Demonstration of the model on readily-available site data for thermal profile prediction. Outline of the input selection, feature detection and pre-processing for optimum performance.

ASME Power: Hybrid Methods and Multi-fidelity Thermal Prediction

Introduction to the Hybrid concept and simultaneous use of multiple fidelity systems for thermal prediction. Demonstration of the autoencoder neural network methods for fidelity switching and latent space discovery.

ASME Power: Thermal Network Model Software, Automated Blocking Solver

Description of the Thermal Network Model Software, including solver matrices and automated blocking strategy. Details of the hash table blocking algorithm and automated process for thermal resistance and capacitance calculation.

ASME Power: Thermal Network Model Software, Passive and Active Control

Demonstration of the Thermal Network Model Software in the design of active and passive control systems. Direct integration of Adam optimisers into the solution to allow automated selection of the heating or cladding.

ASME Power: Thermal Network Model Software, Transient Process Optimisation

Application of the reinforcement learning algorithms to develop an optimised control sequence for fast start of flexible operation power turbines. Definition of the Monte-Carlo policy method for environment observation learning.

References

[1] World Energy Council. *World Energy Scenarios 2016*. 2016. URL: `www.worldenergy.org/publications`.

[2] OECD-IEA. “Key World Energy Statistics”. In: *IEA Publishing* 2017. ISBN:978-87-93180-22-2, p. 58. URL: `www.iea.org/t%7B%5C&%7Dc`.

[3] British Petroleum. “BP Statistical Review of World Energy 2017”. In: *BP Statistical Review* 2017. arXiv: `1011.1669v3`.

[4] BEIS. *Energy Trends March 2018*. Tech. rep. UK Department for Business, Energy & Industrial Strategy, 2018. URL: `www.gov.uk/government/statistics/energy-%20trends-march-2018`.

[5] BEIS. *Energy Trends March 2017*. Tech. rep. UK Department for Business, Energy & Industrial Strategy, 2017. URL: `www.gov.uk/government/statistics/energy-%20trends-march-2017`.

[6] Andy Colthorpe. “Manufacturer reveals involvement in world’s biggest battery energy storage system so far”. In: *Energy Storage News* July 17, 2021. URL: `https://www.energy-storage.news/manufacturer-reveals-involvement-in-worlds-biggest-battery-energy-storage-system-so-far/`.

[7] Andy Colthorpe. “Fire hits construction of 300MW Victorian Big Battery in Australia”. In: *Energy Storage News* July 30, 2021. URL: `https://www.energy-storage.news/fire-hits-construction-of-300mw-victorian-big-battery-in-australia/`.

[8] Andy Colthorpe. “Investigation begins into overheating incident at worlds biggest battery storage facility”. In: *Energy Storage News* Sept. 8, 2021. URL: `https://www.energy-storage.news/investigation-begins-into-overheating-incident-at-worlds-biggest-battery-storage-facility/`.

[9] Cision. “Global Steam Turbines Market Analysis & Trends - Industry Forecast to 2025”. In: *PR Newswire* Mar. 21, 2017. URL: `https://www.prnewswire.com/news-releases/global-steam-turbines-market-analysis-trends--industry-forecast-to-2025-300427192.html`.

[10] David Appleyard. “Advancing Steam Turbine Technology”. In: *Power Engineering International* Apr. 20, 2016. URL: `https://www.powerengineeringint.com/coal-fired/advancing-steam-turbine-technology/`.

[11] Mariusz Banaszkiewicz. “Steam turbines start-ups”. In: *Transactions of the Institute of Fluid-Flow Machinery* 126.126 2014, pp. 169–198.

[12] Grzegorz Nowak et al. “Improving the power unit operation flexibility by the turbine start-up optimization”. In: *Energy* 2020. DOI: 10.1016/j.energy.2020.117303.

[13] General Electric Company. *Forced - air Cooling System for Industrial Steam Turbines*. 2017. URL: `https://www.ge.com/content/dam/gepower-new/global/en%7B%5C_%7DUS/downloads/steam-new-site/services/forced-air-cooling-system.pdf`.

[14] Salvatore A Della Villa. “Energy Innovation: A Focus on Power Generation Data Capture and Analytics in a Competitive Market”. In: *Proceedings of ASME TurboExpo* 2018. GT2018-75030.

[15] Jan Greis, Edwin Gobrecht, and Steffen Wendt. “Flexible And Economical Operation Of Power Plants - 25 Years Of Expertise”. In: *Proceedings of ASME TurboExpo* 2012. GT2012-68716.

