

**ANALYSIS OF WATER COOLING PROCESS OF STEEL STRIPS ON RUNOUT  
TABLE**

by  
Soheyl Vakili

B.Sc., Sharif University of Technology, 2003

M.Sc., Sharif University of Technology, 2006

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF  
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

THE FACULTY OF GRADUATE STUDIES

(Mechanical Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

September 2011

© Soheyl Vakili, 2011

## Abstract

This study engages in the thermal analysis of water jet cooling of a hot moving steel strip on a run-out table. General 3D FE programs are developed for the direct and inverse heat transfer analysis. Studies show that gradient-based inverse algorithms suffer from high sensitivity to measurement noise and instability in small time steps. These two shortcomings limit their application in modeling of the real problems.

Artificial neural network (ANN), genetic algorithm (GA), and particle swarm optimization (PSO) methods are applied to the inverse heat conduction problem in order to overcome the challenges faced by the gradient-based methods. Among them, GA and PSO are found to be effective. CRPSO, a variation of PSO, shows the best computational performance. However, compared to the gradient-based methods, these algorithms are very slow. Thus, a set of modifications were performed in this research to accelerate their convergence rate. Sequential formulation using the future time steps, multi-objective optimization, and inexact pre-evaluation using surrogate models are some of these modifications.

Inverse analysis of experimental data shows that heat transfer behavior on the plate is mainly a function of the surface temperature, and can be categorized into three zones: High, mid, and low temperature. The effects of jet line configuration, jet line spacing, and plate moving speed were studied. The most uniform distribution happens in the case of fully staggered configuration. In higher jet line distances, the interaction effects become less significant, and a more uniform distribution is observed. The plate speed affects the heat transfer rate under the impingement point for the higher surface temperatures. In the high entry temperatures, the impingement heat transfer rate is lower when the plate is moving at a higher velocity. The plate speed does not significantly change the heat transfer behavior in the parallel flow zone.

Finally, the results of the heat transfer analysis were coupled with the microstructure and structure fields, to study the thermal stresses and deflection occurring in the strips during the cooling process. It was found that fully-staggered jet configuration, larger spacing between jet lines, and lower plate speeds result in a less deformed steel strip.

## Preface

Some of the contents of chapter 3 have been published in:

- S. Vakili, and M. S. Gadala (2009) Effectiveness and Efficiency of Particle Swarm Optimization Technique in Inverse Heat Conduction Analysis, *Numerical Heat Transfer, Part B: Fundamentals*, vol. 56 (2) pp.119-141.
- S. Vakili, and M. S. Gadala (2011) Low Cost Surrogate Model Based Evolutionary Optimization Solvers for Inverse Heat Conduction Problem, *Submitted to International Journal of Heat and Mass Transfer*.
- S. Vakili, and M. S. Gadala (2011) A Modified Sequential Particle Swarm Optimization Algorithm with Future Time Data for Solving Transient Inverse Heat Conduction Problems, *Numerical Heat Transfer, Part A: Applications*, vol. 59 (12) pp.911-933.
- S. Vakili, and M.S. Gadala (2010) Low Cost Particle Swarm Algorithm Using Surrogate Model Based Pre-Evaluation for Inverse Heat Conduction Analysis. ASME 2010 International Mechanical Engineering Congress & Exposition, Vancouver, BC, Canada, November 12-18.
- S. Vakili, and M.S. Gadala (2008) Effectiveness of Particle Swarm Optimization Technique in Dealing with Noisy Data in Inverse Heat Conduction Analysis. International Simulation Multiconference, Istanbul, Turkey, July 13-16.
- S. Vakili, and M.S. Gadala (2008) Solving Inverse Heat Conduction Problems Using Particle Swarm Optimization Algorithm. The 16th Annual Conference of the CFD Society of Canada, Saskatoon, Saskatchewan, Canada, June 9 - 11.

Some of the contents of chapter 4 have been published in:

- S. Vakili, and M.S. Gadala (2011) Boiling Heat Transfer of Multiple Impinging Jets on a Hot Moving Plate, *Accepted for publication in Heat Transfer Engineering Journal*.

Some of the contents of chapter 5 have been published in:

- S. Vakili, and M.S. Gadala (2010), Finite Element Modeling of Thermal Stresses and Deformation in Steel Strips under Water Jet Impingement Cooling, *the 23rd Canadian Congress of Applied Mechanics*, Vancouver, BC, Canada, June 5-9.

For all the above manuscripts, I have been responsible for performing the various parts of the research, analysis of the research data, and preparation of the manuscripts. Dr. M.S. Gadala has identified and designed the research program, provided feedback about the outcomes, and have helped in the review of the manuscript.



# Table of Contents

Abstract .....	ii
Preface.....	iii
Table of Contents .....	v
List of Tables.....	ix
List of Figures .....	x
List of Symbols and Abbreviations .....	xiii
Acknowledgements .....	xvi
Dedication .....	xvii
1. Introduction .....	1
1.1 Motivations for Controlled Cooling of Steel.....	1
1.2 Research at UBC .....	3
1.3 The Need for Inverse Heat Conduction Solver .....	6
1.4 Inverse Heat Conduction Analysis Methods .....	8
1.5 Jet Impingement Heat Transfer .....	11
1.6 Thermal Stress and Deflection .....	17
1.7 Objectives of This Study .....	19
2. Three-dimensional Direct and Inverse Heat Conduction Analyses .....	21
2.1 Direct Three-Dimensional Heat Conduction Analysis.....	21
2.1.1 The Need for Three-Dimensional Modeling .....	21
2.1.2 Formulation .....	22
2.1.3 Numerical Modeling .....	23
2.1.4 Special Considerations .....	24
2.1.5 Implementation and Validation .....	24
2.2 Formulation of Inverse Analysis .....	27
2.2.1 Objective Function .....	27
2.2.2 Gradient Based Calculation of Heat Flux.....	31
2.2.3 Convergence Criteria.....	33
2.2.4 Flux Zoning Method (FZM).....	34
2.2.5 Flux Marching Method (FMM).....	34
2.2.6 Different Techniques for Calculating Sensitivities .....	36
2.3 Results and Discussion.....	38

2.3.1 Test Cases.....	38
2.3.2 Noisy Domains.....	44
2.3.3 Effects of the Regularization Parameter and Future Time Steps .....	45
2.3.4 Effect of Time Step Size .....	48
2.3.5 Comparison of 1D, 2D and 3D Algorithms .....	48
2.4 Conclusion.....	50
3. Gradient-Free Methods of Solving the Inverse Heat Conduction Problem .....	51
3.1. Inverse Formulation .....	51
3.2. Artificial Neural Networks.....	52
3.2.1. Feedforward Multilayer Perceptrons (FMLP).....	52
3.2.2. Radial Basis Function Networks (RBFN).....	53
3.2.3. Implementation in Inverse Heat Conduction Problem .....	54
3.4. Genetic Algorithm (GA) .....	55
3.5. Particle Swarm Optimization .....	57
3.5.1. Basic Concepts .....	57
3.5.2. Variations .....	62
3.5.3. Elite Particle and Elite Velocity .....	64
3.6. Application in Inverse Heat Conduction .....	64
3.7. Time Step Size .....	68
3.8. Efficiency .....	69
3.9. Sequential vs. Whole Domain .....	75
3.10. Noisy Domain Solution.....	76
3.11. Future Time Step Regularization .....	83
3.12. Effect of Non-linearity .....	90
3.13. Effect of Elite Members .....	91
3.14. Multi-Criteria Formulation Based on Spatial Location.....	93
3.15. Surrogate Modeling.....	95
3.15.1. Motivation.....	95
3.15.2. Procedure.....	96
3.15.3. Surrogate Model Types .....	96
3.15.4. Surrogate Modeling (SUMO) Toolbox .....	98
3.15.5. Model Management .....	98
3.15.6. Model Configuration.....	100

3.15.7. Results .....	100
3.15.8. Solution Time .....	101
3.15.9. Accuracy and Noise Resistance .....	102
3.15.10. Surrogate Model Management .....	104
3.16. Conclusion.....	107
4. Heat Transfer Behaviour under Multiple Circular Water Jets Impinging on a Moving Plate.....	110
4.1. Experimental Setup .....	110
4.2. Uncertainty Analysis .....	113
4.3. Time Histories of Temperature and Heat Flux.....	114
4.4. Heat Flux vs. Surface Temperature.....	120
4.5. Lateral Distribution of the Heat Flux .....	123
4.6. Lateral Distribution of the Surface Temperature .....	125
4.7. Longitudinal Distribution of the Heat Flux.....	127
4.8. Effect of the Plate Speed.....	130
4.9. Conclusion.....	132
5. Structural Effects of the Thermal Field during Water Jet Cooling .....	134
5.1. Phase Transformation Modeling .....	135
5.2. Thermal Stress Modeling .....	137
5.3. Computational Modeling.....	139
5.4. Validation Cases.....	140
5.4.1. Validation Case I.....	140
5.4.2. Validation Case II .....	141
5.5. Stationary Plate .....	143
5.6. Moving Plate .....	146
5.6.1. Single Passes .....	147
5.6.2. Multiple Passes.....	151
5.6.3. Effect of Plate Thickness.....	155
5.7. Conclusion.....	155
6. Conclusion and Future Work .....	157
6.1. Conclusion.....	157
6.1.1. 3D Finite Element Program for the Inverse Heat Conduction Analysis .....	157
6.1.2. Study of the Neural Network Algorithms for Inverse Heat Conduction Analysis.....	157
6.1.3. Study of the GA and PSO Algorithms for Inverse Heat Conduction Analysis.....	158

6.1.4. Improvements in the Efficiency and Effectiveness of Gradient-free Algorithms for IHC Analysis.....	159
6.1.5. Surrogate Model Based Acceleration of GA and PSO Solvers.....	159
6.1.6. Characterization of the Water Jet Impingement Cooling on a Moving Plate.....	160
6.1.7. Study of the Structural Effects of the Controlled Cooling of Steel Strips.....	161
6.2. Suggestions for Future Work .....	162
6.2.1. Experimental Study of Higher Plate Speeds, Different Flow Rates, and Water Temperatures .....	162
6.2.2. Correlations for Heat Transfer of Multiple Jets Impinging on a Hot Moving Plate .....	162
6.2.3. Coupling of Hydrodynamics and Heat Transfer Models .....	162
6.2.4. Experimental Measurements and Correlations for Thermal Stresses and Deflections .....	162
6.2.5. Developing Mechanistic Models for Moving Plate Jet Impingement Boiling.....	163
Bibliography.....	164
Appendix A. Definition of Terms in the Finite Element Heat Conduction Formulation.....	175

## List of Tables

Table 3.1. Comparison of the Typical Solution Time for Different Inverse Analysis Algorithms for the Whole Cooling Process on a Run-Out Table .....	70
Table 3.2. Performed Comparison Tests.....	70
Table 3.3. Calculated $t$ -Values for the Combination of 2 Solution Methods; $t_{critical} = 1.73$ .....	72
Table 3.4. Calculated U-Values for the Combination of 2 Solution Methods .....	72
Table 3.5. Number of Computational Units of Different Gradient-Free Solution Methods, Their Speedup with Respect to GA, and Their Relative Cost with Respect to the Most Efficient Method .....	74
Table 3.6. Time Required to Perform One Computational Unit for each Test Case .....	75
Table 3.7. Comparison of Efficiency for Whole Domain and Sequential Implementations.....	76
Table 3.8. Effect of Self-Confidence Parameter: $L_2$ Norm of the Error ( $W/m^2$ ) .....	81
Table 3.9. Effect of the Self-Confidence Parameter on the $L_2$ Norm of Error in the Solution .....	82
Table 3.10. The $L_2$ Norm of Error in the Solution in a Noisy Domain for Different Algorithms.....	83
Table 3.11. The $L_2$ Norm of Error in the Solution of Test Case I After Modifications in a Noisy Domain .....	89
Table 3.12. The $L_2$ Norm of Error in the Solution of Test Cases IV and V After Modifications in a Noisy Domain.....	89
Table 3.13. The $L_2$ Norm of Error in the Solution in an Exact Domain for Different Algorithms .....	91
Table 3.14. The Average Speedup in Each Algorithm After Using Elite Particles and Velocities .....	93
Table 3.15. Calculated U-Values to Study the Effect of Elite Particles on the Performance.....	93
Table 3.16. Required Time (in Seconds) to Construct the Model for the Linear Case .....	102
Table 3.17. Required Time (in Seconds) for 1000 New Predictions.....	102
Table 3.18. $R^2$ Values for Different Surrogate Models in the Exact and Noisy Domains for the Linear Case.....	103
Table 3.19. $R^2$ Values for Different Surrogate Models in the Exact and Noisy Domains for the Nonlinear Case.....	104
Table 3.20. Effect of Using Surrogate Models on the Total Solution Time (s) for the Linear Case.....	106
Table 3.21. Effect of Using Surrogate Models on the Total Solution Time (s) for the Nonlinear Case ...	107
Table 4.1. Test Matrix.....	112
Table A. 1. Definition of the Terms in the FE General Heat Conduction Equation .....	175
Table A. 2. Definition of the Terms in the Heat Transfer Equation.....	176

## List of Figures

Figure 1.1. A Typical Production Line of Steel Rolls [2].	2
Figure 1.2. Three Different Types of Cooling on a Run-out Table [12].	2
Figure 1.3. ROT Test Facility at UBC [24].	3
Figure 1.4. Intrinsic thermocouple junction	4
Figure 1.5. Schematic Arrangement of Thermocouples [1].	5
Figure 1.6. Schematic View of Test Plate Setup [26].	5
Figure 1.7. Temperature Profiles During the Water Jet Impingement [25].	7
Figure 1.8. Different Boiling Regions Present During Top Jet Impingement [14].	13
Figure 2.1. Contour Plot Showing Surface Heat Flux ( $W/m^2$ ) in a Staggered Nozzle Configuration [31].	21
Figure 2.2. Geometry and mesh for the validation of the direct 3D code	26
Figure 2.3. The results of validation for the 3D direct code	27
Figure 2.4. FZM Model for Inverse Calculation of the Boundary Heat Flux	34
Figure 2.5. Problem Geometry and Boundary Conditions, Test Case I	39
Figure 2.6. Test Case II; (a) Model, (b) Problem Description [1].	40
Figure 2.7. Temperature Distribution Inside the Slab	41
Figure 2.8. (a) The Whole Block Consisting of Nine Thermocouples; (b) A Close View of the TC Hole from the Bottom	42
Figure 2.9. (a) The Applied Heat Flux on the Top Surface; (b) The Thermocouple Readings Used for Inverse Analysis	43
Figure 2.10. Test Case V; A Block Consisting of Eighteen Thermocouple Zones	44
Figure 2.11. (a) The Applied Heat Flux on the Top Surface; (b) The Thermocouple Readings Used for Inverse Analysis; Second Line of Thermocouples, Test Case V	44
Figure 2.12. Effect of Regularization Parameter; 4 Future Steps; Error level = $\pm 3^\circ C$ ; (a) $\alpha = 1e-12$ ( $L_2$ Norm of Error = $4.63e5$ ); (b) $\alpha = 1e-11$ ( $L_2$ Norm of Error = $2.56e5$ ); (c) $\alpha = 1e-10$ ( $L_2$ Norm of Error = $1.18e5$ ).	46
Figure 2.13. Effect of Future Time Steps; $\alpha = 1e-11$ ; Error Level = $\pm 3^\circ C$ ; (a) 4 Steps ( $L_2$ Norm of Error = $4.63e5$ ); (b) 6 Steps ( $L_2$ Norm of Error = $1.72e5$ ); (c) 8 Steps ( $L_2$ Norm of Error = $1.34e5$ )	47
Figure 2.14. Threshold of Divergence in the Solution of Test Case II Due to Small Time Step Size	48
Figure 2.15. Results of the 1D, 2D, and 3D Inverse Algorithms in Predicting a Known Heat Flux Applied to a Direct 3D Problem	50
Figure 3.1. A Feedforward Network Topology	53
Figure 3.2. An RBF Network Topology	54
Figure 3.3. Flowchart of a General Implementation of Genetic Algorithm (GA)	56
Figure 3.4. Velocity and Position Updates in the Basic Particle Swarm Optimization Algorithm	61
Figure 3.5. Flowchart of the Basic Particle Swarm Optimization Procedure	62
Figure 3.6. Test Case II; Time History of Heat Fluxes in a Typical Run-Out Table Application; Expected Results (Squares) vs. the RBF Network Results (Line)	65
Figure 3.7. Individual Heat Flux Peaks vs. Time from a Typical Run-Out Table Application; Expected Results (Circles) vs. the RBF Network Results (Pluses)	66
Figure 3.8. PSO Results, Test case I: (a) Without Regularization ( $L_2$ Norm of Error = $0.127$ ),	67
Figure 3.9. Test Case III; (a) The Top Plane ( $y = 1$ ) Temperature Contours; Solid Lines: Expected Input; Dotted Lines: Output of the Inverse Analysis; (b) Temperature Distribution (Expected and Restored by PSO) on the Top Surface ( $y=1$ ) Along the $x=1.5$ Line ( $L_2$ Norm of Error = $31.66$ )	67

Figure 3.10. Heat Flux vs. Time: (a) Classical Approach, (b) PSO with the same computational expense, (c) Final results of PSO [112], (d) Final results of GA. ....	69
Figure 3.11. Individual Heat Flux Peaks vs. Time from a Typical Run-Out Table Application; Expected Results (Circles) vs. the RBF Network Results (Pluses); Artificial Noise Added: $c = \pm 0.1\%$ .....	77
Figure 3.12. Individual Heat Flux Peaks vs. Time from a Typical Run-Out Table Application; Expected Results (Circles) vs. the RBF Network Results (Pluses); Artificial Noise Added: $c = \pm 0.3\%$ .....	78
Figure 3.13. Individual Heat Flux Peaks vs. Time from a Typical Run-Out Table Application; Expected Results (Circles) vs. the RBF Network Results (Pluses); Artificial Noise Added: $c = \pm 0.5\%$ .....	78
Figure 3.14. Individual Heat Flux Peaks vs. Time from a Typical Run-Out Table Application; Expected Results (Circles) vs. the RBF Network Results (Pluses); Artificial Noise Added: $c = \pm 1\%$ .....	79
Figure 3.15. Test Case II; Effect of Regularization Parameter (1% added noise); a: $\alpha = 10^{-12}$ ; b: $\alpha = 10^{-11}$ ; c: $\alpha = 10^{-10}$ .....	80
Figure 3.16. Test Case II; Effect of Self-Confidence Parameter (1% added noise); (a) $c_0=0.5$ ; (b) $c_0=0.6$ ; (c) $c_0=1.0$ ; (d) $c_0=1.2$ .....	81
Figure 3.17. PSO Behavior in Handling Noisy Data in Test Case I; (a) No Regularization; (b) Addition of Regularization Term ( $\alpha = 10^{-10}$ ); (c) Tuning of Self-Confidence Parameter ( $c_0 = 1.2$ ); (d) Introduction of Future Time Step Data (10 time steps).....	86
Figure 3.18. PSO Behavior in Handling Noisy Data for Test Case IV; (a) No Regularization; (b) Addition of Regularization Term ( $\alpha = 10^{-10}$ ); (c) Tuning of Self-Confidence Parameter ( $c_0 = 1.2$ ); (d) Introduction of Future Time Step Data (10 time steps) .....	87
Figure 3.19. PSO Behavior in Handling Noisy Data for the 2nd Set of Heat Fluxes in Test Case V; (a) No Regularization; (b) Addition of Regularization Term ( $\alpha = 10^{-10}$ ); (c) Tuning of Self-Confidence Parameter ( $c_0 = 1.2$ ); (d) Introduction of Future Time Step Data (10 time steps).....	88
Figure 3.20. Convergence of Scalar vs Vector Objective Function Formulations; (a) Test Case IV; (b) Test Case V .....	95
Figure 3.21. Results of Surrogate Based PSO for (a) Test Case I; (b) Test Case IV .....	101
Figure 3.22. Effect of Surrogate Model Management on the Performance of Inverse Analyzer; (a) GA Solver; (b) PSO Solver.....	106
Figure 4.1. A Sample Plate, Showing the Dimensions and Thermocouple Locations.....	110
Figure 4.2. Nozzle Configuration with Respect to Thermocouples; (a) In-line; (b) Half-staggered; (c) Full-staggered [31]. ....	111
Figure 4.3. History of (a) Thermocouple Readings; (b) Surface Temperature; (c) Surface Heat Flux; High Entry Temperature .....	117
Figure 4.4. History of (a) Thermocouple Readings; (b) Surface Temperature; (c) Surface Heat Flux; Mid Entry Temperature .....	118
Figure 4.5. History of (a) Thermocouple Readings; (b) Surface Temperature; (c) Surface Heat Flux; Low Entry Temperature .....	119
Figure 4.6. Heat Flux and Heat Transfer Coefficient vs. Surface Temperature Across the Width of the Plate for Different Entry Temperature Ranges.....	122
Figure 4.7. Lateral Heat Flux Distribution at the Moment of Impingement of the First Jet Line; (a) No Stagger; (b) Half Stagger; (c) Full Stagger. ....	124
Figure 4.8. Lateral Temperature Distribution After the Impingement of the Second Jet Line; (a) No Stagger; (b) Half Stagger; (c) Full Stagger. ....	126

Figure 4.9. Longitudinal Distribution of Heat Flux for the High Entry Temperature; (a) No Stagger, Jet Line Spacing: 25.4 cm; (b) Half Stagger, Jet Line Spacing: 25.4 cm; (c) Full Stagger, Jet Line Spacing: 25.4 cm; (d) Half Stagger, Jet Line Spacing: 50.8 cm .....	128
Figure 4.10. Longitudinal distribution of heat flux for the mid entry temperature; (a) No stagger, jet line spacing: 25.4 cm; (b) half stagger, jet line spacing: 25.4 cm; (c) full stagger, jet line spacing: 25.4 cm; (d) half stagger, jet line spacing: 50.8 cm .....	129
Figure 4.11. Longitudinal Distribution of Heat flux for the Low Entry Temperature; (a) No Stagger, Jet Line Spacing: 25.4 cm; (b) Half Stagger, Jet Line Spacing: 25.4 cm; (c) Full Stagger, Jet Line Spacing: 25.4 cm; (d) Half Stagger, Jet Line Spacing: 50.8 cm .....	130
Figure 4.12. (a) and (b) Impingement Heat Transfer Versus Entry Temperature; Jet Line Spacings of 25.4 and 50.8 cm, Respectively; (c) and (d) Jet Interaction Heat Transfer Versus Entry Temperature; Jet Line Spacings of 25.4 and 50.8 cm, Respectively; Solid Lines: No Stagger; Dashed Lines: Half Stagger; Dotted Lines: Full Staggered .....	132
Figure 5.1. Different Kinds of Coupling between the Three Physical Fields in Cooling of Steel Strips..	134
Figure 5.2. Schematic Representation of Cooling on the Run-out Table (Temperature vs. Time).....	136
Figure 5.3. Problem Dimensions and Temperature History in Validation Case I by Yoshida [86].....	140
Figure 5.4. Validation Case I; Comparison between the Current Results and the Experimental and Numerical Results in the Literature [86].....	141
Figure 5.5. Geometry of Validation Case II [89] .....	141
Figure 5.6. Temperature Distribution in Validation Case II in (a) Transverse and (b) Longitudinal Directions [89] .....	142
Figure 5.7. Validation Case II; Comparison between the Current Results and the Numerical Results in the Literature [90] .....	142
Figure 5.8: Schematic Arrangement of Thermocouples under Impinging Jet .....	143
Figure 5.9: Top Surface Heat Flux History Obtained By Inverse Analysis of Temperature Readings at Thermocouple Locations.....	144
Figure 5.10: Plate Geometry; Only the Shaded Area is Modeled .....	144
Figure 5.11: Temperature Distribution in the Plate a Few Seconds after the Start of the Cooling Process .....	145
Figure 5.12: Contours of (a) Z Displacement (Out of Plane) in Millimeters and (b) von Mises Stress in Pascals at the End of the Cooling Process.....	145
Figure 5.13. Maximum Out of Plane Displacement vs. Strip Thickness for the Stationary Plate .....	145
Figure 5.14. The Deformed and the Edge of the Undeformed Strip in a Moving Case.....	146
Figure 5.15. Single Pass Results for High Temperature Range Cooling .....	148
Figure 5.16. Single Pass Results for Mid Temperature Range Cooling.....	149
Figure 5.17. Single Pass Results for Low Temperature Range Cooling.....	150
Figure 5.18. Results for a Plate Cooled Down to 500 °C.....	152
Figure 5.19. Results for a Plate Cooled Down to 300 °C.....	153
Figure 5.20. Results for a Plate Cooled Down to 100 °C.....	154
Figure 5.21. Maximum Out of Plane Displacement vs. Strip Thickness for the Moving Plate .....	155



## List of Symbols and Abbreviations

<b>C</b>	Heat capacity matrix
$c_0$	Self-confidence parameter in particle swarm method; Atomic fraction
$c_1, c_2, c_3, c_4$	Acceleration coefficients in particle swarm method
$c_p$	Specific heat, J/kg.°C
<b>D</b>	Jet diameter
$d_{\text{eff}}^y$	Effective austenite grain size
<b>g</b>	Best global solution; Gravitational acceleration, m/s <sup>2</sup>
<b>h</b>	Heat transfer coefficient, W/m.°C; Step size in Taylor series expansion
<b>H</b>	Vertical distance between nozzle exit to the plate; Latent heat of phase change
<b>i</b>	Time step index
<b>I</b>	Improvement in a predicted value in surrogate models
<b>J</b>	Number of spatial components of heat flux
<b>k</b>	Conductivity, W/m.°C
<b>K</b>	Heat conduction matrix
<b>K</b>	Constriction factor
<b>L</b>	Number of measurement points
<b>N</b>	Total number of time steps
<b>n</b>	Iteration step
$N_E$	Number of function evaluations
<b>p</b>	Best solution of a particle
<b>P</b>	Pressure
<b>Q</b>	Flux load vector, J
<b>q</b>	Heat flux, W/m <sup>2</sup>
<b>q</b>	Heat flux vector, W/m <sup>2</sup>
$q^b$	Heat generation per unit mass, W/kg
<b>r</b>	Normally distributed random vector; Number of future time steps
$R^2$	Deviation in the values, as defined by equation (3.26)
<b>S</b>	Total boundary domain
<b>T</b>	Temperature, °C
<b>T</b>	Temperature vector, °C

$t$	Time, s; Student t-test value
$T_s$	Surface temperature, °C; Transformation start temperature
$U$	Mann-Whitney U value
$v$	Velocity of particles in particle swarm method; Center of receptive field unit in radial basis functions
$V$	Water velocity
$X$	Total sensitivity matrix
$x$	Position (value) of a particle in particle swarm method
$X$	Volumetric rate of phase transformation
$x, y, z$	Cartesian coordinates, m
$Y$	Measured temperature matrix

### **Greek Letters**

$\alpha$	Regularization parameter; Thermal expansion coefficient
$\beta$	Polynomial coefficients in surrogate modeling
$\Delta$	Change in a quantity
$\delta$	Norm of measurement errors
$\varepsilon$	Emissivity; Random error in measurements; Strain
$\rho$	Density, kg/m <sup>3</sup>
$\sigma$	Stefan Boltzmann constant; Standard deviation; Stress
$\psi$	Radial basis function
$\omega$	Weights in radial basis function surrogate model

### **Abbreviations**

ANN	Artificial neural networks
CRPSO	Complete repulsive particle swarm optimization
FE	Finite Element
FF	Feedforward
FMLP	Feedforward multi-layer perceptron
FMM	Flux marching method
FTS	Future time step

FZM	Flux zoning method
GA	Genetic algorithm
HTC	Heat transfer coefficient
IHCP	Inverse heat conduction problem
JMAK	Johnson-Mehl-Avrami-Kolmogorov (phase transformation model)
NN	Neural networks
PSO	Particle swarm optimization
RBF	Radial basis function
ROT	Run-out table
RPSO	Repulsive particle swarm optimization
TC	Thermocouple
TTT	Temperature-time transformation
1D	One dimensional
2D	Two dimensional
3D	Three dimensional

### **Superscripts**

calc	calculated
meas	measured
$\alpha$	Ferrite phase
$\gamma$	Austenite phase

### **Subscripts**

a	Atmospheric
j	Index for space; Jet impingement
n	Nozzle exit
s	Stagnation

## **Acknowledgements**

I offer my enduring gratitude to my supervisor, Dr. Mohamed S. Gadala, for his valuable insight and feedback, as well as his professional and personal support.

Many thanks to Dr. Fuchang Xu for his help in the first steps of this research, to Dr. Fateh Fazeli for his input on the microstructural modeling in chapter 5, to Hossein Mansour for his help in generating the computational grid for simulations, to Geoffrey Franco and Gary Lockhart for their roles in providing the experimental results, and to Dr. Matthias Militzer for coordinating the research in the UBC ROT research group.

And of course, special thanks to my dear family and friends. Without their support and guidance, I would be lost.

To my Mom,

who dragged me back into the classroom after I skipped school in the first grade;

and to my Dad,

who coerced me into memorizing the multiplication table in the third grade.

# **1. Introduction**

## **1.1 Motivations for Controlled Cooling of Steel**

Steel is one of the most widely used materials in today's industries. The development of advanced high strength steels (AHSS) is very crucial in many different aspects of industry. Steel strips are used in various applications ranging from automobile bodies to drink cans. Roughly, 25 to 30% of this production includes the hot rolling process [1]. Hot rolled steel sheets can be used for piping and tubing, automotive parts, rail cars, and many more applications. The above factors stimulate the research and need for more detailed study on the procedure of steel production and how to control the resulting mechanical properties.

The microstructure and mechanical properties of steel strips can be highly modified by the controlled cooling on the runout table. A finer grain can be achieved by using this process, which in turn results in an increase in strength, an improved notch toughness and resistance to brittle fracture, and contributes to a significant reduction in carbon content, while still maintaining the same strength level. The lower carbon content improves formability and contributes to superior notch toughness and weldability. In short, controlled cooling process will assure a balanced combination of mechanical properties. Controlled cooling can also eliminate the troublesome differences in strength among samples taken from the head, body and tail of a steel coil.

Detailed study of the effect of cooling profile on the properties of steel began in the early sixties, mainly in the Swinden Laboratories in England [2]. Reference [3] is an early publication that describes the advantages of this kind of treatment. Recognizing the major influence of "time-temperature cycle during cooling" on the grain properties, a research program at J&L steel company started in 1961, in order to design an optimum cooling curve [2]. Since then there has been a great amount of effort to study and design the cooling process [1,4-11].

In a typical production line, the slabs are reheated in a furnace to a hot rolling temperature close to 850 °C, then rolled through roughing and finishing mills, and then they are cooled down to a coiling temperature of 550 – 650 °C while rolling on a so-called runout table (ROT). Figure 1.1 shows a schematic view of such a configuration in a steel mill.

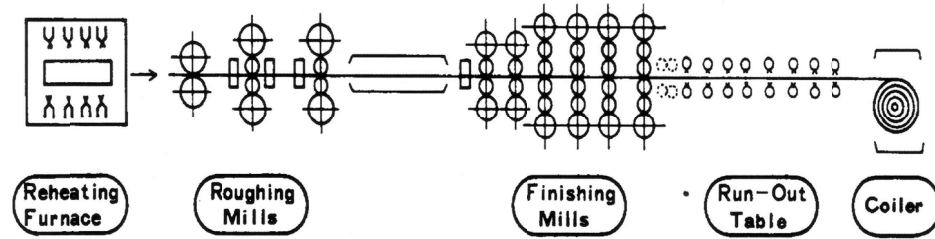


Figure 1.1. A Typical Production Line of Steel Rolls [2].

There are three main types of cooling systems used on the runout table: water spray, laminar flow (circular jet) and water curtain, as shown in Figure 1.2 [12]. Blazevic [5] performed a comparison of the performance of aspired sprays, sprays, and planar and circular jets by using the same volume of water. However, he compared the jets using different conditions, such as areas of cooling, impact time, and surface temperature, which provided some conflicting results. In summary, not enough research has been completed yet on the advantages of each cooling system. Nowadays a combination of these methods is used in steel industries. Relative position of each mechanism and the amount of cooling in each of them vary from one steel mill to another, and is mainly based on their own trial and error experiments.

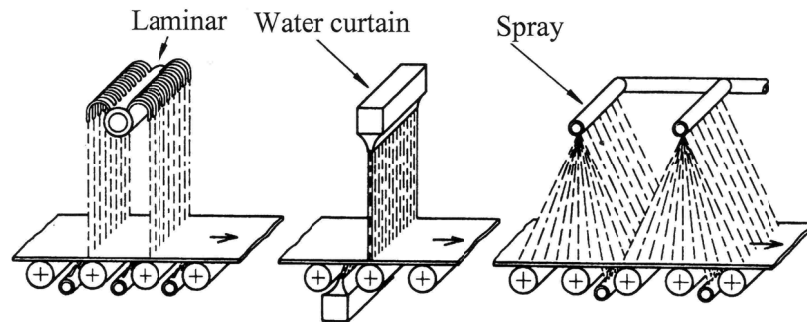


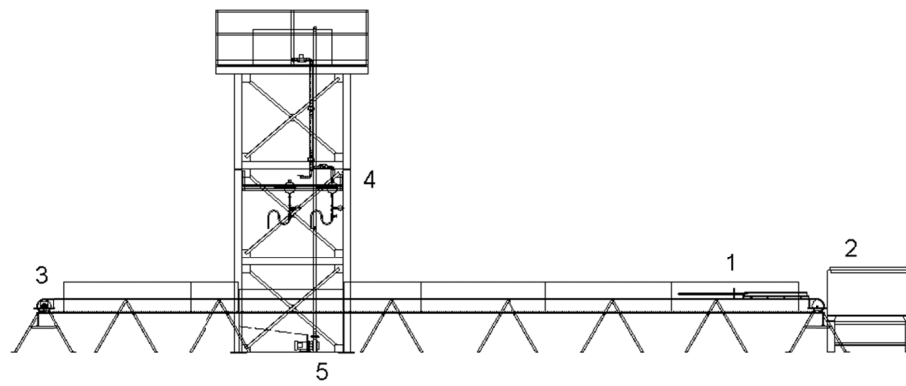
Figure 1.2. Three Different Types of Cooling on a Run-out Table [12].

Although accelerated cooling is already widely adopted by steel manufacturers, there is still a lack of fundamental understanding of the process, which limits the ability of accurately predicting the temperature in the metal during cooling, and hinders the appropriate design of the process. The complications in the process arise from the presence of a huge number of coupled sub processes, the lack of real-life experimental measurements, as well as from the issues regarding the large dimensions and temperatures of the industrial applications. Several

researchers have studied the water jet cooling process experimentally and/or numerically [13-23]. However, there is still a need to study many important aspects of controlled cooling on the runout table.

Proper control of roll and strip cooling process requires achieving several goals. Following a desired time – temperature profile, attaining the desired finishing and coiling temperatures at the end, minimizing temperature gradients in the strip in all directions, and minimizing the coolant consumption are some of the main objectives.

## 1.2 Research at UBC



**Figure 1.3. ROT Test Facility at UBC [24].**

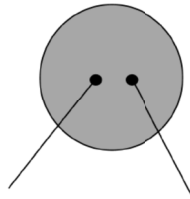
By recognizing the need to further investigate the water jet impingement cooling of steel, a ROT facility with industrial scale has been constructed at the University of British Columbia in the Advanced Materials Processing Lab (AMPL). It consists of two water banks and two headers, one at the top and the other at the bottom, a water pump, pipe circuits, flow control valves, a furnace, and devices for heating water and measuring water temperature. The nozzles can be set up onto the top header and can be easily changed if necessary. Nozzle dimensions are comparable with those used in steel industry. The distance between the top nozzles is adjustable from 50 to 90 mm. The vertical distance between the top nozzle exit and the test plate can be adjusted from 0.6 to 2 m. Only one nozzle is available on the bottom header whose distance to the plate is fixed. The alignment angle of the bottom nozzle to the plate may be changed.

The maximum water flow capacity is 138 l/min. The water can be heated up to 90 °C. The above listed parameters are very close to those actually used in steel industry. The initial

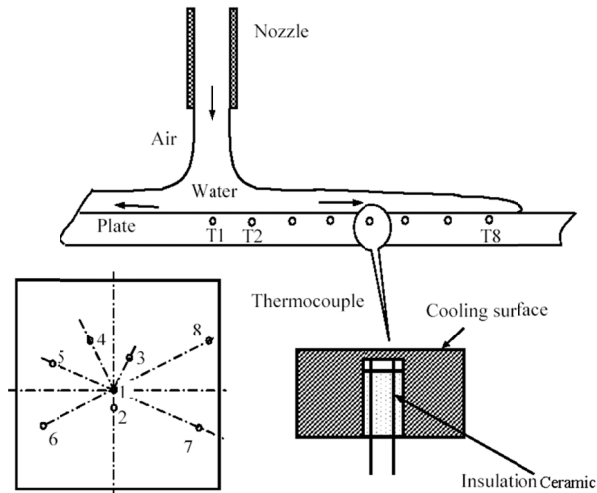


temperatures of the test plates are roughly 700 – 900 °C (the furnace is capable of heating samples up to 1200 °C), to make it more similar to an actual steel mill.

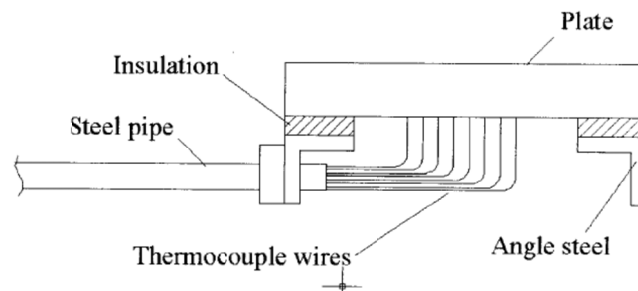
Due to the considerable amount of steam generated during these experiments, as a result of the quick evaporation of water on the hot surface of steel, methods such as infrared measurements that highly rely on a visual inspection of the plate are not applicable. Also, most of these methods normally provide us with a snapshot of the thermal field, rather than a continuous measurement that are required for this research. Thus, thermocouples (TCs) of type K are used in all experiments. These TCs would normally have an error of approximately 0.75% of the largest temperature, when used at a temperature higher than 277 °C. At each sensor location, an internal TC is installed in a blind hole with a diameter of 1.6 mm that is drilled from the plate's bottom surface. The measuring junction is fixed onto the end surface of the hole that is about 1 mm below the plate's top surface. In order to reduce the response time of the thermocouples, the individual wires are spot welded directly to the surface of the metal, as seen in Figure 1.4. This type of junction, called intrinsic junction, can greatly reduce the thermal inertia of junction and hence decrease the response time of TC close to zero [25]. Figure 1.5 shows a schematic arrangement of the thermocouples in the plate, and Figure 1.6 shows a schematic view of the test plate, including the thermocouple wires and the mounting device.



**Figure 1.4. Intrinsic thermocouple junction**



**Figure 1.5. Schematic Arrangement of Thermocouples [1].**



**Figure 1.6. Schematic View of Test Plate Setup [26].**

The establishment of the industrial scale runout table facility in UBC has facilitated intensive research in this area for both stationery and moving plates [1,8,9,27-32]. The research done at UBC may be categorized as:

- Studying and modeling the evolution of austenite microstructure and the optimum “conditioning” of austenite before transformation of optimum grain refinement.
- Studying the thermal behaviour of stationary and moving steel plate, cooled by circular jets under cooling conditions similar to those found in industry.
- Studying and modeling the austenite decomposition of steel under complex cooling conditions.

- Establishing quantitative correlations between chemistry, microstructure, and property as a guide for steel grade (alloy) design and steel processing.
- Developing new methods for inverse heat conduction problem (IHCP) to model the moving heat flux front on the cooled surface.
- Investigation of the errors included in surface temperature measurements due to the presence of thermocouples and possible ways to remedy the problem.
- Developing a special 2D program to model the runout table applications with direct and inverse heat transfer analysis capabilities.
- Conducting experiments on the heat transfer of a single jet or multiple jets impinging on stationary and moving plates.
- Studying the effect of the flow rate and inclination angle of a bottom jet.
- Conducting experiments and numerical simulation of the jet impingement hydrodynamics.

### **1.3 The Need for Inverse Heat Conduction Solver**

Solving the original Navier-Stokes and energy equations directly, in order to have a proper estimate of the heat flux to the model is quite difficult, and requires a lot of simplifications. Instabilities in the vapour / liquid interface are very common due to the larger density of the liquid that overrides the vapour. Turbulence will be present and its modeling is extremely challenging. Heat generation inside the strip due to a phase change in steel is another factor that complicates the study of these problems. Water moving speed is usually approximated with very simplified models and correlations. The different physical fields, i.e. hydrodynamics, heat transfer, microstructural, etc. are highly coupled. Boiling heat transfer is also strongly affected by factors such as surface roughness that are very difficult to implement in the current numerical simulation models. While these mechanistic models [33,34] provide a valuable insight into the physics of the jet impingement boiling, there is still a need to study the jet impingement boiling process in larger scales and using the experimental data. Experimental measurements that may be used to adjust and verify the numerical models are also quite challenging. As shown in Figure 1.7, recent studies in the group [25] show that errors in the thermocouple measurement of the top plate surface may exceed 200 °C. As indicated in Reference [25], there are several parameters that contribute to these significant errors. One of these factors is the thermal mass of

the thermocouples and wires on the plate surface, which causes some change in the thermal field due to the heat conduction through the wires. Also, the presence of wires may break the insulating vapour layer on the surface and create a contact between the liquid water and the plate surface. More importantly, since the thermocouples work through creating an electric current between their two junctions, and water is a conductive medium for electricity, there will be some distortion in the electric current and thus, some error in the measurements. Thus, instead of measuring the surface temperatures using thermocouples, it is more accurate to use the readings of thermocouples inside a plate, and use numerical simulations to find the boundary conditions on the surface.

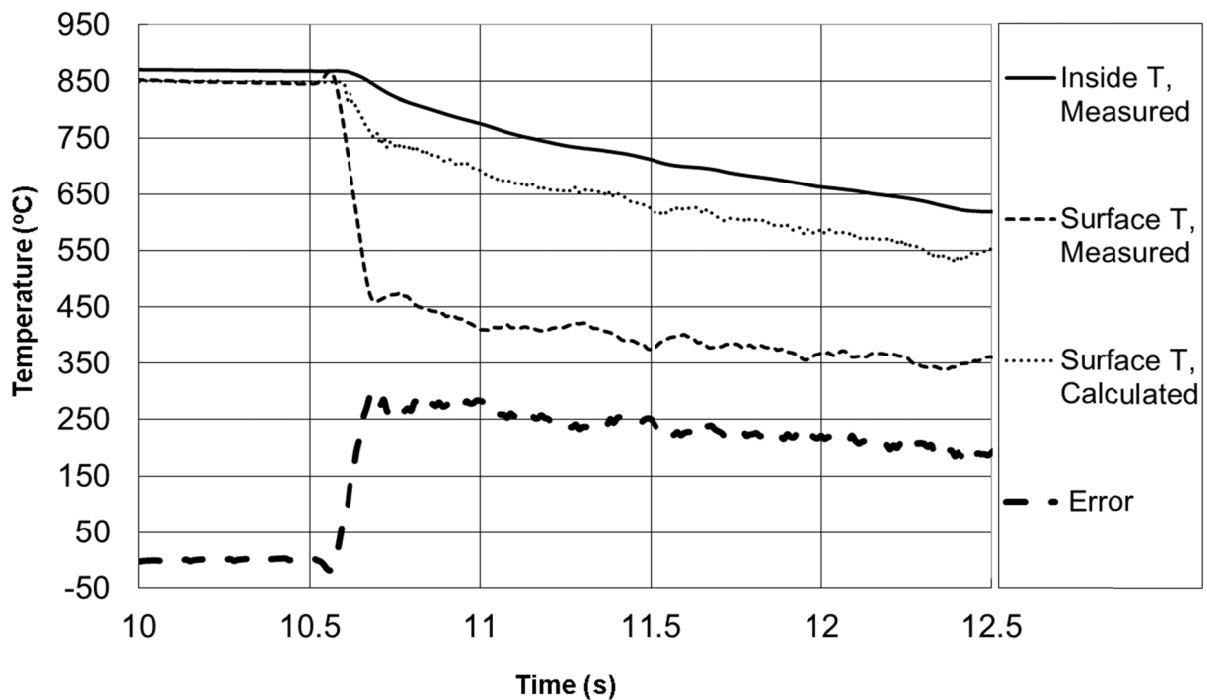


Figure 1.7. Temperature Profiles During the Water Jet Impingement [25].

An approach that may overcome some of the above problems is to use the results of industry-scale experiments and only utilize inner sub-surface temperature measurements in the simulations. This approach requires inverse heat transfer conduction (IHTC) analysis to find the surface temperatures and/or heat fluxes that would be used in consequent simulations.

## 1.4 Inverse Heat Conduction Analysis Methods

In an inverse heat conduction problem, the boundary conditions, initial conditions, or thermophysical properties of material are not fully specified, and they are determined from measured internal temperature profiles. The main problem is that the effect of changes in boundary conditions are normally damped or lagged, i.e. the varying magnitude of the interior temperature profile lags behind the changes in boundary conditions and is generally of lesser magnitude. Therefore, such a problem would be a typically ill-posed problem and would normally be sensitive to the measurement errors. Thus, in general, the uniqueness and stability of the solution are not guaranteed [35-37]. Inverse problems can be formulated either as parameter estimation or function estimation problems.

Inverse heat conduction problems, like most of the inverse problems encountered in science and engineering may be reformulated as an optimization problem. Therefore, many available techniques of solving the optimization problems are available as methods of solving the inverse problems. However, the corresponding objective function of the inverse problems can be highly nonlinear or non-monotonic, may have a very complex form, or in many practical applications, its analytical expression may be unknown. The objective function usually involves the squared difference between measured and estimated unknown variables. If  $\mathbf{Y}$  and  $\mathbf{T}$  are the vectors of the measured and estimated temperatures, then the objective function will be in the form of

$$U = [\mathbf{Y} - \mathbf{T}]^T [\mathbf{Y} - \mathbf{T}] \quad (1.1)$$

However, normally there is need for another term, called “regularization” in order to eliminate the oscillations in the results and make the solution more stable. The effect of this term and the strategy of choosing it will be discussed in details in the subsequent chapters.

The above equation is only valid, if the measured temperatures and the associated errors have the following statistical characteristics [38]:

- The errors are additive, i.e.

$$Y_i = T_i + \varepsilon_i \quad (1.2)$$

where  $\varepsilon_i$  is the random error associated with the  $i^{\text{th}}$  measurement.

- The temperature errors have zero mean.
- The errors have constant variance.

- The errors associated with different measurements are uncorrelated.
- The measurement errors have a normal (Gaussian) distribution.
- The statistical parameters describing the errors, such as their variance, are known.
- Measured temperatures are the only variables that contain measurement errors. Measured time, positions, dimensions, and all other quantities are all accurately known.
- There is no more prior information regarding the quantities to be estimated. If such information is available, it should be utilized to improve the estimates.

While classical methods, such as the least square regularization method [35,39], the sequential function specification method [36,39,40], the space marching method [41], conjugate gradient method [42,43], steepest descent method [44], online input estimation [45,46] and the model reduction model [47-49] are vastly studied in the literature, and applied to the problems in thermal engineering [50-56], including water cooling on run-out table [57], there are still some unsolved problems:

- The solution often shows some kinds of overshoot and undershoot, which may result in non-physical answers.
- Very high heat flux peak values such as those experienced in jet impingement cooling are normally damped and considerably underestimated.
- Results are very sensitive to the quality of input. Measurement errors are intrinsic in laboratory experiments, so we need a more robust approach in solving the inverse problem.
- The time step size that can be used with these methods is bounded from below, and cannot be less than a specific limit [35]. This causes temporal resolutions that are not sufficient for some real world applications, where changes happen very fast, such as a hot steel strip traveling at a high speed on a run-out table.

State of the art and more recent optimization techniques may be used in the solution of the IHTC problem to aid in stability, solution time, and to help in achieving global minimum solutions. Some of these recent techniques are briefly reviewed in the following section:

- ***Genetic Algorithm:*** This technique has been widely adopted to solve inverse problems [58-60]. Genetic algorithms (GAs) belong to the family of computational techniques originally inspired by the living nature. They perform random search optimization algorithms to find the global optimum to a given problem. The main advantage of GAs

may not necessarily be their computational efficiency, but their robustness, i.e. the search process may take much longer than the conventional gradient-based algorithms, but the resulting solution is usually the global optimum. Also, they can converge to the solution when other classical methods become unstable or diverge. However, this process can be time consuming since it needs to search through a large tree of possible solutions. Luckily, they are inherently parallel algorithms, and can be easily implemented on parallel structures. Another remedy is to use them in conjunction with other classical methods to overcome the disadvantages of both methods.

There are mainly two approaches that may be used to overcome the difficulties of GA. One approach is to use the solution of classical approach with large time steps as an initial guess to GA. Since GAs do not require any information about the derivatives, they can be used with smaller time steps. This can also improve the quality of the results in terms of reducing the undershoots and overshoots, and reduce the dispersion and diffusion of the peak values. Another approach is to use the solution of classical approach as the initial guess, then using some previous knowledge of the final answer as a way to find the problematic parts of the curve, then using GAs to modify the values of those sections.

- **Neural Networks:** Artificial neural networks can be successfully applied in the solution of inverse heat conduction problems [61-63]. They are capable of dealing with significant non-linearities and are known to be effective in damping the measurement errors.
- **Self-learning finite elements:** This methodology combines neural network with a nonlinear finite element program in an algorithm which uses very basic conductivity measurements to produce a constitutive model of the material under study. Through manipulating a series of neural network embedded finite element analyses, an accurate constitutive model for a highly nonlinear material can be evolved [64,65]. It is also shown to exhibit a great stability when dealing with noisy data.
- **Maximum entropy method:** This method seeks the solution that maximizes the entropy functional under given temperature measurements. It converts the inverse problem to a non-linear constrained optimization problem. The constraint is the statistical consistency between the measured and estimated temperatures. It can guarantee the uniqueness of the

solution. When there is no error in the measurements, maximum entropy method can find a solution with no deterministic error [66].