[16] Dominik Born, Gabriel Marinescu, and Stefan Koch. “Thermal Modelling of an Intermediate Pressure Steam Turbine By Means of Conjugate Heat Transfer Simulation and Validation”. In: *Proceedings of ASME TurboExpo* 2016. GT2016-57247.

[17] L. He and M. L. G. Oldfield. “Unsteady Conjugate Heat Transfer Modeling”. In: *Journal of Turbomachinery* 133.3 2011. DOI: 10.1115/1.4001245.

[18] Piotr Luczynski. “Unsteady Conjugate Heat Transfer Investigation of a Multistage Steam”. In: *Proceedings of ASME TurboExpo* 2018. GT2018-75926.

[19] M. B. Giles. “Stability analysis of numerical interface conditions in fluid-structure thermal analysis”. In: *International Journal for Numerical Methods in Fluids* 25.4 1997, pp. 421–436.

[20] Tadeusz Chmielniak et al. “Analysis of Thermal and Stress States in Transient Operation of a Turbine Co-operating with Twinboiler”. In: *Heat Transfer Engineering* 7632 2017. DOI: 10.1080/01457632.2017.1363634.

[21] Wolfgang Beer, Lukas Propp, and Lutz Voelker. “A Simplified Analytical Approach For Calculating The Start-Up Time Of Industrial Steam Turbines For Optimal And Fast Start-Up Procedures”. In: *Proceedings of ASME TurboExpo* 2015. GT2015-43284.

[22] Dennis Toebben. “Analytical Heat Transfer Correlation for a Multistage Steam Turbine in Warm-Keeping Operation With Air”. In: *Proceedings of ASME TurboExpo* 2018. GT2018-76091.

[23] Howard M Brilliant and Anil K Tolpadi. “Analytical Approach to Steam Turbine Heat Transfer in a Combined Cycle Power Plant”. In: *Proceedings of ASME Conference* 2004.41685 2004, pp. 401–409.

[24] Debabrata Mukhopadhyay et al. “Development of a conjugate heat transfer simulation methodology for prediction of steam turbine cool- down phenomena and shell deflection”. In: *Proceedings of ASME TurboExpo* 2018. GT2014-25874.

[25] James Spelling, Markus Jöcker, and Andrew Martin. “Thermal Modeling Of A Solar Steam Turbine With A Focus On Start-Up Time Reduction”. In: *Proceedings of ASME TurboExpo* 2011. GT2011-45686.

[26] M. Topel et al. “Geometric modularity in the thermal modeling of solar steam turbines”. In: *Energy Procedia.* DOI: 10.1016/j.egypro.2014.03.184. 2014.

[27] Monika Topel and Björn Laumert. “Improving concentrating solar power plant performance by increasing steam turbine flexibility at start-up”. In: *Solar Energy* 2018.

[28] W. F. Mohr and P. Ruffino. “Experimental investigation into thermal behaviour of steam turbine components, part 1 - Temperature measurements with optical probes”. In: *Proceedings of the ASME Turbo Expo* 2012. GT2012-68703.

[29] Gabriel Marinescu et al. “Experimental Investigation Into Thermal Behavior of Steam Turbine ComponentsTemperature Measurements With Optical Probes and Natural Cooling Analysis”. In: *Journal of Engineering for Gas Turbines and Power* 136.2 2014. DOI: 10.1115/1.4025556.

[30] Gabriel Marinescu, Andreas Ehrsam, and Michael Sell. “Experimental Investigation Into Thermal Behavior Of Steam Turbine Components. Part 3 Startup And The Impact On LCF Life”. In: *Proceedings of ASME TurboExpo* 2013. GT2013-94356.

[31] Gabriel Marinescu, Michael Sell, and Peter Stein. “Experimental Investigation Into Thermal Behavior of Steam Turbine Components . Part 4 Natural Cooling and Robustness of the Overconductivity Function”. In: *Proceedings of ASME TurboExpo* 2014. GT2014-25247.

[32] Krzysztof Kosowski, Karol Tucki, and Adrian Kosowski. “Application of Artificial Neural Networks in Investigations of Steam Turbine Cascades”. In: *Journal of Turbomachinery* 132.1 2010. DOI: 10.1115/1.3103923, pp. 1–5.

[33] Krzysztof Dominiczak, Romuald Rzadkowski, and Wojciech Radulski. “Steam turbine stress control using NARX neural network”. In: *Open Engineering* Volume 5.Issue 1 2015, p. 43.

[34] Vipul Goyal et al. “Prediction of gas turbine performance using machine learning methods”. In: *Proceedings of the ASME Turbo Expo* 6 2020. GT2020-15232, pp. 1–11.