- ***Proper Orthogonal Decomposition:*** Here, the idea is to expand the direct problem solution into a sequence of orthonormal basis vectors, describing the most essential features of spatial and temporal variation of the temperature field. This can result in the filtration of the noise in the field under study [67].
- ***Particle Swarm Optimization (PSO):*** This is a population based stochastic optimization technique developed by Eberhart and Kennedy in 1995 [68], inspired by social behavior of bird flocking or fish schooling. Like GA, the system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. Compared to GA, the advantages of PSO are the ease of implementation and that there are few parameters to adjust. Some researchers showed that it requires less computational expense when compared to GA for the same level of accuracy in finding the global minimum [69]. PSO has been successfully applied in many areas, e.g. function optimization, artificial neural network training, fuzzy system control, and other areas where GA can be applied. It has also been successfully employed by many researchers in solving inverse heat transfer problem, mainly inverse radiation [70].

In this research, we will study the genetic algorithm, neural network, and particle swarm optimization techniques in more details. We will investigate their strengths and weaknesses, and try to modify them in order to increase their efficiency, i.e. to lower their computational cost, and to increase their effectiveness, i.e. to lower their sensitivity to the measurement errors, in solving inverse heat conduction problems.

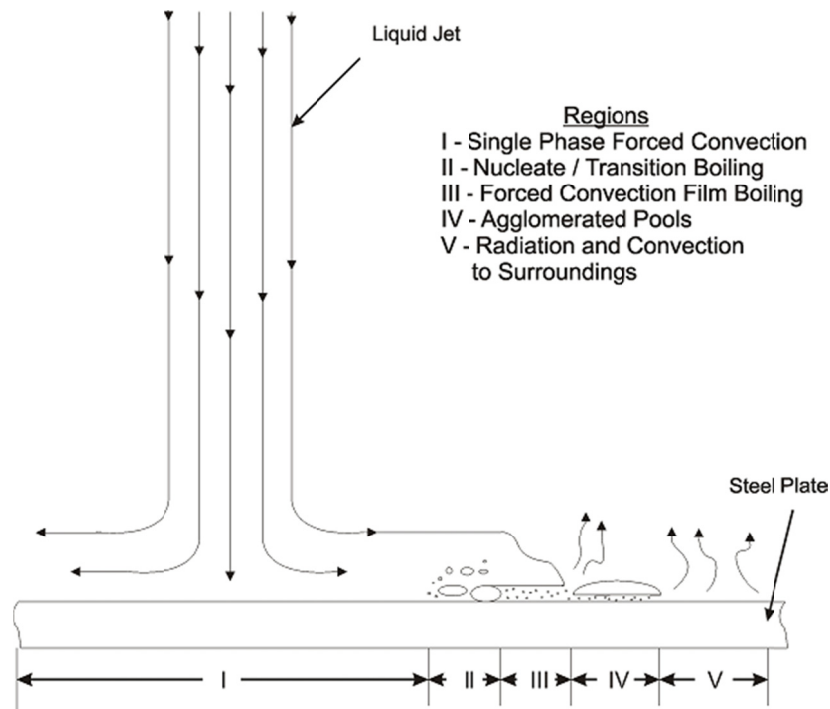
## **1.5 Jet Impingement Heat Transfer**

Jet impingement cooling has been one of the most active areas of research in heat transfer for many years. However, due to the complex nature of the involved phenomena, and the ever increasing applications of this technique, it continues to attract a considerable amount of attention. The major industrial applications of jet impingement are cooling of electronic components and controlled cooling of metals after being subjected to hot processes. In each of



these applications, a large number of sub-processes are involved, and each of them is very complex in nature. Furthermore, in the case of controlled cooling of metals, the hot surface is normally moving, and there are multiple jets in various configurations impinging on its surface. Between different kinds of jets, round jets especially benefit from a higher specific cooling performance (ability of removing more heat per unit volume of the coolant), and can create more uniform cooling patterns on the plate surface, compared to the sprays and planar jets [71]. Hence, a more detailed study of the behavior of multiple circular jets impinging on a moving surface seems to be highly beneficial.

A good amount of research in this field has been focused on understanding the basic underlying mechanism of the jet impingement cooling. Thus, the studies are mainly focused on one single jet impinging on a stationary surface. While many researchers have tried to study the effects of multiple jets, or a moving surface, these two parameters are not adequately considered together in order to understand their coupled impacts. Also, previous studies usually categorize the wet area only into two zones: the impingement zone, and the parallel flow zone. While this may be an acceptable classification system for a stationary plate and a single nozzle, it is not very suitable for multiple jets impinging on a moving surface. The main problem with this classification is that there is no distinction between the parallel flow zone within a row of jets and the parallel flow zone between two consecutive jet rows. Moreover, many researchers have performed their experiments on a scaled down setup, while in this research we are using a large experimental industrial scale facility similar to the runout table facilities in material processing industries.



**Figure 1.8. Different Boiling Regions Present During Top Jet Impingement [14].**

On its simplest form, the area on the top surface of a hot plate that is cooled by a water jet can be divided into five different heat transfer regions, as displayed in Figure 1.8 [14]. The first zone is the impingement zone, and is located just underneath the impinging jet. This region is characterized by single-phase forced convection and a very effective cooling rate. Based on some operating conditions, such as plate speed, flow rate, and jet cross section, this region can extend about 2 to 4 times the nozzle diameter [14,72-74].

With the movement of the waterfront, the water is heated, the surface temperature is kept at a higher level, and the onset of boiling is finally reached, creating a relatively narrow nucleate boiling region. In the third region, the forced convection film boiling occurs, resulting in a reduced heat transfer rate. Thereafter, due to surface tension, the water may agglomerate into pools, overriding the vapour layer. Finally, the last region is where the plate is still dry, Heat transfer in this zone occurs by radiation and convection to the surrounding air.

The study of the water jet impingement cooling in the literature can be divided into two main categories of stationary and moving surfaces. For stationary cases, the cooled area is symmetrical with respect to the water jet line in the cases of water curtains or a row of jets, and axisymmetrical around the stagnation point in the case of a single circular jet. Although, much

easier for experimental investigation and numerical modeling, this case does not exist in industrial applications.

What happens in a typical steel mill is water cooling of a moving hot surface. In the moving case, heat transfer caused by the impinging jets is no longer symmetrical. On one side, where the water flow and the work piece move in the same direction, the vapour layer thickness in the film boiling region decreases. On the other hand, the thickness of the vapour layer is increased in the side that the water flow and the hot surface move in the opposite direction [1]. Another factor that makes the analysis of the cooling process more complicated is the dependency of the thermal conductivity and the specific heat of the steel on temperature. Also, phase transformation in the steel affects the thermal field through the latent heat of transformation.

Ochi et al. [15] experimentally investigated the transient boiling heat transfer to a circular water jet impinging on a hot plate. The temperature at the bottom was measured, and the temperature and flux at the top cooling surface were inversely calculated using the finite difference method. In the experiment, it was observed that a stable film boiling was maintained over the whole cooling surface of the heated plate, and the temperature fell linearly with time.

Vader et al. [73,74] studied the convective nucleate boiling on a heated surface cooled by an impinging planar water jet under a steady-state condition. The temperatures were measured by TCs at the opposite dry face to the cooled top surface. The temperature and heat flux at the cooled surface were found by solving the steady state equation. In the calculations, the boundary conditions were assigned from the least squares cubic spline fit of the measured temperatures.

Kumagi et al. [52] studied the transient cooling of a hot plate with an impinging water jet. Temperatures were measured at four points at different depths along the plate thickness and at nine locations in the width direction. An exponential equation with three coefficients was used to describe the temperature profile along the thickness, and then the coefficients were determined from the measured temperatures and the least squares method. The approximate temperature curve was extrapolated to the surface to obtain the surface temperature. The gradient at the surface produced the heat flux.

Many researchers have studied the jet impingement under steady state conditions, i.e., the sample is constantly heated while being cooled by jet impingement. Some good examples are the works of Robidou et al. [10,75]. They use a 2D inverse heat conduction solver to find the local

heat fluxes and corresponding temperature. They found that in the forced convection regime, heat fluxes increase with an increase of subcooling, jet velocity, and decrease of the distance from the stagnation line, while in the fully developed nucleate boiling regime, no influence of jet velocity, subcooling, and nozzle to plate spacing on the heat flux is observed.

Xu and Gadala [76] used their developed inverse algorithm [1] to study the heat transfer behaviour in the impingement zone under a circular water jet. They found out that the heat transfer behaviour in the stagnation zone is mainly affected by the water flow rate, with mild effect from the steel grades. Their developed code is 2D, so it cannot predict the effect of staggered jet configurations, in addition to not being able of modeling the heat transfer in the width of the strip. The study is also mainly focussed on the stagnation region in the stationary plate with only a single jet.

One of the key research papers studying the interaction zone between multiple jets was published in 1977 by Ishigai et al. [77]. Unfortunately, this research is focused on the non-boiling heat transfer. They found that the radial location of interference is an important factor for both the hydrodynamics and thermal properties of the flow field. Different behaviors were attributed to different values of film Froude number and the Reynolds number based on the radius from the center of the jet. The reported thermal quantities are the surface temperature, and not the heat flux.

Sakhuja et al. [78] used experiments to study the effect of multiple jets on the boiling heat transfer on a steady plate. They found that at smaller jet spacing, the area that is covered by each jet is small, and crossflow causes a reduction in the heat transfer coefficient. On the other hand, a larger distance between the jets results in a smaller ratio of the impinging area to the coverage area, which again, reduces the overall heat transfer coefficient. So an optimum heat transfer is obtained when the jet spacing is in a bounded range. They found this range to be between 8 to 12 times of the one jet diameter.

Monde et al. [79] studied the effect of multiple jets on a stationary plate. Between two and four circular free surface jets were used. While they concluded that the degree of scatter in the results is typical for nucleate boiling, and thus, the number or location of jets has little or no effect on the heat transfer behavior, a more careful scrutiny of their data by Wolf et al. [13], suggested that they actually do have an effect, and additional investigation is required.

Slayzak et al. [80] studied the effects of interaction between adjacent free surface *planar* jets under non-boiling condition. They observed a strong upwelling of flow (called an *interaction fountain*) in the region between two planar jets. Interestingly, they found that the heat transfer under this fountain can be comparable to those related to the jet impingement region. They tried to explain this behavior based on the oscillatory nature of the interaction zone, which can encourage a strong mixing and intermittent disruption of the thermal boundary layers in the interaction region. They also studied the effect of interaction between *adjoining rows* of circular jets under non-boiling condition [81]. Again, an interaction fountain was observed. However, unlike the case of planar jets, where a well-defined interaction zone appears midway between the impinging jets, the interaction zone between circular jets was found to be irregular, and at a location different from the midway point. Also, there is no splattering or hydraulic jump occurring in their tests. Obviously, both the splattering and the hydraulic jump can increase the complexity of the flow and thermal fields.

Filipovic, et al. [71] studied the heat transfer behavior of an array of round jets impinging on a moving surface. They have used some correlations for the boiling heat transfer in different regions, sometimes even from the stationary plates, and have made some assumptions regarding the extent of each zone. These correlations are applied, and the model is solved numerically. In their validation, they only consider the starting and coiling temperatures, and not the transient behavior of the thermal field. They found out that around 50% of the cooling is done by the film boiling mechanism, and the jet impingement regions are responsible for about 30% of the total heat extraction.

The uniformity of temperature distribution across the plate width when using multiple impinging jets was studied by Haraguchi and Hariki [82]. They used a radiation thermometer in their experimental setup on a stationary plate. They found the interaction zone to reduce the heat transfer coefficient around the impingement locations. To obtain a more uniform temperature distribution in the plate, they suggested decreasing the nozzle pitch. Unfortunately, this resulted in a reduction of heat transfer coefficient in those areas as well.

Liu [8] also studied the effect of jet interaction between adjacent water jets on the heat transfer behavior in a simulation of steel runout table. By visual inspection of the experiments, he observed that the dark areas that are associated with the high heat flux regions develop immediately at the stagnation points under the jets, and then gradually move toward the

interaction boundary, where they stabilize. He also observed a strong fountain at the interaction zone, accompanied by sizeable water splashes. However, the cooling intensity of water was weakened at the interaction region when boiling was the mechanism of heat transfer. He also studied the case where jets had different flow rates. In that case, the heat transfer rate under the weaker jet was reduced due to the existence of a cross flow created by the stronger jet.

Jondhale et al. [83] studied the heat transfer behavior on the top surface of a moving hot plate cooled by multiple impinging jets. They observed that there is a huge discrepancy between the heat transfer in the impingement zone and interaction zone, until the plate entry temperature reaches 200°C, when the water can wet the surface between the nozzles. They also observed that larger nozzle spacing in a jet row can actually increase the heat extraction rate under the jet, because it prevents having a submerged jet. The heat flux is however reduced in the interaction zone when having larger jet spacing.

Gradeck et al. [84] studied the effect of moving of the hot surface by impinging a planar water jet on a *rotating cylinder* with an initial temperature of 500 – 600 °C. They observed that the heat transfer is significantly changed when the surface is moving. The first effect was a homogenization of the boiling curves, and elimination of the “shoulder of flux”. They also observed a reduction in the value of the critical heat flux.

In this work, we are going to study the boiling heat transfer characteristics of two rows of three impinging jets on a moving hot plate. Experiments are conducted in an industrial scale setup, and thermocouples are used to measure the temperatures just under the plate surface. Then these temperatures are used as the input in an inverse heat conduction algorithm to find the boundary heat flux variations in space and time across the width and along the length of the moving flat plate. The effects of nozzle configuration, jet line spacing, and plate speed are going to be investigated.

## **1.6 Thermal Stress and Deflection**

As a result of an uneven temperature distribution in the strip, significant thermal stress occurs. This is even more magnified by the steel phase transformation. These stresses can result in local deformation and residual stresses. The flatness of the hot rolled products on the run-out table has not been sufficiently studied in the literature. Only a few researchers have investigated the thermal stresses of hot rolled strips [85].

Yoshida [86] was one of the first researchers to numerically study the non-flatness of hot rolled steel strips after cooling in a run-out table by applying finite-difference methods. Earlier, experiments had shown that the edge wave occurs after cooling in most cases, unless the strip is coiled at temperatures higher than 570 °C. He stated that the non-flatness occurs when the compressive residual thermal stress becomes larger than the critical buckling stress of a long thin plate. The thermal distribution was found using simple correlations for water and air cooling heat transfer coefficient, and the phase change was calculated based on a time-temperature-transformation (TTT) diagram. He found out that reducing the transverse variation of finishing temperature and cooling rate, coiling at higher temperatures and water cooling in the final sections of a run-out table can prevent the occurrence of edge waves. Among them, a uniform cooling rate across the strip was the most beneficial.

Nakata and Yoshida [87] later extended the above study to further study the effect of the cooling uniformity in the strip flatness using both experimental measurements and numerical simulations. The thermal model was validated against the measured surface temperature distributions using an infrared camera. It was confirmed that the non-uniform cooling along the length of the strip can cause non-flatness in the strip.

Han et al. [88], calculated the elastic strain, the volumetric strain caused by the thermal loads and phase transformation, the plastic strain, and the transformation induced superplastic strain, to estimate the total deformation of steel strips on a run-out table. The model was two-dimensional, and cooling conditions were set using simple correlations for two separate zones cooled by air or water.

Zhou et al. [89] calculated the thermal expansion coefficient as a function of the cooling rate, phase composition and temperature. They used a plane stress finite element model based on the assumption that the through-thickness stress is negligible in thin strip. The validation was done only based on qualitative agreement of the predicted results by the results of a simple method in a confidential report. Thermal boundary conditions were applied in the form of simplified temperature distributions. They found that the temperature distribution in the length of the strip has little effect on the residual stress, but the thermal field across the width of the plate has a large contribution to the stress field. In a later study [90], they investigated the flatness of the plate under the controlled cooling process using three-dimensional shell elements. They

found that the severity of buckling in the strip increases with the transverse temperature gradient across the plate, i.e. it resulted in a lower wavelength and higher amplitude.

As discussed above, most of the existing studies in the literature use simplified models for the thermal boundary conditions, usually based on some generalized correlations of heat transfer in two cooling zones (under impingement and parallel flow zones), or temperature measurements in few stages in the strip. They are also normally using simplified models for phase transformation. In the present study, we are going to use the results of our inverse heat conduction solver [76,91] to extract the boundary heat fluxes from the readings of thermocouples in an extensive set of experiments [1,31], and use a more refined and modern algorithm to calculate the phase transformation. We will study the cases of both the stationary and moving plates, and different jet configurations.

### **1.7 Objectives of This Study**

The following steps are taken in this research:

#### **1) Developing a three-dimensional gradient-based inverse heat conduction program**

In order to be able to model different jet configurations accurately, and to obtain accurate heat flux values, a three-dimensional finite element heat conduction simulator is needed. The code should be capable of taking into account the heat generation caused by the steel phase transformation and the temperature and phase dependency of thermophysical properties. The model should be able to capture the interaction between several jets. Unlike the case of interacting planar jets, where a well-defined interaction zone exists in midway between the jets, interaction zones between arrays of circular jets are irregular and do not necessarily occur in the midway [81]. Thus, an iterative and sequential three-dimensional inverse heat conduction code is developed. The algorithm is based on the least-square technique, sequential function specification, and regularization. The code needs to be capable of handling unsteady cases, and moving boundary conditions.

#### **2) Investigating various categories of stochastic inverse heat conduction solvers**

The above mentioned inverse solver, while being computationally efficient, suffers from some problems related to its gradient-based nature, such as sensitivity to the measurement errors, and instability in handling small time steps. Other forms of inverse solvers will be studied and modified to find a more effective algorithm. Modifications will be performed in order to improve not only the stability and effectiveness of the method, but also its computational expense.



### 3) Determination of surface heat flux and heat transfer behaviour in multiple jet impingement on a moving plate

Existing numerical models are mainly focussed on the case of single water jet. While they are useful for understanding the physics and to validate the algorithm, they are far from the real situation in industry. Multiple jets with different configurations should be modelled to make the case closer to the industrial applications. The developed inverse algorithms will be used to determine the heat fluxes occurring during the water jet cooling process. We will be especially focused on cases where multiple jets (several jet lines, each including more than one nozzle) imping on a moving plate, which are closer to what really happens in an industrial run-out table.

### 4) Studying the effect of plate speed, jet line stagger, and jet line spacing on heat transfer

Effect of these parameters on the maximum heat extraction, uniformity of cooling pattern and temperature distribution, and the overall efficiency of the run-out table will be studied.

### 5) Modeling the thermal stresses caused by the water jet cooling of stationary and moving plates

As a result of non-uniform cooling on the plate, thermal stresses and deflection are expected. A model will be devised that will be capable of modeling the phase change process as well.

## 2. Three-dimensional Direct and Inverse Heat Conduction Analyses

In this chapter, we discuss the development of a 3D finite element heat conduction program. Then, a classical gradient-based inverse algorithm is developed based on the 3D code. Some modifications and considerations are discussed. Finally, some test cases are introduced and solved, and the effect of these modifications are observed and discussed.

### 2.1 Direct Three-Dimensional Heat Conduction Analysis

The core of any inverse heat conduction code is a direct heat conduction solver. It is mainly used to evaluate the value of objective functions; thus, it needs to be as accurate and as stable as possible. It should also allow for the modeling of all complications in the geometry, boundary conditions, material properties, and transient behaviour. In this section, the formulation and development of such a solver is discussed.

#### 2.1.1 The Need for Three-Dimensional Modeling

In the case of multiple jets impinging on a moving surface, significantly different heat fluxes may be produced. This can result in a large temperature gradient across a jet line, as well as between two successive jet lines. Figure 2.1 shows the distribution of heat fluxes on the top surface of a moving plate [31]. In addition to this considerable variation in the values across the top surface, there is also a gradient across the thickness of the plate, between the top and bottom surfaces. This is due to the fact that the cooling pattern on the bottom surface is essentially different than the top surface, mainly due to the lack of pool boiling regions [28].

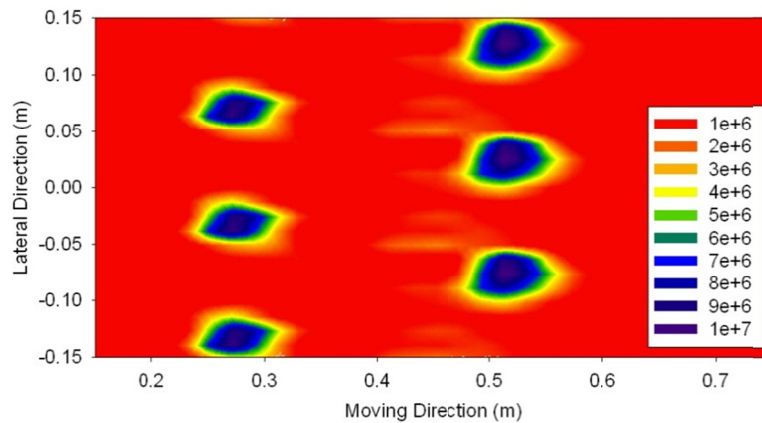


Figure 2.1. Contour Plot Showing Surface Heat Flux ( $\text{W/m}^2$ ) in a Staggered Nozzle Configuration [31]

### 2.1.2 Formulation

The general governing equation for the 3D conduction heat transfer problems, shown in Figure 1, is written in the form:

$$\frac{\partial}{\partial x}(k_x \frac{\partial T}{\partial x}) + \frac{\partial}{\partial y}(k_y \frac{\partial T}{\partial y}) + \frac{\partial}{\partial z}(k_z \frac{\partial T}{\partial z}) + q^b = c_p \rho \frac{\partial T}{\partial t} \quad (2.1)$$

where  $T$  is the temperature, °C;  $q^b$  is the heat generation per unit volume, W/m<sup>3</sup>;  $k_x$ ,  $k_y$ , and  $k_z$  are the conductivities in the  $x$ -,  $y$ -, and  $z$ -directions, respectively, W/m·°C;  $\rho$  is the density, kg/m<sup>3</sup>;  $c_p$  is the specific heat, J /kg·°C;  $t$  is the time, s ; and  $x$ ,  $y$ , and  $z$  are the Lagrangian coordinates of the point.

The boundary conditions may be one or a combination of the followings cases:

**Prescribed temperature:** This is an example of a *Dirichlet* boundary condition (BC). The prescribed temperature  $T_s$  (°C) may be a function of time and boundary coordinate (spatial function):

$$T = T_s(x, y, z, t) \quad (2.2)$$

**Prescribed heat flow of flux:** Specified heat flux ( $q_s$ ) may be a spatial function or a function of time:

$$-\left(k_x \frac{\partial T}{\partial x} + k_y \frac{\partial T}{\partial y} + k_z \frac{\partial T}{\partial z}\right) = q_s(x, y, z, t) \quad (2.3)$$

where  $q_s$  is the specified rate of heat flow per unit area (W/m<sup>2</sup>). Prescribed heat flow is an example of Cauchy's or Neumann boundary conditions. If  $q_s$  is zero, it will represent natural boundary conditions.

**Convection heat exchange:** When there is a convective heat transfer on part of the body surface due to contact with a fluid medium, we have:

$$-\left(k_x \frac{\partial T}{\partial x} + k_y \frac{\partial T}{\partial y} + k_z \frac{\partial T}{\partial z}\right) = h(T_s - T_f) \quad (2.4)$$

where  $h$  is the convection heat transfer or film coefficient (W/m<sup>2</sup>·°C), which may be temperature dependent (nonlinear),  $T_s$  is the surface temperature (°C) and  $T_f$  is the fluid temperature (°C), which may be a spatial or time function.

**Radiation:** Assuming grey body, this boundary condition is given by:

$$-\left(k_x \frac{\partial T}{\partial x} + k_y \frac{\partial T}{\partial y} + k_z \frac{\partial T}{\partial z}\right) = \epsilon \sigma [T_{sr}^4 - T_r^4] \quad (2.5)$$

where  $\varepsilon$  is the emissivity of the surface of the body,  $\sigma$  is the Stefan-Boltzmann constant ( $\text{W/m}^2 \cdot ^\circ\text{K}^4$ ),  $T_{sr}$  is the absolute temperature of surface  $S_5$  ( $^\circ\text{K}$ ) and  $T_r$  is the known absolute temperature of the external radiative source ( $^\circ\text{K}$ ). The radiation boundary condition may be dealt with as a nonlinear convective boundary condition with an equivalent temperature dependent film coefficient,  $\kappa$ , where:

$$\kappa = \varepsilon \sigma (T_{sr}^2 + T_r^2) (T_{sr} + T_r) \quad (2.6)$$

### 2.1.3 Numerical Modeling

Using a weighted residual Galerkin procedure, the final finite element equations may be written as:

$$\mathbf{C} \dot{\mathbf{T}} + \mathbf{K} \mathbf{T} = \mathbf{Q} \quad (2.7)$$

where  $\mathbf{C}$  is the equivalent heat capacity matrix;  $\mathbf{K}$  is the equivalent heat conduction matrix;  $\mathbf{T}$  and  $\dot{\mathbf{T}}$  are vectors of the nodal temperature and its derivatives, respectively; and  $\mathbf{Q}$  is the equivalent load vector. Detailed expressions of the matrices in equation (2.7) are given in Appendix-A.

A general family of solution algorithms may be obtained by introducing a parameter  $\alpha$  where  $(0.0 \leq \alpha \leq 1.0)$  such that

$${}^{t+\alpha\Delta t} \dot{\mathbf{T}} = \frac{1}{\Delta t} ({}^{t+\Delta t} \mathbf{T} - {}^t \mathbf{T}) = \frac{1}{\alpha\Delta t} ({}^{t+\alpha\Delta t} \mathbf{T} - {}^t \mathbf{T}) \quad (2.8)$$

$${}^{t+\alpha\Delta t} \mathbf{T} = \alpha {}^{t+\Delta t} \mathbf{T} + (1 - \alpha) {}^t \mathbf{T} \quad (2.9)$$

If  $\alpha = 0$ , an explicit Euler forward method is obtained; if  $\alpha = 1/2$ , an implicit trapezoidal rule is obtained; and if  $\alpha = 1$ , an implicit Euler backward method is obtained. Substituting equation (2.8) into equation (2.7) and applying Newton-Raphson iterations yields:

$${}^{t+\alpha\Delta t} \left[ \mathbf{K} + \left( \frac{1}{\alpha\Delta t} \right) \cdot \mathbf{C} \right] {}^{\Delta t(i)} \mathbf{T} = {}^{t+\alpha\Delta t} (\mathbf{Q}^b + \mathbf{Q}^s) + {}^{t+\alpha\Delta t} (\hat{\mathbf{Q}}^h + \hat{\mathbf{Q}}^r)^{(i-1)} - {}^{t+\alpha\Delta t} (\hat{\mathbf{Q}}^c + \hat{\mathbf{q}}^c)^{(i-1)} \quad (2.10)$$

where  $\alpha \neq 0$ . The definition of all terms is given in Appendix-A and all quantities at time  $(t + \alpha\Delta t)$  are calculated from a relation similar to equation (2.9).

Depending on the value of  $\alpha$ , the procedure may be either conditionally stable ( $\alpha < 0.5$ ) or unconditionally stable  $\alpha \geq 0.5$ .

### 2.1.4 Special Considerations

The thermal properties of steel normally change with a change in the temperature. This results in a nonlinear behaviour of the thermal field. The exact correlation depends on the chemical composition of each steel sample. For example, for DQSK steel, the relationship between the thermal conductivity and the temperature is:

$$k = 60.571 - 0.03849 \times T \text{ (}^\circ\text{C)} \text{ in } W/m.^\circ\text{C} \quad (2.11)$$

and for SS316 steel, the correlation is:

$$k = 11.141 + 0.014 \times T \text{ (}^\circ\text{C)} \text{ in } W/m.^\circ\text{C} \quad (2.12)$$

These equations are valid from zero to 1000 °C. So the finite element code should be capable of handling these nonlinearities.

It is also important to integrate the microstructure evolution models into the heat transfer analysis [1,92]. Since during the cooling process, the austenite will transform into other phases, a latent heat is released. While some researchers have ignored this effect [93], the accuracy of the simulations can be enhanced by incorporating this heat into the model [1,94]. In our implementation, following the procedure in Reference [1], the heat generation per volume is calculated by:

$$Q = H \cdot \dot{F} \quad (2.13)$$

where H is the mole of material considered, and  $\dot{F}$  is the transformation rate of austenite to ferrite and pearlite per volume. This value is going to be calculated in a subroutine, and the values will be included in the source term  $Q^b$  in equation (2.10).

### 2.1.5 Implementation and Validation

In the developed program, the geometry can be discretized using isoparametric 3D elements. By default, the elements will have eight nodes and will resemble a brick. However, other 3D shapes can also be created. The program is capable of dealing with a mixture of element types. The program is also capable of dealing with all different types of boundary conditions, as mentioned above. The boundary conditions can be a function of space and/or time. The nonlinearities caused by the dependence of the thermophysical properties on the temperature can be handled in a step-wise staggered approach, i.e. the values of the parameters at the current

step are calculated based on the temperature at the previous step, and are assumed to be constant during the current step. The program is capable of handling both the steady and transient problems.

Various verification cases were modeled and the results were compared with the results obtained by the commercial program ANSYS [95]. In all cases, the results were shown to be within less than 1% of each other. Some of the test cases are shown here.

### **Geometry and Mesh**

Figure 2.2 shows a schematic of the geometry and the discretized domain. The width of the block ( $x$  direction) is 1.5 m, the height ( $y$  direction) is 0.5 m, and the length ( $z$  direction) is 5.0 m. The thermal conductivity is following equation (2.11), the density,  $\rho$ , is  $7850 \text{ kg/m}^3$ , and the heat capacity,  $C_p$  is  $475 \text{ J/kgK}$ .

### **Test Case I**

The initial temperature of the block is  $14^\circ\text{C}$ . The top surface of the block is subjected to a convection boundary condition with a film coefficient of  $60 \text{ W/(m}^2\cdot^\circ\text{C)}$  and a fluid temperature of  $40^\circ\text{C}$ . The comparison between the developed code and ANSYS is displayed in Figure 2.3(a).

### **Test Case II**

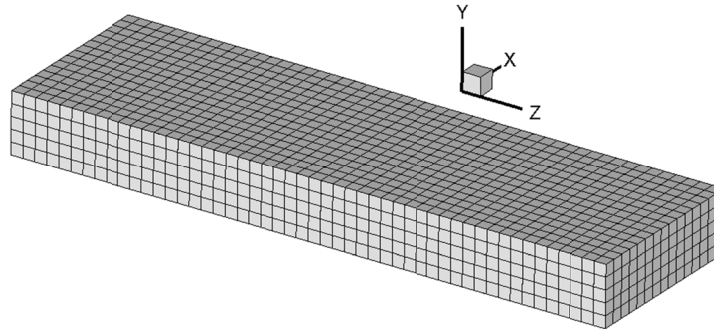
Starting from an initial temperature of  $8^\circ\text{C}$ , the top surface of the block is subjected to a constant heat flux of  $100 \text{ W/m}^2$ . The comparison between the developed code and ANSYS is displayed in Figure 2.3(b).

### **Test Case III**

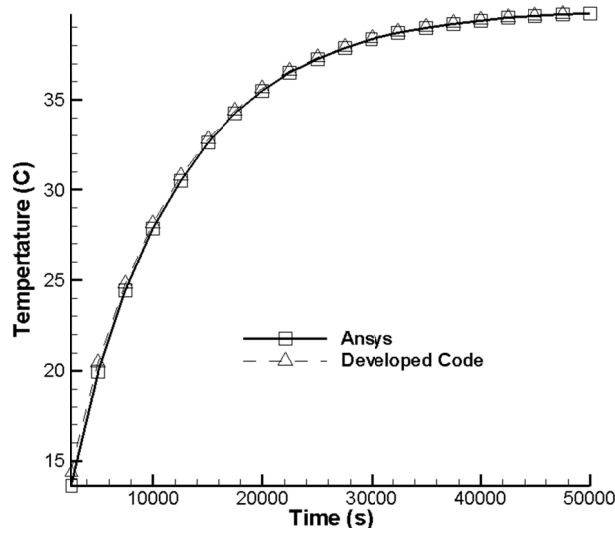
Starting from an initial temperature of  $1^\circ\text{C}$ , the top surface of the block is subjected to a radiation boundary condition with a surrounding temperature of  $100^\circ\text{C}$ . The comparison between the developed code and ANSYS is displayed in Figure 2.3(c).

#### **Test Case IV**

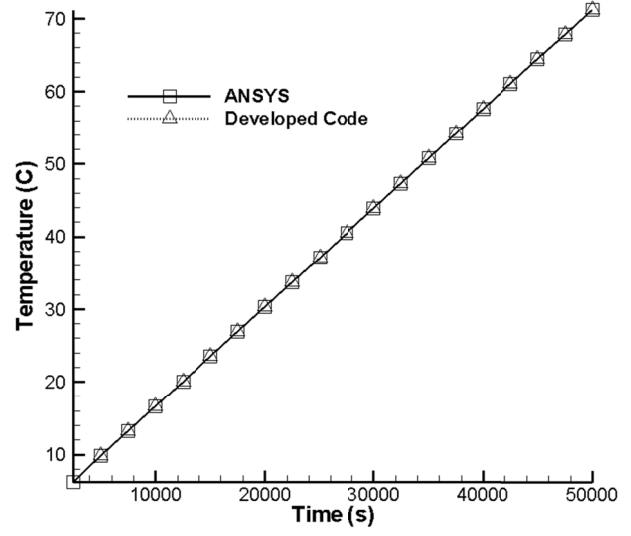
Starting from an initial temperature of 1 °C, the top surface of the block is subjected to a specified temperature boundary condition of  $T_{\text{top}} = 2x + 5z + 10$ . The comparison between the results of the developed code and ANSYS for a point at  $y = 0.25$  m is displayed in Figure 2.3(d).



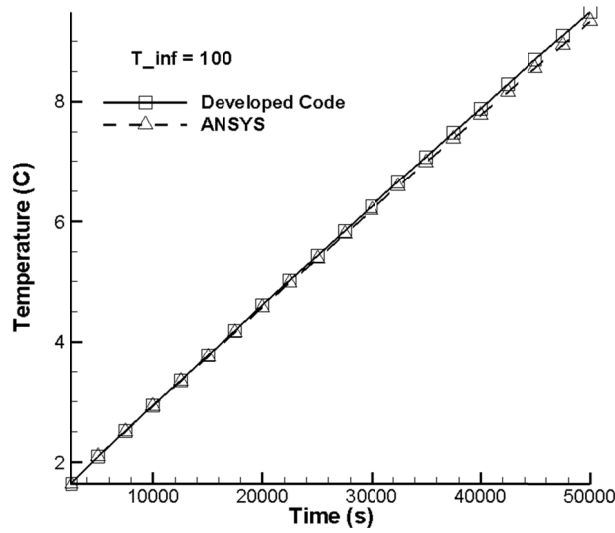
**Figure 2.2. Geometry and mesh for the validation of the direct 3D code**



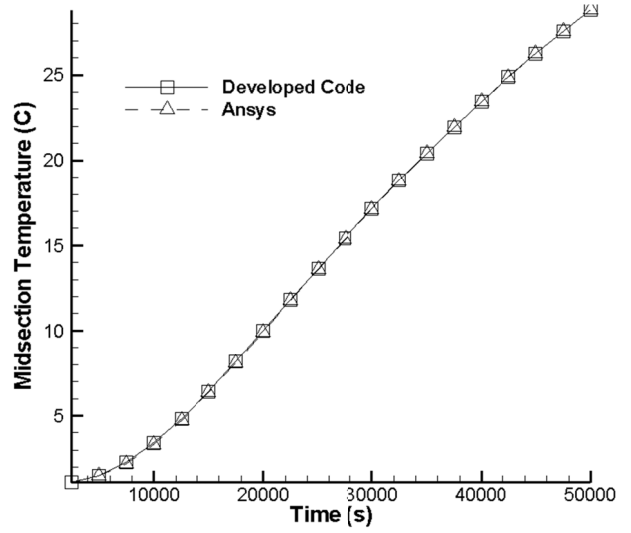
(a)



(b)



(c)



(d)

Figure 2.3. The results of validation for the 3D direct code

## 2.2 Formulation of Inverse Analysis

### 2.2.1 Objective Function

The boundary inverse heat conduction problem can be formulated as an optimization problem in which we try to minimize the norm of difference between the experimentally measured temperatures and the calculated temperatures obtained from a direct solution of the problem with some guessed boundary conditions. In order to dampen the oscillations in the



solution, and make the algorithm more stable in dealing with noisy measurement data, it is very common to include some kind of regularization terms that penalize the large magnitude error in the objective function. A common choice in inverse heat transfer problems is to use a scalar quantity based on the boundary heat fluxes, with a weighting parameter  $\alpha$ , which is normally called the regularization parameter. The regularization term can be related to the values of heat flux, or their first or second derivatives with respect to time or space. Based on the previous experience [76], as well as our own trial of different regularization terms, we choose to use the heat flux values (zeroth-order regularization). The objective function is then

$$f(\mathbf{q}) = \sum_{j=1}^J \left( \sum_{i=1}^N (T_j^{i,meas} - T_j^{i,calc})^2 + \alpha \sum_{i=1}^N (q_j^i)^2 \right) \quad (2.14)$$

where  $T_j^{i,meas}$  and  $T_j^{i,calc}$  are the values of expected (measured) and calculated temperatures at the  $i^{\text{th}}$  time step and the  $j^{\text{th}}$  spatial location, respectively;  $\alpha$  is the regularization coefficient; and  $q_j^i$  is the boundary heat flux vector at the  $i^{\text{th}}$  time step and the  $j^{\text{th}}$  spatial location,  $N$  is the total number of time steps, and  $J$  is the number of spatial components of temperature vectors.

Due to the parabolic nature of the heat equation, there are basically two categories of approaches in solving the inverse transient heat conduction problems. One method is to try to find components of heat flux in each step of the solution. This is possible since the temperature distribution at the end of each time step is only a function of the temperature distribution at the previous time step and the boundary conditions during the present time step. These methods are normally called “sequential”. In these methods, in each step of the inverse analysis, the direct problem is not solved for the whole history, and is solved for only one (or a few) time steps. The optimization problem is then solved for the temperatures at that step, and then the problem is marched in time to the next steps. The other solution method is to treat the whole history of heat fluxes as an unknown in the optimization problem. In this formulation, which is usually called the “whole domain” approach, the direct problem will be solved over the whole time history.

Due to the diffusive nature of transient heat conduction and from a computational expense perspective, sequential estimation techniques are highly preferred over whole-domain inverse analysis techniques [35]. Unfortunately, there is an important drawback in using

sequential methods, and that is the high sensitivity of the solution to the experimental errors. That is why we will be using the future time steps concept.

Due to the fact that inverse problems are generally ill-posed, the solution may not be unique and is generally sensitive to measurement errors. To decrease such sensitivity and improve the simulation in function specification methods, a number of future time steps ( $n_{FTS}$ ) are often utilized in the analysis of each time step [35]. This means that in addition to the measured temperature at the present time step  $T^i$ , the measured temperatures at future time steps,  $T^{i+1}, T^{i+2}, \dots, T^{i+n_{FTS}}$ , are also used to approximate the heat flux  $q^i$ . In this process, a temporary assumption is usually considered for the values  $q^{i+1}, q^{i+2}, \dots, q^{i+n_{FTS}}$ . The simplest and most widely used one is to assume  $q^{i+k} = q^i$  for  $1 \leq k \leq n_{FTS}$ , which is also used in our work. Also in this work, we combine the regularization concept with the future time step data. This is analogous to the combined function specification-regularization method, initially introduced by Beck and Murio [96].

The sequential algorithm for time step  $M$ , when using  $r$  future time steps (the heat flux is known for  $t < t_M$ ), is presented below:

1. An assumption is made for the heat flux function at times  $t_M, t_{M+1}, \dots, t_{M+r}$  (usually  $q_M = q_{M+1} = \dots = q_{M+r}$ ).
2. The direct problem is solved with  $T_{M-1}^{meas}$  as the initial condition and the heat flux values from step 1 as its transient boundary conditions. The temperatures at these  $r$  time steps are calculated ( $T_M^{calc}, T_{M+1}^{calc}, \dots, T_{M+r}^{calc}$ ).
3. The objective function is then evaluated using the heat fluxes from step 1, the calculated temperatures from step 2, and the known temperatures from the measurement ( $T_M^{meas}, T_{M+1}^{meas}, \dots, T_{M+r}^{meas}$ ). The new form of the objective function becomes:

$$f(\mathbf{q}) = \sum_{j=1}^J \left( \sum_{i=M}^{M+r} (T_j^{i,meas} - T_j^{i,calc})^2 + \alpha \sum_{i=M}^{M+r} (q_j^i)^2 \right) \quad (2.15)$$

4. The algorithm is repeated until we reach a desired low value for the objective function, or after a certain number of iterations. The solution vector will be the spatial components of heat fluxes for the  $r$  time steps.
5. The values of  $q_M$ , as obtained from step 4, are used, and the values for the rest of the heat fluxes ( $q_{M+1}, \dots, q_{M+r}$ ) are discarded.
6. The above procedure is repeated ( $M = M+1$ ) as you move to the next time step.

In theory, it is possible to use multiple sensors instead of multiple future time steps to improve the stability of the method. However, this idea is not practical since adding thermocouples to the sample is not only more expensive, but also distorts the temperature field. For a detailed discussion of the effect of thermocouple hole on the thermal field, see References [25,97-101].

Using future time steps introduces a bias in the solution. This bias is normally observed in the form of rounding the spikes in oscillatory boundary condition histories, and an early prediction of these oscillations [35]. To obtain an appropriate level of regularization, the number of future time steps (or more accurately, the size of the look-ahead time window, i.e. the product of the number of future time steps and time step size) and the value of the regularization parameter must be chosen with respect to the errors involved in the temperature readings. The residual principle [36,102] can be used to determine these parameters based on the accuracy of thermocouples in the relative temperature range. In this method, the number of future time steps is increased until the RMS error of the estimation falls below the standard deviation of noise in the measured temperatures.

Choosing proper values for the regularization parameter and the number of future time steps is a very challenging task, and although there are some methods to find the range of appropriate values, the fine tuning still remains more like an art rather than an exact science. Basically, there are five main approaches to choose a close to optimal value for these parameters [103]. They are the maximum likelihood (ML), the ordinary cross-validation (OCV), the generalized cross-validation (GCV), the L-curve, and the discrepancy principle (DP) methods. All these methods require some kind of knowledge about the solution characteristics. One important piece of information is the level of errors in the solution domain. Luckily, the thermocouples that are normally used in experiments are well documented, and their accuracy levels are well investigated and understood. In our experiments, we are using type K

thermocouples that have an error level of  $\pm 2.5$  °C in temperatures below 333 °C, and an error level of  $\pm 0.75\%$  of the measured temperatures between 333 °C and 1200 °C. Among these five methods, the discrepancy principle (DP) method is known to produce the best estimate of the regularization parameter for inverse heat conduction problems [103]. This method requires that the problem be solved so that the residual norm is the same as the norm of errors in the measurements, denoted by  $\delta$ :

$$\sqrt{\sum_{i=1}^L (\overline{T}_m^i - T_c^i)^2} = \delta \quad (2.16)$$

When the values of the regularization parameter and the number of future time steps for one case are obtained, the same values can be normally used for similar problems.

### 2.2.2 Gradient Based Calculation of Heat Flux

The heat flux and temperature vectors are

$$\mathbf{q} = [\mathbf{q}^1 \quad \mathbf{q}^2 \quad \dots \quad \mathbf{q}^N]^T \quad (2.17)$$

$$\mathbf{q}^i = [q_1^i \quad q_2^i \quad \dots \quad q_J^i]^T \quad (2.18)$$

$$\mathbf{T}_m^i = [T_{1m}^i \quad T_{2m}^i \quad \dots \quad T_{Lm}^i]^T \quad (2.19)$$

$$\mathbf{T}_c^i = [T_{1c}^i \quad T_{2c}^i \quad \dots \quad T_{Lc}^i]^T \quad (2.20)$$

where  $L$  is the number of measurement points,  $J$  the number of heat flux components that can be determined for the flux space distribution on a surface. It should be noted that  $J$  must be less than or equal to  $L$ , the number of measurement points. It should also be noted that the dimensions of the heat flux vector  $\mathbf{q}^i$  at each step are  $L \times J$  while the total heat flux vector  $\mathbf{q}$  is  $J \times N$  as it includes the data in  $N$  steps; and the temperature vector  $\mathbf{T}^i$  at each step is  $L \times L$ .

Also, the temperature  $\mathbf{T}^k$  is only determined or affected by the heat fluxes  $\mathbf{q}^m$  where  $m \leq k$ . Mathematically, we may express  $\mathbf{T}^k$  as an implicit function of the heat flux:

$$\mathbf{T}_c^k = f(\mathbf{q}^1, \mathbf{q}^2, \dots, \mathbf{q}^k) \quad (2.21)$$

or in a successive form as:

$$\begin{aligned}
\mathbf{T}_c^k &= f(\mathbf{T}_c^{k-1}, \mathbf{q}^k) \\
\mathbf{T}_c^{k-1} &= f(\mathbf{T}_c^{k-2}, \mathbf{q}^{k-1}) \\
&\vdots \\
\mathbf{T}_c^2 &= f(\mathbf{T}_c^1, \mathbf{q}^2) \\
\mathbf{T}_c^1 &= f(\mathbf{T}_c^0, \mathbf{q}^1)
\end{aligned} \tag{2.22}$$

and the following equation is also valid

$$\mathbf{T}_c^k = \mathbf{T}_c^{k*} + \frac{\partial \mathbf{T}_c^1}{\partial \mathbf{q}^1}(\mathbf{q}^1 - \mathbf{q}^{1*}) + \frac{\partial \mathbf{T}_c^2}{\partial \mathbf{q}^2}(\mathbf{q}^2 - \mathbf{q}^{2*}) + \dots + \frac{\partial \mathbf{T}_c^k}{\partial \mathbf{q}^k}(\mathbf{q}^k - \mathbf{q}^{k*}) \tag{2.23}$$

The values with a ‘\*’ superscript in the above equation may be considered as initial guess values ultimately lead to the temperature  $\mathbf{T}_c^{k*}$ .

Now we define the first derivative of temperature  $\mathbf{T}_c^i$  with respect to heat flux  $\mathbf{q}^i$  as the sensitivity matrix:

$$\begin{aligned}
\mathbf{X}^i &= \frac{\partial \mathbf{T}_c^i}{\partial \mathbf{q}^i} \\
&= \begin{bmatrix} a_{11}(i) & a_{12}(i) & \dots & a_{1J}(i) \\ a_{21}(i) & a_{22}(i) & \dots & a_{2J}(i) \\ \vdots & \vdots & \ddots & \vdots \\ a_{L1}(i) & a_{L2}(i) & \dots & a_{LJ}(i) \end{bmatrix} \\
a_{rs}(i) &= \frac{\partial T_{cr}^i}{\partial q_s^i}
\end{aligned} \tag{2.24}$$

where  $i=1, 2 \dots N$ ,  $r=1, 2 \dots L$  and  $s=1, 2 \dots J$ . The sensitivity matrix  $\mathbf{X}^i$  is an  $L \times J$  matrix.

The optimality of the objective function may be obtained by letting  $\partial f / \partial \mathbf{q} = 0$ , and we get the following set of equations (note that  $\partial f / \partial \mathbf{q}$  should be done with respect to each component  $\mathbf{q}^i$ , with  $i=1, 2 \dots N$ ):

$$\left\{ \sum_{i=1}^N \left( \frac{\partial \mathbf{T}_c^i}{\partial \mathbf{q}^j} \right)_{\mathbf{q}^j = \mathbf{q}^{j*}}^T \left( \frac{\partial \mathbf{T}_c^i}{\partial \mathbf{q}^j} \right)_{\mathbf{q}^j = \mathbf{q}^{j*}} + \alpha \mathbf{I} \right\} (\mathbf{q}^j - \mathbf{q}^{j*}) = \sum_{i=1}^N \left( \frac{\partial T_{cr}^i}{\partial \mathbf{q}^j} \right)_{\mathbf{q}^j = \mathbf{q}^{j*}}^T (\mathbf{T}_m^i - \mathbf{T}_c^{i*}) - \alpha \mathbf{q}^{j*} \tag{2.25}$$

$j = 1, 2, \dots, N$

where  $\mathbf{q}^{j*}$  is the initial guess of heat fluxes, and  $\mathbf{T}_c^{i*}$  is the calculated temperature vector with the initial guess values.

Recalling equations (2.20) and (2.21), equation (2.22) may be rearranged and written in the following form:

$$(\mathbf{X}_{\mathbf{q}=\mathbf{q}^*}^T \mathbf{X}_{\mathbf{q}=\mathbf{q}^*} + \alpha \mathbf{I})(\mathbf{q} - \mathbf{q}^*) = \mathbf{X}^T \Delta \mathbf{T} - \alpha \mathbf{q}^* \quad (2.26)$$

where  $\mathbf{X}$  is labeled as the total sensitivity matrix for a multi-dimensional problem and has the following form:

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}^1 & 0 & 0 & 0 \\ \mathbf{X}^2 & \mathbf{X}^1 & 0 & 0 \\ \vdots & \vdots & \ddots & 0 \\ \mathbf{X}^N & \dots & \mathbf{X}^2 & \mathbf{X}^1 \end{bmatrix} \quad (2.27)$$

and

$$\Delta \mathbf{T} = (\mathbf{T}_m^1 - \mathbf{T}_c^{1*} \quad \mathbf{T}_m^2 - \mathbf{T}_c^{2*} \quad \dots \quad \mathbf{T}_m^N - \mathbf{T}_c^{N*})^T \quad (2.28)$$

Note that the dimension of matrix  $\mathbf{X}$  is  $(L \times N) \times (J \times N)$  and  $\Delta \mathbf{T}$  has dimensions of  $(L \times N)$ . Also, note that the calculations in equation (2.23) may easily be done in the time domain and no function specification for  $\mathbf{q}^i$  is needed. If the total sensitivity is known, no iteration is required to get a final solution.

### 2.2.3 Convergence Criteria

In each time step, the iterative procedure is used until the inversely predicted temperature  $\mathbf{T}_c$  converges to the measured temperature  $\mathbf{T}_m$ . Convergence criteria used to define the acceptance of the predicted temperature are based on an error norm defined by:

$$\text{Error-norm}^n = \|\Delta \mathbf{T}^n\| \quad (2.29)$$

Two convergence criteria for ending the iteration process at each time step are used:

$$\text{Error-norm}^n \leq \delta T \quad (2.30)$$

or

$$\frac{|\text{Error-norm}^{n+1} - \text{Error-norm}^n|}{\text{Error-norm}^n} \leq \varepsilon \quad (2.31)$$

The values of  $\delta T$  and  $\varepsilon$  depend on the measurement error level. The rationale behind using absolute criteria is that while the norm at a given previous iteration is already very small, the relative norm criterion is still not satisfied in the last iteration.

### 2.2.4 Flux Zoning Method (FZM)

The boundary heat flux, which is the output of the developed inverse algorithm, will be in vector form. We need to assign this discretized vector form to the target surface. A common technique is to divide the target surface into several zones, each corresponding to one temperature measurement. The heat fluxes for each zone will be constant for that zone, and may differ from one to the next, as shown in Figure 2.4. This approach is called the Flux Zoning Method (FZM) [1].

This approach is fairly easy to understand and implement. It is a very good solution for the case where the heat flux is nearly constant or does not dramatically change spatially in a subregion. However, some researchers have recognized it as inappropriate in the case of a moving boundary heat flux, such as the case with the moving waterfront [1].

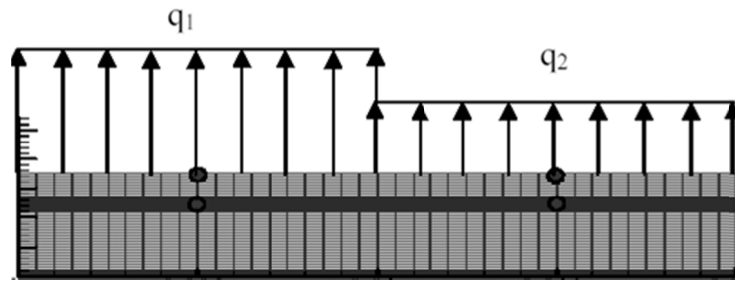


Figure 2.4. FZM Model for Inverse Calculation of the Boundary Heat Flux

### 2.2.5 Flux Marching Method (FMM)

An alternative approach to the FZM is the Flux Marching Method (FMM) [1], which tries to match the movement of the heat flux on the boundary. In its original form, this method determines the heat flux distribution in each region using the following algorithm:

1. The heat flux at each given node at the current time step is generally assumed to be equal to the heat flux on the neighbouring upstream node at the previous time step.
2. The heat flux on the other nodes at the current time step is generally assumed to be the value on the prior node at the several previous time steps.

While in simple cases, and mostly in theoretical testing, this method may perform better than the FZM, there are several serious problems that hinder its application in real cases. Some of these problems are:

a) The FMM heavily relies on the flow direction on the boundary. In three-dimensional cases, and especially in the cases with moving plates and multiple impinging jets in each row and multiple successive jet rows, it is increasingly hard to determine the upstream and downstream directions. The flow pattern is very complex, with stagnation zones in the area between two impinging jets. The shape and location of these areas are also irregular [81].

b) In order to obtain accurate results, the exact value of the water velocity must be known, since it is the speed at which the nodal heat flux values move from one node to the other. Again, there is currently no accurate way to predict these speeds, and while recent studies [32] have started looking in more details at the velocity of the water propagation, we are still far from having a robust and useful correlation for the water velocity, especially in cases with multiple jets and a moving surface. This is especially important when one notices that the “step mismatch”, i.e. moving of the heat flux from one node to the other at a speed higher (overstepping) or lower (understepping) than the real velocity, may produce lower quality results than the FZM [1].

c) One of the main characteristics of the heat transfer in the jet impingement cooling is the sharp heat flux peaks in the impingement zone. The FMM is, in nature, an interpolating and smoothing scheme, which tends to change the sudden behaviours into gradual changes. This has caused a dampening of the heat flux peak values, as well as a virtual negative heat flux before the real sharp increase [1]. Adjusting the waterfront speed may reduce the magnitude of this problem, but then it will be another parameter that is needed to be addressed and will increase the complexity of the program, and make it harder to be used as an off-the-shelf code.

d) In the FMM, the following equation is used to find the time step at which the heat fluxes should move to the next node and the solution should be repeated [1]:

$$\Delta t_i = \frac{d_j}{v_{ij}} \quad (2.32)$$

where  $\Delta t_i$  is the time step size in the  $i$ th step,  $d_j$  is the  $j$ th spatial distance between thermocouples, and  $v_{ij}$  is the water moving speed over this distance in that time step. So in the cases of higher water moving speed, the denominator of the above equation becomes large, resulting in a very small time step size. This will result in either a large computational cost, or in many cases, to the instability of the gradient-based inverse algorithm, as will be discussed later in Section 2.3.4.



e) It is very crucial to notice that the problem with the FZM can be mitigated if smaller time steps are used. By refining the temporal resolution, the assumption that the heat flux is not moving in each smaller time step, will not significantly reduce the accuracy. Fortunately, the data provided by the data acquisition system is in very small intervals, so the limitation will not be caused by the experiments, but by the failure of the gradient-based inverse solvers to deal with smaller time steps. The remedy may be using inverse solvers that are not gradient-based. This will be studied in the next chapter.

### **2.2.6 Different Techniques for Calculating Sensitivities**

The form of the governing differential equation for both the temperature  $T(x, t)$  and the sensitivity coefficient  $X(x, t)$  is the same [35] and, therefore, the same finite element program may be used to calculate them. While this is efficient from the programming point of view, it may not be practical, especially when the temperature time history is somewhat long and the density of mesh is high, and also because of the fact that not all the coefficients at all points are needed to get the heat flux  $q$ .

An alternative way is that the sensitivity matrices at each time step for the measurement points be calculated. Such a procedure would still require intensive calculation time and cost when the number of total time steps considered and/or the number of heat flux components are reasonably large.

A perturbation algorithm [8] was used to obtain the sensitivity matrices. First, a given value is assumed for all components of the heat flux vector as input in the direct heat transfer calculation to get the temperature distribution for the given future steps at each thermocouple location. Then, one of the components of the heat flux vector is increased by a reasonable amount, such as 10%, to obtain new temperatures. While a perturbation of 10% looks larger than normal, we should notice that if the perturbation is small, the change in inside temperatures will be very small, due to the lagging and dampening of boundary changes in internal points, and the resulting sensitivity will be very small as well, and sometimes very close to zero. This can result in divergence or “division-by-zero” errors. The ratios of temperature difference at each thermocouple location to the applied increase in heat flux component are the sensitivity coefficients. Such perturbation is repeated for each component of the heat flux until all the sensitivity matrix components are obtained.

One important consideration in calculating the sensitivity is the nonlinearity. The whole sensitivity matrix is independent of the heat flux only if the thermal properties of the material are constant with temperature. For most materials, the thermophysical properties are temperature dependent. In such cases, all properties should be updated at the beginning of each time step, which is time consuming, especially for large size models. Moreover, such changes in properties would not be very large and would not significantly change the magnitude of the sensitivity coefficients. Also, updating the material properties at the beginning of each time step would be based on the temperatures  $\mathbf{T}^{k*}$  obtained from the initially given values of the heat flux  $\mathbf{q}^*$ , which is essentially an approximation. So, we may opt for updating the sensitivity matrix every  $M$  steps (in our numerical experiments,  $M=10$ ). The results obtained under this assumption are very close to those obtained by updating the values at each step, so the assumption is justified.

Another approach was also investigated in this research to replace the normal perturbation method in calculating the sensitivity coefficients. The suggested new approach is called the complex step method, and is basically based on a complex number, instead of a real number, perturbation. To introduce this method, we first write the Taylor series expansion of the real valued function  $f$  about both a real  $(x+h)$  and a complex  $(x+ih)$  numbers.

$$f(x+h) = f(x) + hf'(x) + h^2 \frac{f''(x)}{2!} + h^3 \frac{f'''(x)}{3!} + \dots \quad (2.33)$$

$$f(x+ih) = f(x) + ihf'(x) - h^2 \frac{f''(x)}{2!} - ih^3 \frac{f'''(x)}{3!} + \dots \quad (2.34)$$

Based on these expansions, the following three approximations of the first derivative of  $f$  with respect to  $x$  is possible:

$$f'(x) = \frac{f(x+h) - f(x)}{h} - h \frac{f''(x)}{2!} - h^2 \frac{f'''(x)}{3!} - \dots \quad (2.35)$$

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - h^2 \frac{f'''(x)}{3!} - \dots \quad (2.36)$$

$$f'(x) = \frac{\text{Im}[f(x+ih)]}{h} + h^2 \frac{f'''(x)}{3!} + \dots \quad (2.37)$$

The first one is a normal forward difference scheme, which is similar to the perturbation scheme that was previously described, the second expression is a central difference scheme, and the third one that uses the complex number perturbation is the basis for our complex step method.

When compared with the forward difference scheme, the complex step method is of higher order (2 instead of 1), and it does not involve a difference operation, so we can choose small steps, with no loss of accuracy due to subtractive cancellation. In comparison with the central difference algorithm, the complex step again does not involve a difference operation, so we can choose small steps, with no loss of accuracy. Also, unlike the central difference scheme, the complex step method does not need to evaluate the function at a third point ( $x-h$ ).

In our inverse problem, the sensitivity matrix,  $\mathbf{X}$ , is calculated using a perturbation method, which is similar to the forward difference algorithm:

$$X = \frac{\partial T}{\partial q} = \frac{T(q^* + \Delta q) - T(q^*)}{\Delta q} \quad (2.38)$$

where  $q^*$  is the initial guess for the value of the heat flux, and  $\Delta q$  is the perturbation, which is typically about 10% of the value of heat flux. Using the complex step idea, the sensitivity will be calculated as

$$X = \frac{\partial T}{\partial q} = \frac{\text{Im}[T(q^* + i\Delta q)]}{\Delta q} \quad (2.39)$$

This approach was tested in our problem. We found that unless the problem is highly nonlinear (more than what heat conduction problems normally are), there is no tangible difference between the first order perturbation method, and the second order complex step algorithm. This approach also could not solve the problem of gradient based inverse heat conduction algorithms, such as time step size limitation, or sensitivity to the measurement errors.

## 2.3 Results and Discussion

In this section, a few test cases are introduced and developed algorithms are tested in their capability of solving one-, two-, and three-dimensional problems. The effect of noise in the domain on the performance of the inverse algorithm is studied. Also, the effects of using regularization parameters, future time steps, and different time step sizes are also investigated. Finally, 1D, 2D, and 3D inverse algorithms are compared.

### 2.3.1 Test Cases

Five test cases are introduced. They include one-, two-, and three-dimensional heat conduction problems, in transient or steady conditions, and both linear and nonlinear in nature.

Notice that these test cases are also going to be used in the next chapter, and will not be repeated there.

### **Test Case I: 1D Transient Problem**

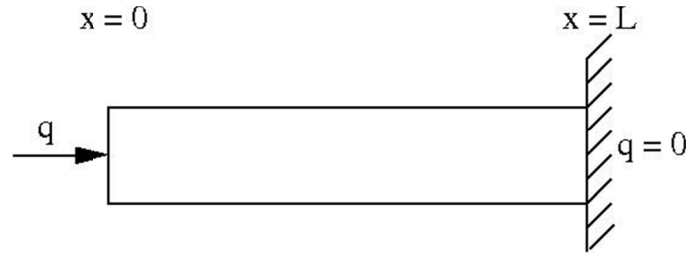
In this case, the triangular heat flux history, popularized by Beck et al. [35] is used as the input heat flux at the left end of the sample. For simplicity, the parameters are first taken as  $\rho c = L = 1$ . The problem was once solved with a constant  $k = 1$  (linear case), and then with the thermal conductivity varying with the temperature at  $x = L$  as:

$$k = 0.82 + T(x = L) \quad (2.40)$$

The heat flux history is given in the following form:

$$q(t) = \begin{cases} 0 & t < 0.24 \\ t - 0.24 & 0.24 \leq t < 0.84 \\ -t + 1.44 & 0.84 \leq t < 1.44 \\ 0 & t \geq 1.44 \end{cases} \quad (2.41)$$

Figure 2.5 shows the geometry of the problem, and its boundary conditions.



**Figure 2.5. Problem Geometry and Boundary Conditions, Test Case I**

The direct problem is solved and the temperatures at the right end ( $x = L$ ) are obtained and stored. The goal is to minimize the difference between these temperatures (simulated measurements) and those that can be obtained by applying an arbitrary input heat flux.

### **Test Case II: 2D Transient Problem**

In this transient 2D problem, the heat flux at the top surface (see Figure 2.6(a)) is chosen to resemble the one in water cooling of steel strips on the run-out table in a steel mill. In these experiments, in order to determine the surface heat flux, or the heat transfer coefficient during

the water cooling process, thermocouples are placed on the opposite side of the plate, or within the plate and 1.0 mm below the top surface, as depicted in Figure 2.6(b). Inverse analysis is then used to obtain the boundary conditions on the surface, based on the readings of these internal thermocouples. As shown in Figure 2.6(b), the thermocouple is located in a blind hole, with a diameter of 1.6 mm that is drilled from the bottom surface of the plate. The measurement junction of the thermocouple is fixed to the end surface of the hole, which is about 1 mm below the top surface of the plate. The problem is solved as an axisymmetric case. The boundary condition of the top surface is prescribed heat flux. The other boundaries are assumed to be adiabatic. The left boundary is axisymmetric. The density,  $\rho$ , is  $7850 \text{ kg/m}^3$ ,  $C_p$  is  $475 \text{ J/kgK}$ , and the thermal conductivity,  $k$ , is  $44.5 \text{ W/mK}$ . These are very close to the physical properties of the steel strips that are used in our controlled cooling experiment. Results are obtained at a point inside the object, which is the assumed position of a thermocouple. Inverse analysis is conducted to obtain the transient heat flux profile at the top surface.

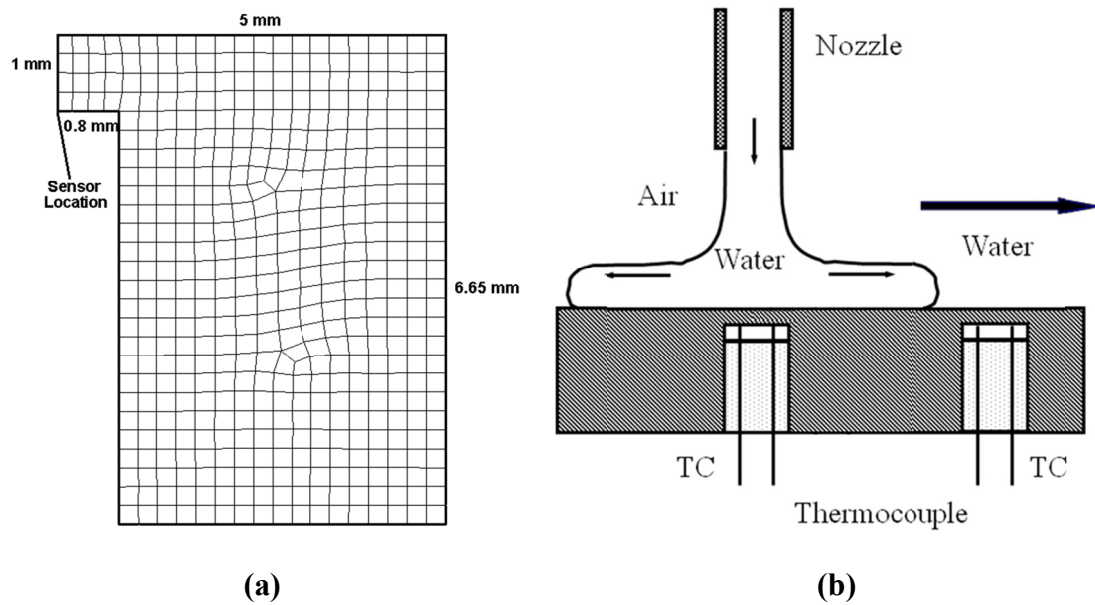
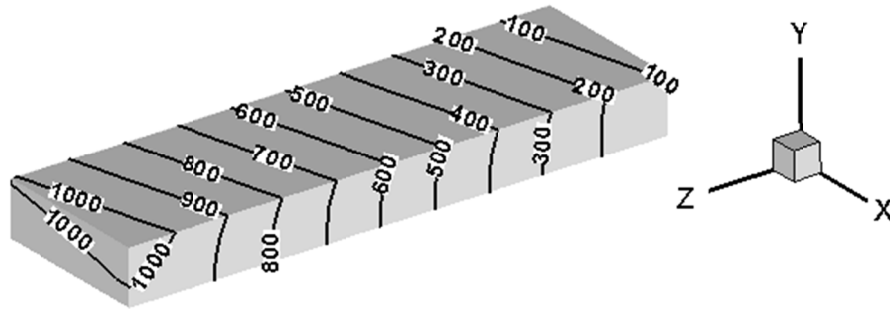


Figure 2.6. Test Case II; (a) Model, (b) Problem Description [1]

### **Test Case III: 3D Steady Problem**

The third test case is a 3D steady heat conduction problem in a slab, with dimensions of 3, 1, and 10, in the  $x$ ,  $y$ , and  $z$  directions, respectively. The top surface of the slab ( $y = 1$ ) is subjected to a specified temperature boundary condition that varies as  $T_{Top} = 25x + 100z + 10$ .

The other boundaries are adiabatic. The material properties are the same as in the previous test case. Figure 2.7 displays the temperature distribution inside the slab, obtained by 3D finite element modeling.



**Figure 2.7. Temperature Distribution Inside the Slab**

Unlike the previous test cases where there was only one data point in the solution field, in this problem, there are three sensors located inside the domain. They are located at the centerline with respect to the  $x$  and  $y$  coordinates ( $y = 0.5$ ,  $x = 1.5$ ), and at the  $z$  locations of 2, 5, and 8.

#### **Test Case IV: 3D Transient Problem – Nine Thermocouples**

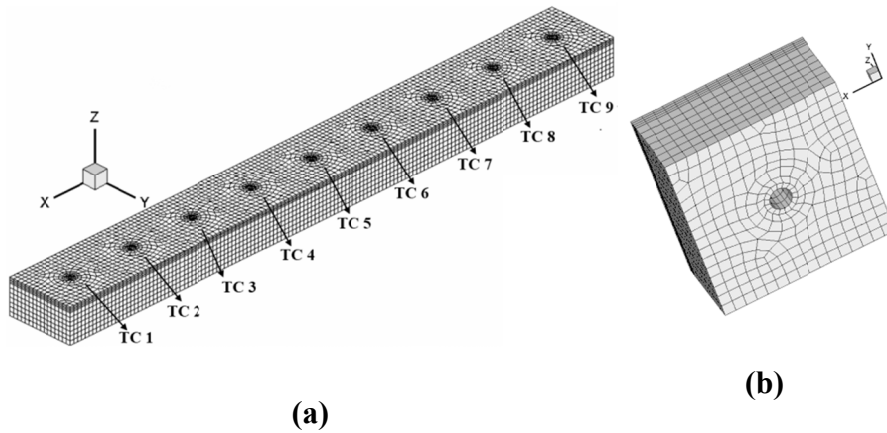
A block containing nine thermocouples is modeled for each pass of water jet cooling of a steel strip. The length of the block is 114.3 mm (9 sections of each 12.7 mm). The width and thickness are 12.7 mm and 6.65 mm, respectively. To model the thermocouple hole, a cylinder of radius 0.5 mm and height 5.65 mm is taken out of the block. Isoparametric eight-node brick elements are used to discretize the domain. Figure 2.8(a) shows the whole domain, and Figure 2.8(a) is a close-up view of one of the TC holes.

The boundary condition of the top surface is prescribed heat flux which is chosen to resemble the one in water cooling of steel strips. Figure 2.9(a) shows the applied heat flux at five of the nine thermocouple locations. It is very similar to an actual heat flux happening on a run-out table with two rows of staggered circular jets, impinging on the third and seventh thermocouple locations [91]. Figure 2.9(b) is the temperature history at five of the nine thermocouple locations inside the plate, obtained from direct finite element simulation. The direct finite element code has been validated, and grid independence studies were performed, i.e. the mesh was refined until the changes in the results were less than 5%. However, due to space limitations, we are not presenting them here. The other boundaries are assumed to be adiabatic.

The density,  $\rho$ , is  $7850 \text{ kg/m}^3$ ,  $C_p$  is  $475 \text{ J/KgK}$ , and the thermal conductivity,  $k$ , is first assumed to be constant and equal to  $40 \text{ W/m} \cdot ^\circ\text{C}$  (linear problem), and later changed to be depending on temperature (nonlinear problem) as

$$k = 60.571 - 0.03849 \times T \text{ W/m} \cdot ^\circ\text{C} \quad (2.42)$$

These are the physical properties of the steel strips that are used in our controlled cooling experiment. Results are obtained at the top of the cylindrical hole, which is the assumed position of a thermocouple. Inverse analysis is conducted to obtain the transient heat flux profile at the top surface.



**Figure 2.8. (a) The Whole Block Consisting of Nine Thermocouples; (b) A Close View of the TC Hole from the Bottom**

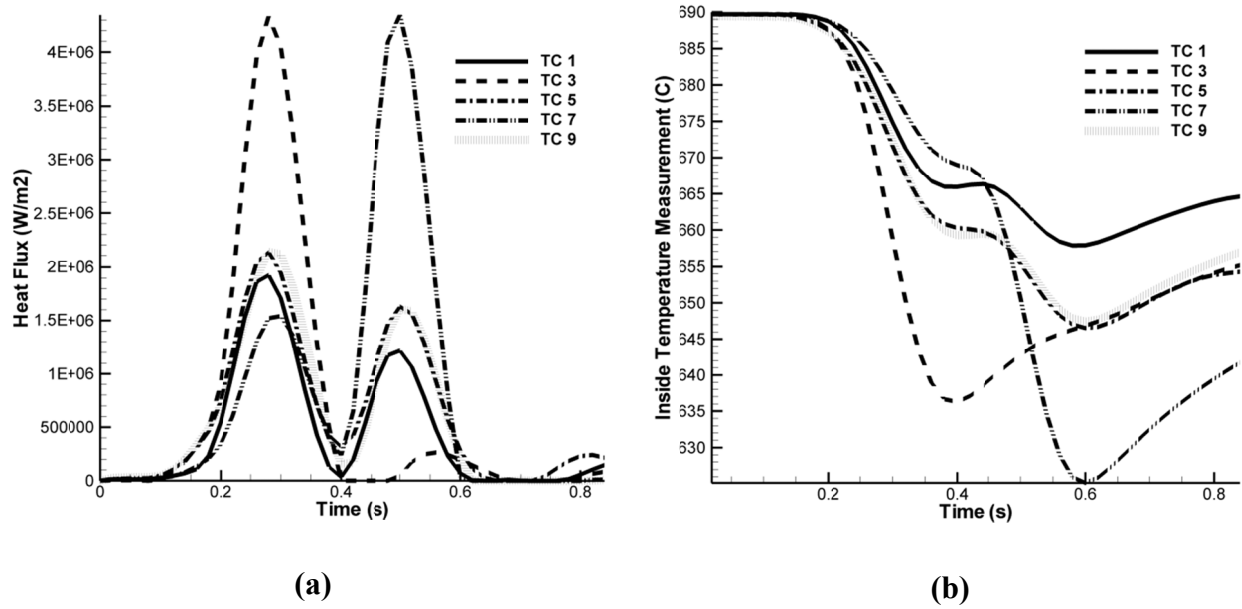


Figure 2.9. (a) The Applied Heat Flux on the Top Surface; (b) The Thermocouple Readings Used for Inverse Analysis

### **Test Case V: 3D Transient Problem – Eighteen Thermocouples**

Another row of thermocouples is added to the previous test case in the  $y$ -direction, as shown in Figure 2.10, increasing the width of the block to 25.4 mm. Again the top surface boundary condition is set to resemble the heat fluxes occurring in the water cooling of steel strips by impinging jets. However, in this test case, each thermocouple line is under a different line of impinging jets [91], i.e. there are two lines of nine thermocouples, and the heat flux applied on the top surface of each line is set to resemble the water jet cooling heat fluxes. Figure 2.11 shows the boundary heat fluxes on the top surface of five of the thermocouples and the temperature histories for the new line of added thermocouples. The main reason for using this test case is to study the effect of the number of thermocouples on the performance of the designed inverse analyzer, especially to investigate the effect of multi-criteria objective function formulation, which will be discussed in the next chapter.



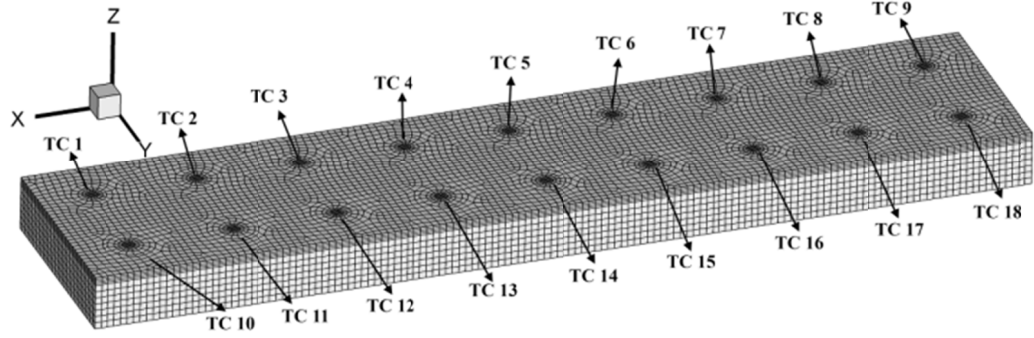


Figure 2.10. Test Case V; A Block Consisting of Eighteen Thermocouple Zones

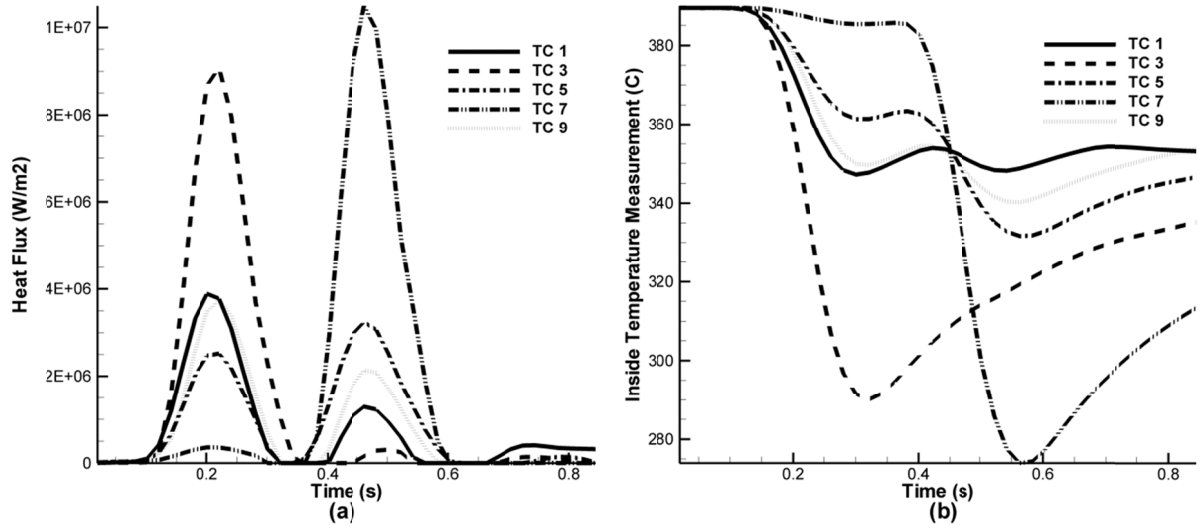


Figure 2.11. (a) The Applied Heat Flux on the Top Surface; (b) The Thermocouple Readings Used for Inverse Analysis; Second Line of Thermocouples, Test Case V

### 2.3.2 Noisy Domains

To study the effects of measurement errors in internal temperatures on the inversely calculated boundary conditions, random errors are imposed onto the calculated exact internal temperatures with the following equation:

$$T_m = T_{exact} + \sigma \cdot r \quad (2.43)$$

where  $T_m$  is the virtual internal temperature that is used in the inverse calculations instead of the exact temperature,  $T_{exact}$ ;  $r$  is a normally distributed random variable with zero mean and unit

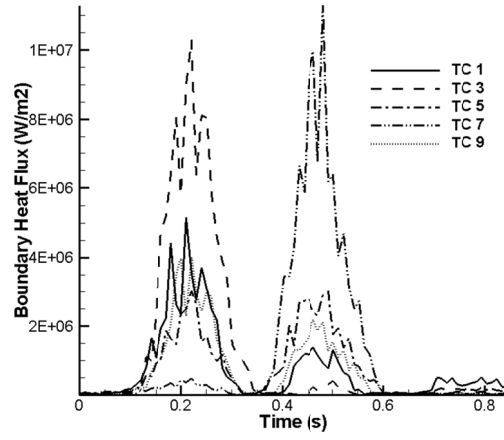
standard deviation; and  $\sigma$  is the standard deviation. In this work, the maximum additive random error is  $\pm 3$  °C.

There are several ways to make an inverse algorithm more stable when dealing with noisy data. For example, Gadala and Xu [57] have shown that increasing the number of “*future time steps*” in their sequential function specification algorithm results in greater stability. They have also demonstrated that increasing the regularization parameter,  $\alpha$ , improves the ability of the algorithm to handle noisy data. However, these approaches were also shown to greatly increase the required number of iterations, and in many cases the solution may diverge.

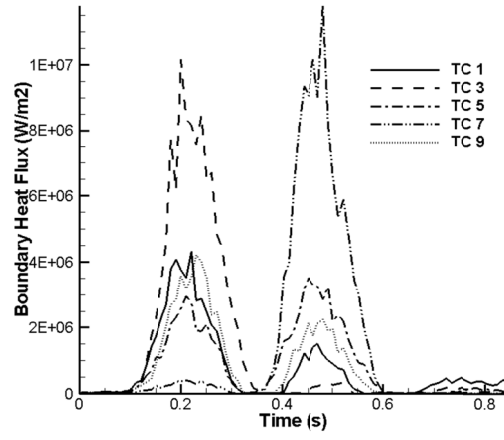
### 2.3.3 Effects of the Regularization Parameter and Future Time Steps

Both the regularization method, proposed by Tikhonov, and the future time steps concept in sequential function specification, proposed by Beck, have the same purpose of stabilizing the solution of the inverse heat conduction problem. In this research, these two methods have been combined, and their effects cannot be separated from each other. This approach is based on the idea of a regularized function specification method, and is shown to be more effective and more computationally efficient than using just one of these methods [104].

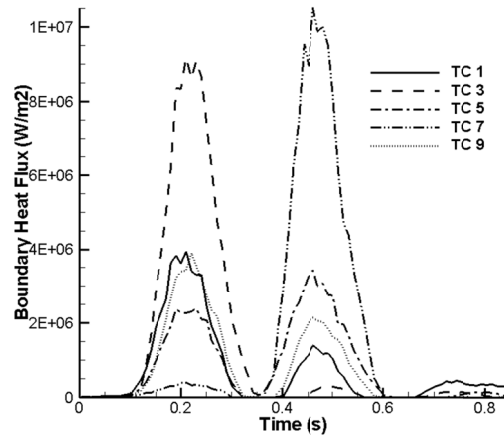
Notice that the future time steps concept is only applicable to the transient problems, so in test case III, which is a steady-state problem, only the Tikhonov regularization can be used. For the other test cases, while the exact value of the regularization parameter and the number of future time steps vary slightly, the same trends are always observed. We chose test case IV to show the effect of using a regularization parameter in Figure 2.12, and the effect of using more future time steps in Figure 2.13. As seen in Figure 2.12, increasing the value of the regularization coefficient results in a smoother profile for the predicted heat flux. Increasing the number of future steps has the same effect, as seen in Figure 2.13. These two effects can be combined. However, there are two main problems in using these techniques. First of all, the computational expense increases significantly when a higher number of future time steps are used. The other problem, which is more significant, is that there is a limit to the improvement that can be obtained by these two techniques. Increasing either of these parameters after a certain limit can cause divergence of the whole inverse solver.



(a)

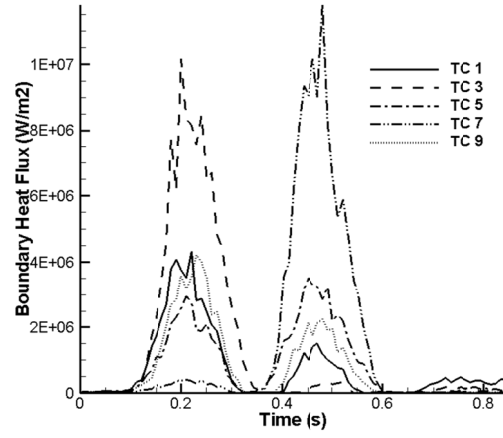


(b)

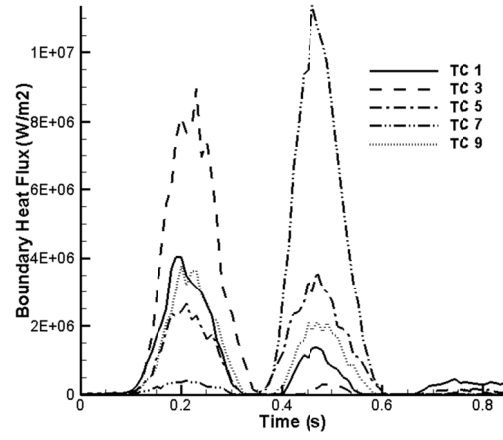


(c)

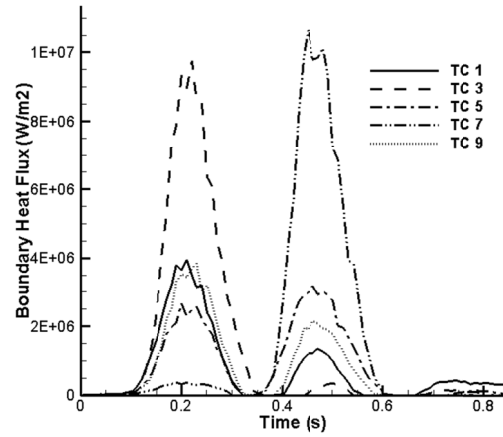
Figure 2.12. Effect of Regularization Parameter; 4 Future Steps; Error level =  $\pm 3^\circ\text{C}$ ; (a)  $\alpha = 1\text{e-}12$  ( $L_2$  Norm of Error =  $4.63\text{e}5$ ); (b)  $\alpha = 1\text{e-}11$  ( $L_2$  Norm of Error =  $2.56\text{e}5$ ); (c)  $\alpha = 1\text{e-}10$  ( $L_2$  Norm of Error =  $1.18\text{e}5$ ).



(a)



(b)

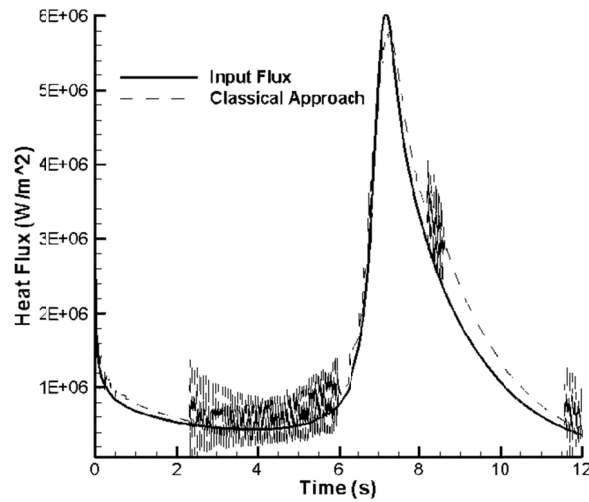


(c)

Figure 2.13. Effect of Future Time Steps;  $\alpha = 1e-11$ ; Error Level =  $\pm 3^\circ\text{C}$ ; (a) 4 Steps ( $L_2$  Norm of Error =  $4.63e5$ ); (b) 6 Steps ( $L_2$  Norm of Error =  $1.72e5$ ); (c) 8 Steps ( $L_2$  Norm of Error =  $1.34e5$ )

### 2.3.4 Effect of Time Step Size

It is well known that gradient based algorithms for solving inverse problems become unstable when the time step size becomes smaller than a certain limit. This is due to the ill-conditioning of the sensitivity matrix in smaller time steps. Basically, by reducing the time step size, huge non-physical oscillations are produced in the results (see Figure 2.14 for the oscillations that occurred in the solution of Test Case II). If we continue reducing the time step size, the whole program diverges.



**Figure 2.14. Threshold of Divergence in the Solution of Test Case II Due to Small Time Step Size**

This time step limitation can result in an unacceptably low temporal resolution, which may cause us to miss some of the main features of the heat flux profile. For example, in the cooling of steel products on the run-out table, if the sample is moving at a speed of 10 m/s, and the distance between two jet lines is 10 cm, we need time steps of less than 0.01 s, just in order to have one data point for each jet impingement moment. This is quite a low value for the gradient-based inverse solvers in problems with the thermophysical properties of steel. One way to overcome this problem is to use inverse solvers that are not gradient-based. This will be discussed in more details in the next two chapters.

### 2.3.5 Comparison of 1D, 2D and 3D Algorithms

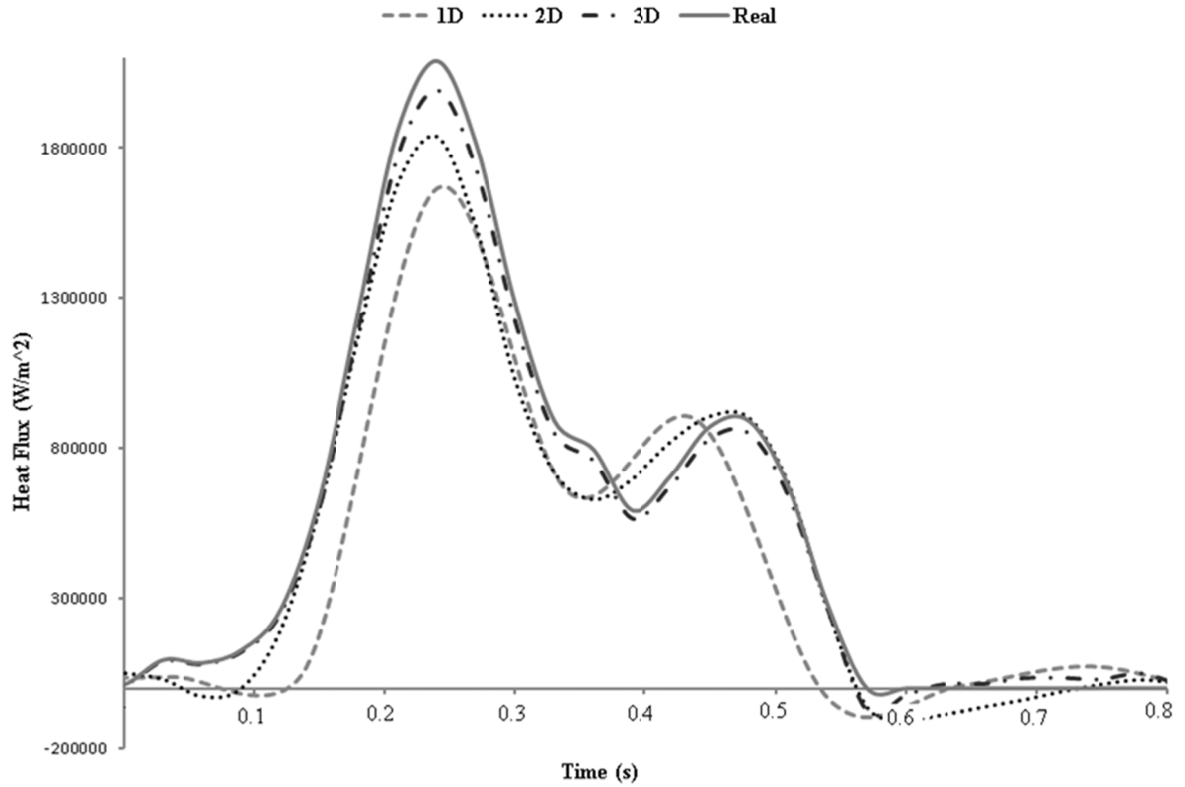
One of the reasons for using a 3D inverse solver instead of a 2D model [1] is the temperature gradients between different thermocouple locations. These gradients can exist in a

jet line between several impinging jets, or between two successive rows of jets. The simple 1D model assumes that the heat transfer is only happening in the normal direction to the plate. The 2D model also considers the heat flow between two neighbouring thermocouples. However, the 3D model is capable of modeling the heat flow in all directions, and is supposed to be a more physical simulation for the problem.

Due to different modeling assumptions, especially different treatment of the boundary conditions, the values of heat fluxes that will be obtained from the same set of temperature readings will be different for each of these models. For example, because in the 1D and 2D models, one or two sides of the domain are assumed adiabatic, i.e. there is no heat exchange through them, the heat flux in the cooling process needs to cool down a smaller thermal mass, so the values that will be obtained in these cases for the peak heat flux will be lower.

To investigate this, a set of known boundary heat fluxes are applied to the top surface of test case V, and the problem is directly simulated. The temperatures at the internal sensor locations are calculated and stored. Then the 1D, 2D, and 3D inverse algorithms are employed to use the stored temperatures and recover the missing boundary conditions. The time step size in the initial direct problem is selected to be smaller than the one used in inverse algorithms, in order to avoid the so-called “inverse crime” [105].

Figure 2.15 shows the result of the 1D, 2D, and 3D algorithms, in addition to the known applied heat flux. Note that due to the ill-posed nature of the inverse problem, even in the case of the 3D algorithm, some discrepancy between the expected heat flux and the results is observed. However, the 1D and 2D algorithms show a significant dampening in the predicted value of the heat flux. As mentioned previously, these algorithms study each thermocouple location in complete isolation (1D), or just consider the heat transfer in one lateral direction to the neighboring thermocouple(s) (2D). This means that a lower amount of thermal mass needs to be cooled. Thus, the predicted heat fluxes are smaller. Also, some of the details of the heat flux profiles, e.g. the shoulder before the first rise, and the small shoulder after it are not sensed in the 1D and 2D models. These effects are caused by the cooling of the neighboring thermocouple locations, and are not sensed in these algorithms.



**Figure 2.15. Results of the 1D, 2D, and 3D Inverse Algorithms in Predicting a Known Heat Flux Applied to a Direct 3D Problem**

## 2.4 Conclusion

In this chapter, equations for the direct and inverse simulation of the heat conduction problem were introduced, and a gradient-based inverse solver was extended to 3D to obtain the missing boundary conditions based on the readings of internal thermocouples, especially for applications in the controlled cooling of steel on a run-out table. The results show that the method is very sensitive to measurement errors, and becomes unstable when small time steps are used. In the next chapter, we will try to find algorithms that are capable of solving the inverse heat conduction problem without the shortcomings of the gradient-based methods.

### 3. Gradient-Free Methods of Solving the Inverse Heat Conduction Problem

As shown in Chapter-2, the gradient-based methods for solving the inverse heat conduction problem suffer from some shortcomings, such as instability in the presence of noise and in the smaller time step sizes. This can hinder their application in real-world problems, such as in the characterization of the jet impingement boiling heat transfer on the run-out table in a steel mill, where huge changes in the boundary conditions happen in short intervals, and the thermocouple readings are always erroneous. In this chapter, we try to investigate some gradient-free optimization methods and utilize them in the inverse heat conduction problem to mitigate these shortcomings, and to find a more suitable inverse solver for our applications. Three major techniques that are studied are: the artificial neural networks (ANN), the genetic algorithm (GA), and the particle swarm optimization (PSO). First, we introduce the basic algorithm and its formulation. Then, we test it in the solution of an inverse heat conduction problem, and try to optimize the performance parameters associated with that method to make it both more efficient, i.e. requiring lower computational expense, and more effective in handling noisy data. Finally, comparisons are made between different methods and their variations, and their advantages and disadvantages are studied. Also, in the last part of this chapter, a general modification is introduced that can be applied to the stochastic methods in inverse problems to accelerate their convergence.

#### 3.1. Inverse Formulation

The boundary inverse heat conduction problem can be formulated as an optimization problem in which we try to minimize the norm of difference between the experimentally measured temperatures and the calculated temperatures obtained from a direct solution of the problem with some guessed boundary conditions. In order to dampen the oscillations in the solution, and make the algorithm more stable in dealing with noisy measurement data, it is very common to include some kind of regularization terms that make the problem more stable, especially in the case of noisy domains. As mentioned in the previous chapter, based on previous experience [76], as well as our own trial of different regularization terms, we chose to use the heat flux values (zeroth-order regularization). The objective function will then be



$$f(\mathbf{q}) = \sum_{j=1}^J \left( \sum_{i=1}^N (T_j^{i,meas} - T_j^{i,calc})^2 + \alpha \sum_{i=1}^N (q_j^i)^2 \right) \quad (3.1)$$

where  $T_j^{i,meas}$  and  $T_j^{i,calc}$  are the values of expected (measured) and calculated temperatures at the  $i^{\text{th}}$  time step and the  $j^{\text{th}}$  spatial location, respectively;  $\alpha$  is the regularization coefficient; and  $q_j^i$  is the boundary heat flux vector at the  $i^{\text{th}}$  time step and the  $j^{\text{th}}$  spatial location,  $N$  is the total number of time steps, and  $J$  is the number of spatial components of temperature vectors.

### 3.2. Artificial Neural Networks

Artificial Neural Networks (ANN) are motivated by the efficiency of brain in performing computations. These networks are made of a large number of processing units (neurons) that are interconnected through weighted connections, similar to synapses in the brain. In order for the network to perform the expected tasks, it should first go through a “learning” process. There are two main categories of learning: supervised, or unsupervised. In supervised learning, the network learning is achieved by practicing on pre-designed training sets, while in unsupervised learning, the network is presented with a set of patterns, and learns to group these patterns into certain categories. The supervised learning is useful in function fitting and prediction, while unsupervised learning is more applicable to pattern recognition and data clustering. Since the learning process in our application is a supervised one, we focus on this type of learning process.

While there are several major classes of neural networks, in this research we studied only two of them, which are introduced in this section.

#### 3.2.1. Feedforward Multilayer Perceptrons (FMLP)

In a feedforward network, the nodes are arranged in layers, starting from the input layer, and ending with the output layer. In between these two layers, a set of layers called hidden layers, are present, with the nodes in each layer connected to the ones in the next layer through some unidirectional paths. See Figure 3.1 for a presentation of the topology. It is common to have a different number of elements in the input and output vectors. These vectors can occur either concurrently (order is not important), or sequentially (order is important). In inverse heat transfer applications, normally the order of elements is important, so sequential vectors are used.

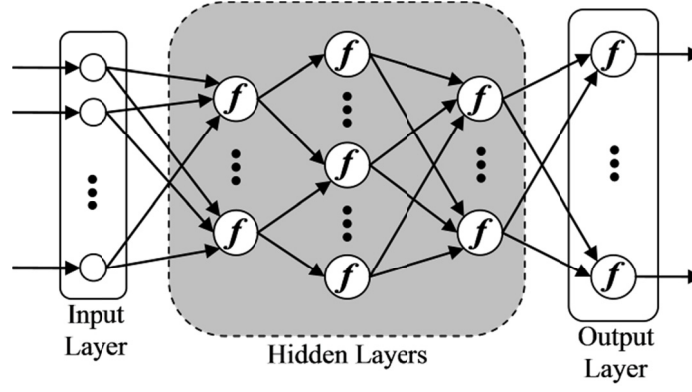


Figure 3.1. A Feedforward Network Topology

### 3.2.2. Radial Basis Function Networks (RBFN)

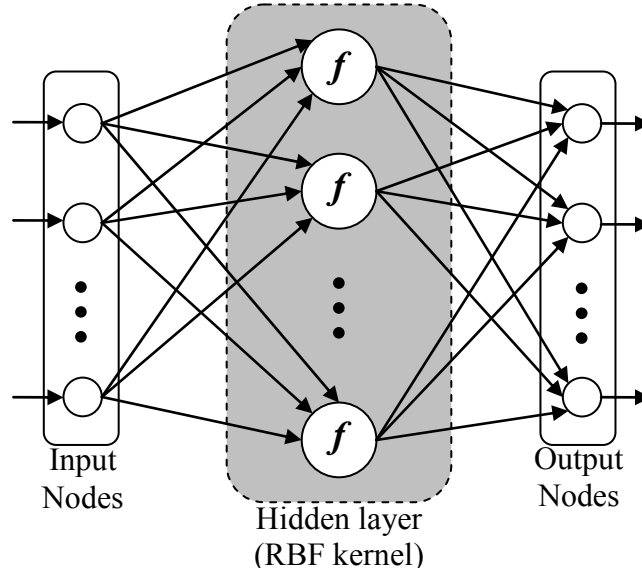
The basic RBFN includes only an input layer, a single hidden layer, and an output layer. See Figure 3.2 for a visual representation. The form of the radial basis function can be generally given by

$$f_i(\mathbf{x}) = r_i \left( \frac{\|\mathbf{x} - \mathbf{v}_i\|}{\sigma_i} \right) \quad (3.2)$$

in which  $\mathbf{x}$  is the input vector, and  $\mathbf{v}_i$  is the vector denoting the center of the receptive field unit  $f_i$  with  $\sigma_i$  as its unit width parameter. The most popular form of this function is the Gaussian kernel function, given as

$$f_i(\mathbf{x}) = \exp \left( \frac{-\|\mathbf{x} - \mathbf{v}_i\|^2}{2\sigma_i^2} \right) \quad (3.3)$$

These networks normally require more neurons than the feedforward networks, but they can be designed and trained much faster. However, in order to have good performance, the training set should be available in the beginning of the process.



**Figure 3.2. An RBF Network Topology**

### 3.2.3. Implementation in Inverse Heat Conduction Problem

In order to use the artificial neural networks in the inverse heat conduction problem, we first started with a direct heat conduction finite element code, and applied several sets of heat fluxes in the boundary. The resulting temperatures in locations inside the domain, which correspond to the thermocouple locations in the experiments, were obtained. The neural network was then trained using the internal temperature history as an input, and the corresponding applied heat flux as the target. The assumption was that this way, the neural network should be able to act as an inverse analysis tool, and given a set of measured thermocouple readings, be able to reproduce the heat fluxes.