[35] Ibrahem M.A. Ibrahem et al. “An ensemble of recurrent neural networks for real time performance modelling of three-spool aero-derivative gas turbine engine”. In: *Proceedings of the ASME Turbo Expo* 6 2020. DOI: 10.1115/GT2020-15756, pp. 1–13.

[36] V. Panov and S. Cruz-Manzo. “Gas turbine performance digital twin for real-time embedded systems”. In: *Proceedings of the ASME Turbo Expo* 2020. DOI: 10.1115/GT2020-14664.

[37] Milton Abramowitz and Irene A Stegun. *Handbook of mathematical functions : with formulas, graphs, and mathematical tables [electronic resource]*. 10th print. Applied mathematics series (Washington, D.C.) ; 55. Washington, D.C., 1972, p. 1026.

[38] M. L. G. Oldfield. “Impulse Response Processing of Transient Heat Transfer Gauge Signals”. In: *Journal of Turbomachinery* 130.2 2008. DOI: 10.1115/1.2752188.

[39] D L Schultz and T V Jones. “Heat Transfer Measurements in Short-duration Hypersonic Facilities”. In: *Agard Library* No.165 1973, pp. 5–8.

[40] SciPy 1.7.1. *SciPy*. Aug. 1, 2021. URL: `https://docs.scipy.org/doc/scipy/reference/`.

[41] Faisal Shaikh. “Unsteady phenomena and realistic geometry effects at the combustor-turbine interface of a large gas turbine”. DPhil thesis, University of Oxford, 2020.

[42] David R.H Gillespie and Peter T Ireland. “Intricate internal cooling systems for gas turbine blading”. DPhil thesis, University of Oxford, 1998.

[43] Roberto Maffulli. *Conjugate heat transfer in high pressure turbines*. 2016.

[44] W M Kays and M E Crawford. *Convective heat and mass transfer*. 2nd ed. McGraw-Hill series in mechanical engineering. New York ; London: McGraw-Hill, 1980.

[45] W. C. Reynolds, W. M. Kays, and S. J. Kline. “National Aeronautics Space Administration Reflection”. In: *NTRS - NASA Technical Reports Server* 1999011604.June 1959 1959. URL: `https://ntrs.nasa.gov/citations/19990116044`.

[46] J. E. Doorly and M. L.G. Oldfield. “The theory of advanced multi-layer thin film heat transfer gauges”. In: *International Journal of Heat and Mass Transfer* 30.6 1987. DOI: 10.1016/0017-9310(87)90045-7, pp. 1159–1168.

[47] Matthew Collins, Kam Chana, and Thomas Povey. “New technique for the fabrication of miniature thin film heat flux gauges”. In: *Measurement science and technology* 26.2 2015. ISSN:0957-0233.

[48] John Lekner. *Theory of reflection : reflection and transmission of electromagnetic, particle and acoustic waves.* 2nd edition. Springer series on atomic, optical, and plasma physics. ISBN:9783319236278 [electronic book]. Cham, 2016.

[49] Dong Ming Li. “Measurement of thermal conductivities of polymer films with the TC-1000 thermal comparator”. Thesis, Rochester Institute of Technology. 1990.

[50] American Society for Testing and Materials. *ASTM D1505: Standard Test Method for Density of Plastics by the Density-Gradient Technique.* 2012.

[51] Carlos E.S. Bernardes, Abhinav Joseph, and Manuel E.Minas da Piedade. “Some practical aspects of heat capacity determination by differential scanning calorimetry”. In: *Thermochimica Acta* 2020. DOI: 10.1016/j.tca.2020.178574.

[52] Gerald W Recktenwald. *Finite-Difference Approximations to the Heat Equation.* 2011. URL: `http://www.nada.kth.se/%7B~%7Djjalap/numme/FDheat.pdf`.

[53] Mark Holmes. *tridiag.m.* MathWorks File Exchange. 2013. URL: `https://uk.mathworks.com/matlabcentral/fileexchange/40722-tridiag-m?focused=3780252%7B%5C&%7Dtab=function`.

[54] R I Hickson et al. *Finite difference schemes for multilayer diffusion.* DOI: 10.1016/j.mcm.2011.02.003. 2011.

[55] S S Askar and A A Karawia. “On Solving Pentadiagonal Linear Systems via Transformations”. In: *Mathematical Problems in Engineering* 2015. Ed. by George S Dulikravich. DOI: 10.1155/2015/232456.

[56] K Benkert and R Fischer. “An efficient implementation of the thomas-algorithm for block penta-diagonal systems on vector computers”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics).* Vol. 4487. 2007, pp. 144–151.