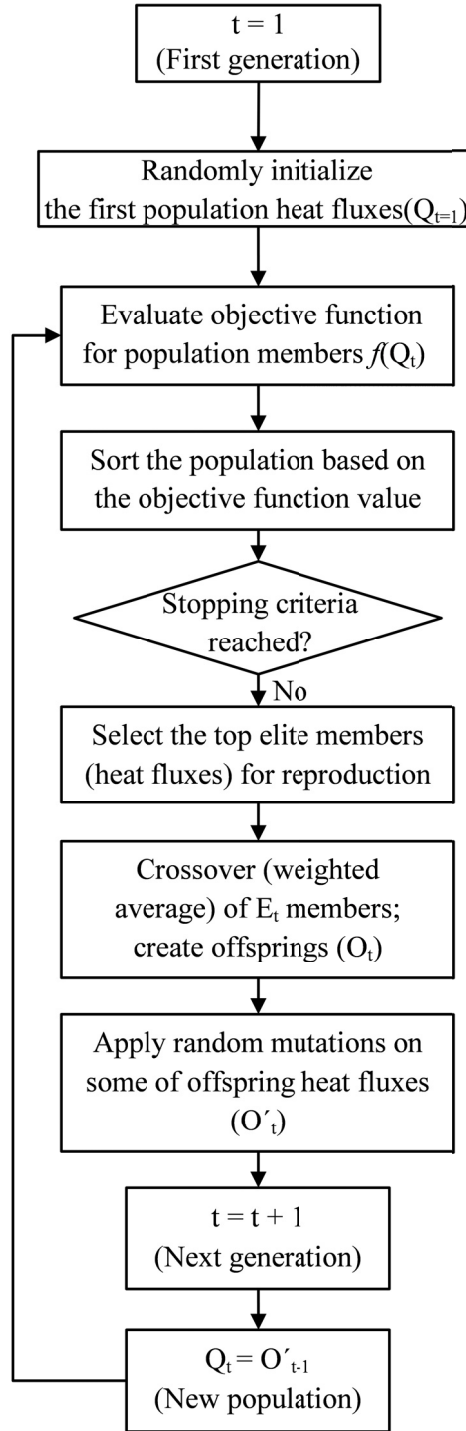
The obtained results were, however, far from satisfactory. It seemed that the relationship between the actual values of temperatures and heat fluxes is a complicated one, which is very hard for the neural networks to understand and simulate, at least when using a reasonably small number of layers. Thus, we decided to reformulate the problem, and use the change in the temperature in each time step as the input. In this formulation, neural networks performed much better, and a good quality was achieved in the solution in a reasonable amount of time.

Further investigations showed that if the time step size is varying, we can use a derivative of temperature with respect to the heat flux as the input, i.e. divide the temperature change by the time step size. The results were again satisfactory; however, more bookkeeping is needed, which complicates the implementation and makes the algorithm more prone to coding errors. This

practice is not normally recommended, unless it can result in a considerable reduction in the solution time.

### **3.4. Genetic Algorithm (GA)**

Genetic algorithm is probably the most popular stochastic optimization method. It is also widely used in many heat transfer applications, including inverse heat transfer analysis [106]. Figure 3.3 shows a flowchart of the basic GA. Similar to PSO, GA starts its search from a randomly generated population. This population evolves over successive generations (iterations) by applying three major operations. The first operation is “Selection”, which mimics the principle of “Survival of the Fittest” in nature. It finds the members of the population with the best performance, and assigns them to generate the new members for future generations. This is basically a sort procedure based on the obtained values of the objective function. The number of elite members that are chosen to be the parents of the next generation is also an important parameter. Usually, a small fraction of the less fit solutions are also included in the selection, to increase the global capability of the search, and prevent a premature convergence. The second operator is called “Reproduction” or “Crossover”, which imitates mating and reproduction in biological populations. It propagates the good features of the parent generation into the offspring population. In numerical applications, this can be done in several ways. One way is to have each part of the array come from one parent. This is normally used in binary encoded algorithms. Another method that is more popular in real encoded algorithms is to use a weighted average of the parents to produce the children. The latter approach is used in this research. The last operator is “Mutation”, which allows for global search of the best features, by applying random changes in random members of the generation. This operation is crucial in avoiding the local minima traps. More details about the genetic algorithm may be found in references [107,108].



**Figure 3.3. Flowchart of a General Implementation of Genetic Algorithm (GA)**

Among the many variations of GAs, in this study, we use a real encoded GA with roulette selection, intermediate crossover, and uniform high-rate mutation [107]. The crossover

probability is 0.2, and the probability of adjustment mutation is 0.9. These settings were found to be the most effective based on our experience with this problem. A mutation rate of 0.9 may seem higher than normal. This is because we start the process with a random initial guess, which needs a higher global search capability. However, if smarter initial guessing is utilized, a lower rate of mutation may be more effective. Genes in the present application of GA consist of arrays of real numbers, with each number representing the value of the heat flux at a certain time step, or a spatial location.

### 3.5. Particle Swarm Optimization

We start this section by giving a description of the basic concepts of the algorithm. Then a brief description of the three variations of the PSO algorithm that are used in this study is given. Finally we investigate some modifications in the PSO algorithm to make it a more robust and efficient solver of the inverse heat conduction problem.

#### 3.5.1. Basic Concepts

Particle swarm optimization (PSO) is a high-performance stochastic search algorithm that can also be used to solve inverse problems. The method is based on the social behavior of species in nature, e.g., a swarm of birds or a school of fish. It was originally developed by Eberhart and Kennedy in 1995 [68].

In the basic PSO algorithm, if a member of the swarm finds a desirable position, it will influence the traveling path of the rest of the swarm members. Every member searches in its vicinity, and not only learns from its own experience (obtained in the previous iterations), but also benefits from the experiences of the other members of the swarm, especially from the experience of the best performer. The original PSO algorithm includes the following components [109]:

- *Particle Position Vector  $\mathbf{x}$* : For each particle, this vector stores its current location in the search domain. These are the values for which the value of the objective function is calculated, and the optimization problem is solved.
- *Particle Velocity Vector  $\mathbf{v}$* : For every particle, this vector determines the magnitude and direction of change in the position of that particle in the next iteration. This is the factor that causes the particles to move around the search space.

- *Best Solution of a Particle  $p$* : For each particle, this is the position that has produced the lowest value of the objective function (the best solution with the lowest error in our case). So if  $f$  is the objective function that is supposed to be minimized,  $i$  is the index for each particle, and  $m$  is the iteration counter, then:

$$p_i^m = \arg \min_{0 \leq s \leq m} (f(x_i^s)) \quad (3.4)$$

- *Best Global Solution  $g$* : This is the best single position found by all particles of the swarm, i.e., the single  $p$  point that produces the lowest value for the objective function, among all the swarm members. In other words, if  $n$  is the swarm size, then:

$$g^m = \arg \min_{0 \leq s \leq m, 1 \leq k \leq n} (f(x_k^s)) \quad (3.5)$$

The number of particles in the swarm ( $n$ ) needs to be specified at the beginning. Fewer particles in the swarm results in lower computational effort in each iteration, but possibly a higher number of iterations is required to find the global optimum. On the other hand, a larger population will have a higher computational expense in each iteration, but will likely require less iterations to reach the global optimum point. Earlier studies have shown that a smaller population is normally preferred [110,111]. This was also observed in our study; however, its effect on the convergence speed seems to be insignificant.

The steps involved in the basic PSO algorithm are detailed below [109]:

1. Randomly initialize the positions and velocities for all the particles in the swarm.
2. Evaluate the fitness of each swarm member (objective function value at each position point).
3. At iteration  $m$ , the velocity of the particle  $i$ , is updated as:

$$v_i^{m+1} = c_0 v_i^m + c_1 r_1 (p_i^m - x_i^m) + c_2 r_2 (g^m - x_i^m) \quad (3.6)$$

where  $x_i^m$  and  $v_i^m$  are the position and velocity of particle  $i$  at the  $m$ -th iteration, respectively;  $p_i^m$  and  $g^m$  are the best positions found up to now by this particle (local memory) and by the whole swarm (global memory) so far in the iterations, respectively;  $c_0$  is called the inertia coefficient or

the self-confidence parameter and is usually between zero and one;  $c_1$  and  $c_2$  are the acceleration coefficients that pull the particles toward the local and global best positions; and  $r_1$  and  $r_2$  are random vectors in the range of (0,1). The ratio between these three parameters controls the effect of the previous velocities and the trade-off between the global and local exploration capabilities.

4. Update the position of each particle using the updated velocity and assuming unit time:

$$x_i^{m+1} = x_i^m + v_i^{m+1} \quad (3.7)$$

5. Repeat (2) – (4) until a convergence criterion (an acceptable fitness value or a certain maximum number of iterations) is satisfied.

The first step of the above algorithm may be improved, if instead of using random numbers, we initialize the algorithm with some meaningful initial guess, i.e. a set of values close to the ones that we expect the final solution to be. We chose not to do so in this research, because we wanted to compare the efficiency of different algorithms, and for that reason, decided to let them perform with minimum supervision. On the other hand, efficiency of the gradient based algorithms is less affected by the values of the initial guess. However, in those methods one may fall into the local minima of the function. The iterations in this research were carried out until the best value of the objective function normalized with respect to the best value of the objective function in the first iteration was less than  $10^{-6}$  or the average value of the best objective function became less than the variance of added artificial noise.

There are some considerations that must be taken into account when updating the velocity of the particles (step 3 of the above algorithm). First, we need a value for the maximum velocity. A rule of thumb requires that, for a given dimension, the maximum velocity,  $v_{i,\max}$ , should be equal to one-half the range of possible values for the search space. For example, if the search space for a specific dimension is the interval  $[0, 100]$ , we will take a maximum velocity of 50 for this dimension. If the velocity obtained from Equation (3.6) is higher than  $v_{i,\max}$ , then we will substitute the maximum velocity with  $v_i^{m+1}$ . The reason for having this maximum allowable velocity is to prevent the swarm from “*explosion*” (divergence). Another popular way of preventing divergence is a technique called “*constriction*”, which dynamically scales the velocity update [109]. The first method was used in the previous research by the authors [112].



However, further investigation showed that a better performance is obtained when combining the constriction technique while limiting the maximum velocity. In this research, the velocity updates are done using constriction and can be written as:

$$v_i^{m+1} = K \left( v_i^m + c_1 r_1 (p_i^m - x_i^m) + c_2 r_2 (g^m - x_i^m) \right) \quad (3.8)$$

where  $K$  is the constriction factor, and is calculated as [109]:

$$K = \frac{2}{\left| 2 - \varphi - \sqrt{\varphi^2 - 4\varphi} \right|} \quad (3.9)$$

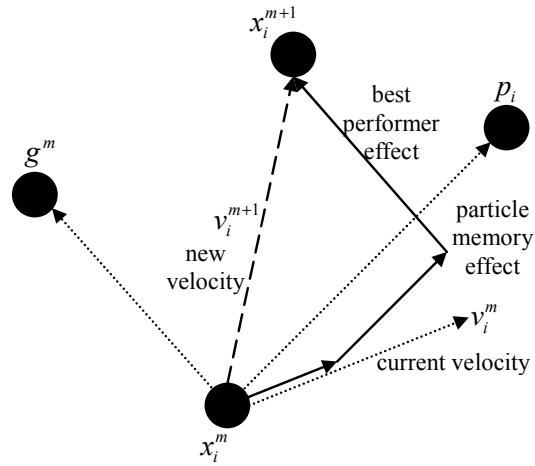
where  $\varphi = c_1 + c_2$ . Here, following the recommendations by Clerc [109], the initial values for  $c_1$  and  $c_2$  are set to 2.8 and 1.3, respectively. These values will be modified in subsequent iterations, as discussed below.

As mentioned above, the relation between the self-confidence parameter,  $c_0$ , and the acceleration coefficients,  $c_1$  and  $c_2$ , determines the trade-off between the local and global search capabilities. When using the constriction concept, the constriction factor is responsible for this balance. As we progress in time through iterations, we get closer to the best value. Thus, a reduction in the value of the self-confidence parameter will limit the global exploration, and a more localized search will be performed. In this research, if the value of the best objective function is not changed by a certain number of iterations (10 iterations in our case), the value of  $K$  is multiplied by a number less than one (0.95 for our problems) to reduce it (i.e.  $K^{new} = 0.95K^{old}$ ). These numbers are mainly based on the authors' experience, and the performance is not very sensitive to their exact values. Some other researchers have used a linearly decreasing function to make the search more localized after the first few iterations [110]. These techniques are called “*dynamic adaptation*”, and are very popular in recent implementations of PSO [113].

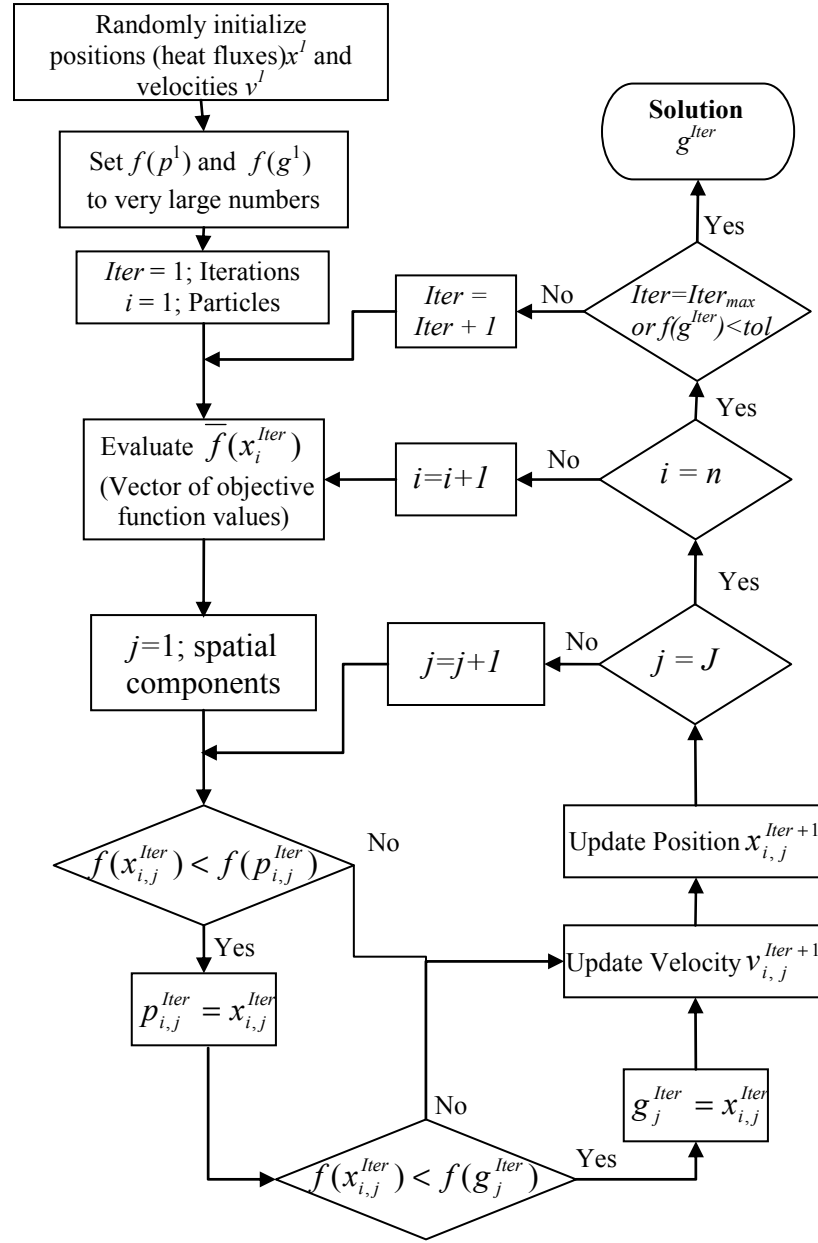
Also, in updating the positions, one can impose a lower and upper limit for the values, usually based on the physics of the problem. For example, in a heat transfer application, if we know that cooling is happening at the boundary, then the heat fluxes are going to be negative, and we can assign an upper limit of zero for these values. If the position values fall outside this range, several treatments are possible. In this research, we set the value to the limit that has been passed by the particle. Other ideas include substituting that particle with a randomly chosen

particle in the swarm, or penalizing this solution by increasing the value of the objective function.

The above steps are illustrated in Figure 3.4. Figure 3.5 shows a flowchart of the whole process of basic PSO algorithm for each time step.



**Figure 3.4. Velocity and Position Updates in the Basic Particle Swarm Optimization Algorithm**



**Figure 3.5. Flowchart of the Basic Particle Swarm Optimization Procedure**

### 3.5.2. Variations

Unfortunately, the basic PSO algorithm may get trapped in a local minimum, which can result in a slow convergence rate, or even premature convergence, especially for complex problems with many local optima. Therefore, several variants of PSO have been developed to improve the performance of the basic algorithm [114]. Some variants try to add a chaotic acceleration factor to the position update equation, in order to prevent the algorithm from being

trapped in local minima [115]. Others try to modify the velocity update equation to achieve this goal. One of these variants is called the Repulsive Particle Swarm Optimization (RPSO), and is based on the idea that repulsion between the particles can be effective in improving the global search capabilities and finding the global minimum [70,116]. The velocity update equation for RPSO is

$$v_i^{m+1} = c_0 v_i^m + c_1 r_1 (p_i^m - x_i^m) + c_2 r_2 (p_j^m - x_i^m) + c_3 r_3 v_r \quad (3.10)$$

where  $p_j^m$  is the best position of a randomly chosen particle among the swarm,  $c_3$  is an acceleration coefficient,  $r_3$  is a random vector in the range (0,1), and  $v_r$  is a random velocity component. Here  $c_2$  is -1.43, and  $c_3$  is 0.5. These values are based on recommendations by Clerc [109]. Based on our own experience, minor changes in these values do not significantly impact the performance. The newly introduced third term on the right-hand side of Eq. (3.10), with a negative coefficient ( $c_2$ ), causes repulsion between the particle and the best position of another randomly chosen particle. The third term's role is to prevent the population from being trapped in a local minimum. The fourth term generates noise in the particle's velocity in order to take the exploration to new areas in the search space. Once again, we are gradually decreasing the weight of the self-confidence parameter. Note that the third term on the right-hand side of Eq. (3.6), i.e., the tendency toward the global best position, is not included in a repulsive particle swarm algorithm in most of the literature.

The repulsive particle swarm optimization technique does not benefit from the global best position found. A modification to RPSO that also uses the tendency towards the best global point is called the "Complete Repulsive Particle Swarm Optimization" or CRPSO [112]. The velocity update equation for CPRSO will be:

$$v_i^{m+1} = c_0 v_i^m + c_1 r_1 (p_i^m - x_i^m) + c_2 r_2 (g^m - x_i^m) + c_3 r_3 (p_j^m - x_i^m) + c_4 r_4 v_r \quad (3.11)$$

In CRPSO, by having both an attraction toward the particle's best performance, and a repulsion from the best performance of a random particle, we are trying to create a balance between the local and global search operations.

### 3.5.3. Elite Particle and Elite Velocity

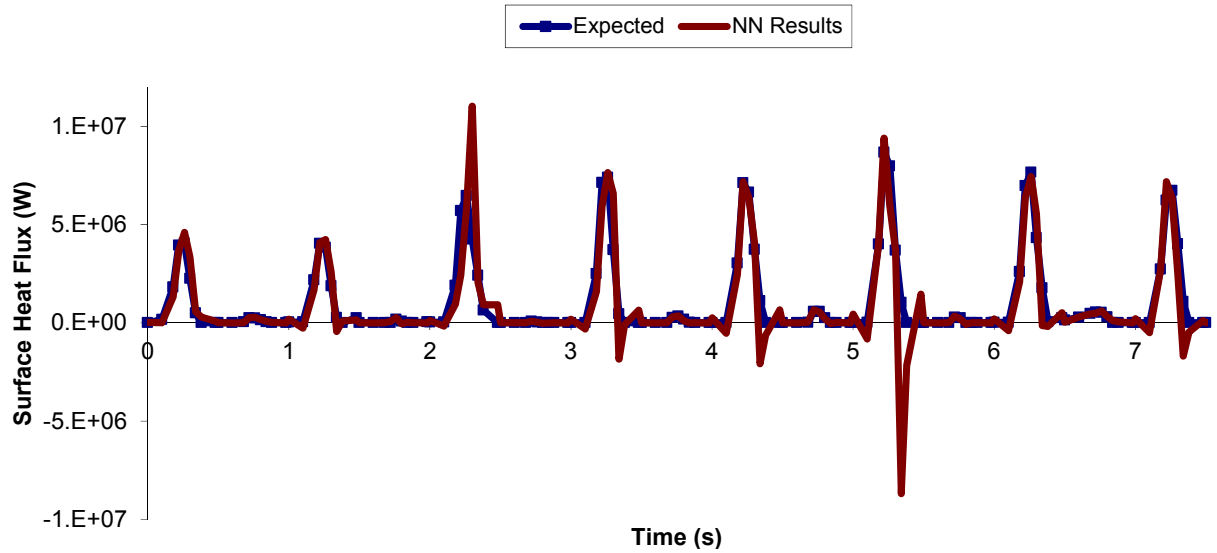
This modification has been proposed by Fourie and Groenwold [117] in structural optimization applications, and is borrowed from the genetic algorithm, where the gene that has the best characteristics never leaves the population. Here, we replace the particle with the worst value of objective function with the best global solution,  $g$ . Also, if a particle's velocity caused an improvement in the best global solution during the  $m^{\text{th}}$  iteration, it is allowed to continue with the same velocity, without being influenced by other particles. In other words, if  $v_k^m$  caused an improvement on  $g$ , then:

$$\begin{aligned} g &= x_k^m \\ x_k^{m+1} &= g + c_0 v_k^m \end{aligned} \tag{3.12}$$

### 3.6. Application in Inverse Heat Conduction

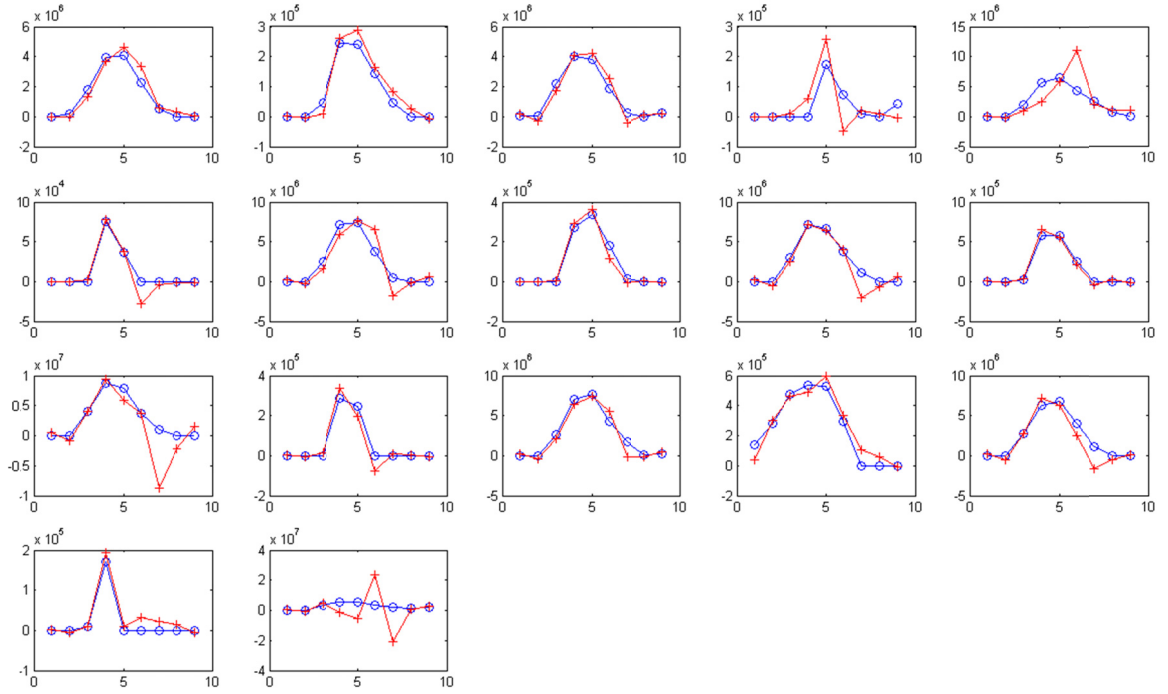
As the first step in our study of these techniques, we investigate their capability in solving a general inverse heat conduction problem similar to the problem on the run-out table, such as the test cases introduced in the previous chapter. In order to save space and since the results of PSO and GA are very similar, we only present the graphs of some of the test cases. However, other test cases will be used later in this manuscript to study the different modifications in the original algorithm.

We start by applying the artificial neural networks to the inverse heat conduction problem. This is different from GA and PSO, since those methods perform a stochastic search and are similar in many aspects, while the artificial neural networks are more like a correlation between the inputs and outputs. Figure 3.6 shows the result of the application of the radial basis function neural networks for the whole history of the heat fluxes on the runout table. Temperatures start at 700 °C and go down to 176 °C. The heat flux vs. time profile is plotted in Figure 3.6. As can be seen from this figure, neural networks are generally capable of dealing with the whole range of the cooling history.



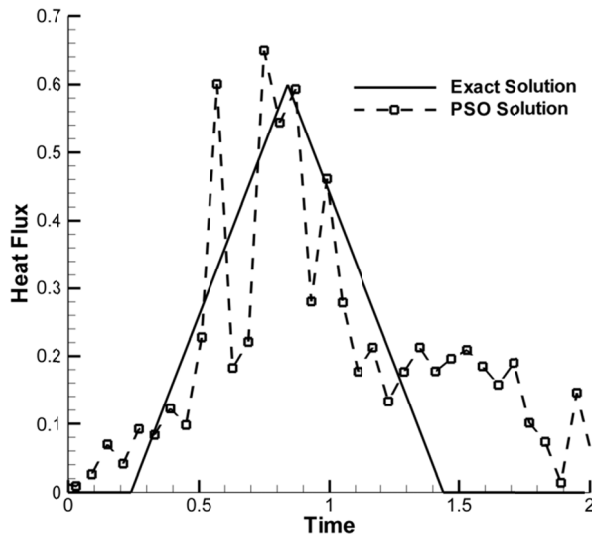
**Figure 3.6. Test Case II; Time History of Heat Fluxes in a Typical Run-Out Table Application; Expected Results (Squares) vs. the RBF Network Results (Line)**

However, this method has limitations, as observed in Figure 3.6, and in more details in Figure 3.7. The latter figure shows a close-up view of the peaks of heat flux that happen during the cooling process on the run-out table, i.e. the peaks in Figure 3.6. The circles are the expected heat flux, and the plusses are the result of NNs. The top left sub-figure is the first peak heat flux in time, and then it moves to the right, and then to the next row. Note that each even sub-figure ( $2^{\text{nd}}$ ,  $4^{\text{th}}$ , and so on) is a very smaller peak which is associated with the second row of jets. These peaks are not very obvious in Figure 3.6, due to the scaling. Going through the subfigures of heat fluxes, it is apparent that the success or failure of NNs is not that much related to the temperature range, or the magnitude of heat fluxes, but on the actual shape of the heat flux profile. If the heat flux has a clear thin peak and two tails before and after the peak, the NN is doing a good job. However, the existence of other details in the heat flux profile reduces the quality of the NN predictions. Also, considering the ill-posed nature of the problem, and all the complications that are involved, we can generally say that in most cases (about 75% of the cases) it does a decent job. Overall we can say that NNs are more useful in getting a general picture of the solution, rather than producing a very accurate and detailed answer to the IHCP. Also, it should be noted that if parameters such as the number of jets, plate speed, or the distance between jet rows are changed, we need to train a new neural network.

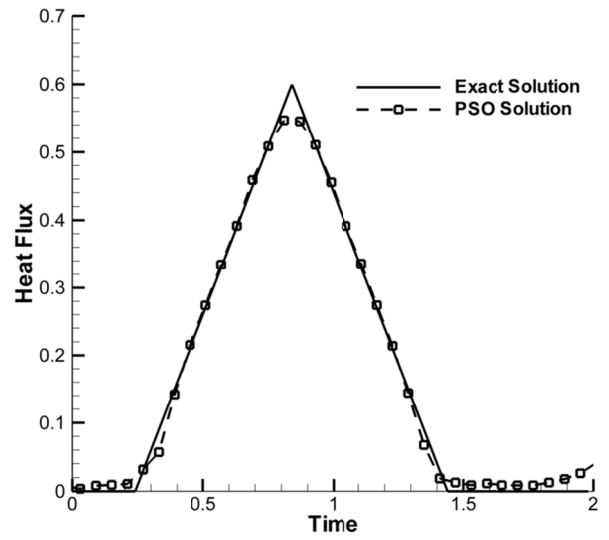


**Figure 3.7. Individual Heat Flux Peaks vs. Time from a Typical Run-Out Table Application; Expected Results (Circles) vs. the RBF Network Results (Pluses)**

On the other hand, GA and PSO algorithms show reasonably good predictions of the details of the missing boundary conditions. Notice that we still need to have some form of regularization for these methods to work properly. See Figure 3.8 and Figure 3.9 for the solution to test cases I and III. As mentioned before, the solutions to the other test cases will be shown in the next subsections, while studying the effect of modifications in the algorithms.

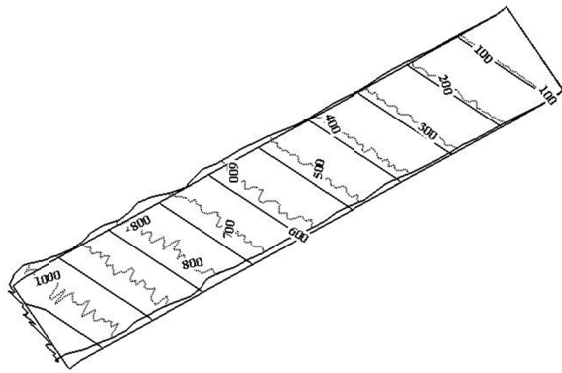


(a)

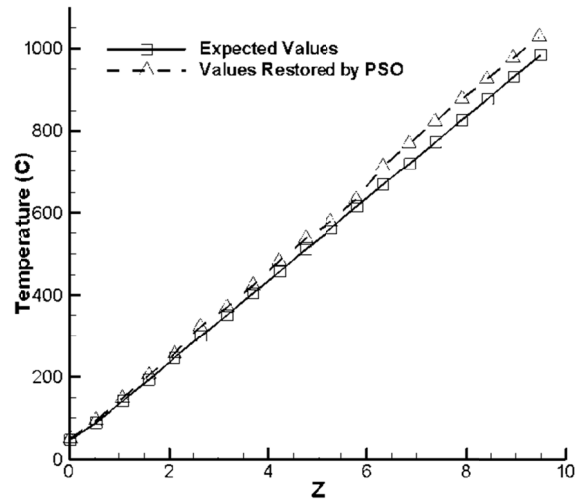


(b)

Figure 3.8. PSO Results, Test case I: (a) Without Regularization ( $L_2$  Norm of Error = 0.127),  
(b) With Regularization ( $L_2$  Norm of Error = 0.014)



(a)



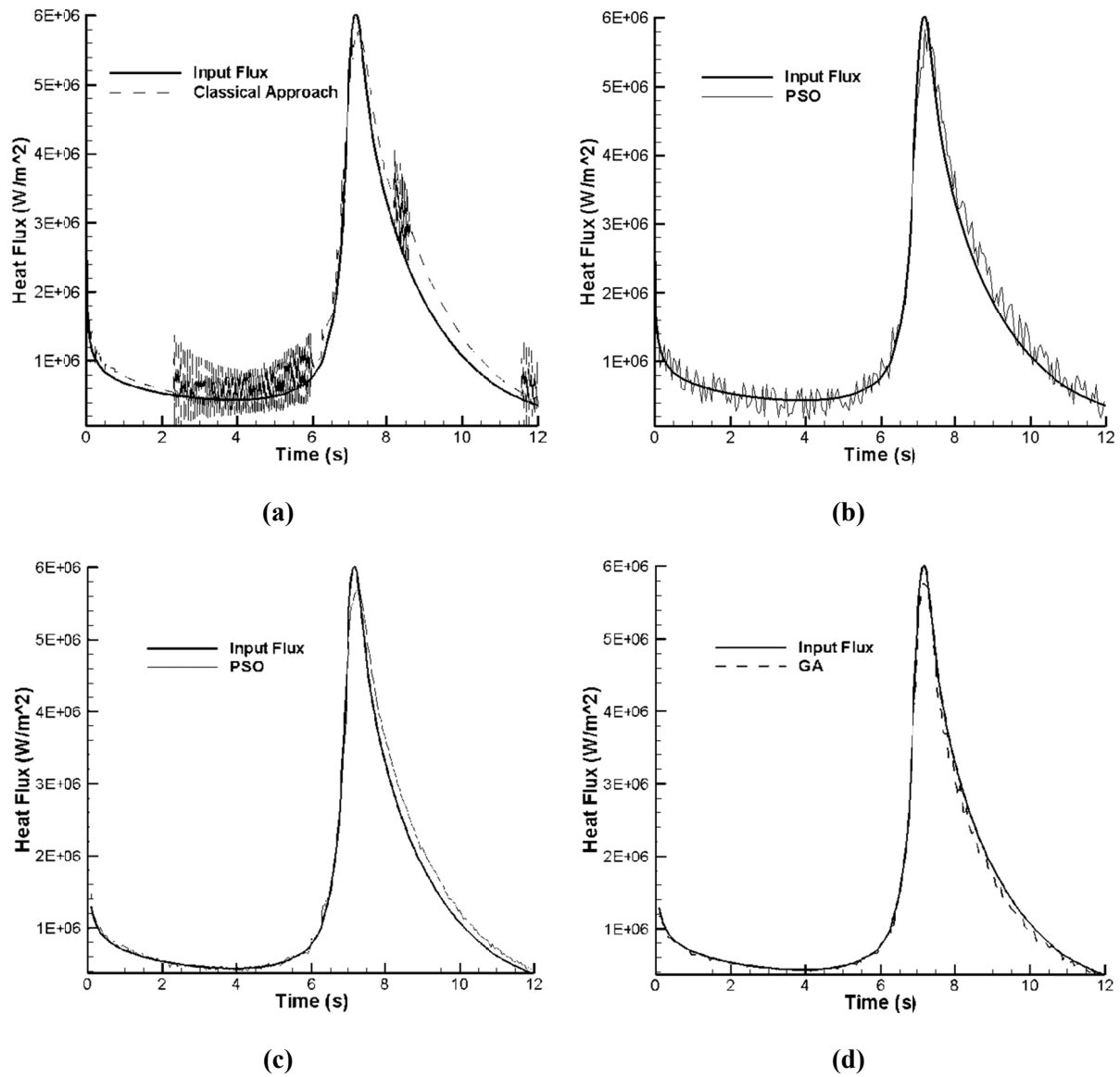
(b)

Figure 3.9. Test Case III; (a) The Top Plane ( $y = 1$ ) Temperature Contours; Solid Lines: Expected Input;  
Dotted Lines: Output of the Inverse Analysis; (b) Temperature Distribution (Expected and Restored by PSO)  
on the Top Surface ( $y=1$ ) Along the  $x=1.5$  Line ( $L_2$  Norm of Error = 31.66)



### 3.7. Time Step Size

One of the main problems of the classical approaches, such as the sequential function specification method, is their instability when small time steps are used. Unlike direct problems where the stability requirement gives the upper limit of the time step size, in inverse problems the time step is bounded from below. Figure 3.10(a) [112] shows the oscillation in the results obtained by the function specification method and a time step size of 0.01 (s), which corresponds to the onset of instability. For time steps smaller than this, the whole process diverges. Luckily, PSO, GA, and NNs can successfully produce the results for the same time step size, as shown in Figure 3.10(b) for PSO. Note that the oscillations here are not due to the instability caused by the time step size, and can be improved by performing more iterations, as is shown in Figure 3.10(c). Figure 3.10(d) shows the final results of the GA method. It is, however, important to mention that the time requirements for these techniques are much higher than those of the classical function specification approaches.



**Figure 3.10. Heat Flux vs. Time: (a) Classical Approach, (b) PSO with the same computational expense, (c) Final results of PSO [112], (d) Final results of GA.**

### 3.8. Efficiency

In this section, we first compare the solution time required for the gradient-based function specification method with GA, the three variations of PSO, and the feedforward and radial basis function neural networks. We assume that there is no noise in the solution, and we compare the time that is required to get a certain accuracy in the heat flux predictions. Table 3.1 compares the solution time for different inverse analysis algorithms for solving the whole thermal history problem of cooling steel on a run-out table. The fastest solution technique is the gradient-based function specification method. The stochastic methods such as GA and PSO

variants suffer from high computational cost. RBF neural networks perform much faster than GA and PSO, but they are still slower than the gradient-based methods, such as function specification.

**Table 3.1. Comparison of the Typical Solution Time for Different Inverse Analysis Algorithms for the Whole Cooling Process on a Run-Out Table**

<b>Function Specification</b>		<b>GA</b>	<b>PSO</b>	<b>RPSO</b>	<b>CRPSO</b>	<b>FMLP</b>	<b>RBFN</b>
<b>Method</b>							
<b>Solution Time</b>							
<b>(s)</b>	1406	8430	6189	5907	6136	7321	2316

However, due to the limitations mentioned in the previous chapter for the gradient-based methods, and in the previous sections of this chapter for the neural networks, in many cases one has to choose between GA and PSO variations. Then, among the slower algorithms, the efficiency of the algorithm becomes even more important. Thus, in the remainder of this section, we will compare the performance of GA and PSO variations in more details.

The six different comparison tests performed are presented in Table 3.2. Because of the stochastic nature of these methods, they may perform differently for various runs. Therefore, a statistical approach is needed to compare their performances.

**Table 3.2. Performed Comparison Tests**

<b>Test #</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>
<b>Method1</b>	GA	GA	GA	PSO	PSO	RPSO
<b>Method2</b>	PSO	RPSO	CRPSO	RPSO	CRPSO	CRPSO

The statistical  $t$ -test [118] is used to compare the number of function evaluation calls (number of times that the direct solver is executed) of 10 runs of each algorithm, which means a sample size of 10 is used. The purpose of performing this test is to determine whether or not the difference between our two averages is large in comparison with the standard deviation. All of the runs used in the performance evaluation test are tested for the accuracy of the results, and have proven to produce acceptable results.

If we represent the number of function evaluations for methods 1 and 2 with  $N_{E1}$  and  $N_{E2}$ , respectively, then the average number of function evaluations for method 1 is

$$\overline{N_{E1}} = \frac{\sum_{i=1}^{n_1} N_{E1_i}}{n_1} \quad (3.13)$$

where  $n_1$  is the number of executions of method 1 (in these examples,  $n_1 = n_2 = 10$ ). The same equation holds for method 2. The standard deviation of each sample can be calculated as

$$s(N_{E1}) = \sqrt{\frac{\sum_{i=1}^{n_1} (N_{E1_i} - \overline{N_{E1}})^2}{n_1 - 1}} \quad (3.14)$$

The  $t$ -value can then be calculated as

$$t = \frac{\overline{N_{E1}} - \overline{N_{E2}}}{\bar{s}(N_E) \sqrt{1/n_1 + 1/n_2}} \quad (3.15)$$

where  $\bar{s}$  is the standard deviation of both samples and is calculated as

$$\bar{s}(N_E) = \sqrt{\frac{(n_1 - 1)s^2(N_{E1}) + (n_2 - 1)s^2(N_{E2})}{n_1 + n_2 - 2}} \quad (3.16)$$

The objective of the test is to check whether generally  $N_{E2} < N_{E1}$ . The null hypothesis in this test will be:  $N_{E2} \geq N_{E1}$ . The test is a one-sided test of significance of comparison of two means [118], with a significance level of 5%, and a  $10+10-2=18$  degrees of freedom, which gives a critical  $t$ -value of 1.73.

Table 3.3 shows the results of the comparison tests. The shaded values show the cases where the null hypothesis is accepted, i.e., the second method does not perform better than the first one. In the other cases, which are the majority of the tests, using the second method improves the performance, i.e., reduces the number of function evaluations, at least in 95% of the occurrences.

**Table 3.3. Calculated  $t$ -Values for the Combination of 2 Solution Methods;  $t_{\text{critical}} = 1.73$** 

	<b>Test Case I</b>	<b>Test Case II</b>	<b>Test Case III</b>	<b>Test Case IV</b>	<b>Test Case V</b>
<b>Test 1</b>	4.88	10.08	8.26	10.20	13.43
<b>Test 2</b>	6.60	12.52	9.41	15.05	15.46
<b>Test 3</b>	10.78	17.76	13.41	19.55	12.44
<b>Test 4</b>	1.49	3.32	1.27	4.39	8.21
<b>Test 5</b>	4.63	7.59	7.92	9.71	10.90
<b>Test 6</b>	3.04	3.45	7.75	10.01	9.22

Since an important requirement for the validity of the  $t$ -test results is the normality of the distributions for both categories that are compared, and it normally requires a larger number of simulations, the  $t$ -test statistics may not be a perfectly sound comparison tool. Thus, we have also used the Mann-Whitney non-parametric statistical test [118] to investigate our results. Table 3.4 shows the  $U$ -values for the Mann-Whitney  $U$ -test. Again the lowest  $U$  values occur at the same cases as the  $t$ -test, and only these two aren't acceptable with a significance level of 5%, for a one-sided Mann-Whitney  $U$ -test.

**Table 3.4. Calculated  $U$ -Values for the Combination of 2 Solution Methods**

	<b>Test Case I</b>	<b>Test Case II</b>	<b>Test Case III</b>	<b>Test Case IV</b>	<b>Test Case V</b>
<b>Test 1</b>	97	100	100	100	100
<b>Test 2</b>	100	100	100	100	100
<b>Test 3</b>	100	100	100	100	100
<b>Test 4</b>	72	87.5	62	88	94
<b>Test 5</b>	94	100	100	94	100
<b>Test 6</b>	84	88	99	94	94

The following observations can be made regarding the results:

1. PSO variations are generally less computationally expensive than GA in solving inverse heat conduction problems.
2. The advantage of PSO over GA is more pronounced in more complex cases (more unknowns and fewer data points).
3. Between the three variations of PSO, the difference is less significant, but generally, CRPSO performs better than the other two variations.

Table 3.5 shows the required number of computational units (i.e., solutions of the direct problem, and obtaining the norm of difference between that solution and the expected one), the relative speedup of each variation of PSO with respect to GA, and the relative computational cost of each method. From the table, speedups of up to 36% are possible when using some PSO variations instead of GA in an inverse heat conduction problem.

**Table 3.5. Number of Computational Units of Different Gradient-Free Solution Methods, Their Speedup with Respect to GA, and Their Relative Cost with Respect to the Most Efficient Method**

		<b>GA</b>	<b>PSO</b>	<b>RPSO</b>	<b>CRPSO</b>
<b>Test Case I</b>	<b>No. of Direct Calculations</b>	5075	4582	4427	4155
	<b>Speedup</b>	1.00	1.11	1.15	1.22
	<b>Relative Cost</b>	1.22	1.10	1.06	1.00
<b>Test Case II</b>	<b>No. of Direct Calculations</b>	13499	11382	10659	9939
	<b>Speedup</b>	1.00	1.19	1.27	1.36
	<b>Relative Cost</b>	1.36	1.14	1.07	1.00
<b>Test Case III</b>	<b>No. of Direct Calculations</b>	23903	21119	20872	19611
	<b>Speedup</b>	1.00	1.13	1.15	1.22
	<b>Relative Cost</b>	1.22	1.08	1.06	1.00
<b>Test Case IV</b>	<b>No. of Direct Calculations</b>	31050	24187	23451	23218
	<b>Speedup</b>	1.00	1.28	1.32	1.34
	<b>Relative Cost</b>	1.34	1.04	1.01	1.00
<b>Test Case V</b>	<b>No. of Direct Calculations</b>	57438	47497	44321	42475
	<b>Speedup</b>	1.00	1.21	1.30	1.35
	<b>Relative Cost</b>	1.35	1.12	1.04	1.00

Table 3.6 shows the required time to perform one computational unit for each test case. Note that these values have no impact on the performance comparison of the tested optimization algorithms, and are similar for all the optimization methods.

**Table 3.6. Time Required to Perform One Computational Unit for each Test Case**

	Test Case I	Test Case II	Test Case III	Test Case IV	Test Case V
<b>Time for one computational unit (s)</b>	1.65	10.82	5.41	5.06	8.11

The difference between the behaviors of PSO and GA can be attributed to the different underlying mechanisms in these two evolutionary algorithms. For example, mutation in GA occurs in all directions, while the mutation-like behavior in PSO is directional.

### 3.9. Sequential vs. Whole Domain

Due to the parabolic nature of the heat equation, there are basically two categories of approaches in solving the inverse transient heat conduction problems. One method is to try to find components of heat flux in each step of the solution. This is possible since the temperature distribution at the end of each time step is only a function of the temperature distribution at the previous time step and the boundary conditions during the present time step. These methods are normally called “sequential”. In these methods, in each step of the inverse analysis, the direct problem is not solved for the whole history, and is solved for only one (or a few) time steps. The optimization problem is then solved for the temperatures at that step, and then the problem is marched in time to the next steps. The other solution method is to have the whole history of heat fluxes as the unknown in the optimization problem. In this formulation, which is usually called the “whole domain” approach, the direct problem will be solved over the whole time history.

Due to the diffusive nature of transient heat conduction, from the computational expense perspective, sequential estimation techniques are highly preferred over the whole-domain inverse analysis techniques [35]. Unfortunately, there is an important drawback in using sequential methods, and that is the high sensitivity of the solution to experimental errors. That is why we will be using the concept of future time steps.

Table 3.7 shows the average solution time for ten simulations of the test cases using different solution methods and whole domain and sequential formulations. Test case III is a steady-state problem, and thus is not included here. As seen in the table, sequential formulation always accelerates the solution process. Here, the speedup is defined as the whole domain



solution time over the sequential solution time, and is usually somewhere between 30 and 60 percent.

**Table 3.7. Comparison of Efficiency for Whole Domain and Sequential Implementations**

		<b>Whole Domain</b>	<b>Sequential Solution</b>	
		<b>Solution Time (s)</b>	<b>Time (s)</b>	<b>Speedup</b>
<b>Test</b>	<b>PSO</b>	8316	5904	1.41
<b>Case</b>	<b>RPSO</b>	8107	5107	1.59
<b>I</b>	<b>CRPSO</b>	8020	5293	1.52
<b>Test</b>	<b>PSO</b>	15052	11759	1.28
<b>Case</b>	<b>RPSO</b>	14501	11069	1.31
<b>II</b>	<b>CRPSO</b>	14115	10942	1.29
<b>Test</b>	<b>PSO</b>	79978	58384	1.37
<b>Case</b>	<b>RPSO</b>	76784	56820	1.35
<b>IV</b>	<b>CRPSO</b>	74266	55208	1.34
<b>Test</b>	<b>PSO</b>	121870	81348	1.50
<b>Case</b>	<b>RPSO</b>	109539	82910	1.32
<b>V</b>	<b>CRPSO</b>	116449	81824	1.42

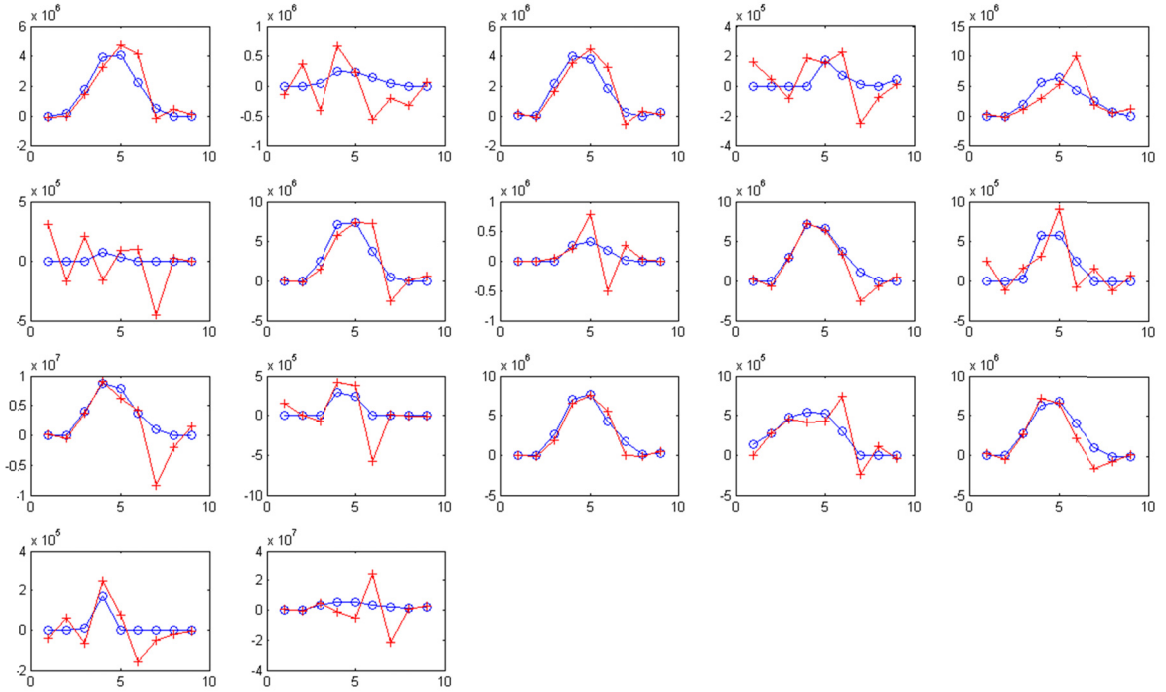
### 3.10. Noisy Domain Solution

To investigate the behavior of different inverse algorithm variations in dealing with noise in the data, a known boundary condition is first applied to the direct problem. The temperature at some internal point(s) will be calculated and stored. Then random errors are imposed onto the calculated exact internal temperatures with the following equation:

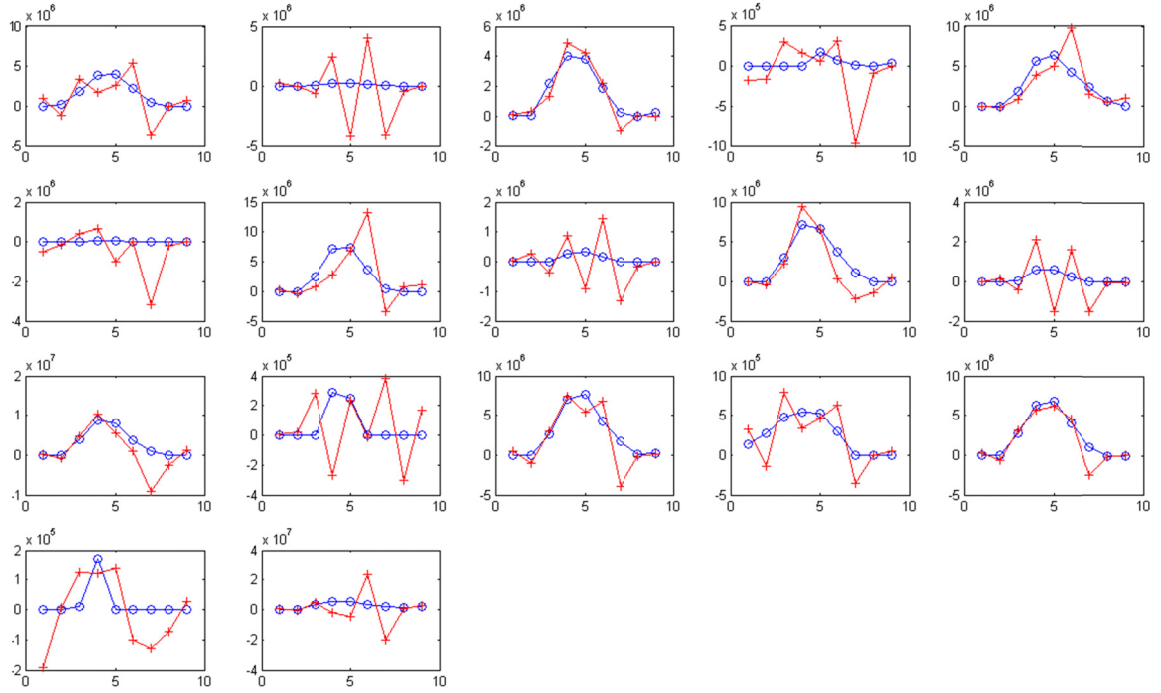
$$T_m = T_{exact} + \sigma \cdot r \quad (3.17)$$

where  $T_m$  is the virtual internal temperature that is used in the inverse calculations instead of the exact temperature,  $T_{exact}$ ;  $r$  is a normally distributed random variable with zero mean and unit standard deviation; and  $\sigma$  is the standard deviation. Virtual errors of 0.1%, 0.3%, 0.5%, and 1% of the temperature magnitude are investigated in this research. Since the temperature is changing from 700 to 176 °C, the magnitude of the noise can be as high as 7 °C for the initial peaks.

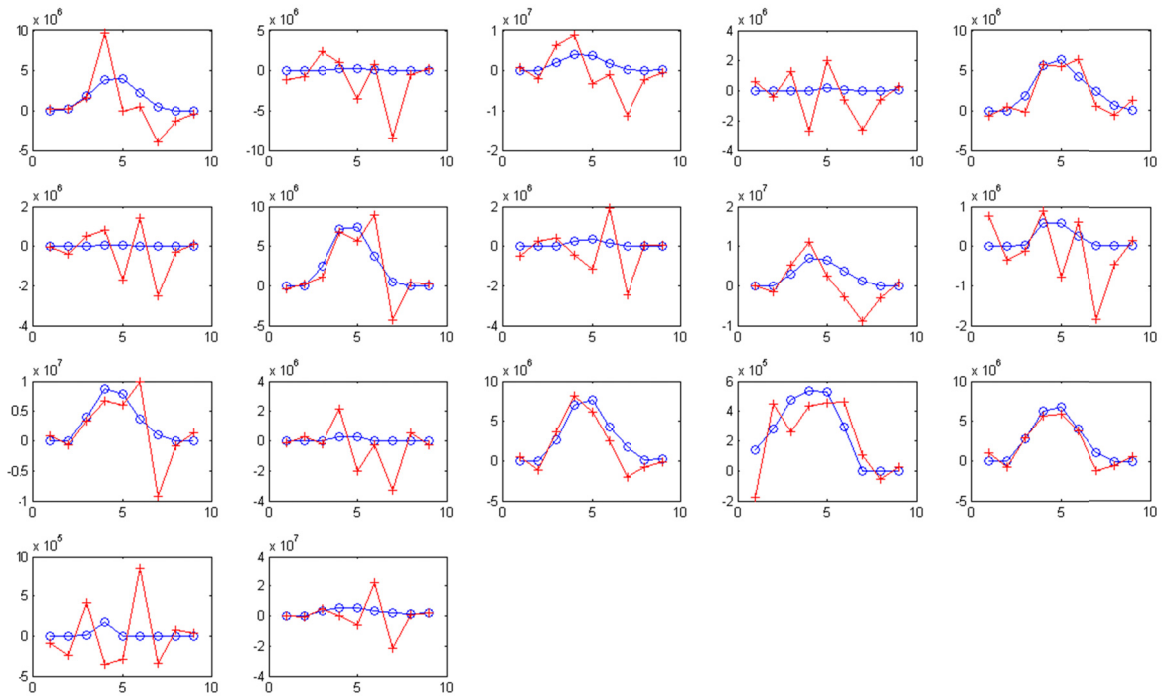
We start by studying the effectiveness of the neural networks in handling noisy domains. Generally, the stability of the neural networks is on the same order as other inverse methods. It may be possible to tune the parameters to make them a little bit more stable, but generally it does not look promising in terms of noise resistance, since such modifications exist for almost all other methods, such as PSO. Figure 3.11 - Figure 3.14 show the results of the RBF network (pluses) versus the expected results (circles) for individual heat flux peaks during the cooling history of the plate. The amount of added noise in these figures is  $\pm 0.1\%$ ,  $\pm 0.3\%$ ,  $\pm 0.5\%$ , and  $\pm 1\%$ , respectively.



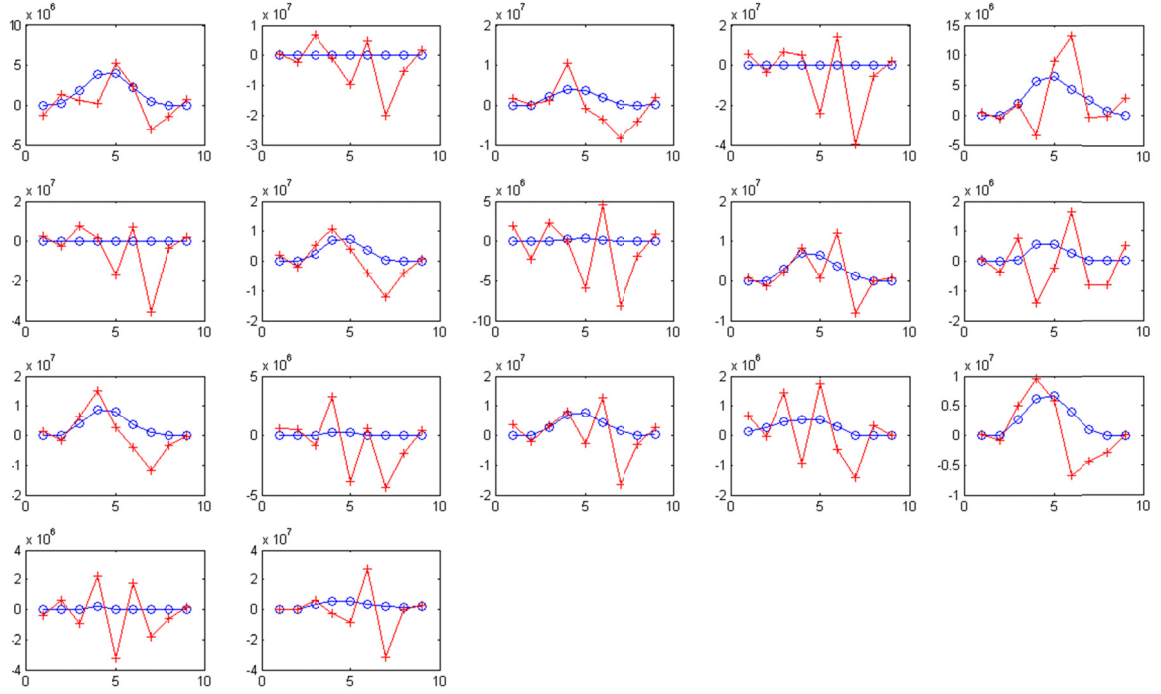
**Figure 3.11. Individual Heat Flux Peaks vs. Time from a Typical Run-Out Table Application; Expected Results (Circles) vs. the RBF Network Results (Pluses); Artificial Noise Added:  $c = \pm 0.1\%$**



**Figure 3.12. Individual Heat Flux Peaks vs. Time from a Typical Run-Out Table Application; Expected Results (Circles) vs. the RBF Network Results (Pluses); Artificial Noise Added:  $c = \pm 0.3\%$**



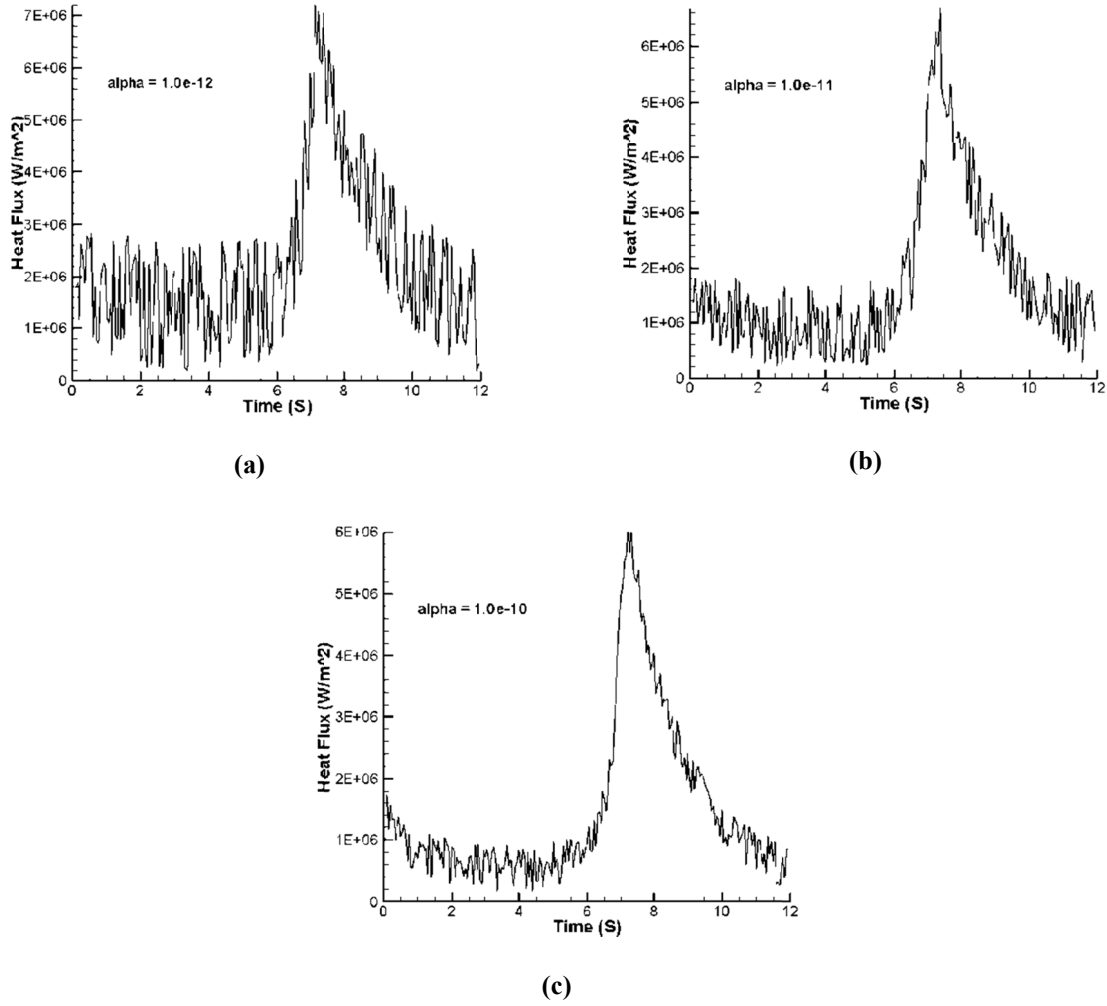
**Figure 3.13. Individual Heat Flux Peaks vs. Time from a Typical Run-Out Table Application; Expected Results (Circles) vs. the RBF Network Results (Pluses); Artificial Noise Added:  $c = \pm 0.5\%$**



**Figure 3.14. Individual Heat Flux Peaks vs. Time from a Typical Run-Out Table Application; Expected Results (Circles) vs. the RBF Network Results (Pluses); Artificial Noise Added:  $c = \pm 1\%$**

There are several ways to make an inverse algorithm more stable when dealing with noisy data. For example, Gadala and Xu [57] have shown that increasing the number of “*future time steps*” in their sequential function specification algorithm resulted in greater stability. They have also demonstrated that increasing the regularization parameter,  $\alpha$ , improves the ability of the algorithm to handle noisy data. However, the latter approach was shown to greatly increase the required number of iterations, and in many cases the solution may diverge. In this work, we first examine the effect of the regularization parameter, and then investigate an approach unique to the PSO method, to improve the effectiveness of the inverse algorithm in dealing with noise.

Figure 3.15 shows the effect of varying the regularization parameter value on the reconstructed heat flux, using the basic particle swarm optimization technique. Stable and accurate results are obtained for a range of values of  $\alpha = 10^{-12}$  to  $10^{-10}$ . These results are very close to those reported in [57], i.e., the proper values of  $\alpha$  are very similar for the sequential specification approach and PSO.



**Figure 3.15. Test Case II; Effect of Regularization Parameter (1% added noise); a:  $\alpha = 10^{-12}$ ; b:  $\alpha = 10^{-11}$ ; c:  $\alpha = 10^{-10}$**

Another factor that can affect the performance of a PSO inverse approach in dealing with noisy data is the value of the self-confidence parameter,  $c_0$ , or the ratio between this parameter and the acceleration coefficients. To show the effect of this parameter, test case II is studied again, with the same level of noise as in Figure 3.15, and a regularization parameter equal to  $10^{-10}$  (Figure 3.15(c)). The acceleration coefficients are set to the default value of 1.42. The initial value of the self-confidence parameter,  $c_0$ , is changed from the default value of 0.7. The results are shown below.

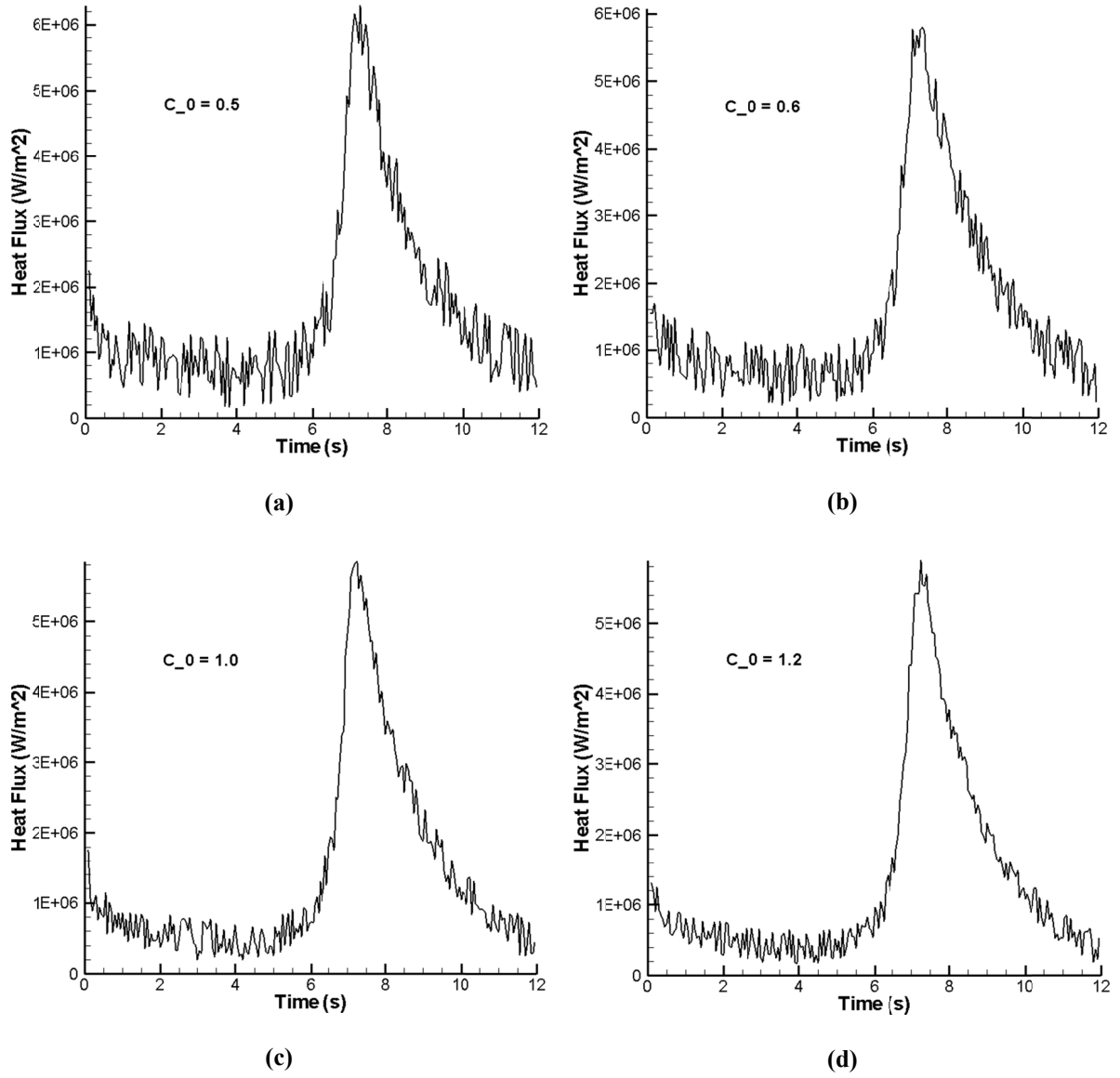


Figure 3.16. Test Case II; Effect of Self-Confidence Parameter (1% added noise); (a)  $c_\theta=0.5$ ; (b)  $c_\theta=0.6$ ; (c)  $c_\theta=1.0$ ; (d)  $c_\theta=1.2$

Table 3.8. Effect of Self-Confidence Parameter:  $L_2$  Norm of the Error ( $W/m^2$ )

$c_\theta$	0.5	0.6	0.7	1.0	1.2
$\alpha = 10^{-12}$	5.88e+5	5.24e+5	4.47e+5	3.81e+5	2.98e+5
$\alpha = 10^{-10}$	4.85e+5	4.30e+5	3.19e+5	2.36e+5	2.10e+5

As can be seen in Figure 3.16 (for  $\alpha = 10^{-10}$ ), and more quantitatively in Table 3.8 (for  $\alpha = 10^{-10}$  and  $10^{-12}$ ), increasing the value of the self-confidence parameter results in better handling of the noisy data. This trend was observed for values up to approximately 1.3, after which the results become worse, and diverge. The same trend was also observed in the other test cases. One possible explanation is that increasing the ratio of the self-confidence parameter with respect to the acceleration coefficients results in a more global search in the domain, and therefore increases the capability of the method to escape from the local minima caused by the noise, and find values closer to the global minimum. This effect was observed to be weaker in highly noisy domains. However, in the presence of a moderate amount of noise (which is the case in most experimental setups), increasing the self-confidence ratio results in more effectiveness in dealing with noisy data. Table 3.9 shows the same trend when using different variations of the PSO algorithm on test cases I, IV, and V. As can be seen in Table 3.9, the best effectiveness is normally obtained by RPSO, closely followed by CRPSO. Considering the higher efficiency of CRPSO, it is still recommended for the inverse heat conduction analysis.

**Table 3.9. Effect of the Self-Confidence Parameter on the  $L_2$  Norm of Error in the Solution**

<b><math>C_0</math></b>		<b>0.7</b>	<b>0.8</b>	<b>0.95</b>	<b>1.1</b>	<b>1.2</b>
<b>Test Case I</b>	<b>PSO</b>	0.07466	0.06722	0.06113	0.05651	0.05397
	<b>RPSO</b>	0.05546	0.05010	0.04362	0.04096	0.03728
	<b>CRPSO</b>	0.05798	0.05163	0.04415	0.04262	0.04043
<b>Test Case II</b>	<b>PSO</b>	8.105e+4	7.532e+4	7.079e+4	6.823e+4	6.257e+4
	<b>RPSO</b>	7.577e+4	7.064e+4	6.685e+4	6.346e+4	5.816e+4
	<b>CRPSO</b>	7.611e+4	6.739e+4	6.117e+4	5.999e+4	5.822e+4
<b>Test Case III</b>	<b>PSO</b>	1.641e+5	1.477e+5	1.322e+5	1.282e+5	1.222e+5
	<b>RPSO</b>	1.328e+5	1.272e+5	1.203e+5	1.184e+5	1.158e+5
	<b>CRPSO</b>	1.361e+5	1.288e+5	1.221e+5	1.184e+5	1.143e+5

Table 3.10 shows the value of  $L_2$  norm of error in the solution, for  $\pm 1\%$  added noise, and for different algorithms. It can be seen that the RBF neural networks perform better than the

function specification method, and somewhere between the genetic algorithm and PSO variants. The most noise resistant algorithms are PSO variants, and the least stable algorithm is the gradient-based function specification method.

**Table 3.10. The  $L_2$  Norm of Error in the Solution in a Noisy Domain for Different Algorithms**

<b>Function Specification Method</b>		<b>GA</b>	<b>PSO</b>	<b>RPSO</b>	<b>CRPSO</b>	<b>FMLP</b>	<b>RBFN</b>
<b><math>L_2</math> Norm of Error</b>	9.14e4	6.61e4	5.24e4	4.82e4	5.02e4	8.91e4	5.91e4

### 3.11. Future Time Step Regularization

There are also other ways to make an inverse algorithm more stable when dealing with noisy data. One of them is using the concept of future time steps that was also discussed in the function specification method in the previous chapter. Due to the fact that inverse problems are generally ill-posed, the solution may not be unique and would, in general, be sensitive to measurement errors. To decrease such sensitivity and to improve the simulation in function specification methods, a number of future time steps ( $n_{FTS}$ ) are often utilized in the analysis of each time step [35]. This means that in addition to the measured temperature at the present time step  $T^i$ , the measured temperatures at future time steps,  $T^{i+1}, T^{i+2}, \dots, T^{i+n_{FTS}}$ , are also used to approximate the heat flux  $q^i$ . In this process, a temporary assumption would usually be considered for the values of  $q^{i+1}, q^{i+2}, \dots, q^{i+n_{FTS}}$ . The simplest and the most widely used one is to assume  $q^{i+k} = q^i$  for  $1 \leq k \leq n_{FTS}$ , which is also used in our work. Also in this work, we combine the regularization concept with the future time step data. This is analogous to the combined function specification-regularization method, initially introduced by Beck and Murio [96].

The sequential algorithm for time step  $M$  when using  $r$  future time steps (the heat flux is known for  $t < t_M$ ) is presented below:

1. An assumption is made for the heat flux function at times  $t_M, t_{M+1}, \dots, t_{M+r}$  (usually  $q_M = q_{M+1} = \dots = q_{M+r}$ ).



2. Direct problem is solved with  $T_{M-1}^{meas}$  as the initial condition and the heat flux values from step 1 as its transient boundary conditions. The temperatures at these  $r$  time steps are calculated  $(T_M^{calc}, T_{M+1}^{calc}, \dots, T_{M+r}^{calc})$ .
3. The objective function is then evaluated using the heat fluxes from step 1, the calculated temperatures from step 2, and the known temperatures from the measurement  $(T_M^{meas}, T_{M+1}^{meas}, \dots, T_{M+r}^{meas})$ . The new form of the objective function will be:

$$f(\mathbf{q}) = \sum_{j=1}^J \left( \sum_{i=M}^{M+r} (T_j^{i,meas} - T_j^{i,calc})^2 + \alpha \sum_{i=M}^{M+r} (q_j^i)^2 \right) \quad (3.18)$$

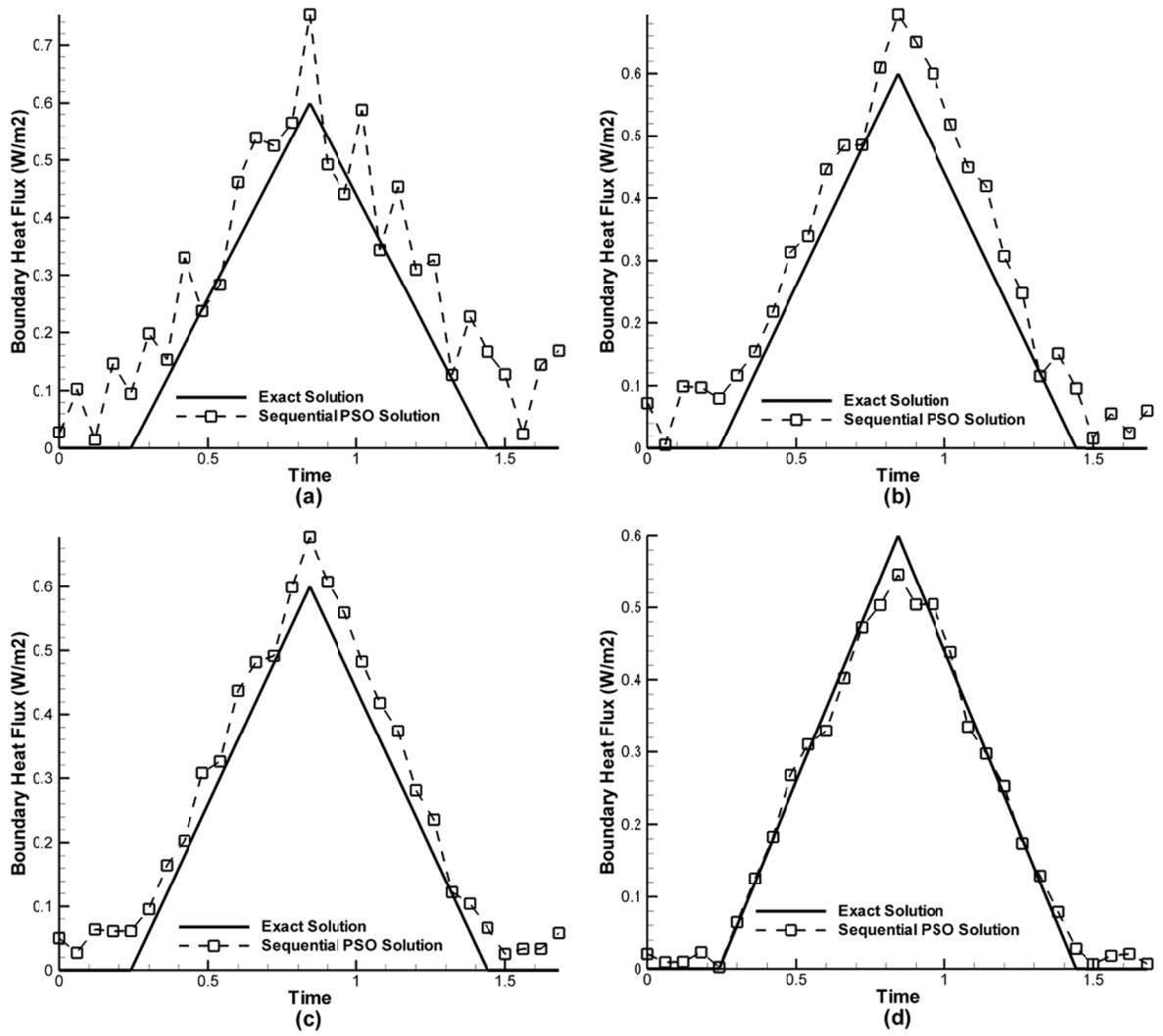
4. Steps 1 to 3 are repeated for each swarm member, and the GA or PSO algorithm is repeated until we reach a desired low value for the objective function, or after a certain number of iterations. The solution vector will be the spatial components of heat fluxes for the  $r$  time steps.
5. Values of  $q_M$  are used as obtained from step 4 and the values for the rest ( $q_{M+1}, \dots, q_{M+r}$ ) are discarded.
6. The above procedure ( $M = M+1$ ) is repeated as you move to the next time step.

In theory, it is possible to use multiple sensors instead of multiple future time steps to improve the stability of the method. However, this idea is not practical since adding thermocouples to the sample is not only more expensive, but also distorts the temperature field.

Using future time steps introduces a bias in the solution. This bias is normally observed in the form of rounding the spikes in oscillatory boundary condition histories, and an early prediction of these oscillations [35]. To obtain an appropriate level of regularization, the number of future time steps (or more accurately, the size of the look-ahead time window, i.e. the product of the number of future time steps and time step size) and the value of the regularization parameter must be chosen with respect to the errors involved in the temperature readings. The residual principle [36,102] can be used to determine these parameters based on the accuracy of thermocouples in the relative temperature range. In this method, the number of future time steps

is increased until the RMS error of the estimation falls below the standard deviation of noise in the measured temperatures.

Figure 3.17 shows the results of applying the basic PSO-based inverse analyzer to the first test case, when a moderate amount of noise is added to the temperature data. As can be seen in part (a) of the figure, the method with default settings and no regularization term shows a very unruly behavior. While addition of the zeroth-order Tikhonov regularization term and the tuning of the self-confidence parameter can enhance the behavior, as seen in parts (b) and (c) of Figure 3.17, there is a limit to these modifications, i.e. after a certain limit the method will become very slow or will diverge. However, using the future time step concept, can increase the stability of the algorithm, as seen in Figure 3.17(d). Table 3.11 also shows the  $L_2$  norm of error in the solution after each stage of algorithm modifications. Figure 3.18, Figure 3.19 and Table 3.12 show the same trend for the first and second set of heat fluxes applied to the three-dimensional problem. As expected, the norm of error in the solution in each case decreases after each set of modifications is applied. Also, investigating the numbers in Table 3.11 and Table 3.12 shows that among the three variations of PSO, RPSO is the most effective one in handling noisy data, closely followed by CRPSO. This is in agreement with the findings in Reference [112], shown earlier in this chapter. As can be seen from the figures, in the cases without the future time step method, the predicted heat fluxes overestimate the peak heat flux values, but when future time steps are used, due to the anticipation of the temperature in the following time steps, the predicted peak better matches the expected results.



**Figure 3.17. PSO Behavior in Handling Noisy Data in Test Case I; (a) No Regularization; (b) Addition of Regularization Term ( $\alpha = 10^{-10}$ ); (c) Tuning of Self-Confidence Parameter ( $c_0 = 1.2$ ); (d) Introduction of Future Time Step Data (10 time steps)**

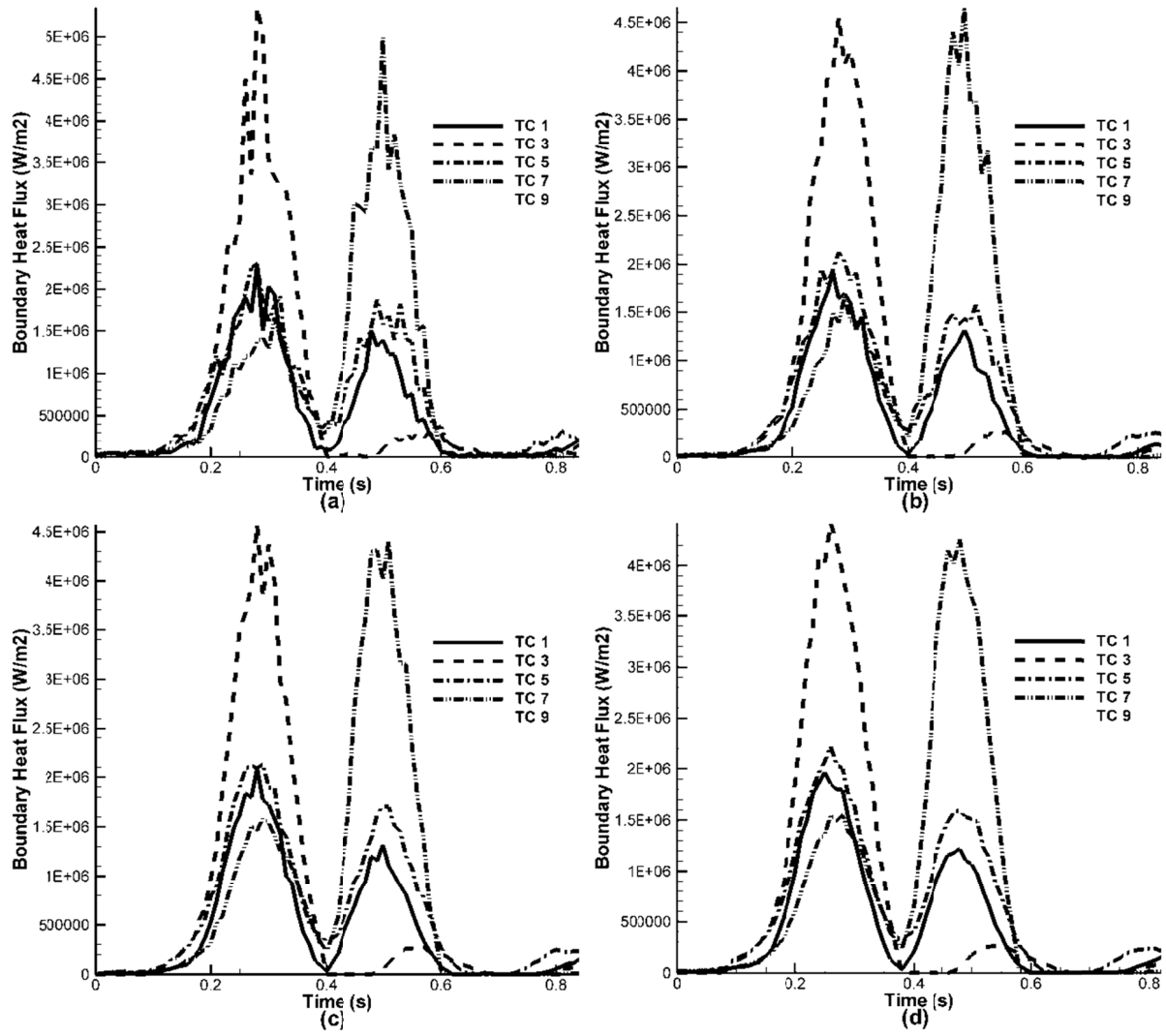
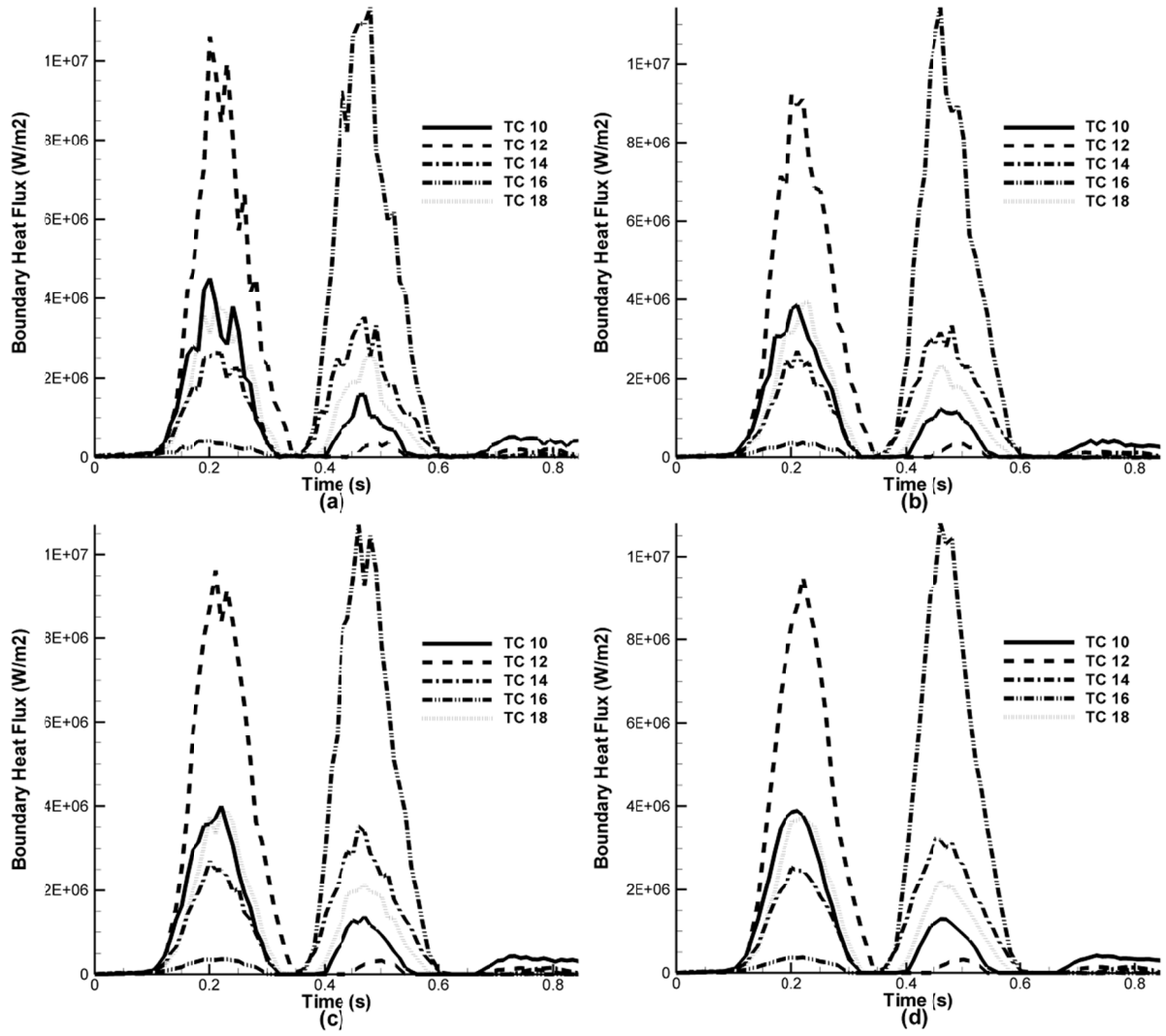


Figure 3.18. PSO Behavior in Handling Noisy Data for Test Case IV; (a) No Regularization; (b) Addition of Regularization Term ( $\alpha = 10^{-10}$ ); (c) Tuning of Self-Confidence Parameter ( $c_0 = 1.2$ ); (d) Introduction of Future Time Step Data (10 time steps)



**Figure 3.19. PSO Behavior in Handling Noisy Data for the 2nd Set of Heat Fluxes in Test Case V; (a) No Regularization; (b) Addition of Regularization Term ( $\alpha = 10^{-10}$ ); (c) Tuning of Self-Confidence Parameter ( $c_0 = 1.2$ ); (d) Introduction of Future Time Step Data (10 time steps)**

**Table 3.11. The  $L_2$  Norm of Error in the Solution of Test Case I After Modifications in a Noisy Domain**

<b>Method</b>	<b>No Regularization</b>	<b>Tikhonov Regularization</b>	<b>Tikhonov Regularization + Tuning of the <math>c_0</math> parameter</b>	<b>Tikhonov Regularization + Tuning of the <math>c_0</math> parameter + Future time steps</b>
<b>PSO</b>	0.10845	0.07466	0.05397	0.02108
<b>RPSO</b>	0.10070	0.05546	0.03728	0.01855
<b>CRPSO</b>	0.10148	0.05798	0.04043	0.01876

**Table 3.12. The  $L_2$  Norm of Error in the Solution of Test Cases IV and V After Modifications in a Noisy Domain**

		<b>No Regularization</b>	<b>Tikhonov Regularization</b>	<b>Tikhonov Regularization + Tuning of the <math>c_0</math> parameter</b>	<b>Tikhonov Regularization + Tuning of the <math>c_0</math> parameter + Future time steps</b>
<b>1<sup>st</sup> Set of Heat Fluxes</b>	<b>PSO</b>	1.504E+05	8.105E+04	6.257E+04	3.179E+04
	<b>RPSO</b>	1.393E+05	7.577E+04	5.816E+04	2.813E+04
	<b>CRPSO</b>	1.418E+05	7.611E+04	5.822E+04	2.837E+04
<b>2<sup>nd</sup> Set of Heat Fluxes</b>	<b>PSO</b>	2.712E+05	1.641E+05	1.222E+05	6.305E+04
	<b>RPSO</b>	2.448E+05	1.328E+05	1.158E+05	5.400E+04
	<b>CRPSO</b>	2.495E+05	1.361E+05	1.143E+05	5.440E+04

While using Tikhonov regularization, tuning the value of the self-confidence parameter and using future time steps can improve the quality of the solution when dealing with noisy data, but it also increases the number of parameters that should be controlled. Luckily, once these parameters are set for one simulation of a specific test case, they can be used again, and there is no need to determine the most effective values in each run. Also, these modifications can increase the required solution time, so there will always be a compromise between the level of accuracy and the computational expense, which should be considered according to the requirements of each application.

### 3.12. Effect of Non-linearity

While methods such as GA and PSO are based on a stochastic search of the whole domain, and therefore are not significantly affected by nonlinearity, the neural networks are more sensitive to nonlinearity. In other words, since the same drop in temperature values can be caused by different values of heat flux, a neural network that is trained with the relationship between the temperature change values and heat flux magnitudes may not be correctly capable of recognizing this nonlinear pattern, and as a result the performance will suffer. To investigate this effect, two kinds of expressions are used for thermal conductivity in this study. In one, we assume a constant thermal conductivity of  $40 \text{ W/m.}^{\circ}\text{C}$ , while in the other a temperature-dependent expression is used:

$$k = 60.571 - 0.03849 \times T \text{ W/m.}^{\circ}\text{C} \quad (3.19)$$

As expected, the nonlinearity will weaken the performance of both feedforward and radial basis function neural networks. The effect is seen as the training of the network stalls after a number of epochs. In order to deal with this, increasing the number of hidden layers, increasing the number of neurons in each layer, and choosing different types of transfer functions were investigated. However, none of these methods showed a significant improvement in the behavior of the network. The other methods of solving the inverse problem that are introduced in this research, are much less sensitive to the effect of nonlinearity. Table 3.13 compares the error in the solution for both the linear and nonlinear cases, if the same numbers of iterations, generations, and epochs are used for different methods of solving the inverse heat conduction. As can be seen, the neural networks perform very poorly in the nonlinear cases, while the other methods, either gradient based or stochastic, are immune to the problems caused by

nonlinearity. Basically, neural networks, at least in the form that is used in this research, see nonlinearity as a kind of noise. It should be noted that neural networks can be useful in making rough estimates of the answer, or combined with some other techniques employed as an inverse solver for nonlinear cases [64]. However, on their own, they are not a suitable choice for an accurate prediction of the boundary conditions in a nonlinear inverse heat conduction problem.

**Table 3.13. The  $L_2$  Norm of Error in the Solution in an Exact Domain for Different Algorithms**

Function Specification Method		GA	PSO	RPSO	CRPSO	FMLP	RBFN
<b>Linear</b>	1.81e2	7.62e2	3.85e2	3.42e2	3.17e2	9.90e2	5.35e2
<b>Nonlinear</b>	2.14e2	7.71e2	4.46e2	5.12e2	4.26e2	3.57e4	2.76e4

### 3.13. Effect of Elite Members

In this section, the effect of including the elite particle's position and velocity in the performance of the sequential PSO-based inverse solver is studied. We first take a look at the average speedup as the result of introducing the elite members into the algorithm. Here, the speedup is defined as the ratio of the required solution time in the case without the elite members to the solution time after the elite members are utilized. Obviously, a higher speedup means that the concept of using the elite particles was more effective. Table 3.14 shows the average speedup of 10 runs for each algorithm. As can be seen from the table, in some cases, an improvement of 14% can be achieved as a result of using elite particles and velocities.

While the introduction of this concept usually improved the performance, this improvement did not take place in all the simulations. One must especially remember that the performance of PSO, like any other random search algorithm, can vary from one simulation to another. In order to take care of this randomness in the simulations, a Mann-Whitney non-parametric statistical test [118] is used to compare the solution time for the 10 runs of each algorithm for each test case and assess the improvement caused by the inclusion of the elite particle concept. All of the runs used in the performance evaluation test are tested for result accuracy, and proved to produce acceptable solutions.



The objective of the test is to check whether, generally, the algorithm with the elite member is faster than the same variation without the elite member. The null hypothesis in this test is that the introduction of the elite particle concept does not improve the solution time, i.e.:  $SolutionTime_{without-elite-members} \leq SolutionTime_{with-elite-members}$ . The test is a one-sided test of significance of the comparison of two means [118]. If for each simulation with elite members, we count the number of simulations without elite members that are slower than it, and then sum these counts for all the simulation set, we obtain the  $U$ -value. A significance level of 5%, and an equal number of samples for both sides ( $n_1 = n_2 = 10$ ) gives a critical  $U$ -value of 73.

Table 3.15 shows the  $U$ -values for the Mann-Whitney  $U$ -test. The only case that does not pass the test is the basic PSO algorithm applied to the last test case. As seen in the table, RPSO always performs better when the elite particles are introduced. Based on the results of this table, including the elite particle in the PSO algorithm to solve the inverse heat conduction problem is justified.

**Table 3.14. The Average Speedup in Each Algorithm After Using Elite Particles and Velocities**

		<b>Whole Domain</b>	<b>Sequential</b>
<b>Test Case I</b>	<b>PSO</b>	1.126	1.129
	<b>RPSO</b>	1.173	1.133
	<b>CRPSO</b>	1.138	1.131
<b>Test Case II</b>	<b>PSO</b>	1.020	1.174
	<b>RPSO</b>	1.152	1.147
	<b>CRPSO</b>	1.011	1.279
<b>Test Case III</b>	<b>PSO</b>	1.132	1.237
	<b>RPSO</b>	1.205	1.065
	<b>CRPSO</b>	1.129	1.086
<b>Test Case IV</b>	<b>PSO</b>	1.052	1.105
	<b>RPSO</b>	1.112	1.116
	<b>CRPSO</b>	1.093	1.104
<b>Test Case V</b>	<b>PSO</b>	1.043	1.050
	<b>RPSO</b>	1.052	1.096
	<b>CRPSO</b>	1.082	1.101

**Table 3.15. Calculated U-Values to Study the Effect of Elite Particles on the Performance**

	<b>Test Case I</b>	<b>Test Case II</b>	<b>Test Case III</b>	<b>Test Case IV</b>	<b>Test Case V</b>
<b>PSO</b>	100	88	88	84	72
<b>RPSO</b>	100	100	100	100	100
<b>CRPSO</b>	100	88	84	97	88

### **3.14. Multi-Criteria Formulation Based on Spatial Location**

It is common in inverse heat conduction applications to have more than one sensor inside the sample. In the case of inverse boundary problems, the number of these sensors is often equal to the number of spatial components of the missing boundary heat flux. The formulation of Eq. (3.1) uses a norm of the difference between the calculated and measured temperatures that take

into account all of the thermocouples. In other words, the objective is to minimize some sort of overall difference between these two sets of temperatures. Thus, when searching for the best solution, the method does not distinguish between different spatial components. One way to devise a more intelligent search algorithm is to define a multi-criteria formulation, in which each of the spatial components of the missing boundary condition is treated as an individual search space. This way, if one of the spatial components of the heat flux vector gets close to the expected value, it will not be discarded due to the poor quality of the other components. In this formulation, the objective function will be a vector, and each of its components will represent one of the sensor locations. We can write this objective function as:

$$\bar{f}(q) = \begin{pmatrix} \sum_{i=M}^{M+r} (T_1^{i,meas} - T_1^{i,calc})^2 + \alpha \sum_{i=M}^{M+r} (q_1^i)^2 \\ \sum_{i=M}^{M+r} (T_2^{i,meas} - T_2^{i,calc})^2 + \alpha \sum_{i=M}^{M+r} (q_2^i)^2 \\ \vdots \\ \sum_{i=M}^{M+r} (T_J^{i,meas} - T_J^{i,calc})^2 + \alpha \sum_{i=M}^{M+r} (q_J^i)^2 \end{pmatrix} \quad (3.20)$$

It should be noted that this is not equal to decoupling of the thermocouples, because the direct problem would still be solved as a whole.

The performance of multi-criteria optimization (vector objective function) is compared with the case where the objective function is a single scalar quantity. Test cases IV and V are solved, and the history of the best value of the objective function in the swarm is normalized with respect to the best value of the objective function in the first iteration, and is plotted versus the iteration number. As can be seen in Figure 3.20, this modification greatly improves the convergence rate of the PSO-based inverse analyzer.

Comparing the two parts of Figure 3.20, we observe that while the increased number of thermocouples causes a lower rate of convergence, both when the vector or scalar formulation is used, the relative speedup of the vector formulation stays in the same range.

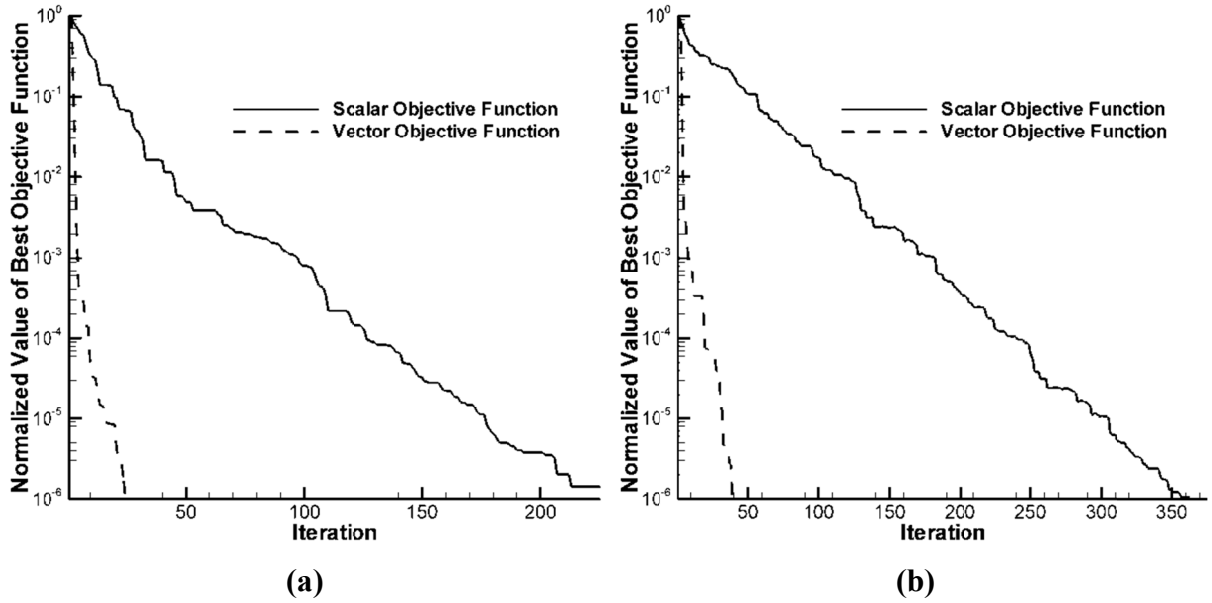


Figure 3.20. Convergence of Scalar vs Vector Objective Function Formulations; (a) Test Case IV; (b) Test Case V

### 3.15. Surrogate Modeling

#### 3.15.1. Motivation

In many engineering applications, when we are faced with an inverse problem, or we simply want to find the optimum design of a system, it is necessary to perform multiple direct simulations. While conventional methods of numerical simulation (finite difference, finite volume, finite element, etc.) are capable of producing very accurate results in most cases, the process is normally very computationally expensive. While it is very common to simplify the numerical simulations using simplified physical (inviscid flow instead of viscous), or geometrical (two-dimensional instead of 3D) models, they are still time-demanding. This is even more unreasonable in the case of rank-based evolutionary algorithms, such as genetic algorithm or particle swarm. In these techniques, the actual value of the objective function is not important, but only the sorting of the members matters. The exact magnitude of the objective function is normally discarded, especially in the initial steps of the algorithm. So, it seems logical not to spend a large amount of computational resources on exactly evaluating these cost function values. Instead an inexact pre-evaluation approach using surrogate models may be adopted to identify the promising members, and later perform the full evaluations only for these candidate solutions.