[57] H S Carslaw and J C Jaeger. *Conduction of heat in solids.* Oxford: Clarendon Press, 1947.

[58] D. R. Buttsworth and T. V. Jones. *Radial conduction effects in transient heat transfer experiments.* 1997.

[59] Szymon Rusinkiewicz. “Estimating curvatures and their derivatives on triangle meshes”. In: *Proceedings - 2nd International Symposium on 3D Data Processing, Visualization, and Transmission. 3DPVT 2004.* DOI: 10.1109/TDPVT.2004.1335277. 2004.

[60] Yizhak Ben Shabat and Anath Fischer. “Design of porous micro-structures using curvature analysis for additive-manufacturing”. In: *Procedia CIRP.* DOI: 10.1016/j.procir.2015.01.057. 2015.

[61] Eric Johnson. *STL File Reader.* MathWorks File Exchange. 2011. URL: `https://uk.mathworks.com/matlabcentral/fileexchange/22409-stl-file-reader?focused=5193625%7B%5C&%7Dtab=function`.

[62] K. Crane. *Discrete Differential Geometry: An applied Introduction.* 3.4.1. Principal, Mean, and Gaussian Curvature. Feb. 25, 2021. URL: `https://www.cs.cmu.edu/~kmcrane/Projects/DDG/paper.pdf`.

[63] H. Jiang et al. “Analytical-Solution Based Corner Correction for Transient Thermal Measurement”. In: *Journal of Heat Transfer* 2015. DOI: 10.1115/1.4030980.

[64] Cláudia Mori and Estaner Romão. “Numerical Simulation by FDM of Unsteady Heat Transfer in Cylindrical Coordinates”. In: *Applied Mechanics and Materials* 851 2016, pp. 322–325.

[65] Piotr Duda. “Finite element method formulation in polar coordinates for transient heat conduction problems”. In: *Journal of Thermal Science* 2016. DOI: 10.1007/s11630-016-0850-2.

[66] Andrew Ng. *Machine Learning.* Stanford University. 2017. URL: `https://www.coursera.org/learn/machine-learning`.

[67] Skymind. *A Beginner’s Guide to LSTMs.* Aug. 20, 2018. URL: `https://skymind.ai/wiki/lstm%7B%5C#%7Dlong`.

[68] A. Verma. *Building A Neural Net from Scratch Using R Part 1.* July 20, 2020. URL: `https://rviews.rstudio.com/2020/07/20/shallow-neural-net-from-scratch-using-r-part-1/`.

[69] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *CoRR* 2014. arXiv: `1412.6980`.

[70] Sebastian Ruder. *An overview of gradient descent optimization algorithms.* Jan. 19, 2016. URL: `https://ruder.io/optimizing-gradient-descent/index.html#adam`.

[71] Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *International Conference on Learning Representations (ICRL)* 2015. arXiv: `1409.1556`.

[72] scikit-learn 1.0.1. *scikit-learn, Machine Learning in Python.* Oct. 1, 2021. URL: `https://scikit-learn.org/stable/`.

[73] Google AI. *Tensorflow.* 2018. URL: `https://www.tensorflow.org/`.

[74] Jakob Aungiers. *Installing TensorFlow GPU Natively on Windows 10.* 2017. URL: `http://www.jakob-aungiers.com/articles/a/Installing-TensorFlow-GPU-Natively-on-Windows-10`.

[75] Enzo Losi et al. “Anomaly Detection in Gas Turbine Time Series by Means of Bayesian Hierarchical Models”. In: *Journal of engineering for gas turbines and power* 141.11 2019. ISSN:0742-4795.

[76] Luca Mantelli et al. “A Degradation Diagnosis Method for Gas TurbineFuel Cell Hybrid Systems Using Bayesian Networks”. In: *Journal of engineering for gas turbines and power* 143.5 2021.

[77] Mohammad Sadeghi, Fereidoon Behnia, and Rouhollah Amiri. “Window Selection of the Savitzky-Golay Filters for Signal Recovery from Noisy Measurements”. In: *IEEE Transactions on Instrumentation and Measurement* 2020. DOI: 10.1109/TIM.2020.2966310.

[78] Ian T. Jollife and Jorge Cadima. “Principal component analysis: A review and recent developments”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 2016. DOI: 10.1098/rsta.2015.0202.

[79] General Electric Company. *GE Digital Twin - Analytic Engine for the Digital Power Plant.* 2016. URL: `https://www.ge.com/digital/sites/default/files/download_assets/Digital-Twin-for-the-digital-power-plant-.pdf`.