### 3.15.2. Procedure

The procedure for constructing the surrogate models has five major stages:

1. *Design of experiments.* In this stage, we choose the samples in the design space that are going to be used for constructing the surrogate model. Since the number of chosen samples is much smaller than all the possible cases, it should be a good representative of the domain. Some of the methods that can be used for the design of experiments are random, error-based, density-based, gradient-based, and some hybrids of these. A random sample selection scheme is used in this study.
2. *Full numerical simulation of the selected samples.* The high-fidelity and expensive numerical simulation technique is used to evaluate the samples that were chosen in the previous stage.
3. *Model selection.* In this step, we choose between several types of available surrogate models, such as polynomial, radial basis function, neural networks, etc. Choosing the most suitable surrogate model type is a problem-related decision in most cases. There are a number of criteria that can be used to rank different surrogate modeling techniques. Some of the most common measures are the accuracy, efficiency, and robustness of the models. Other criteria may include transparency, i.e. capability of explicitly showing the relationship between the input and output, and the ease of implementation [119]. In this research, we are going to compare the implemented methods in term of robustness and efficiency, which are more crucial in our inverse heat conduction applications.
4. *Model identification.* Here we tune the parameters of the selected model type to maximize the capability of the algorithm. Another evolutionary optimization algorithm may be applied to find the best set of parameters. The available options are different for each model type.
5. *Model validation.* In this stage, we make sure that the model is capable of producing acceptable results for some other samples that were not used in the previous steps. Some available algorithms are validation set, cross-validation, leave-one-out, and model difference. Cross-validation is used in this research.

### 3.15.3. Surrogate Model Types

Four types of surrogate models were tested in this research: Polynomial models, Radial Basis Function models, Kriging models, and Feedforward Neural Networks. Some researchers have previously compared the performance of several surrogate model types in solving some

benchmark optimization problems [119]. In this study, we directly focus on their performance in inverse heat conduction problems.

### Polynomial Models

A second-order polynomial model is the most popular polynomial model, and is also used in this research. It can be written in the following form:

$$\hat{y} = \beta_0 + \sum_{1 \leq i \leq n} \beta_i x_i + \sum_{1 \leq i \leq j \leq n} \beta_{n-1+i+j} x_i x_j \quad (3.21)$$

where  $\hat{y}$  is the estimated response,  $x$  is the vector of inputs,  $\beta$  values are the coefficients that need to be estimated, and  $i$  and  $j$  are the indices of the input parameters. This is normally done using the least square method [120].

### Radial Basis Function

Using the same definitions for  $x$  and  $y$  as above, the radial basis function model can be expressed as:

$$\hat{y} = \sum_{1 \leq i \leq n} \omega_i \psi(\|x_i - c_i\|) \quad (3.22)$$

where  $\omega_i$  is the selected weight for each parameter,  $c_i$  denotes the  $i$ th basis function center, and  $\psi$  is the basis function. The main strength of this model arises from the fact that it is linear in terms of the basis function weights; however it is capable of predicting highly nonlinear responses [120].

### Kriging Models

One of the most popular types of surrogate models is the Kriging method. In this variation, the basis function is of the form:

$$\psi^{(i)} = \exp \left( - \sum_{j=1}^k \theta_j |x_j^{(i)} - x_j|^{p_j} \right) \quad (3.23)$$

The vector  $\boldsymbol{\theta} = \{\theta_1, \theta_2, \dots, \theta_k\}$  represents the width of the basis function, and it can change from variable to variable. Also, the exponent values,  $\mathbf{p}$ , can also vary from one variable to another. Basically, a higher value for the exponent increases the smoothness of the function around that

variable [120]. In estimating the output of the Kriging model, it is necessary to perform matrix inversions [121]. This can significantly increase the computational costs of this method when the dimensions are high.

### **Feedforward Neural Networks**

The MATLAB implementation of neural networks is also used as a surrogate model. A feedforward multilayer perceptron uses three or more layers of neurons (nodes) with nonlinear activation functions. The learning process in this type of neural network takes place by altering the connection weights after each segment of data is processed, based on the amount of error in the output compared to the expected result.

#### **3.15.4. Surrogate Modeling (SUMO) Toolbox**

In this research, the SURrogate MOdeling (SUMO) MATLAB toolbox [122] is used to construct the surrogate models. This open source toolbox is developed by the IBCN research group of the Department of Information Technology (INTEC) at Ghent University. It has excellent extensibility, making it possible for the user to add, customize and replace any element of the modeling procedure. It also has a wide range of built-in test functions and test cases, as well as support for many different model types.

In SUMO, first, an initial design is generated and evaluated. Then, a set of models is constructed, and the accuracy of these models is approximated using a set of measures, such as cross-validation or an external validation test set. Each type of model has a number of parameters which can be customized, e.g. degrees of freedom for rational models, smoothness for RBF models, number and size of hidden layers in neural networks, etc. These parameters are adjusted using an optimization method, and additional models are produced until no additional improvement can be made by changing the model parameters. If the desired accuracy has not so far been reached, an adaptive sampling algorithm is employed to generate a new set of sample locations, and the algorithm starts from the beginning.

#### **3.15.5. Model Management**

One drawback of using surrogate models in the context of evolutionary algorithms is the convergence to a “false optimum”, i.e. a solution that is an optimum to the approximate surrogate model, and not to the original objective function [123]. In order to overcome this

problem, it is crucial to use the surrogate model with the original objective function. The approach that is used to select some (or all) of the individuals to be used with the original fitness function is called model management [124]. It can be divided into two major forms:

- *Fixed Evolution Control.* A fixed percentage of the individuals are chosen to be evaluated by the original model. The individual members can be chosen randomly, or based on the value of the approximate objective function, i.e. choosing the ones that seem to be more promising based on the pre-evaluation using the surrogate model. If a lower portion of the individuals are chosen, it results in a faster iteration, but it is possible to miss the global optimum, and a larger number of iterations may be required.
- *Adaptive Evolution Control.* In this strategy, the number of members that are chosen to be evaluated by the original model can be different in each iteration. The selection criteria can be a function of the values generated by the surrogate model, the iteration number, the degree of improvement in the results, or other factors based on the nature of the problem and the chosen optimization algorithm. In this research, we investigate the usage of two different strategies of adaptive evolution control. The first one is based on the concept of “index of improvement” [125]. If  $f_{min}$  is the current minimum of the objective function, and  $\hat{f}(x)$  is the predicted value of the objective function by a new point  $x$ , then the index of improvement can be defined as

$$I(x) = \begin{cases} 0 & \text{if } \hat{f}(x) > f_{min} \\ f_{min} - \hat{f}(x) & \text{otherwise} \end{cases} \quad (3.24)$$

If a point has a large value of this index of improvement, then it is more likely to improve the value of the objective function and should be evaluated using the full simulation method. In our research we started testing all the samples with  $I(x) \geq 0.1 f_{min}$  in the initial iterations, and then gradually relaxed this condition in later iterations, where the values are closer to each other, and improvements are smaller. The second type of adaptive evolution control, employed in this research, is based on the iteration number, i.e. a smaller ratio of the solutions were fully tested in the first iterations, and then this number was gradually increased to include nearly all of the solutions in the final iterations. This strategy is based on the observation that the candidate solutions in the later iterations generally have better qualities, and also, that the search in these later stages is more localized. Since the values of the objective function are going to be closer to each other, surrogate modeling may not be able to correctly sort them, and a high fidelity solution is more desirable.



### 3.15.6. Model Configuration

In the inverse problems with multiple sensors, such as in test cases IV and V, there are two different strategies in designing the surrogate model. One is to construct a simple model for a single thermocouple location and its respective boundary heat flux. In this case, we will need to run this model for the number of sensors in the problem. The other strategy is to create a larger model that takes the whole boundary heat flux matrix as its input (with rows for different time steps and columns for different sensor locations), and its output is also a matrix including the temperature responses in all the thermocouple locations. In this strategy, only one surrogate model is needed, but it will be larger and more complicated. This latter form is a more physical representation of the real problem. In this research, for test cases IV and V, we use both strategies. The first one will be called “3D – 1 sensor”, and the second method will be called “3D – 9 sensors” and “3D – 18 sensors” for test cases IV and V, respectively. We will also study the application of surrogate modeling in a simple case, by investigating test case I. In this test case, since there is only one sensor in the domain, only one kind of surrogate model is used; “1D – 1 sensor”. Their performances will be compared in terms of the time needed to construct the model, the prediction time, the total solution time, and their accuracy in predicting the results and in dealing with noisy domains.

### 3.15.7. Results

The iterations were continued until the normalized value of the best objective function with respect to the best value of the objective function in the first iteration became smaller than  $10^{-7}$ . Hence, the qualities of the results in all cases were very similar. However, the time and computational expense required for the method to reach such a solution were very different. Figure 3.21 shows the typical results of the applied algorithm.

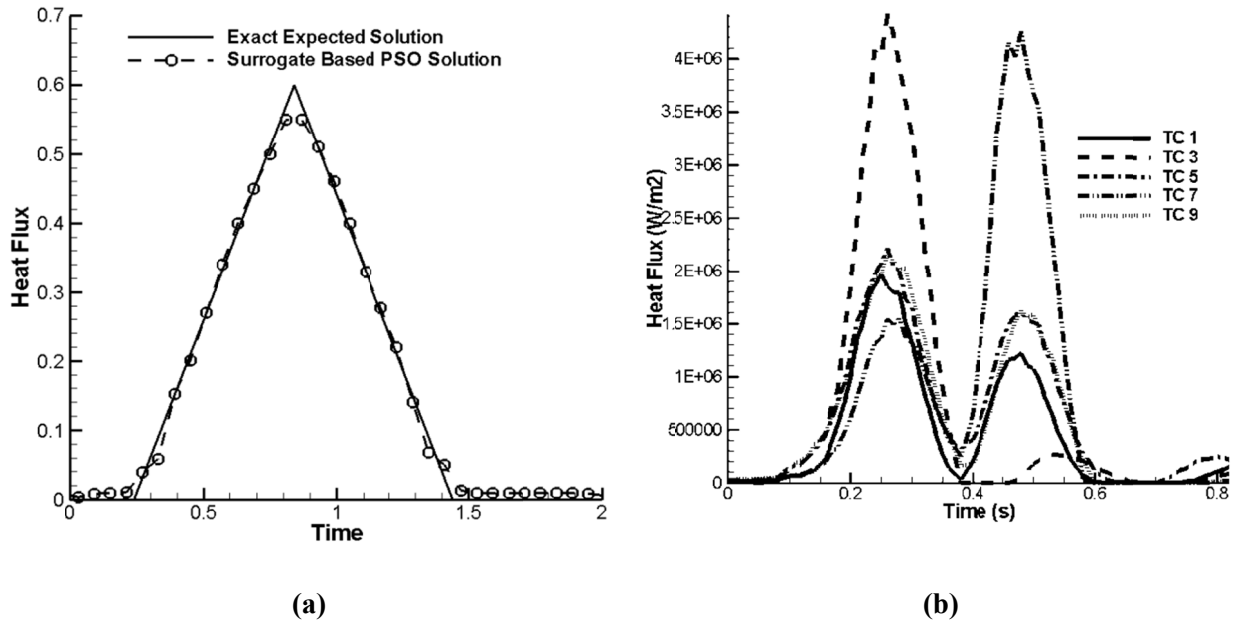


Figure 3.21. Results of Surrogate Based PSO for (a) Test Case I; (b) Test Case IV

### 3.15.8. Solution Time

First, we are going to compare the four previously mentioned surrogate modeling techniques (polynomial, radial basis function, Kriging, and feedforward neural networks) and modeling strategies (1 sensor, 9 sensors, and 18 sensors) in terms of computational efficiency. Efficiency is simply related to the amount of time required for a method to reach a certain goal. However, in surrogate modeling there are two distinct kinds of efficiency. The first one is related to the time it takes to construct a model. The second kind of computational efficiency is related to the time required to perform new predictions using the constructed model. In this research, we compare both kinds of efficiency of the four surrogate modeling techniques. All of the simulations are done on a PC with an Intel Pentium D 2.80 GHz CPU, with 2 GB of RAM, and running Microsoft Windows XP Professional. Table 3.16 and Table 3.17 show the required time for constructing and implementing the models, respectively.

**Table 3.16. Required Time (in Seconds) to Construct the Model for the Linear Case**

	<b>Polynomial</b>	<b>RBF</b>	<b>Kriging</b>	<b>FF Neural Network</b>
<b>1D – 1 sensor</b>	< 1	4.42	43.70	9.63
<b>3D – 1 sensor</b>	12.14	15.46	306.73	43.81
<b>3D – 9 sensors</b>	96.68	125.38	2183.97	316.53
<b>3D – 18 sensors</b>	146.47	187.44	2915.53	550.76

**Table 3.17. Required Time (in Seconds) for 1000 New Predictions**

	<b>Polynomial</b>	<b>RBF</b>	<b>Kriging</b>	<b>FF Neural Network</b>
<b>1D – 1 sensor</b>	< 1	9.13	6.65	9.55
<b>3D – 1 sensor</b>	< 1	11.19	7.43	14.47
<b>3D – 9 sensors</b>	5.14	90.75	68.44	119.45
<b>3D – 18 sensors</b>	9.05	147.92	125.24	205.45

As can be seen from Table 3.16 and Table 3.17, the polynomial regression model is the fastest model to develop and implement. Radial basis function and feedforward neural networks have comparable implementation times (both much more than the polynomial method), but in terms of model construction, the radial basis function model is faster. The construction time for the Kriging model is high, especially for the cases with more sensors. As expected, using a more complex model (more sensors) will increase the time required for both constructing the model and making predictions.

### **3.15.9. Accuracy and Noise Resistance**

Another important measure for evaluating the surrogate modeling methods is their accuracy in predicting the exact values, and also their effectiveness in dealing with noisy domains. To examine this latter property, we add artificial noise to the results of the direct solution for the points that are used to construct the model. Then the model is evaluated for

another set of points to see how it is affected by the noise. Random errors are imposed onto the calculated exact internal temperatures with the following equation:

$$T_m = T_{exact} + \sigma \cdot r \quad (3.25)$$

where  $T_m$  is the modified temperature that is used instead of the exact temperature,  $T_{exact}$ ;  $r$  is a normally distributed random variable with zero mean and unit standard deviation; and  $\sigma$  is the standard deviation.

To compare the methods, the  $R^2$  value is calculated for 20 samples:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (3.26)$$

where  $\hat{y}_i$  is the predicted value, corresponding to the expected value  $y_i$ , and  $\bar{y}$  is the mean of the expected values. Obviously, a higher value of  $R^2$  (closer to 1) is more desirable. Table 3.18 shows the value of  $R^2$  for different methods, for both the exact and noisy domains, for the linear case, i.e. when the thermal conductivity is a constant value. Table 3.19 shows the same results, but for the nonlinear case, i.e. the temperature dependent thermal conductivity.

**Table 3.18.  $R^2$  Values for Different Surrogate Models in the Exact and Noisy Domains for the Linear Case**

		<b>Polynomial</b>	<b>RBF</b>	<b>Kriging</b>	<b>FF Neural Network</b>
<b>1D – 1 sensor</b>	<b>exact</b>	0.99	0.99	0.99	0.99
	<b>noisy</b>	0.90	0.95	0.84	0.96
<b>3D – 1 sensor</b>	<b>exact</b>	0.99	0.98	0.99	0.99
	<b>noisy</b>	0.88	0.93	0.81	0.96
<b>3D – 9 sensors</b>	<b>exact</b>	0.99	0.99	0.99	0.99
	<b>noisy</b>	0.90	0.94	0.85	0.97
<b>3D – 18 sensors</b>	<b>exact</b>	0.99	0.99	0.99	0.99
	<b>noisy</b>	0.91	0.94	0.85	0.97

**Table 3.19.  $R^2$  Values for Different Surrogate Models in the Exact and Noisy Domains for the Nonlinear Case**

		<b>Polynomial</b>	<b>RBF</b>	<b>Kriging</b>	<b>FF Neural Network</b>
<b>1D – 1 sensor</b>	<b>exact</b>	0.69	0.87	0.92	0.78
	<b>noisy</b>	0.64	0.80	0.70	0.75
<b>3D – 1 sensor</b>	<b>exact</b>	0.58	0.86	0.90	0.75
	<b>noisy</b>	0.56	0.82	0.71	0.73
<b>3D – 9 sensors</b>	<b>exact</b>	0.60	0.86	0.91	0.77
	<b>noisy</b>	0.57	0.83	0.71	0.74
<b>3D – 18 sensors</b>	<b>exact</b>	0.60	0.87	0.92	0.77
	<b>noisy</b>	0.57	0.83	0.72	0.75

As seen from Table 3.18, when there is no noise in the domain, and the problem is linear, the three methods produce results with very similar levels of accuracy. However, in the case of noisy domains, the feedforward neural networks are the safest choice, followed closely by RBF. Polynomials and Kriging show the worst performance. Comparing Table 3.19 with Table 3.18 shows that the nonlinearity in the problem considerably lowers the accuracy of all surrogate techniques. This is especially pronounced in the case of polynomials. The Kriging model performs better than the rest in the case where the data is exact. However, addition of noise significantly reduces its accuracy. In the nonlinear problem with noisy data, the RBF model is the most accurate one. Since most real-world inverse heat conduction problems are nonlinear and noisy, the RBF method seems most suitable.

### **3.15.10. Surrogate Model Management**

Since the most significant part of the computational expense is created by the full exact solution of the direct problem, the effect of model management (number of the most promising particles that have been solved exactly using finite element techniques) is studied in Figure 3.22. The RBF model is used in this section, since in the previous section, it was found to be the most robust surrogate model for inverse heat conduction problems. The “Adaptive I” strategy is based on the concept of the index of improvement, as mentioned in Equation (3.24). The “Adaptive II”

strategy is based on changing the percentage of exactly evaluated members based on the iteration number. The figure shows the value of the objective function attained by the best performing particle in each iteration, normalized to the best value in the first generation, versus the total number of full direct simulations of the direct problem using the 3D finite element code for test case IV. As is seen in Figure 3.22, the computational expense is reduced considerably if only a smaller portion of promising members are evaluated (e.g., 20% in this problem). However, choosing a very small percentage (10% in this example) is inefficient, since it requires more iterations, and the total number of direct solutions will be more than the 20% case. Also notice that the strategy of evaluating 100% of the particles is equivalent to not using surrogate modeling at all. In this sense, we can say that using surrogate modeling can speed up the PSO-based inverse algorithm up to around 3 times and the GA-based inverse algorithm up to around 4.5 times.

Table 3.20 shows the effect of using the surrogate models on CPU time. A considerable amount of speedup is observed when using a suitable ratio for the number of exactly evaluated swarm members. Also, it seems that using simpler model configurations (1 sensor instead of 9 or 18) is preferable. This is reasonable considering that the nature of surrogate modeling is based on approximation, and not a full detailed solution. As can be seen from Table 3.21, the speedup of the nonlinear case is similar to the one in the linear case, and RBF seems to perform a very good job there as well. This can also be explained as the main role of the surrogate models is ranking of candidates, and even though in the nonlinear case, the accuracy of the prediction is lower, the RBF model can still choose which candidates will perform better.

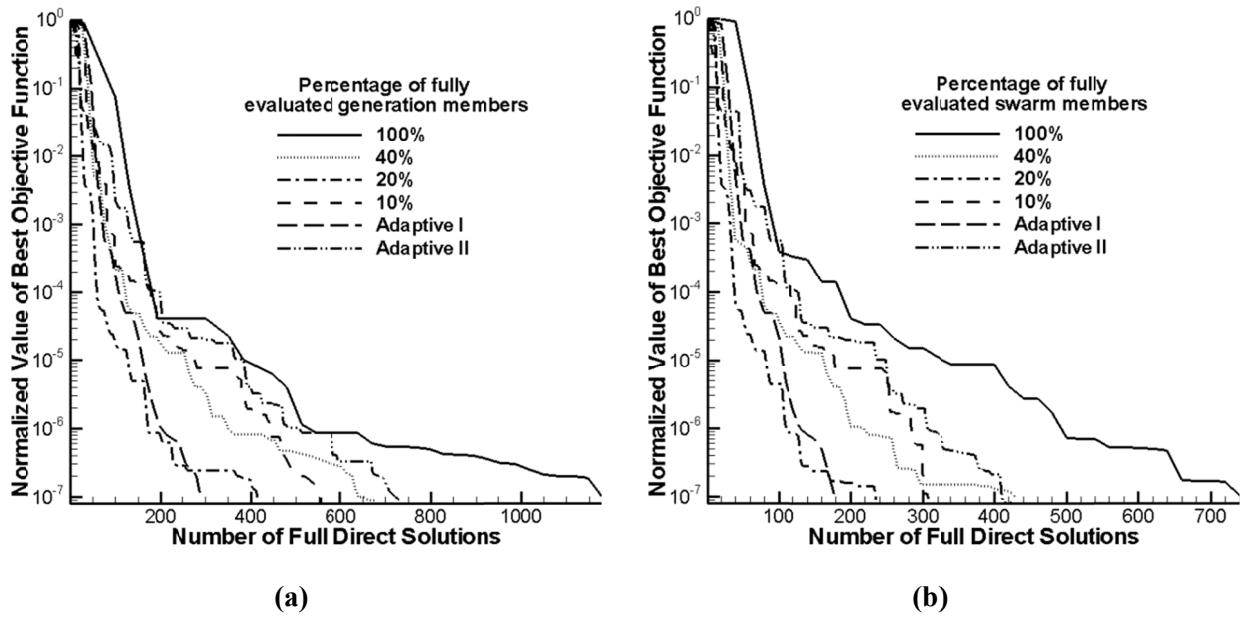


Figure 3.22. Effect of Surrogate Model Management on the Performance of Inverse Analyzer; (a) GA Solver; (b) PSO Solver

Table 3.20. Effect of Using Surrogate Models on the Total Solution Time (s) for the Linear Case

% of fully evaluated members		100%	40%	20%	10%	Adaptive I	Adaptive II
1D – 1 sensor	GA	5831	3341	2317	3386	2241	3712
	PSO	3394	1986	1268	1811	1152	2105
3D – 1 sensor	GA	15301	8869	5460	7345	4060	9895
	PSO	8951	5006	2685	3498	2194	4715
3D – 9 sensors	GA	15301	11404	5253	6977	3820	8324
	PSO	8951	6257	2891	3374	1889	3822
3D – 18 sensors	GA	15301	11852	5323	6120	3569	6809
	PSO	8951	7010	3090	3187	1812	3683

**Table 3.21. Effect of Using Surrogate Models on the Total Solution Time (s) for the Nonlinear Case**  
**% of fully evaluated members**   **100%**   **40%**   **20%**   **10%**   **Adaptive I**   **Adaptive II**

<b>1D – 1 sensor</b>	<b>GA</b>	6036	3584	2354	3397	2422	4070
	<b>PSO</b>	3483	1995	1268	1864	1162	2256
<b>3D – 1 sensor</b>	<b>GA</b>	16071	9394	5511	8139	4189	10103
	<b>PSO</b>	9174	5282	2991	3537	2473	5325
<b>3D – 9 sensors</b>	<b>GA</b>	16071	11821	5625	7117	4281	9360
	<b>PSO</b>	9174	6409	3178	3424	2102	4306
<b>3D – 18 sensors</b>	<b>GA</b>	16071	12818	5634	6713	3629	7254
	<b>PSO</b>	9174	7054	3251	3567	1915	3710

### 3.16. Conclusion

In order to overcome the shortcomings associated with the gradient-based inverse heat conduction solvers in our steel cooling applications, such as the instabilities due to the thermocouple reading noise, and low temporal resolution, some gradient-free algorithms were investigated in this chapter. These methods are artificial neural networks, genetic algorithms, and particle swarm optimization.

The artificial neural networks are capable of capturing the whole thermal history on the run-out table, but are not very effective in restoring the detailed behavior of the boundary conditions. Also, they behave poorly in nonlinear cases and where the boundary condition profile is different. Thus, their application in the run-out table heat transfer analysis will be limited to a general prediction of the whole history.

GA and PSO are more effective in finding a detailed representation of the time-varying boundary conditions, as well as in nonlinear cases. However, their convergence takes longer. Comparison was made between GA and PSO variations in terms of both the computational efficiency and effectiveness in dealing with noisy data. PSO was found to improve efficiency



(i.e., reduce the required computational effort), especially in more complex test cases. A variation of the basic PSO, called CRPSO in this research, showed the best performance among the three versions. The effectiveness of PSO was also studied in the presence of noise. PSO proved to be effective in handling noisy data, especially when its performance parameters were tuned. The proper choice of the regularization parameter helped PSO deal with noisy data, similar to the way it helps the classical function specification approaches. An increase in the self-confidence parameter was also found to be effective, as it increased the global search capabilities of the algorithm. RPSO was the most effective variation in dealing with noise, closely followed by CRPSO. The latter variation is recommended for inverse heat conduction problems, as it combines the efficiency, i.e. a lower computational expense, and effectiveness, i.e. resistance to the measurement errors, required by these problems.

Sequential formulation, i.e. solving the inverse problem for each separate time step was found more efficient than a whole domain implementation. Also, in the case of multi-dimensional problems with multiple sensors, the proposed multi-objective optimization was very successful in accelerating the solution. The relative speedup seemed to be independent of the number of spatial components. Introducing the concept of elite members to the algorithm also resulted in a faster convergence; however, the improvement was of less magnitude. It also looked more effective in simpler cases, i.e. smaller problems with fewer internal measurement points. In order to stabilize the algorithm, especially making it more effective in dealing with noisy measurement data, the concept of future time steps was borrowed from the function specification methods, and was applied to the sequential algorithm. It was very helpful in making the method more stable, especially when used in combination with a Tikhonov regularization term and tuning of the self-confidence parameter in PSO. The developed modified formulation for the inverse heat conduction problem and PSO implementation seem to be promising in terms of stability and performance for the cases similar to those studied in this research.

While evolutionary optimization methods are very effective in dealing with inverse heat conduction problems in terms of stability and handling of noisy data, their very high computational expenses hinder the wide usage of these algorithm. Inexact pre-evaluation using surrogate models can be used to enhance the speed of evolutionary algorithms in solving the inverse heat conduction problems. In this strategy, the entire candidate solutions are first roughly evaluated using a much simpler surrogate model, and only the top performing members are

exactly evaluated using the full direct solvers, e.g. finite element method. Polynomial methods are the cheapest selection, both during the model construction phase, and the implementation. Radial basis function and feedforward neural networks have similar construction time, but in making predictions, RBF models perform faster. Kriging model is the slowest, especially when the number of unknowns increases. All the investigated surrogate models perform very well in predicting linear cases with no added noise. Nonlinear domains are much harder to predict. The most suitable model for noisy nonlinear cases is the RBF model. Since real-world cases of inverse heat conduction problem are normally nonlinear and noisy, this method is recommended for these applications. The best performing model management strategy is an adaptive selection scheme, based on the expected improvement of the objective function. Also, the most appropriate way to construct the surrogate model is to make a separate model for each sensor. If done properly, surrogate modeling can speed up the stochastic methods of solving the inverse heat conduction problem up to around 5 times.

Finally, based on the performance that was measured on the test cases in this research, a sequential implementation of the CRPSO algorithm, using a multi-criteria formulation of the objective function, and using future time steps, Tikhonov regularization, and a tuned self-confidence parameter, in conjunction with a single-sensor RBF surrogate model, using the adaptive selection based on the expected improvement of the objective function, is the most appropriate solution method for solving inverse heat conduction problems similar to the ones studied in this research, with characteristics like the ones encountered on a steel run-out table.

## 4. Heat Transfer Behaviour under Multiple Circular Water Jets Impinging on a Moving Plate

In this chapter, we study the boiling heat transfer characteristics of two rows of three impinging jets on a moving hot plate. Experiments are conducted in an industrial scale setup, and thermocouples are used to measure the temperatures just under the plate surface. Then these temperatures are used as the input in our inverse heat conduction algorithm to find the boundary heat flux variations in space and time across the width and along the length of the moving flat plate. The effects of nozzle configuration, jet line spacing, and plate speed are investigated. The results in this chapter are obtained using our developed CRPSO algorithm, as introduced in the previous chapter. The sequential implementation is always used, and multi-criteria formulation of the objective function, future time steps, Tikhonov regularization, and a tuned self-confidence parameter are adopted to increase the stability of the method. Also, a single-sensor RBF surrogate model, using the adaptive selection based on the expected improvement of the objective function is utilized to reduce the computational costs.

### 4.1. Experimental Setup

The general characteristics of the experimental run-out table facility were described in chapter 1. In the experiments that are analyzed in this chapter, there are eighteen thermocouples attached to each plate. The ones that are located along the lateral direction are 12.6 mm apart from each other. The distance between the two sets of thermocouples is either 25.2 cm, or 50.4 cm, based on the distance between jet lines for that experiment. Figure 4.1 shows a top view of the TC configuration on the plate.

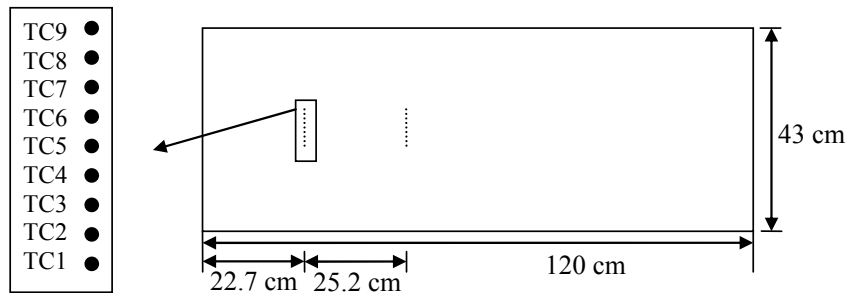
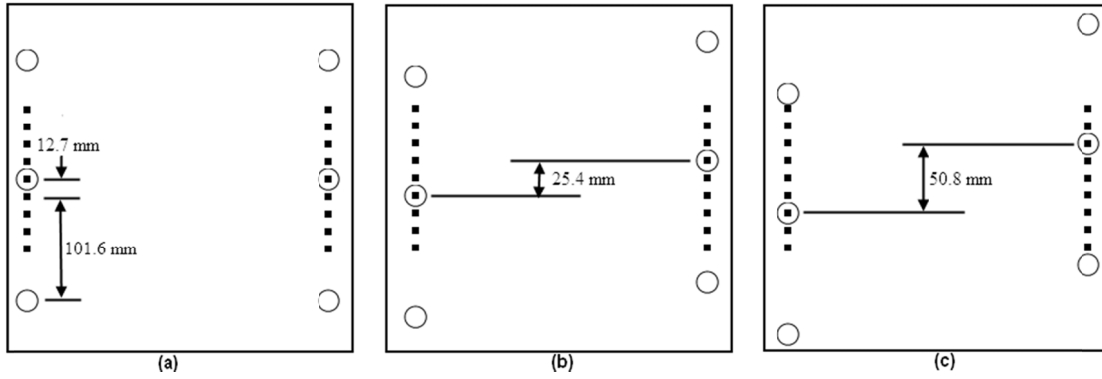


Figure 4.1. A Sample Plate, Showing the Dimensions and Thermocouple Locations

Thermocouple signals, as well as the flow rates from the flow meter, are gathered by two data acquisition boards, and are sent to a PC that uses DASyLab<sup>®</sup> 7.0 data acquisition software. The same software also controls the water flow rate that exits the nozzles. In our experiments, the nozzles are placed 1.5 m above the plate. The spacing between the two successive jet lines is either 25.4 cm or 50.8 cm. The nozzle diameter is 19 mm. There are three possible jet configurations: in-line (no stagger), half staggered, and fully staggered. These three configurations are shown in Figure 4.2. The plate speed is set to either 0.35 or 1.0 m/s. The water flow rate is constant at 15 l/min, which gives an outlet velocity of 0.88 m/s. The water temperature is 25 °C.



**Figure 4.2. Nozzle Configuration with Respect to Thermocouples; (a) In-line; (b) Half-staggered; (c) Full-staggered [31].**

The jet velocity (impact velocity) is the velocity at which the water hits the plate, and can be calculated by

$$V_j = \sqrt{V_n^2 \pm 2gH} \quad (4.1)$$

where  $V_j$  is the jet velocity, m/s;  $V_n$  is the water velocity at nozzle exit,  $g$  is the gravitational acceleration,  $\text{m/s}^2$ ; and  $H$  is the vertical distance from the nozzle exit to the plate surface,  $m$ . For our experiments the jet velocity will be 5.49 m/s.

The diameter of the water jet  $D_j$  can be calculated by

$$D_j = D_n \cdot \sqrt{V_n/V_j} \quad (4.2)$$

where  $D_n$  is the nozzle diameter. In our case, the water jet diameter at the stagnation point will be 7.6 mm.

The pressure at the stagnation point is given by

$$P_s = P_a + \frac{1}{2} \rho V_j^2 \quad (4.3)$$

where  $P_a$  is the atmospheric pressure and  $\rho$  is the density of water. In this case, the pressure at the stagnation point will be about 116,389 Pa. The saturation temperature of water at this condition can be found from a saturation table of water, and is around 103.6 °C.

The plate passes through the system several times, so the input plate temperature varies from around 700 °C, down to just slightly more than 100 °C. This gives us the opportunity of studying different boiling heat fluxes, due to the various levels of temperature gradient. The experimental database used in this research is developed by Franco [31].

Twelve different experimental setups are analyzed, each with a different combination of jet line spacing, nozzle stagger, and plate speed. Table 4.1 shows these setups. In the following sections, we are going to study the effect of these parameters on the heat transfer behavior on the plate surface.

**Table 4.1. Test Matrix.**

<b>Set-up</b>	<b>Jet Line Spacing (cm)</b>	<b>Nozzle Stagger (mm)</b>	<b>Plate Speed (m/s)</b>
S1	25.4	0	0.35
S2	25.4	0	1.0
S3	25.4	25.4	0.35
S4	25.4	25.4	1.0
S5	25.4	50.8	0.35
S6	25.4	50.8	1.0
S7	50.8	0	0.35
S8	50.8	0	1.0
S9	50.8	25.4	0.35
S10	50.8	25.4	1.0
S11	50.8	50.8	0.35
S12	50.8	50.8	1.0

In order to determine the heat fluxes caused by impinging water jets on the surface of hot moving plate, we need to find the surface temperatures. Unfortunately, measuring the surface temperatures directly is not accurate. Our initial experiments with placing thermocouples on the plate surface showed an error of around 250 °C, when the surface temperature was around 700 °C [25]. Other methods such as infrared thermography may not be influenced by the water interaction on the surface, but are very hard to get in our application, since the water vapor covers the plate surface. Also these methods provide a snapshot of the surface temperature at each step rather than a continuous recording of temperature history during time. As a result, researchers use thermocouple readings from inside the plate to obtain the boundary heat flux values. The inverse algorithms that were developed in the previous chapters are used to obtain the surface temperature and heat fluxes during the water cooling process.

## **4.2. Uncertainty Analysis**

As it was mentioned before, type K thermocouples that are used in this study, have an error of approximately 2 °C. There is also some uncertainty in the values of thermophysical properties, such as heat conduction coefficient, and heat capacity. Thermocouple position and depth are the other sources of uncertainty. Since these values are the inputs of the inverse heat conduction analysis, their uncertainties will propagate through the algorithm, and will result in an uncertainty band around the output values, such as the surface heat flux and temperature. While it is possible to use analytical equations to calculate the uncertainty in boundary heat fluxes for simple linear inverse heat conduction problems, in case of nonlinear problems with multiple sensors, a numerical procedure is normally required [126]. This approach is called the “computerized uncertainty analysis” [127]. While there are several methods available for this approach [128], in this research, we have used a finite difference method, based on the sequentially perturbing the inputs, as detailed in [127].

Our calculations indicated that the uncertainty in the values of surface heat flux and temperature is between  $\pm 8\%$  for the parallel flow zones, away from the stagnation point, and  $\pm 16\%$  under the impinging jet. Also, it should be noted that the several sets of repeatability measurements were performed, and the data showed excellent repeatability [31].

### 4.3. Time Histories of Temperature and Heat Flux

Based on our observations, the most significant factor affecting the heat transfer behavior is the range of plate surface temperature when entering the jet line. Thus, we have presented many of our results in three different entry temperature ranges: High (500 – 750 °C), mid (250 – 500 °C), and low (100 – 250 °C). Figure 4.3 - Figure 4.5 show the time history of thermocouple readings (inside temperature) from the experiments, and surface temperature just above the thermocouple and the heat flux at the location of thermocouple, both obtained by our inverse analysis, at five cross-section locations, for three different entry temperature ranges: high temperature (around 700 °C, Figure 4.3), mid temperature (around 400 °C, Figure 4.4), and low temperature (around 120 °C, Figure 4.5). The data shown here is from the S6 setup, i.e. a jet line spacing of 25.4 cm, a nozzle stagger of 50.8 cm (full stagger), and a plate speed of 1.0 m/s are used. That means that for the first pass, the 3<sup>rd</sup> thermocouple is under the impingement, and for the 2<sup>nd</sup> pass, the 7<sup>th</sup> one is in the impingement zone.

As it can be seen in Figure 4.3(a), the inside temperature is initially around 700 °C (A). All temperatures fall after the line reaches the first jet line (B). The magnitude of the temperature drop is much higher for the TC location under the jet (E). For the rest of the TCs, the magnitude of the drop is more or less of the same order, but very gradually decreases when we move away from the impingement point (TCs 1 and 7). When the plate moves away from the jet line, the temperature partially recovers, due to the thermal mass of the plate (C). However, the existence of the second jet line stops the recovery process, and reduces the temperature once again (D). Note that the second jet line is impinging on a plate with a quite non-uniform temperature distribution, which is also lower than the initial entry temperature. Due to this non-uniformity, the history of the second jet line shows a larger diversity in behavior. Once again, the most significant temperature drop occurs at the impingement location (TC 7 this time). After moving away from the 2<sup>nd</sup> jet line, temperature again recovers, and due to the lack of a new jet line, the temperature distribution starts to become uniform again (G).

Figure 4.3(b) shows the surface temperatures at the location of each TC. These values are obtained by using the inverse analysis. As expected these values are smaller than the inner temperatures, and if examined carefully, exhibit an earlier response to the jet line. This is of course due to the time lag that exists for the boundary conditions to be sensed inside the plate. It is also observed that the amount of the temperature drop is larger for the surface compared to the

temperature drop inside the plate. This is also related to the damping of changes in boundary condition inside the plate. These two effects (time lag and damping) are always present in inverse heat transfer analysis and contribute to the complexity of such problems.

Figure 4.3(c) displays the surface heat flux at the location of each TC for the same experiment, again obtained by the inverse analysis. The heat fluxes are initially very small, and close to zero (A). After the TCs reach the first jet line, there is a rise in the amount of heat fluxes (B), with the impingement zone, experiencing the largest value of heat flux (E). The same thing happens for the second jet line, however since now the temperature distribution is not uniform anymore; there is a larger diversity in the heat flux behaviors.

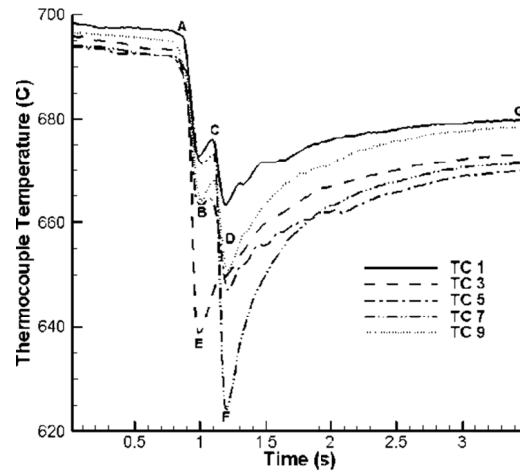
Figure 4.4(a), (b), and (c) show similar results as Figure 4.3(a), (b) and (c), respectively, but for an entry temperature of around 400 °C. The trends are qualitatively similar to the higher entry temperature case, but values are different. This is especially obvious when comparing the heat flux curves. As is seen in Figure 4.4(c), the peak heat flux value reaches a value more than twice of the peak heat flux of Figure 4.3(c). This shows that a more effective boiling mechanism is present in the mid range of surface temperatures. This can be related to a thinner vapor layer on the plate surface in this temperature range. This layer is thicker in the high temperature range, which means a film boiling heat transfer mechanism. In the mid temperature range, the momentum of the vertical impinging jet can break the thinner layer apart, and liquid can touch a wider area, so a partially nucleate boiling mechanism is responsible for higher heat fluxes.

Finally, Figure 4.5(a) represents the last series of TC readings for the low entry temperature range. With an initial inner temperature of 120 °C (A), i.e. just about 20 °C higher than the boiling point, the plate is subjected to the first row of jets. As shown in Figure 4.5(b), this causes the surface temperatures to drop below the boiling point. Interestingly, in this case, there is not a major difference between the cross sectional locations, i.e. the impingement and parallel flow zones exhibit a more uniform temperature drop. This is more clear in Figure 4.5(c). As is seen in this figure, the peak values of heat flux (B) are not as different as previous cases. This is in accordance with the findings of Jondhale [83] that the significance of boiling heat transfer in the impingement zone compared to the boiling in the parallel flow zone is reduced when the surface temperature falls below 200 °C. This is even more pronounced when the (even colder) plate passes the second jet line (D). Interestingly, in both cases of peak heat flux (B and D), the maximum heat flux is not at the location of impingement, but at somewhere between the

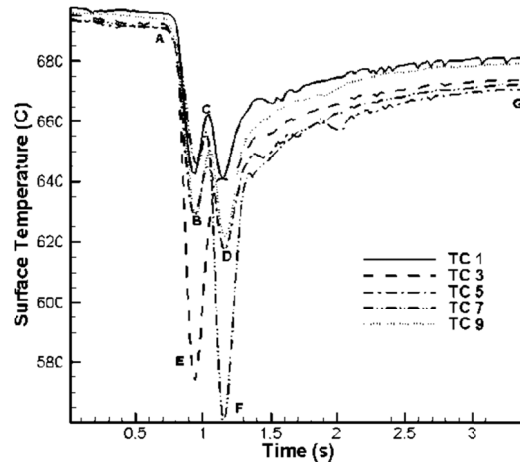


two adjacent impinging circular jets. This is in agreement with the findings of Slayzack et al. [81] about the water fountain in the interaction zone between two jets.

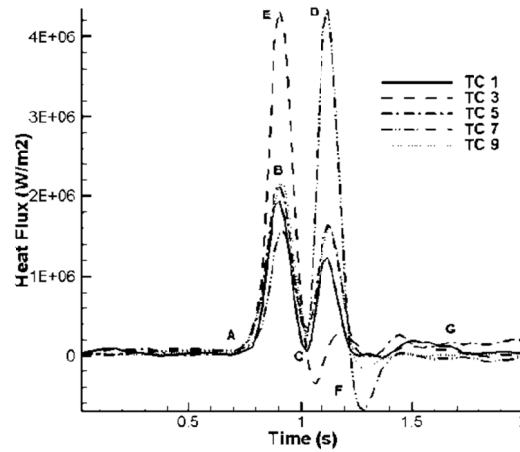
Therefore, the high entry temperature range causes a relatively thick vapor layer that acts as insulation, and the heat transfer mechanism is film boiling, unless when an impinging jet with high momentum breaks down that layer. In the mid entry temperature range, the thinner vapor layer is a weaker insulation, and the dominant mode of heat transfer is mostly transition boiling. In the low entry temperature range, the local surface temperature falls quickly below the boiling point, and the heat transfer mode is non-boiling forced convection.



(a)

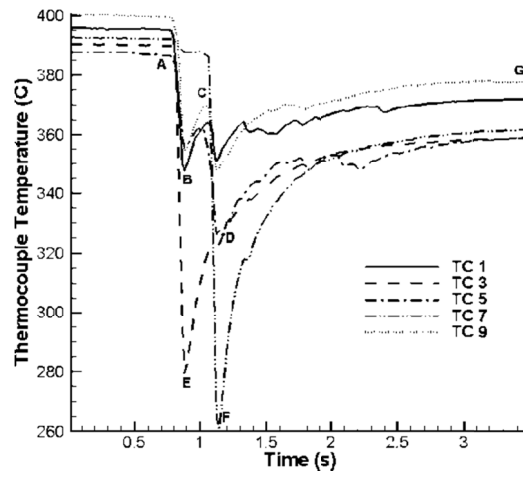


(b)

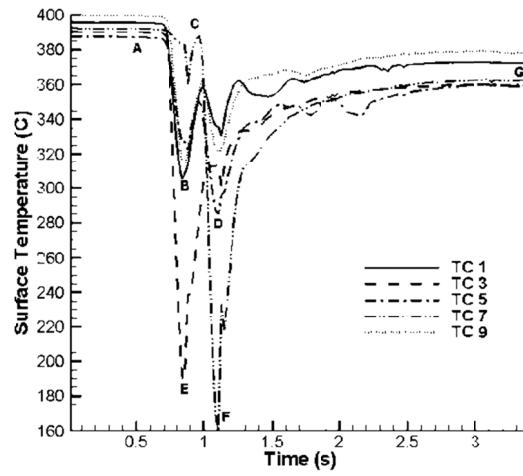


(c)

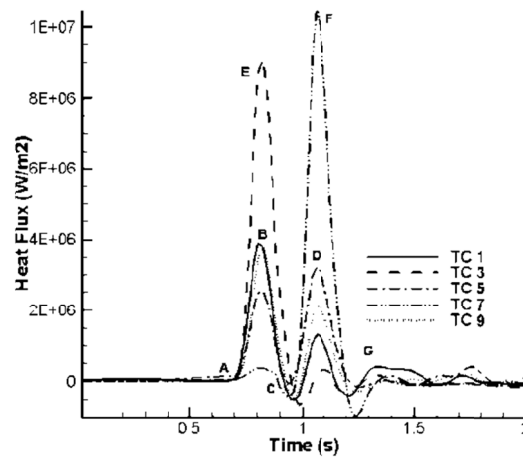
Figure 4.3. History of (a) Thermocouple Readings; (b) Surface Temperature; (c) Surface Heat Flux; High Entry Temperature



(a)

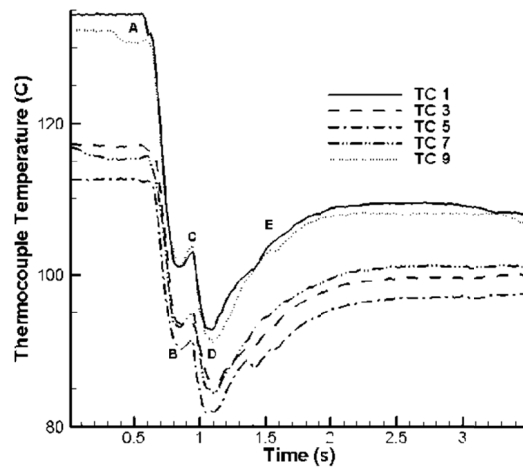


(b)

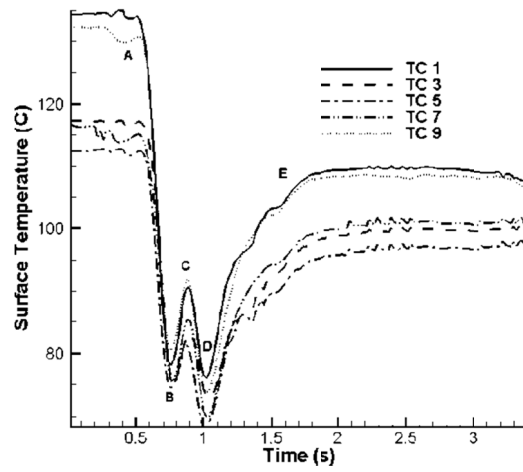


(c)

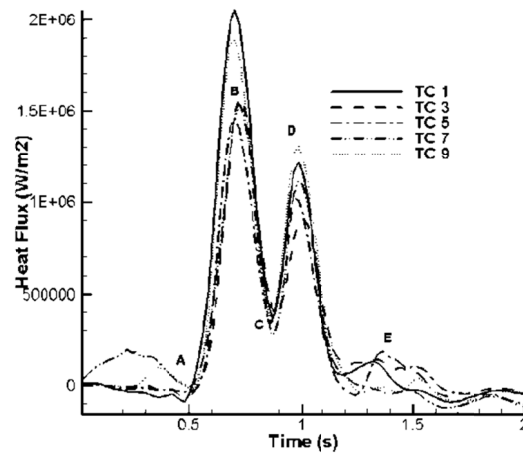
Figure 4.4. History of (a) Thermocouple Readings; (b) Surface Temperature; (c) Surface Heat Flux; Mid Entry Temperature



(a)



(b)



(c)

Figure 4.5. History of (a) Thermocouple Readings; (b) Surface Temperature; (c) Surface Heat Flux; Low Entry Temperature

#### 4.4. Heat Flux vs. Surface Temperature

Figure 4.6 show the variation of surface heat flux and heat transfer coefficient ( $h$ ) versus the surface temperature, during a two jet line experiment for high, mid, and low entry temperatures, and for five of the thermocouples across the width of the plate. Remember that both the surface heat flux and surface temperature are obtained by running an inverse analysis on the readings of internal thermocouples. The surface heat transfer coefficient is simply obtained by dividing the values of heat flux by the temperature difference between the surface and the cooling water. Since the cooling water in all these experiments has a temperature of 25 °C, the heat transfer coefficient will be

$$h = \frac{q_{surface}}{(T_{surface} - T_{water})} = \frac{q_{surface}}{(T_{surface} - 25)} \quad (4.4)$$

Figure 4.6(a) and (b), respectively show the heat flux and heat transfer coefficient vs. surface temperature for the high entry temperature range (around 700°C). The letters on the graph correspond to the same letters in Figure 4.3. As we move from (A) to (B), the temperature will fall very fast, as the impingement point will produce the highest heat flux (E) for the thermocouple that is directly under the jet. After passing the first jet line, the surface temperature partially recovers (C). The same thing happens during the impingement of the second jet line, however, with a lower initial surface temperature and a more non-uniform distribution. The plate finally exits both jet lines with a surface temperature of around 670 °C (G).

Figure 4.6(c) and (d) show the same thing but for the mid entry temperature range (around 400 °C). Here the behavior is very much like the previous case, however, the heat transfer rate is much higher. This is especially significant in the under impingement area (E and F). Also during the temperature recovery after impingement (moving toward points C and G), there is a dent in the graph, i.e. the temperature starts to recover at a high rate, then stops, a little bit of cooling occurs, and then temperature recovers at a slower rate. This should be related to the lower thermal energy that is stored in the plate.

Finally, Figure 4.6(e) and (f) show the same quantities, but for a low entry temperature experiment (around 130 °C). Here the behavior is very different from the previous two cases, especially because after the first jet line, the surface temperature falls below the boiling point, so for the second jet line, the heat transfer mechanism is not boiling, but a forced convection by the water flowing over the plate. Again, for both jet lines, maximum heat transfer is not happening

in the impingement point, but somewhere between the two jets that the interaction fountain is formed.

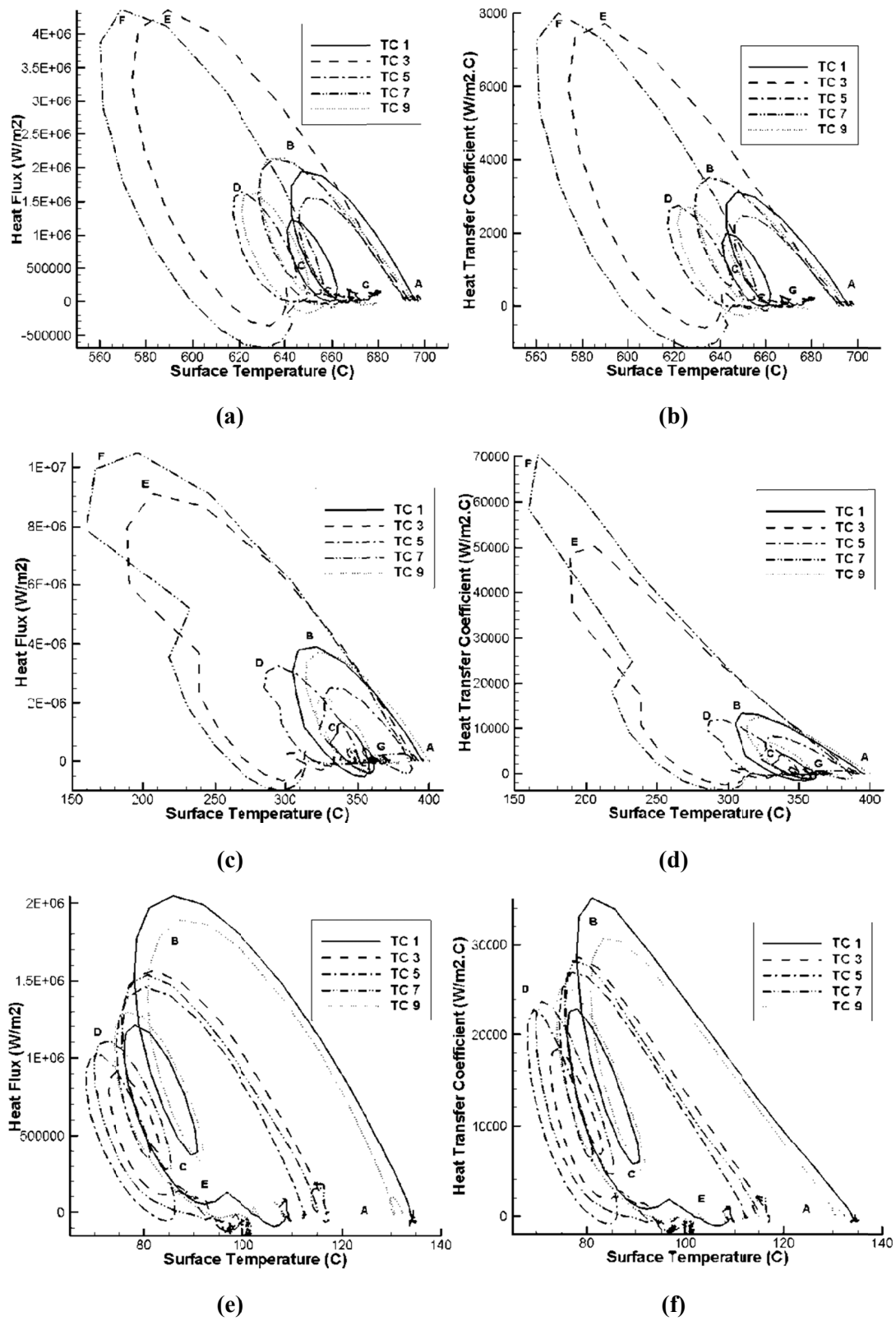


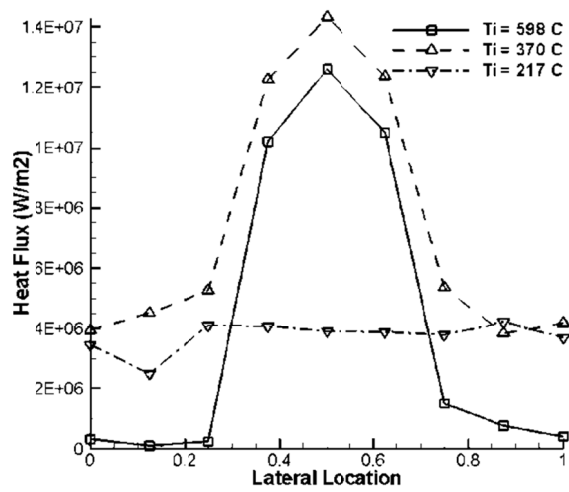
Figure 4.6. Heat Flux and Heat Transfer Coefficient vs. Surface Temperature Across the Width of the Plate for Different Entry Temperature Ranges

#### 4.5. Lateral Distribution of the Heat Flux

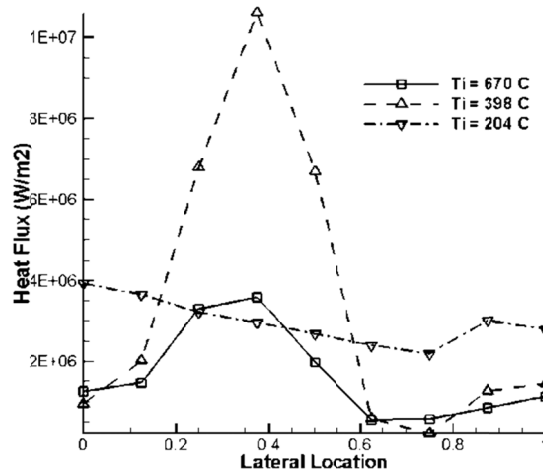
Since we only have nine thermocouples that are clustered in the central part of the plate (see Figure 4.1), we are going to use the three nozzle configurations (see Figure 4.2) to study the variation of the heat flux under and between the nozzles. The lateral location between the jets is normalized with respect to the distance between the first and last thermocouples (101.6 mm) with TC1 and TC9 at the normalized locations of zero and one, respectively. The lateral distribution of heat flux is displayed in Figure 4.7, for the moment of impingement of the first jet line. In the three cases, the jet impinges on thermocouples 5 (Figure 4.7 (a)), 4 (Figure 4.7 (b)), and 3 (Figure 4.7 (c)). For each case, three heat flux distributions are displayed, one for each of the three temperature ranges (high, mid, and low).

As is observed in these figures, the maximum heat flux values happen at the mid entry temperature range. Here, we see that the heat flux distribution is more non-uniform in the mid entry temperature range as well, with the peak heat flux occurring at the point of impingement. Also, we see that at low temperature range, the maximum heat flux is not occurring at the impingement point. The heat flux distribution is nearly symmetrical around the impingement point for the inline nozzle configuration. However, half-staggered and full-staggered nozzle configurations result in more complicated flow fields that result in an unsymmetrical heat transfer behavior around the impingement zone. This is especially more pronounced when the entry temperature is in the high or low temperature ranges. These profiles are going to be especially useful in studying the thermal stresses in plates during a cooling process on the runout table in a material processing mill, which may sometime result in a deformation in the cooled plate.

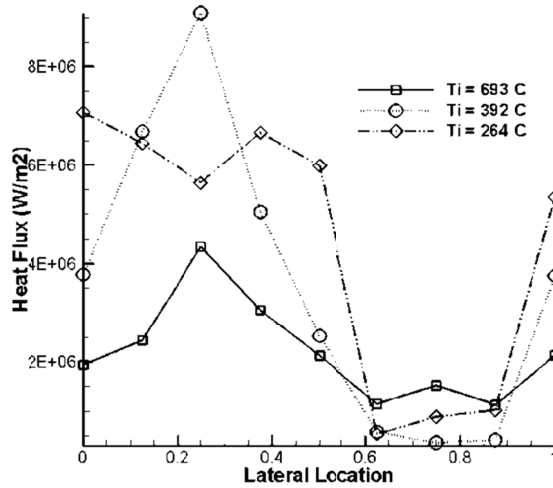




(a)



(b)



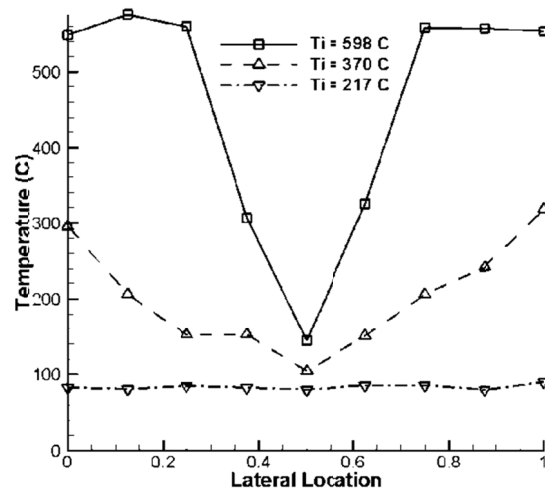
(c)

Figure 4.7. Lateral Heat Flux Distribution at the Moment of Impingement of the First Jet Line; (a) No Stagger; (b) Half Stagger; (c) Full Stagger.

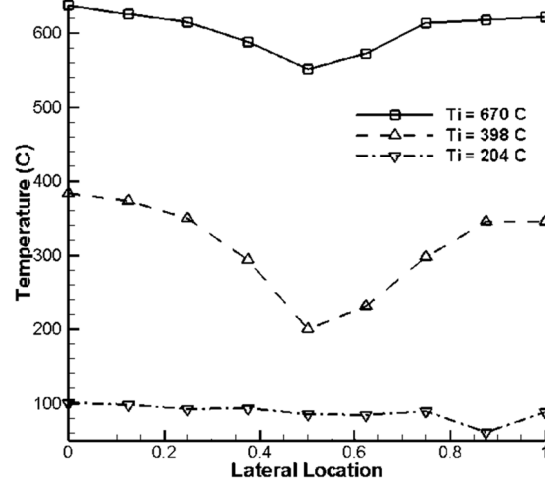
#### **4.6. Lateral Distribution of the Surface Temperature**

Figure 4.8 shows the surface temperature distribution across the plate after the second jet line has passed versus the normalized lateral location between TC1 and TC9. This data is also going to be highly useful in studying the thermal stresses in the plate, as well as making sure that different parts of the plate follow a similar temperature history profile, and thus, the mechanical properties are going to be uniform. Again for each nozzle configuration, three lateral temperature distributions are displayed, one for each of the three temperature ranges (high, mid, and low).

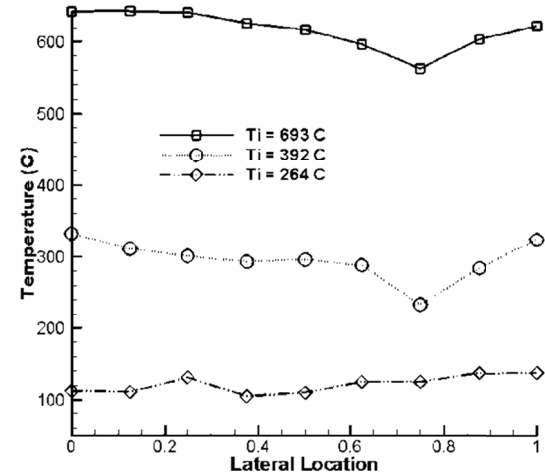
As it is seen in Figure 4.8(c), the most uniform distribution is happening in the case of fully staggered nozzle configuration. This is especially true at low entry temperature range. The most severe case of non-uniform temperature distribution is usually the case of mid entry temperature. Since this figure shows the temperature profile after both jet lines have hit the plate, one can also judge the time that it takes for the plate to reach a more homogeneous temperature distribution. As is seen in parts (b) and (c) of Figure 4.8, where the lateral location of the second line of nozzles is different than the first, the effect of the first jet line is dissipated more slowly in mid surface temperature range. This can be due to the fact that at mid surface temperature range, there is a very high heat flux occurring at the impingement point, resulting in a very effective temperature drop.



(a)



(b)



(c)

Figure 4.8. Lateral Temperature Distribution After the Impingement of the Second Jet Line; (a) No Stagger; (b) Half Stagger; (c) Full Stagger.

#### 4.7. Longitudinal Distribution of the Heat Flux

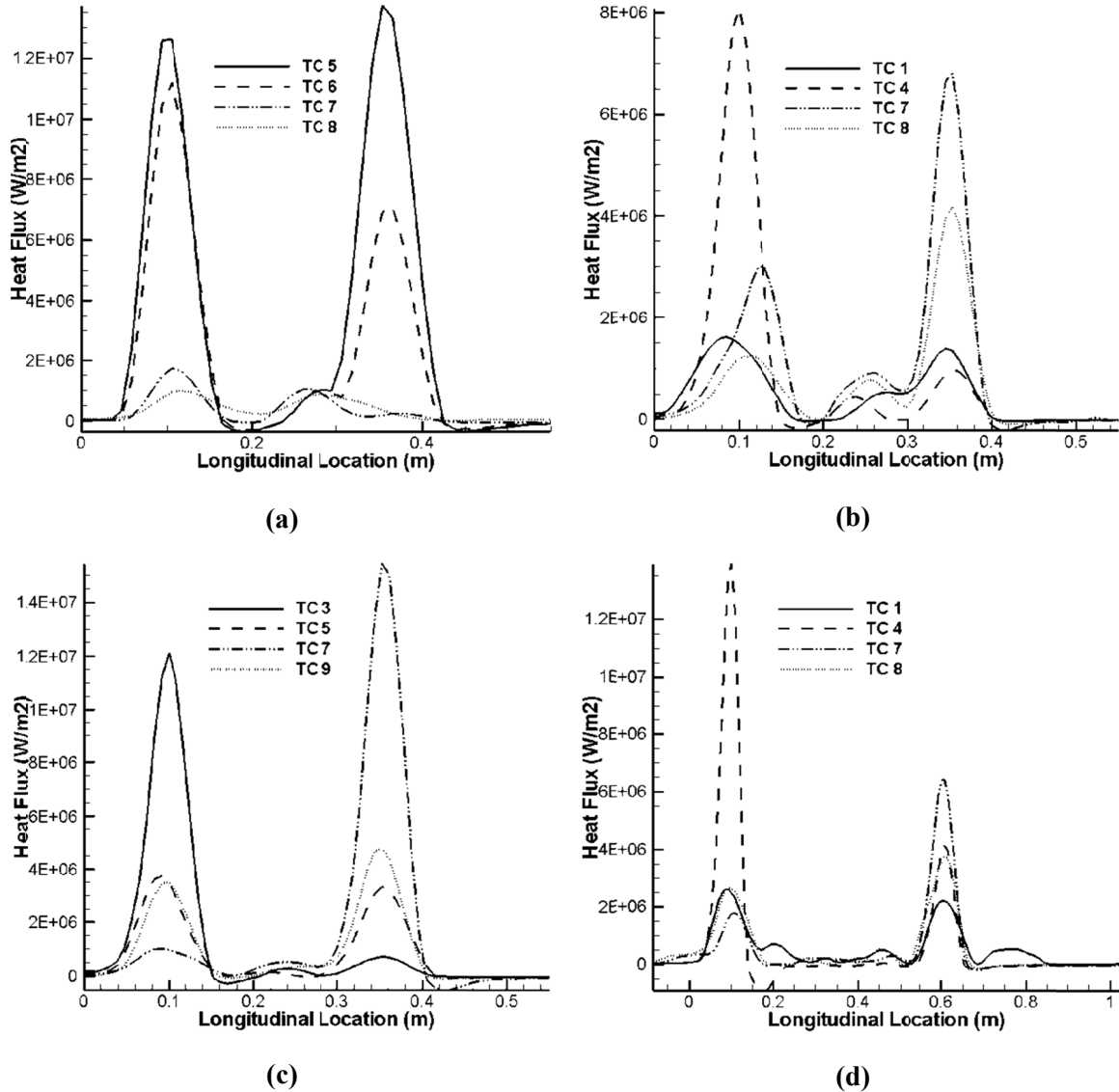
As discussed earlier, although many researchers do not distinguish the two different parallel flow zones, the flow and heat transfer fields between two successive jet rows is different from the ones between two nozzles in one jet line. One reason is the jet rows have more distance from each other compared to the nozzles in one jet line. Also, the plate velocity vector is now oriented in a different direction with respect to the spreading water. Nozzle stagger configuration and jet line spacing are the two parameters that affect the longitudinal distribution of heat flux.

Figure 4.9 shows the heat flux distribution between two successive jet lines for some of the thermocouples (across the plate width) for the high temperature range. Figure 4.9(a), (b), and (c) show the heat flux distribution for jet line spacing of 25.4 cm, and for no stagger, half-staggered, and full-staggered configurations. Figure 4.9(d) shows the distribution for the half-staggered configuration, but for a jet line spacing of 50.8 cm. As it can be seen in this figure, the heat flux is more or less uniform between two successive jet lines in this temperature range. Only in the half-staggered case, with a jet line spacing of 25.4 cm (Figure 4.9(b)), there is an increase in the heat flux somewhere between the two jet lines, probably due to some kind of a stagnation line. Comparing parts (b) and (d) of this figure shows that increasing the jet line spacing may result in the dissipation of this heat flux spike. Also, notice that in all cases the heat flux between two jet lines is smaller than the one between two impinging jets in one jet line.

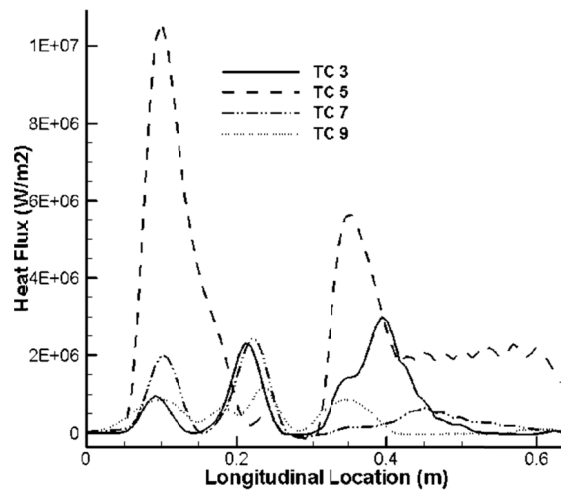
Figure 4.10 displays the same graphs but for the mid temperature range. As shown in these figures (in this temperature range and for a low nozzle spacing, cases-a, b, and c) the heat flux in the interaction zone between two jet lines rises to a value that is comparable to the value of heat flux in the interaction zone between two nozzles of a same jet line. This effect is, however, dissipated when the jet line spacing is increased, probably due to a weaker interaction fountain between two successive jet rows. Also notice that the heat flux values are higher to the right of the peak values in the impingement point. It means that the heat transfer in the downstream of the impingement line (where the plate motion and the water spreading directions are the same) is higher.

Figure 4.11 shows the same graphs for the low entry temperature range. Here, there is a considerable amount of heat flux happening in the region between two jets. The reason is that in this region, the water remains longer on the surface before evaporation and the pool of water is still cooling the plate when the thermocouple is between two impinging jets. Also the vapor layer

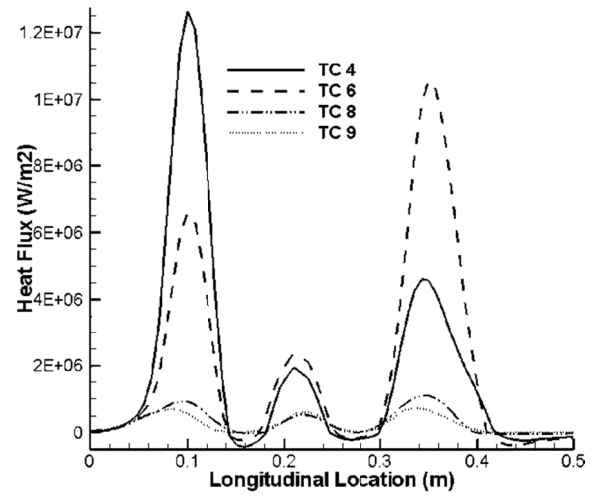
on the surface is not present or is very thin, so the vertical momentum of the water jet is not necessary in order to break it down in the low entry temperatures. Notice that in this case and for some of the thermocouples, the peak heat flux is not happening on the water impingement line, but in the parallel flow zone after the jet line. This is similar to the observation that for this range of entry temperature, the peak heat flux across the plate width is not under the impingement point.



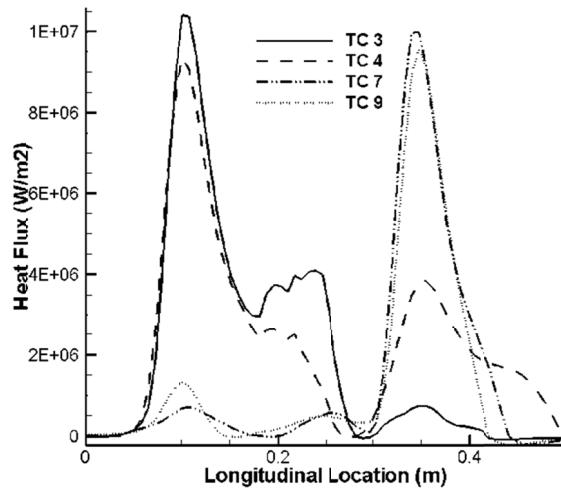
**Figure 4.9. Longitudinal Distribution of Heat Flux for the High Entry Temperature; (a) No Stagger, Jet Line Spacing: 25.4 cm; (b) Half Stagger, Jet Line Spacing: 25.4 cm; (c) Full Stagger, Jet Line Spacing: 25.4 cm; (d) Half Stagger, Jet Line Spacing: 50.8 cm.**



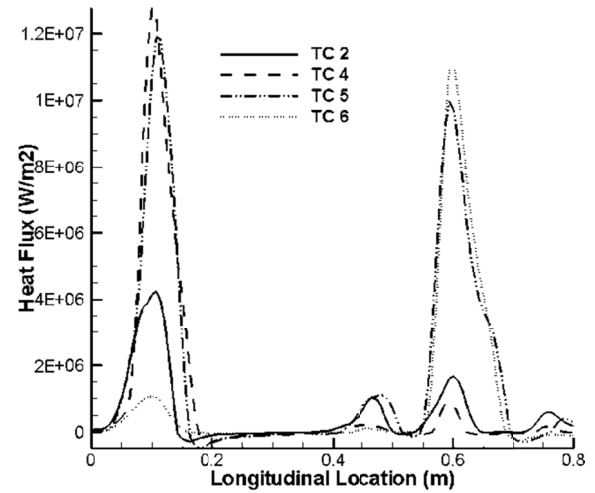
(a)



(b)

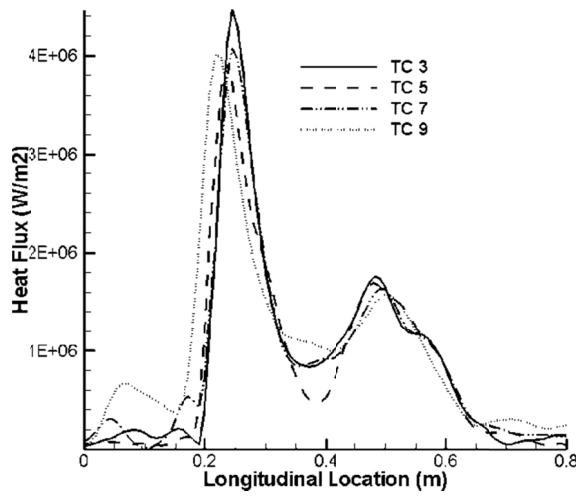


(c)

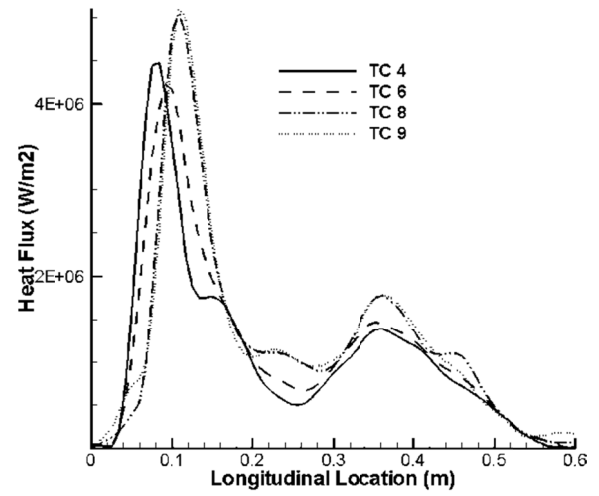


(d)

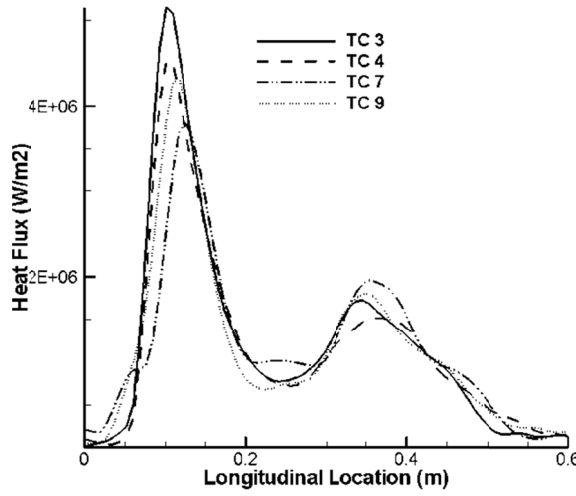
Figure 4.10. Longitudinal distribution of heat flux for the mid entry temperature; (a) No stagger, jet line spacing: 25.4 cm; (b) half stagger, jet line spacing: 25.4 cm; (c) full stagger, jet line spacing: 25.4 cm; (d) half stagger, jet line spacing: 50.8 cm



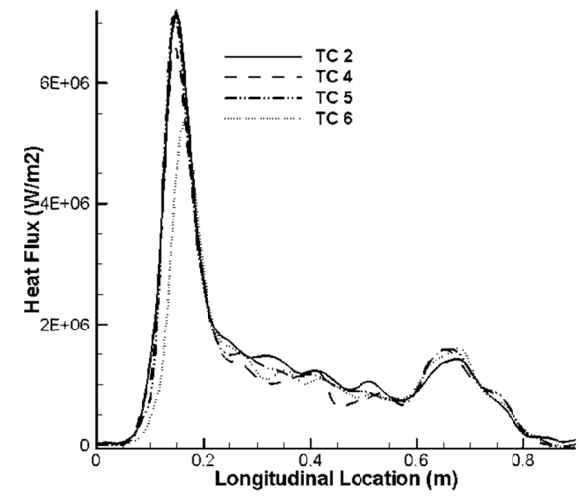
(a)



(b)



(c)



(d)

Figure 4.11. Longitudinal Distribution of Heat flux for the Low Entry Temperature; (a) No Stagger, Jet Line Spacing: 25.4 cm; (b) Half Stagger, Jet Line Spacing: 25.4 cm; (c) Full Stagger, Jet Line Spacing: 25.4 cm; (d) Half Stagger, Jet Line Spacing: 50.8 cm

#### 4.8. Effect of the Plate Speed

The ratio between the plate speed and jet line spacing determines the time it takes between two impinging jet lines to hit the plate, and hence the amount of temperature recovery, which in turn changes the surface temperature for the second jet line, and hence, affects the whole heat transfer process. Also, plate speed has an effect on the values of the heat flux versus

the surface temperature. Figure 4.12(a) and (b) display the value of heat flux under the impingement zone, each for 6 sets of experiments, with jet line spacings of 25.4 and 50.8 cm, respectively. The symbols on the graph show the four different combinations of plate speed and jet line spacing, and the connecting line pattern is representing the nozzle stagger configuration. Solid lines represent the cases with no stagger, while dashed and dotted lines stand for the half-staggered and full staggered configurations, respectively. As it can be observed in these figures, there is not a clear difference between different setups in the temperatures below 400 °C. However, in the temperatures higher than this there is a clear difference between six setups with a plate speed of 0.35 m/s (higher branch) and the ones with a plate velocity of 1 m/s (lower branch). In other words, in the high entry temperatures, the heat transfer rate is much lower when the plate is moving at a higher velocity. This can be related to the fact that at high plate speeds, the vapor layer that is formed on the surface, and works like an insulator, is more strongly attached to the plate, so the impinging jet has a bigger challenge in breaking down the vapor layer and getting in touch with plate. As a result, a more stable film boiling heat transfer will be present, which causes lower heat transfer rates. This temperature range in which the plate speed affects the heat transfer rate is in accordance with the experiments done by Gradeck et al. [84]. Obviously, in order to better understand this behavior, more experiments are needed with a wider range and variety of plate speeds. Notice that this effect is only limited to the impingement zone. Figure 4.12 (c) and (d) display the variation of the surface heat flux between two nozzles versus the surface temperature. Here, however, there is not any clear correlation between the plate speed and the heat transfer rate.



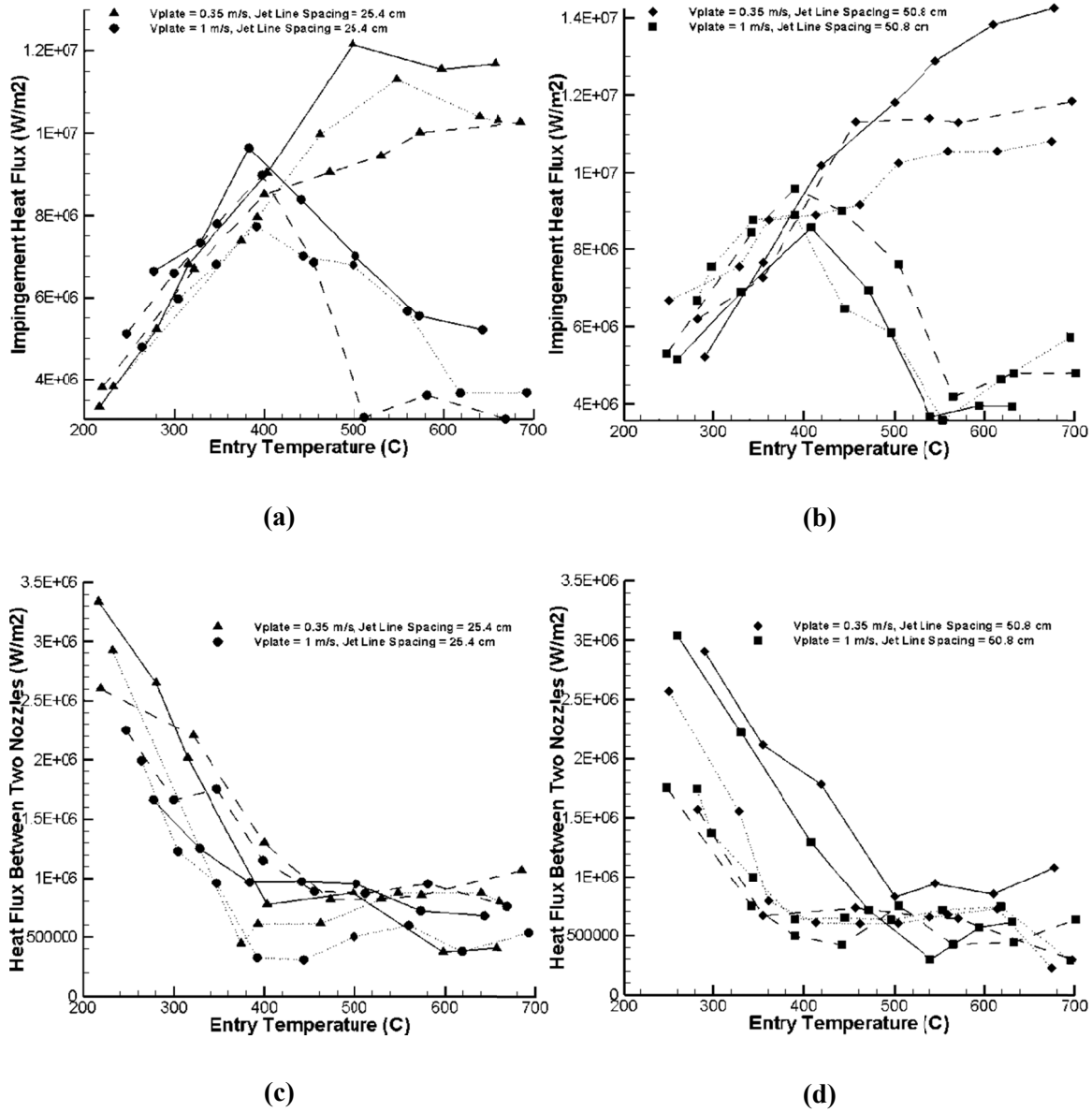


Figure 4.12. (a) and (b) Impingement Heat Transfer Versus Entry Temperature; Jet Line Spacings of 25.4 and 50.8 cm, Respectively; (c) and (d) Jet Interaction Heat Transfer Versus Entry Temperature; Jet Line Spacings of 25.4 and 50.8 cm, Respectively; Solid Lines: No Stagger; Dashed Lines: Half Stagger; Dotted Lines: Full Staggered

## 4.9. Conclusion

Boiling heat transfer on a hot plate cooled by multiple water jets and multiple jet lines in an industrial scale experimental setup is studied. Nozzle configuration, plate speed, and jet line spacing are varied in the experiments. Thermocouple readings are analyzed using an inverse heat conduction procedure. Surface temperature is the most significant factor affecting the heat

transfer behaviour on the plate. At high temperatures, the plate surface is covered with a condensed vapour layer that acts as an insulation blanket, and requires an impinging jet with high momentum to break it down. This translates into a stable film boiling regime, with reduced values of heat fluxes. Heat transfer rate increases at medium temperatures where the vapour thickness is smaller. For low entry temperatures in a multi jet line experiment, the heat transfer is lowered again. In the low temperature region, the maximum heat flux does not occur at the impingement point, but at a location between the jets, due to formation of an interaction water fountain. The most important effect of nozzle stagger is the uniformity of heat transfer across the width of the plate. The most uniform distribution is happening in the case of fully staggered configuration. The heat transfer between two successive rows of jet lines is affected by the jet line distance. In higher jet line distances, the interaction effects become less significant, and a more uniform distribution is observed. However, when the jet lines are closer, the effect of interaction is more pronounced. In medium and high temperatures, there is a spike in the heat flux values in the interaction zone. In the low temperature range, because the water does not evaporate quickly, a pool of water forms on the plate surface, resulting in a flattened heat flux versus time profile. The plate speed affects the heat transfer rate under the impingement point for the higher surface temperatures. In the high entry temperatures, the impingement heat transfer rate is lower when the plate is moving at a higher velocity. The plate speed however does not significantly change the heat transfer behavior in the other parts of the plate surface.

## 5. Structural Effects of the Thermal Field during Water Jet Cooling

In this chapter, we investigate the effect of thermal field during the water impingement cooling of steel strips on the structure of the strips. A finite element based approach is presented that uses the heat flux histories during the controlled cooling of steel strips (as discussed in the previous chapter) as the input, and calculates the steel phase composition in each time step, and simulates the structural behavior, including the stress and deflection values. The DQSK steel is used in all the studies in this chapter, for both the stationary and moving plates. For the validation cases, the steel properties are chosen based on the test case in the literature.

It is important to notice that there are three different types of physics involved in this problem. These are the thermal, microstructural, and structural fields, and all three of them are mutually coupled. However, as Figure 5.1 shows, the levels of these couplings are different [129]. The thermal field strongly affects the other two physics as described in details later in this chapter. Microstructure field also strongly affects the mechanical field through changing the mechanical properties, as well as change in the volume. This also is described in details later in this chapter. Changing the phase composition also changes the thermal properties, such as the heat conduction coefficient, and thermal capacity. However, this last effect is relatively less significant, and is handled through the temperature dependency of the thermal properties. Finally, the effect of the mechanical field on the thermal and microstructural fields is very weak, and is often neglected. In this study, we model the strong and medium couplings, but neglect the weak ones.

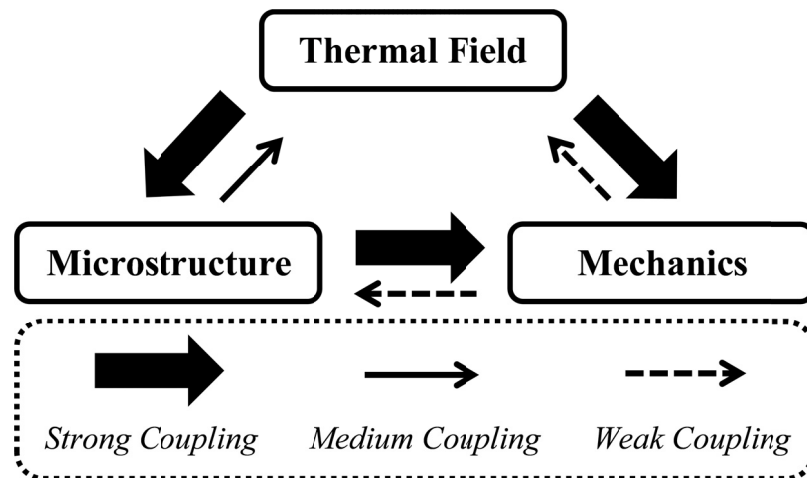


Figure 5.1. Different Kinds of Coupling between the Three Physical Fields in Cooling of Steel Strips

## 5.1. Phase Transformation Modeling

There are three main approaches to find the state of phase transformation during the cooling:

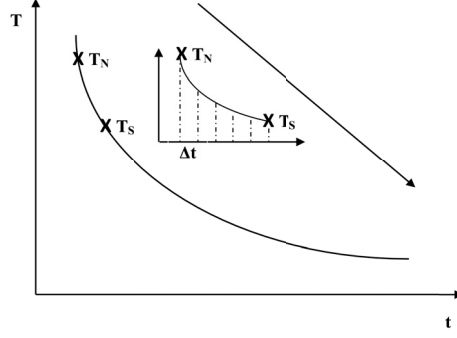
1. Using the laws of thermodynamics, kinetics, and diffusion.
2. Using the continuous cooling transformation (CCT) diagram:

$$X = X_0 \left\{ 1 - \exp \left[ b \left( \frac{T_s - T}{T_s - T_f} \right)^n \right] \right\} \quad (5.1)$$

where  $X_0$  is described as the volumetric rate of intended phase at transformation termination,  $T_s$  and  $T_f$  are the transformation start and finish temperatures, respectively, and  $n$  and  $b$  are constants. Because the cooling rate at each point is different, we will have different values for  $T_f$ ,  $T_s$ ,  $n$  and  $b$ . This approach was developed by Boyadjiev, et al. [130].

3. Using the isothermal temperature-time transformation (TTT) diagrams: The initial time should be defined as the time that the transformation begins under continuous cooling conditions.

The first category of the models are the most accurate ones, but have not been used extensively in the literature due to the lack of information about the exact characteristics of thermal field history and the model calibration coefficients. Previous research (in UBC group) has provided the required information to use in the above models [131,132]. The algorithm that is used here is based on the work of Liu, et al. [132], and falls in the first family of the models. In this algorithm, the static recrystallization kinetics is simulated by the Johnson-Mehl-Avrami-Kolmogorov (JMAK) theory. The transformation start temperature was also found following the method suggested in the work of Militzer, et al. [131].



**Figure 5.2. Schematic Representation of Cooling on the Run-out Table (Temperature vs. Time)**

The first step in this algorithm is finding the transformation start temperature,  $T_S$ . In Figure 5.2,  $T_N$  is the temperature where the nucleation starts, and is known and equal to 797 °C. Starting from this point, we divide the cooling path to small segments, each with the length of  $\Delta t$ . Then, the following summation value is calculated along the cooling path, starting from  $T_N$ :

$$R_a = \sqrt{\sum_{i=1}^n 2D_c(T_i) \frac{c_{eq}^{\gamma}(T_i) - c_0}{c_{eq}^{\gamma}(T_i) - c_{eq}^{\alpha}(T_i)} \Delta t} \quad (5.2)$$

where

$$D_c(T) = 453000 \left( 1 + \frac{8339.9c_0(1-c_0)}{T+273} \right) \exp \left[ - (17767 - 26436c_0) \left( \frac{1}{T+273} - 0.00022 \right) \right] \quad (5.3)$$

and  $c_0$  is the atomic fraction,  $c_{eq}^{\gamma}$  and  $c_{eq}^{\alpha}$  are the equilibrium concentrations of carbon in austenite and ferrite, respectively, which are known as a function of temperature for each steel, and  $T_i$  is the temperature at the  $i$ th time step in degrees Celsius. Transformation starts, i.e.  $T_i = T_S$ , when the value of  $R_a$  becomes larger than a critical value,  $R^*$ , where:

$$R^* = \frac{0.25c_0}{c_{eq}^{\gamma}(T_i) - c_0} d_{eff}^{\gamma} \quad (5.4)$$

where  $d_{eff}^{\gamma}$  is the effective austenite grain size, and is again known for each steel composition.

The next step would be to calculate the fraction of austenite that is transformed to ferrite,  $f_a$ . These calculations start at  $T = T_S$ . In each time step, we define the equilibrium transformed fraction as:

$$f_{\alpha}^{eq} = \frac{c_{eq}^{\gamma}(T_i) - c_0}{c_{eq}^{\gamma}(T_i) - c_{eq}^{\alpha}(T_i)} \quad (5.5)$$

and the real transformed fraction can be obtained as:

$$f_{\alpha} = F_{\alpha} \cdot f_{\alpha}^{eq} \quad (5.6)$$

where at  $T = T_s$  the value of  $F_{\alpha}$  is 0.001, and at each next time step, it is updated through:

$$F_{\alpha}^{new} = F_{\alpha}^{old} + \frac{dF_{\alpha}}{dt} \Delta t \quad (5.7)$$

where

$$\frac{dF_{\alpha}}{dt} = b_{\alpha}(T_i)(1 - F_{\alpha}) \quad (5.8)$$

and

$$b_{\alpha}(T_i) = \exp \left\{ 4.4 - 0.001(860 - T_i) - 11.24 \frac{c_0}{1 - f_{\alpha}} \right\} \quad (5.9)$$

For a more detailed discussion of this model, as well as the physical significance of various terms, refer to [131,132].

## 5.2. Thermal Stress Modeling

If the temperature in a homogenous body changes by  $\Delta T$ , the total strain in the body is consisted of two parts, the free thermal expansion, plus the strains related to the stress state in the body, and can be expressed by the following equations:

$$\left\{ \begin{array}{l} \varepsilon_x = \alpha \Delta T + \left[ \sigma_x - \nu (\sigma_y + \sigma_z) \right] / E \\ \varepsilon_y = \alpha \Delta T + \left[ \sigma_y - \nu (\sigma_x + \sigma_z) \right] / E \\ \varepsilon_z = \alpha \Delta T + \left[ \sigma_z - \nu (\sigma_x + \sigma_y) \right] / E \\ \varepsilon_{xy} = \sigma_{xy} / G \\ \varepsilon_{yz} = \sigma_{yz} / G \\ \varepsilon_{zx} = \sigma_{zx} / G \end{array} \right. \quad (5.10)$$

where  $\varepsilon$  is the strain,  $\sigma$  is the stress,  $\nu$  is the Poisson's ratio,  $E$  is the Young's modulus, and  $G$  is the shear modulus. Solving the above equations for stress yields:

$$\begin{cases} \sigma_x = (\lambda + 2\mu)\varepsilon_x + \lambda(\varepsilon_y + \varepsilon_z) - \beta\Delta T \\ \sigma_y = (\lambda + 2\mu)\varepsilon_y + \lambda(\varepsilon_x + \varepsilon_z) - \beta\Delta T \\ \sigma_z = (\lambda + 2\mu)\varepsilon_z + \lambda(\varepsilon_x + \varepsilon_y) - \beta\Delta T \\ \sigma_{xy} = \mu\varepsilon_{xy}; \sigma_{yz} = \mu\varepsilon_{yz}; \sigma_{xz} = \mu\varepsilon_{xz} \end{cases} \quad (5.11)$$

where  $\lambda$ ,  $\mu$ , and  $\beta$  are defined as:

$$\begin{cases} \lambda = \nu E / (1 + \nu)(1 - 2\nu) \\ \mu = E / 2(1 + \nu) \\ \beta = E\alpha / (1 - 2\nu) \end{cases} \quad (5.12)$$

The equations of equilibrium for temperature change in the body are:

$$\begin{cases} \frac{\partial \sigma_x}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z} + X = 0 \\ \frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + \frac{\partial \sigma_{yz}}{\partial z} + Y = 0 \\ \frac{\partial \sigma_{xz}}{\partial x} + \frac{\partial \sigma_{yz}}{\partial y} + \frac{\partial \sigma_z}{\partial z} + Z = 0 \end{cases} \quad (5.13)$$

where  $X$ ,  $Y$ , and  $Z$  are the body forces per unit volume. The boundary conditions are given by:

$$\begin{cases} \sigma_x l + \sigma_{xy} m + \sigma_{xz} n = X_s \\ \sigma_{xy} l + \sigma_y m + \sigma_{yz} n = Y_s \\ \sigma_{xz} l + \sigma_{yz} m + \sigma_z n = Z_s \end{cases} \quad (5.14)$$

where  $l$ ,  $m$ , and  $n$  are the direction cosines of the outward drawn normal and  $X_s$ ,  $Y_s$ , and  $Z_s$  are the surface forces per unit area.

In cases where the boundary conditions, temperature distribution, geometry, or material properties are more complicated, using analytical methods to solve for the thermo-mechanical properties is impossible. Using numerical methods, such as finite elements is very common to study these problems.

While in the above equations, we assumed that the change in the temperature affects the strain values only through the thermal expansion coefficients, however in these controlled cooling applications, there are normally two main sources for the strain in the steel. One is the thermal strain that is related to the temperature gradient and the thermal expansion coefficient of the mixture of phases (obtained using the rule of mixtures). The other one is the volumetric

strain associated with the phase transformation, i.e. the change in the volume of the same amount of steel during austenite to ferrite transformation.

In this research, we combine the effects of these sources by creating a total coefficient of expansion:

$$\alpha_{total} = \alpha_{thermal} + \alpha_{phase-change} \quad (5.15)$$

where  $\alpha_{total}$  is the total expansion coefficient,  $\alpha_{thermal}$  is the weighted average of the thermal expansion coefficients of the phases in each location in the steel sample, and  $\alpha_{phase-change}$  is the phase change expansion coefficient. This latter component can be calculated in each time step  $i$  as:

$$\alpha_{phase-change} = \left[ \left( \frac{V_i - V_{i-1}}{V_{i-1}} \right) / 3 \right] / \Delta T \quad (5.16)$$

where  $V_i$  and  $V_{i-1}$  are the mixture specific volumes in the current and the last steps, respectively. In each step, the mixture specific volume can be obtained using the rule of mixtures:

$$V_i = V_\alpha \cdot X_i + V_\gamma \cdot (1 - X_i) \quad (5.17)$$

and  $V_\alpha$  and  $V_\gamma$  are the specific volumes of ferrite and austenite, respectively. These values are related to the lattice parameters of ferrite and austenite ( $a_\alpha$  and  $a_\gamma$ , respectively), as:

$$V_\alpha = (1/2)a_\alpha^3 \text{ and } V_\gamma = (1/4)a_\gamma^3 \quad (5.18)$$

The lattice parameters vary with temperature, and in the case of austenite also with the carbon fraction value [133].

### 5.3. Computational Modeling

A commercial finite element software is used to model the coupled thermal and structural field. Shell elements are used to discretize the domain. On the top surface of the plate, the heat fluxes are applied as the thermal boundary condition. After the thermal and microstructure simulations are performed, results are written to a file, which is subsequently read by the structural simulation. Since the effect of structural field on the thermal and microstructural fields is weak (see Figure 5.1), there is no need to feed the structural results back into the thermal/microstructural models. In all the following validation cases, as well as the new simulations, grid and time step independency has been investigated, i.e the mesh is refined until changes in the results were less than 5%.



## 5.4. Validation Cases

In order to validate the modeling of thermal stresses on the run-out table, two of the similar cases from literature are simulated here, and the results are compared with the ones available in the literature.

### 5.4.1. Validation Case I

This test case is based on the work of Yoshida [86]. As was mentioned in Chapter 1, he uses a convection correlation to model the thermal boundary condition on the top surface of the moving plate, a simplified time-temperature-transformation (TTT) for the phase transformation model, and a two-dimensional finite difference method to simulate the thermal and structural fields in the plate. Figure 5.3 shows a schematic of the problem, and the simple thermal boundary conditions. Residual stresses are then plotted versus the distance from the edge of plate at different locations and under different cooling strategies, e.g. uniform, early stage, and later stage cooling.

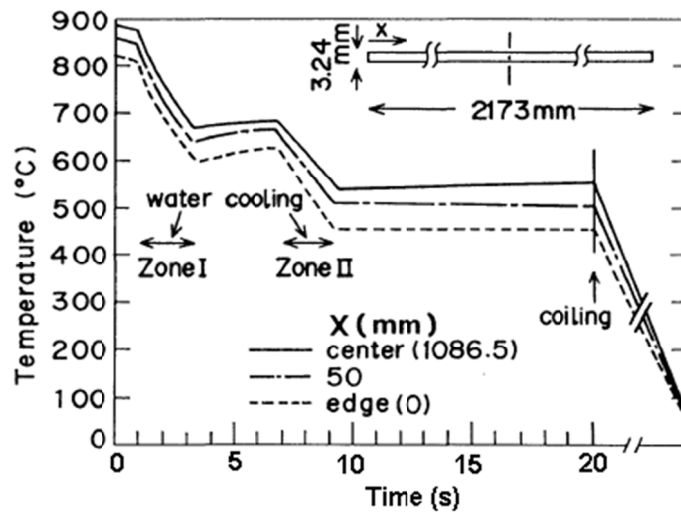


Figure 5.3. Problem Dimensions and Temperature History in Validation Case I by Yoshida [86].