[80] D. T. Lee and B. J. Schachter. “Two algorithms for constructing a Delaunay triangulation”. In: *International Journal of Computer & Information Sciences* 1980. DOI: 10.1007/BF00977785.

[81] Mark A. Kramer. “Nonlinear principal component analysis using autoassociative neural networks”. In: *AIChE Journal* 1991. DOI: 10.1002/aic.690370209.

[82] Venkat Roy, Andrea Simonetto, and Geert Leus. “Spatio-temporal field estimation using kriged kalman filter (KKF) with sparsity-enforcing sensor placement”. In: *Sensors, Switzerland* 2018. DOI: 10.3390/s18061778.

[83] Margaret A. Oliver and R Webster. *Basic steps in geostatistics : the variogram and kriging.* SpringerBriefs in agriculture. ISBN:9783319158655 [electronic book]. Cham, Switzerland, 2015.

[84] PyKrige developers. *PyKrige, Kriging Toolkit for Python, revision f5b39204.* Oct. 1, 2021. URL: `https://geostat-framework.readthedocs.io/projects/pykrige/en/stable/`.

[85] Richard Hartley and Andrew Zisserman. “Multiple View Geometry”. In: *Computer Vision and Pattern Recognition, IEEE Computer Society.* 1999. URL: `https://www.robots.ox.ac.uk/~az/tutorials/tutoriala.pdf`.

[86] Blender Foundartion. *Blender.* 2018. URL: `https://www.blender.org/`.

[87] Herbert J. Bernstein and Lawrence C. Andrews. “Accelerating k-nearest-neighbor searches”. In: *Journal of Applied Crystallography* 2016. DOI: 10.1107/S1600576716011353.

[88] *OpenGlContext, Introduction to Shaders.* 2008. URL: `http://pyopengl.sourceforge.net/context/documentation.html`.

[89] John Vince. *Mathematics for Computer Graphics.* 5th ed. 2017. Undergraduate topics in computer science. ISBN:9781447173366 [electronic book]. London, 2017.

[90] Alexander V. Murray, Peter T. Ireland, and Eduardo Romero. “An experimentally validated low order model of the thermal response of double-wall effusion cooling systems for hp turbine blades”. In: *Proceedings of the ASME Turbo Expo* 2020. GT2020-14603.

[91] Hugo F.M Milan and Kifle G Gebremedhin. “Tetrahedral Node for Transmission-Line Modeling (TLM) applied to Bio-heat Transfer”. In: *Computers in biology and medicine* 79 2016, pp. 243–249.

[92] Nico Schlömer. *meshio: Tools for mesh files.* DOI: 10.5281/zenodo.1173115. 2021. URL: `https://github.com/nschloe/meshio`.

[93] Thomas H Cormen. *Introduction to algorithms [electronic resource].* 3rd ed. Ebook central. Cambridge, Mass.: MIT Press, 2009, pp. 658–659.

[94] Developers of Thermal Analysis Kit (TAK). *Thermal Network Modeling Handbook*. Westminster, CO 80021, USA, 2000. URL: `http://www.tak2000.com/data/handbookx.pdf`.

[95] Alex Holkner. *Pyglet*. 2021. URL: `https://pyglet.readthedocs.io/en/latest/`.

[96] ANSYS Inc. *ANSYS FLUENT 12.0/12.1 Documentation*. 2009. Chap. 4.12.2 Standard Wall Functions.

[97] Vincent Francois-Lavet et al. *An Introduction to Deep Reinforcement Learning*. Foundations and Trends in Machine Learning: Vol. 11, No. 3-4. 2018.

[98] DeepMind Technologies Limited. *DeepMind AI Reduces Google Data Centre Cooling Bill by 40%*. July 20, 2016. URL: `https://deepmind.com/blog/article/deepmind-ai-reduces-google-data-centre-cooling-bill-40`.

[99] OpenAI. *Gym*. 2018. URL: `https://gym.openai.com/`.

[100] Oriol Vinyals et al. “Grandmaster level in StarCraft II using multi-agent reinforcement learning”. In: *Nature* 2019. DOI: 10.1038/s41586-019-1724-z.

[101] David Silver et al. “Mastering the game of Go with deep neural networks and tree search”. In: *Nature* 2016. ISSN:14764687, DOI: 10.1038/nature16961.

[102] David Silver et al. “A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play”. In: *Science* 2018. ISSN:10959203, DOI: 10.1126/science.aar6404.

[103] Daniel Barley. “Enabling Flexible Operation of Steam Turbines”. MEng thesis, University of Oxford, 2020.

[104] Prasad Umavannan. “Flexible Operation of Conventional Power Plants”. MEng thesis, University of Oxford, 2021.