Figure 5.4 shows the results of Yoshida's study (strain gauge readings as well as the numerical simulation results), and the results obtained by the present study. A very good agreement is observed between the results.

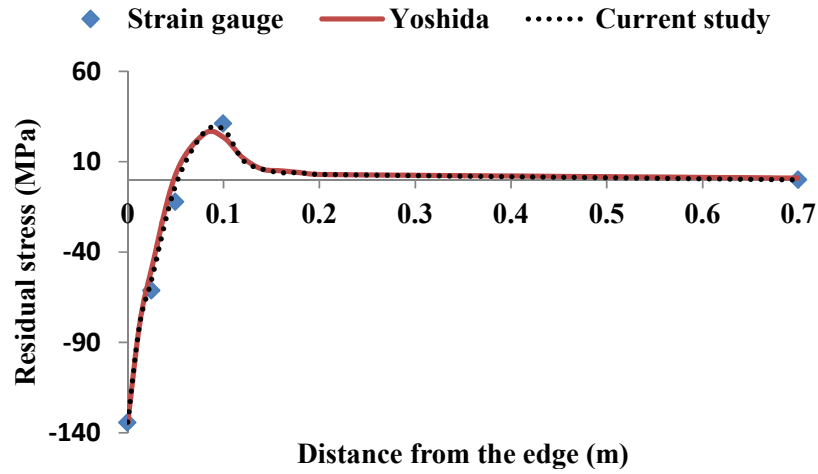


Figure 5.4. Validation Case I; Comparison between the Current Results and the Experimental and Numerical Results in the Literature [86].

#### 5.4.2. Validation Case II

The work done by Zhou, et al. [90] is the basis of the second validation case. They apply a simplified temperature-time curve as the cooling boundary condition in their model. Figure 5.5 shows the geometry, and Figure 5.6 displays the transverse and longitudinal thermal boundary conditions. Again, a TTT diagram is used for finding the phase transformation during the process. ABAQUS finite element code is used to simulate the thermal and structural fields, and the residual stresses and displacements are reported for different cooling strategies, and at different locations.

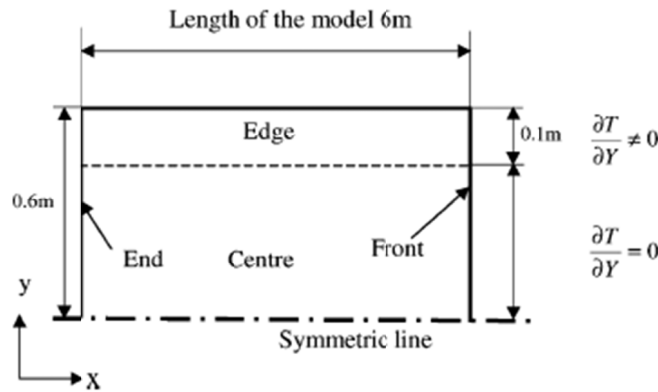


Figure 5.5. Geometry of Validation Case II [89]

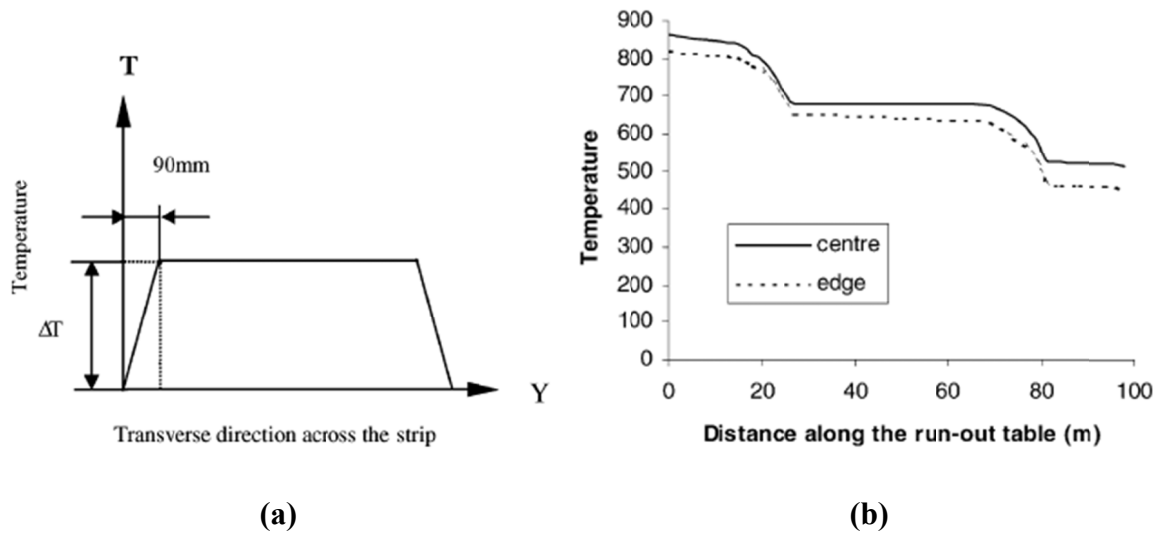


Figure 5.6. Temperature Distribution in Validation Case II in (a) Transverse and (b) Longitudinal Directions [89]

Figure 5.7 shows Zhou's results, as well as the results obtained by this study. The maximum difference is less than 7%, which validates our solution method.

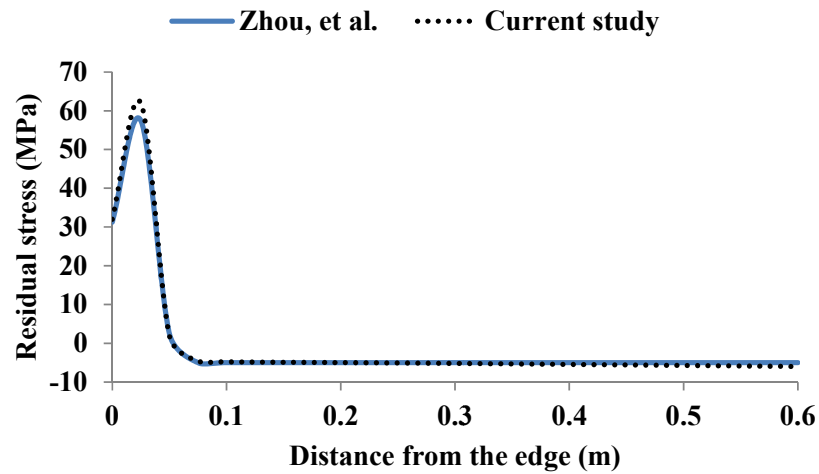
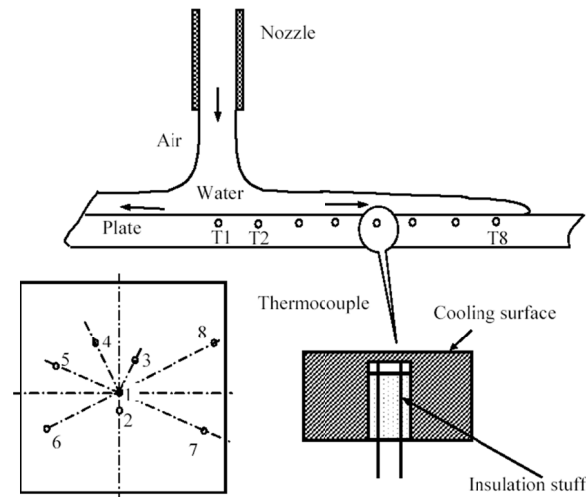


Figure 5.7. Validation Case II; Comparison between the Current Results and the Numerical Results in the Literature [90]

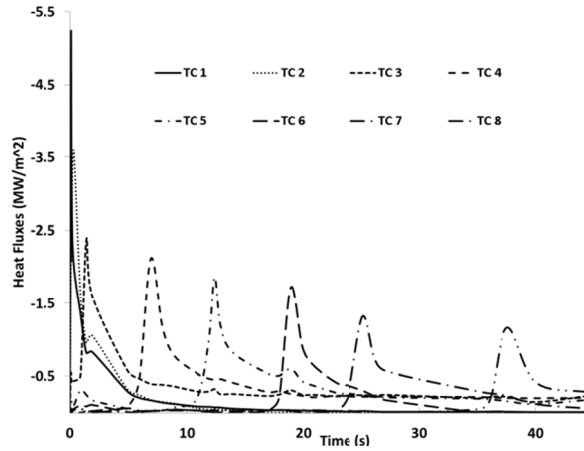
## 5.5. Stationary Plate

In this section, we have used the heat fluxes obtained by Xu and Gadala [76] to study the structural field in a hot stationary steel plate cooled by a circular impinging water jet. This, to our knowledge, has not been studied in the literature.

Figure 5.8 shows the steel plate under the impinging jet, and the eight thermocouples that are implemented inside of it at different radii. Since the plate is stationary and the water jet is impinging on its center, we are assuming that the temperature distribution will be symmetric. The plate surface is 240 by 240 mm, and is 6 mm thick. The thermocouples are 16 mm apart, and the tip of the thermocouple is 1 mm under the top surface of the plate. The plate is originally heated up to 900 °C, and then is cooled down by the impinging water jet. Thermocouple readings are fed into the CRPSO inverse solver that was developed in Chapter 3, which produces the heat fluxes for each thermocouple location, i.e. different radial locations. Figure 5.9 shows the readings of the first to the eighth thermocouple, in order from left to right.



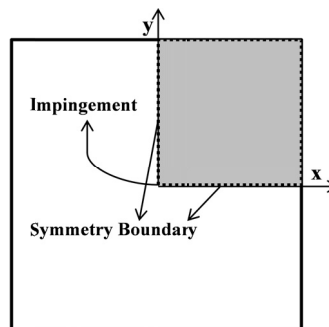
**Figure 5.8: Schematic Arrangement of Thermocouples under Impinging Jet**



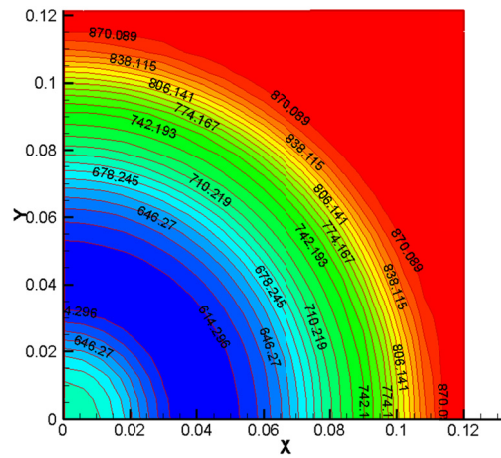
**Figure 5.9: Top Surface Heat Flux History Obtained By Inverse Analysis of Temperature Readings at Thermocouple Locations.**

Based on the symmetry assumption, only one quarter of the plate is modeled in the structural analysis. Figure 5.10 shows the whole geometry and the modeled area (shaded). The jet is impinging on the center and the water moves radially over plate. In order to discretize the domain, shell elements are used.

Figure 5.11 shows the temperature distribution on the top surface of the plate a few seconds after the start of the cooling process. Figure 5.12(a) shows the contours of the vertical, out of the plane displacement (in mm) at the end of the cooling process when the temperature over the plate reaches a nearly uniform value of around 130 °C. Figure 5.12(b) shows the contours of von Mises stress at the same time. Figure 5.13 shows the maximum out of plane deflection vs. the strip thickness at the end of the cooling process. As expected, in thicker strips the difference in temperature across the plate increases, which results in a higher deformation.



**Figure 5.10: Plate Geometry; Only the Shaded Area is Modeled.**

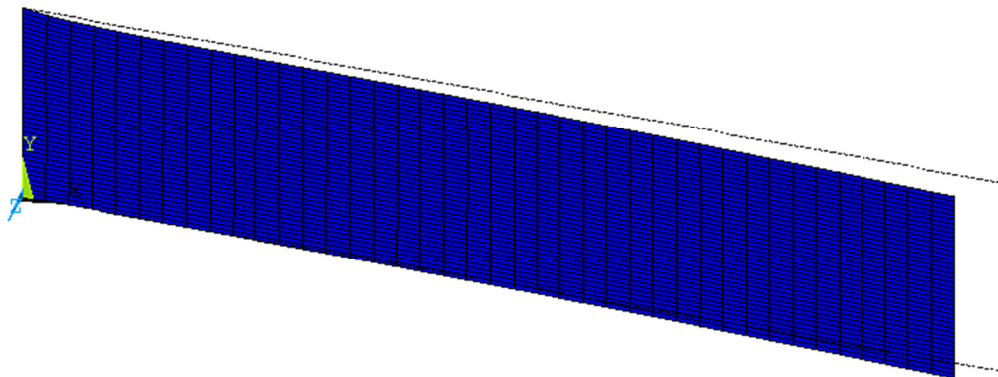


## 5.6. Moving Plate

In this section, we use the results of Chapter 4 (heat fluxes vs. time for moving plates) to model the structural field during the cooling process. The parameters that were studied in the previous chapter will also be studied here. These parameters are the plate moving speed, the jet configuration (inline, half-staggered, and full-staggered), and jet line spacing.

There are several ways that one can study the history of the structural field. We can study each pass in isolation, starting with a flat plate with no residual stress in it. While this may be interesting from an academic point of view, it is not a good representation of what happens in reality. The other approach would be to start with a hot flat plate with no residual stress, just out of the furnace, and then apply the boundary conditions in time, until the temperature reaches the coiling temperature, or any lower temperature that we might be interested in. In this research, both of these strategies are used. First, we apply three isolated passes of the heat flux, each in one of the three temperature zones discussed in the previous chapter (high, mid, and low temperature). Then, we start from the hot plate coming out of the furnace, and apply the heat fluxes until it reaches temperatures of around 500, 300, and 100 °C. Note that in all these simulations, instead of a moving plate, the heat fluxes move over the plate with respect to time.

Figure 5.14 shows an oblique view of the deformed moving strip, as well as the edge of the undeformed one.



**Figure 5.14. The Deformed and the Edge of the Undeformed Strip in a Moving Case**

### **5.6.1. Single Passes**

Single pass simulations were performed for three temperature zones (high, mid, and low temperatures). Figure 5.15 - Figure 5.17 show the results of these simulations for different setups. In each case, we are assuming that the plate is entering the cooling station with uniform temperature distribution and no initial stress or deflection.



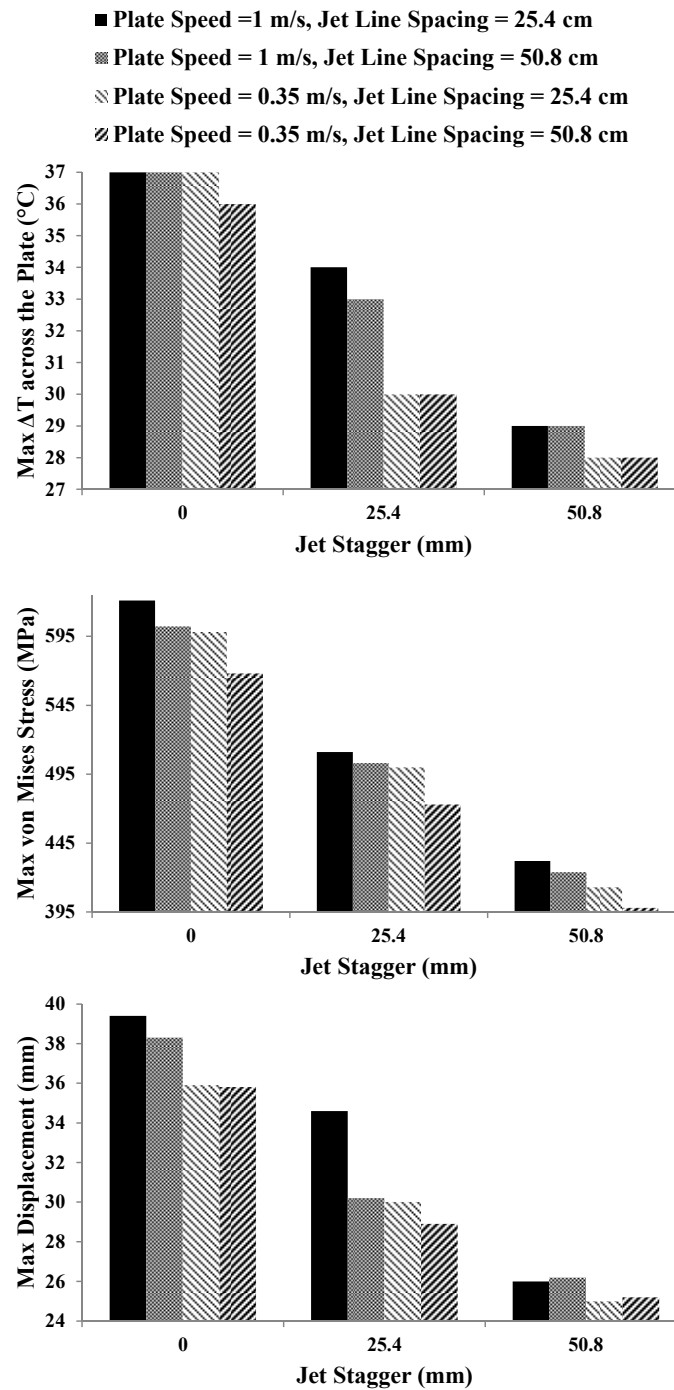


Figure 5.15. Single Pass Results for High Temperature Range Cooling

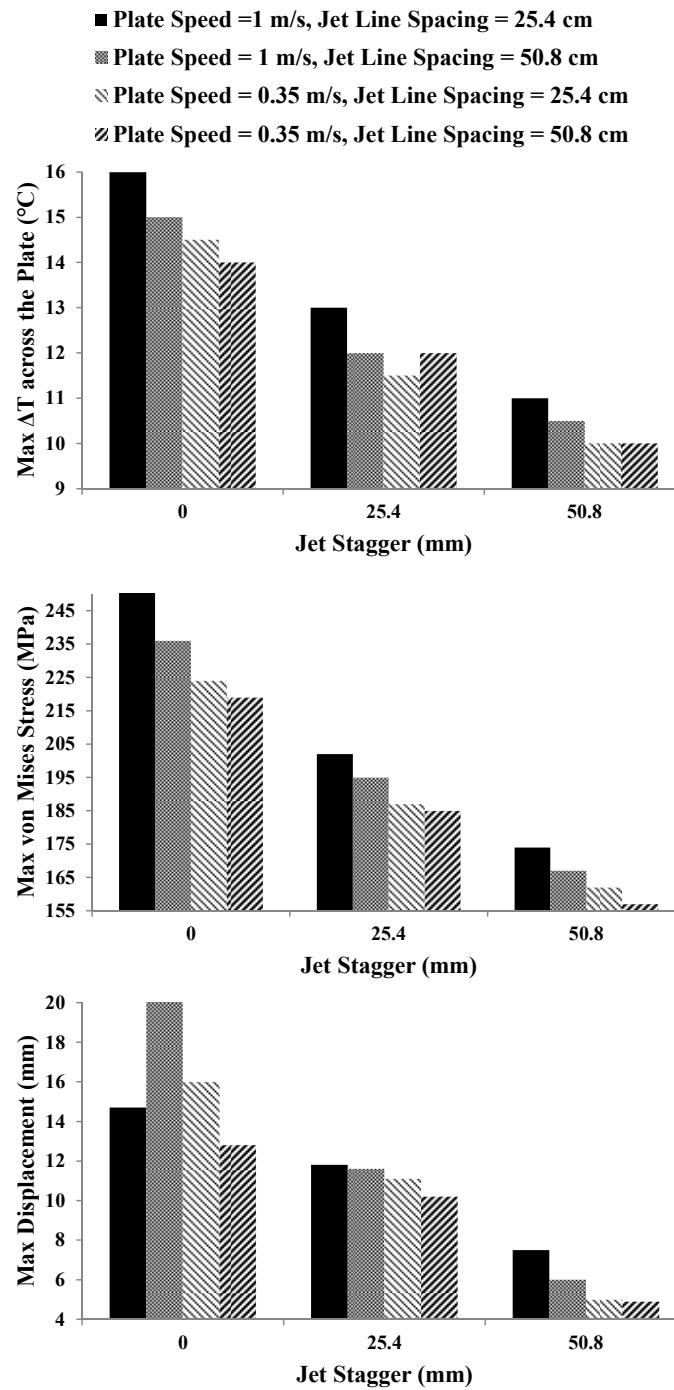


Figure 5.16. Single Pass Results for Mid Temperature Range Cooling.

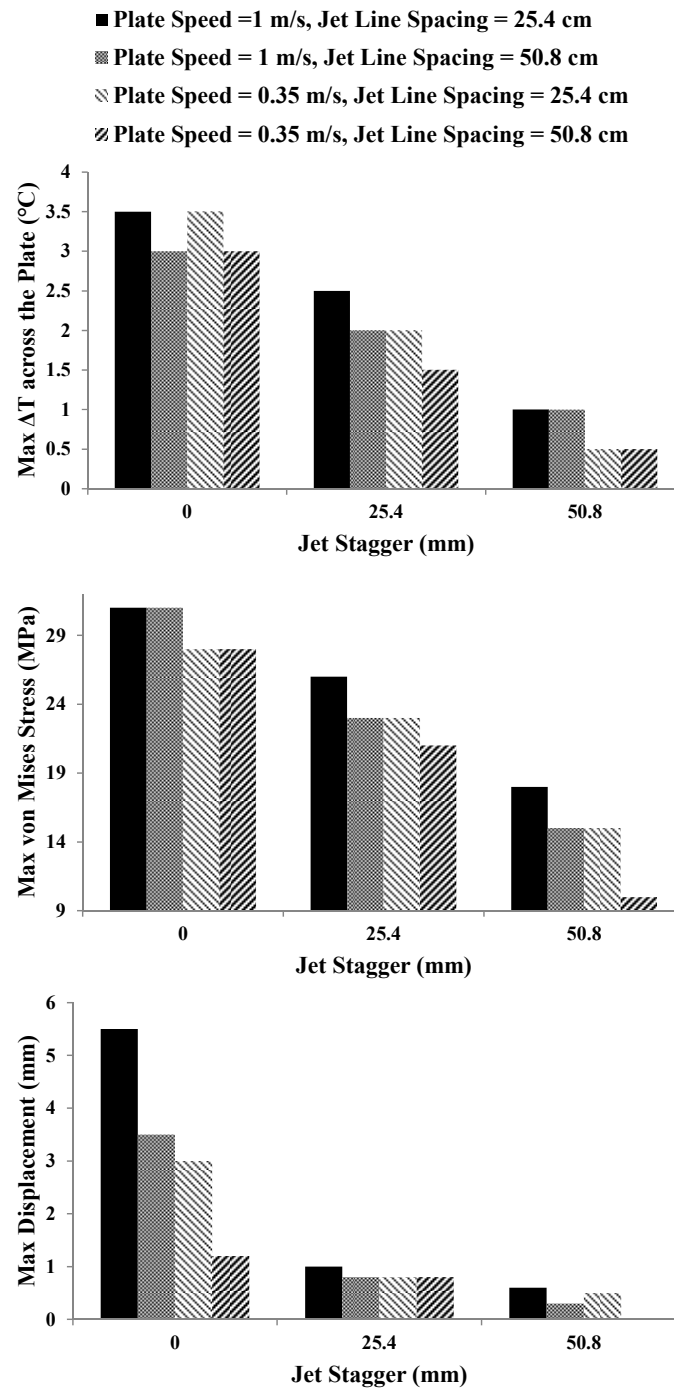


Figure 5.17. Single Pass Results for Low Temperature Range Cooling

As seen in the above figures, higher maximum temperature difference across the plate width results in a higher maximum von Mises stress, and maximum displacement. In each of these figures, values are typically larger for the no-stagger (inline) cases (S1, S2, S7, and S8), followed by the half-staggered ones (S3, S4, S9, and S10). The fully-staggered setups (S5, S6, S11, and S12) have the lowest values of the lateral temperature variations, as well as the lowest values of the von Mises stress and displacement. The most significant factor in controlling the stress and flatness in the strips is the jet configuration, and the most desirable configuration is the one that promotes a more uniform distribution of temperature across the plate width.

Among the cases with the same nozzle configuration, plate speed is the next important factor in determining the values of stress and deformation. Plates traveling at slower speeds are cooled down more uniformly, and will have less stress and deflection in them. Finally, longer jet spacing gives the opportunity of a more uniform cooling and causes less deflection in the plate. However, the effect of jet line spacing and plate speed is much less significant than the effect of jet stagger.

As expected, the cooling in the low temperature zone has the lowest effect on the plate deformation. It should be noted that in this temperature range, there is practically no phase transformation and already all of the steel is transformed to ferrite.

### **5.6.2. Multiple Passes**

For the results in the previous section, we were assuming an initial uniform temperature distribution, and no initial stress or deflection in the plate. In Figure 5.18 - Figure 5.20, we will study the effect of the initial non-uniform temperature distribution and stress and deflection in the plate. In order to do this, we start with a uniform hot plate with no stress and deflection, coming out of furnace at about 750 °C, and then through multiple passes through the cooling station, we lower the temperature to 500, 300, and 100 °C, and display the results in Figure 5.18, Figure 5.19, and Figure 5.20, respectively.

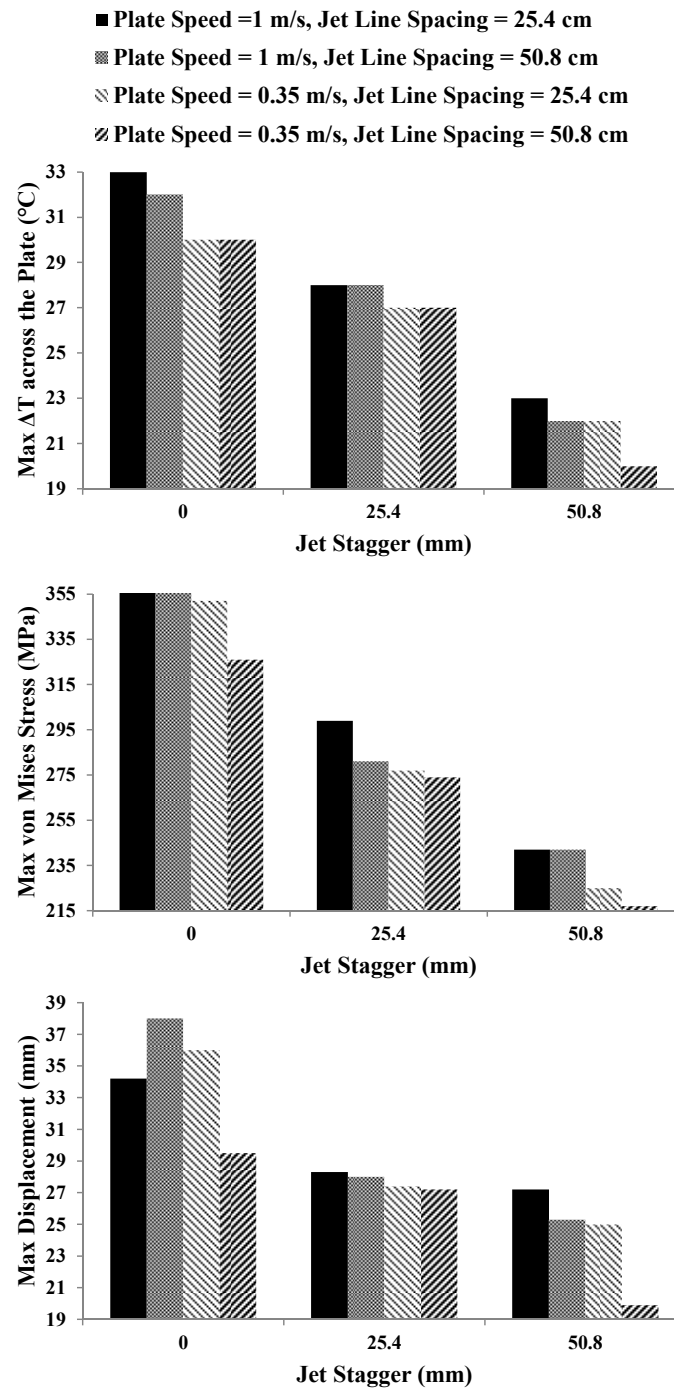


Figure 5.18. Results for a Plate Cooled Down to 500 °C

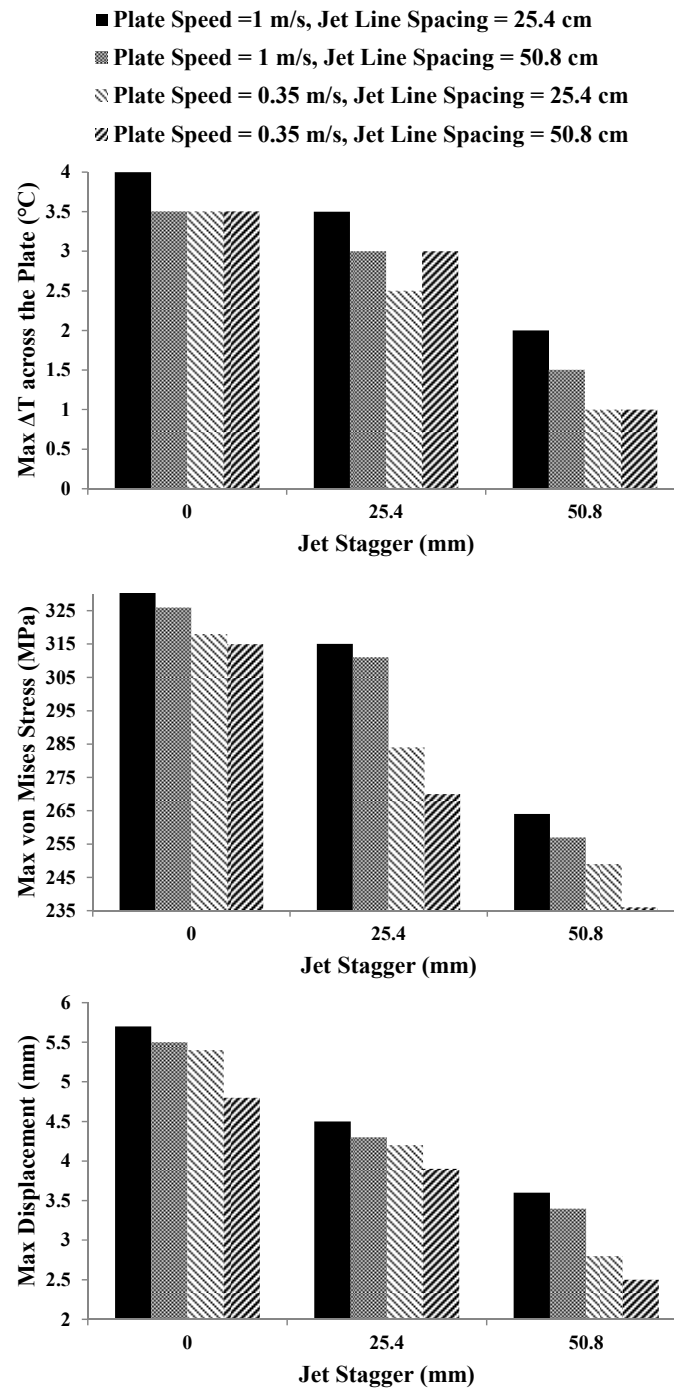


Figure 5.19. Results for a Plate Cooled Down to 300 °C

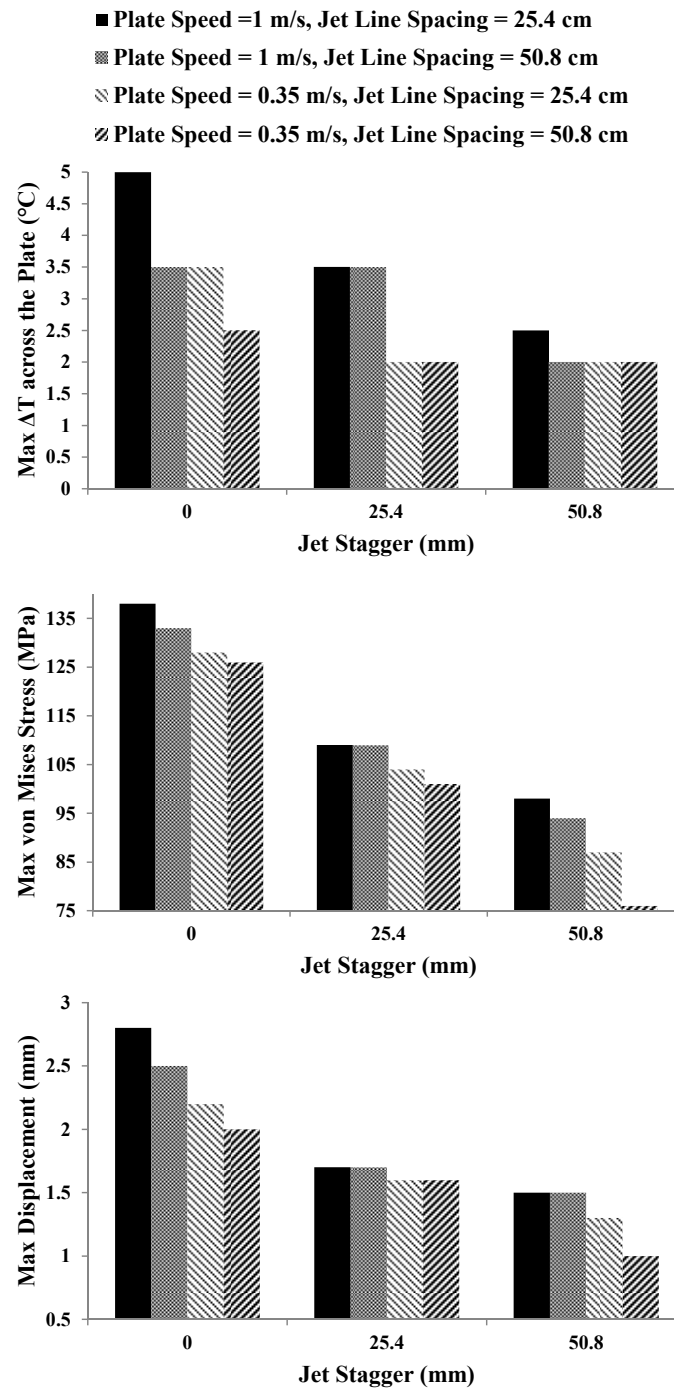


Figure 5.20. Results for a Plate Cooled Down to 100 °C

The trend for the effect of jet stagger, jet line spacing, and plate speed are similar to those observed in section 5.6.1. It is also observed that lowering the final coiling temperature will result in lowering the stress and deflection. This is due to the fact that thermal contraction of ferrite will cancel out some of the volumetric expansion due to the austenite to ferrite transformation.

### 5.6.3. Effect of Plate Thickness

Figure 5.21 shows the variation of the maximum out of plane displacement for the moving plate versus the strip thickness, for case S1, when continuously cooled down to around 300 °C. The trends for all of the moving plate cases are similar, and resemble the trend for the stationary plate, as depicted in Figure 5.13.

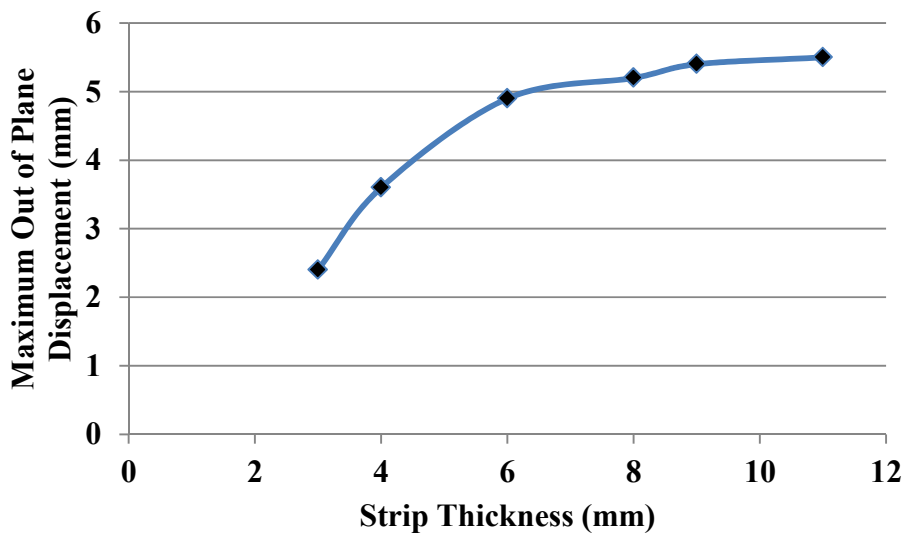


Figure 5.21. Maximum Out of Plane Displacement vs. Strip Thickness for the Moving Plate

## 5.7. Conclusion

In this chapter, the structural behavior of a hot steel strip when subjected to cooling by circular water jets was studied, by combining the heat transfer, phase transformation, and structural models. The heat transfer model uses the heat fluxes obtained by an inverse analysis of the thermocouple readings, and the phase change model is a detailed simulation using the



Johnson-Mehl-Avrami-Kolmogorov (JMAK) theory. The model is validated using sample cases in the literature.

The cooling of both the stationary and the moving plates were studied. It was found that the maximum temperature difference across the plate width is directly affecting the maximum von Mises stress and displacement. Also, these values are typically larger for the no-stagger (inline) cases, followed by the half-staggered ones. The fully-staggered setups have the lowest values of the lateral temperature variations, as well as the lowest values of the von Mises stress and displacement. The most significant factor in controlling the stress and flatness in the strips is the jet configuration, and the most desirable configuration is the one that promotes a more uniform distribution of temperature across the plate width.

Plate speed is the next important factor in determining the values of stress and deformation. Plates traveling at slower speeds are cooled down more uniformly, and will have less stress and deflection in them. Finally, longer jet spacing gives the opportunity of a more uniform cooling and causes less deflection in the plate. It was also observed that increasing the plate thickness, increases the maximum deflection, due to a larger temperature gradient through the thickness.

## **6. Conclusion and Future Work**

### **6.1. Conclusion**

The controlled cooling of steel strips on run-out table is a crucial procedure that can result in obtaining desired mechanical properties. Despite being used in steel mills for decades, the process is still mainly based on trial-and-error. This is due to the complicated nature of the events that happen, and the highly coupled nature of the thermal, fluid flow, microstructure, and mechanical fields. Thus, there is still a need to study this process in more details.

The present study is a part of a comprehensive research project undertaken in UBC [1,8,9,27-32]. Here, we first established a reliable inverse analysis tool that can be used to obtain the boundary heat fluxes from the temperature history of thermocouples implemented inside the plate. Then, these heat fluxes were studied to understand the heat transfer behavior of boiling water impinged on a moving hot plate through multiple circular jets. Finally, the heat fluxes were coupled with the microstructure and mechanical fields to study the thermal stresses and deflection or warping in the steel plates. Recommendations were made to the industry about the ways to improve the cooling performance, and reduce the adverse structural effects. The following subsections summarize the findings and contributions of this research.

#### **6.1.1. 3D Finite Element Program for the Inverse Heat Conduction Analysis**

A gradient-based algorithm for the inverse heat conduction analysis is developed. In this method, the least-square method, future time step technique, zeroth order regularization, and sequential function specification are used. The program can account for the heat generation during the phase transformation as a source term in the heat transfer equation. The results showed that the method is highly sensitive to the measurement errors, and becomes unstable in small time steps, thus limiting its application in real run-out table studies.

As a remedy for the shortcomings related to the gradient-based inverse heat conduction solvers in our steel cooling applications, such as the instabilities due to the thermocouple reading noise and low temporal resolution, three gradient-free algorithms were studied. These methods are the artificial neural networks, genetic algorithms, and particle swarm optimization.

#### **6.1.2. Study of the Neural Network Algorithms for Inverse Heat Conduction Analysis**

Several types of artificial neural networks were studied in this research. Our investigations showed that a suitable formulation for the inverse heat conduction problems in

neural networks should relate the gradient (and not the values) of the inside temperature to the boundary heat fluxes. Also, it was found that the radial basis function is the most suitable type of neural networks for inverse heat conduction problems.

The artificial neural networks were shown to be capable of capturing the whole thermal history on the run-out table, but not very effective in restoring the detailed behavior of the boundary conditions. Also, they behave poorly in nonlinear cases and where the boundary condition profile is different. Thus, their application in the run-out table heat transfer analysis will be limited to a general prediction of the whole history.

### **6.1.3. Study of the GA and PSO Algorithms for Inverse Heat Conduction Analysis**

While GA has been previously used for inverse heat conduction analysis, the application of PSO to these problems was first studied in this research. GA and PSO are more effective in finding a detailed representation of the time-varying boundary conditions, as well as in nonlinear cases. However, their convergence takes longer. For the first time, comparison was made between GA and PSO variations in terms of both the computational efficiency and effectiveness in dealing with the noisy data in inverse heat conduction problems. PSO was found to improve efficiency (i.e., reduce the required computational effort), especially in more complex test cases. In addition to the basic PSO algorithm, two other variations were studied in this research. One of them, the complete repulsive PSO (CRPSO), was introduced for the first time in this work. CRPSO showed the best performance among three other versions. The effectiveness of PSO was also studied in the presence of noise in inverse heat conduction applications. PSO proved to be effective in handling noisy data, especially when its performance parameters were tuned. The proper choice of the regularization parameter helped PSO deal with noisy data, similar to the way it helps the classical function specification approaches. This research showed that an increase in the self-confidence parameter is also effective, as it increases the global search capabilities of the algorithm. RPSO was found to be the most effective variation in dealing with noise, followed closely by CRPSO. The latter variation was recommended for inverse heat conduction problems, as it combines the efficiency and effectiveness required by these problems.

#### **6.1.4. Improvements in the Efficiency and Effectiveness of Gradient-free Algorithms for IHC Analysis**

This research showed that while both GA and PSO are capable of overcoming many of the shortcomings faced by the classical inverse analysis algorithms, they suffer from a slow convergence rate. Several modifications were proposed for the first time in this research that resulted in a faster algorithm for inverse analysis of heat transfer. Sequential formulation, i.e. solving the inverse problem for each separate time step was found more efficient than a whole domain implementation. Also, in the case of multi-dimensional problems with multiple sensors, we proposed a multi-objective optimization formulation that was very successful in accelerating the solution. The relative speedup seemed to be independent of the number of spatial components. Introducing the concept of elite members to the PSO algorithm also resulted in a faster convergence; however, the improvement was of less magnitude. It also looked to be more effective in simpler cases, i.e. smaller problems with fewer internal measurement points.

Also, the concept of future time steps was borrowed from the function specification methods, and for the first time was applied to the sequential stochastic inverse algorithms. It proved to be very helpful in making the method more stable, especially when used in combination with a Tikhonov regularization term and tuning of the self-confidence parameter in PSO. The developed modified formulation for the inverse heat conduction problem and PSO implementation seem to be promising in terms of stability and performance, and make an effective and efficient algorithm for inverse heat conduction problem.

#### **6.1.5. Surrogate Model Based Acceleration of GA and PSO Solvers**

In order to further improve the convergence characteristics of the proposed stochastic inverse heat conduction algorithms, the concept of inexact pre-evaluation using surrogate models was introduced and used to enhance the speed of evolutionary algorithms in solving the inverse heat conduction problems. In this strategy, the entire candidate solutions are first roughly evaluated using a much simpler surrogate model, and only the top performing members are exactly evaluated using the full direct solvers, such as finite element method. It was found that polynomial methods are the cheapest selection, both during the model construction phase, and the implementation. Radial basis function (RBF) and feedforward neural networks have similar construction time, but in making prediction, RBF models perform faster. Kriging model is the slowest, especially when the number of unknowns increases. All the investigated surrogate

models perform very well in predicting linear cases with no added noise. Nonlinear domains are much harder to predict. The most suitable model for noisy nonlinear cases is the RBF model. Since real-world cases of inverse heat conduction problem are normally nonlinear and noisy, this method was recommended for these applications in this study. An adaptive selection scheme based on the expected improvement of the objective function was found to be the best performing model management strategy. Also, making a separate model for each sensor was found to be the most appropriate way to construct the surrogate model. If done properly, surrogate modeling can speed up the stochastic methods of solving the inverse heat conduction problem up to around 5 times. Creating the structure for a stable, accurate, and fast inverse heat conduction solver is one of the major contributions of this research.

#### **6.1.6. Characterization of the Water Jet Impingement Cooling on a Moving Plate**

Boiling heat transfer on a hot plate cooled by multiple water jets and multiple jet lines in an industrial scale experimental setup was studied in this research. Nozzle configuration, plate speed, and jet line spacing were varied in the experiments. Thermocouple readings were analyzed using a 3D inverse heat conduction procedure. It was found that the surface temperature is the most significant factor affecting the heat transfer behaviour on the plate. At high temperatures, the plate surface is covered with a condensed vapour layer that acts like an insulation blanket, and requires an impinging jet with high momentum to break it down. This translates into a stable film boiling regime, with reduced values of heat fluxes. Heat transfer rate was shown to increase at medium temperatures where the vapour thickness is smaller. For low entry temperatures in a multi jet line experiment, the heat transfer is lowered again. One of the very interesting findings of this study was the fact that in the low temperature region, the maximum heat flux does not occur at the impingement point, but at a location between the jets, due to the formation of an interaction water fountain.

The most important effect of nozzle stagger is the uniformity of heat transfer across the width of the plate. The most uniform distribution is happening in the case of fully staggered configuration. The heat transfer between two successive rows of jet lines is affected by the jet line distance. In higher jet line distances, the interaction effects become less significant, and a more uniform distribution is observed. However, it was shown that when the jet lines are closer, the effect of interaction is more pronounced. In this study, it was observed that in medium and high temperatures, there is a spike in the heat flux values in the interaction zone. In the low

temperature range, because the water does not evaporate quickly, a pool of water forms on the plate surface, resulting in a flattened heat flux versus time profile. The plate speed affects the heat transfer rate under the impingement point for the higher surface temperatures. In the high entry temperatures, the impingement heat transfer rate is lower when the plate is moving at a higher velocity. The plate speed however does not significantly change the heat transfer behavior in the other parts of the plate surface.

#### **6.1.7. Study of the Structural Effects of the Controlled Cooling of Steel Strips**

It is desirable that the strips maintain their flatness throughout the controlled cooling process. Thus, it was necessary to study the effect of cooling patterns on the structure of the strips. A literature survey shows that the previous researches in this field are very preliminary, and include major simplifications. One of the major simplifications in the previous studies is the oversimplified phase transformation model. For the first time in this work, an advanced phase transformation model, developed in UBC, is used in association with the thermal and structural models. In this algorithm, the static recrystallization kinetics is simulated by the Johnson-Mehl-Avrami-Kolmogorov (JMAK) theory. This model is based on the real physics of the problem, and enhances the accuracy of our simulations.

Another improvement in the structural modeling in this research is the use of real heat flux data as the boundary conditions in the thermal field. The other researches assume a very simplified distribution of temperature or heat flux across the plate, and assume a linear decrease in temperature during the cooling on the run-out table. This research uses the results of a 3D inverse analysis of the thermal field, and applies an accurate heat flux boundary condition on the surface of the plate.

Also, it was shown that the most significant factor in controlling the stress and flatness in the strips is the jet configuration, and the most desirable configuration is the one that promotes a more uniform distribution of temperature across the plate width. Plate speed was found to be the next important factor in determining the values of stress and deformation. Plates traveling at slower speeds are cooled down more uniformly, and will have less stress and deflection in them. Finally, longer jet spacing gives the opportunity of a more uniform cooling and causes less deflection in the plate. It was also observed that increasing the plate thickness, increases the maximum deflection, due to a larger temperature gradient through the thickness.

## **6.2. Suggestions for Future Work**

### **6.2.1. Experimental Study of Higher Plate Speeds, Different Flow Rates, and Water Temperatures**

Using a higher plate speed will make the simulation closer to the real case happening in a steel mill. Also, water flow rate and temperature are the other parameters that can be used in a steel mill, in order to control the characteristics of the heat transfer. Thus, an experimental study of the effect of these parameters will be beneficial.

### **6.2.2. Correlations for Heat Transfer of Multiple Jets Impinging on a Hot Moving Plate**

More data points are needed to create correlations for the important heat transfer characteristics, such as the maximum heat flux and total extracted heat, as functions of the plate speed, number and configuration of the jets, and the water temperature and flow rate.

### **6.2.3. Coupling of Hydrodynamics and Heat Transfer Models**

A proper heat transfer model requires information about the water velocity and pressure distribution on the plate surface. The experimental and numerical study of the hydrodynamics of the water jet impingement on a moving plate has been started in UBC [32]. It should be coupled with the heat transfer studies, in order to provide a more comprehensive model for the phenomena in the jet impingement cooling of the hot moving plates.

### **6.2.4. Experimental Measurements and Correlations for Thermal Stresses and Deflections**

The thermal stresses and deflection on a water cooled hot plate should be investigated experimentally. The results of the existing numerical model should be compared with the experimental values. Also, similar to the heat transfer case, correlations should be developed in order to find the effect of cooling parameters on the structural field. The effect of steel composition and physical properties on the structural results should also be investigated to make future predictions possible, regardless of the type of steel that is used.

#### **6.2.5. Developing Mechanistic Models for Moving Plate Jet Impingement Boiling**

The knowledge that is gained by studying the heat transfer and hydrodynamics of the jet impingement on a moving plate can be used to develop and validate a mechanistic model for the different zones of the jet impingement boiling heat transfer.



## Bibliography

- [1] F. Xu, Finite Element Simulation of Water Cooling Process of Steel Strips on Runout Table, Ph.D. Thesis, University of British Columbia, 2005.
- [2] M. Korchynsky, Development of Controlled Cooling Practice, in: *Accelerated Cooling of Steel : Proceedings of a Symposium Sponsored by the Ferrous Metallurgy Committee of the Metallurgical Society of AIME and Held in Pittsburgh, Pennsylvania, August 19-21*, Anonymous PA: Warrendale, pp. 3-14, 1985.
- [3] E. R. Morgan, T. E. Dancy, and M. Korchynsky, Improved High Strength Low Alloy Steels through Hot Strip Mill Controlled Cooling, *Blast Furnace and Steel Plant*, vol. 53 (10) pp.921-929, 1965.
- [4] J. Filipovic, R. Viskanta, and F. P. Incropera, A Parameter Study of the Accelerated Cooling of Steel Strip, *Steel Research*, vol. 63 pp.496-499, 1992.
- [5] D.T. Blazevic, Comparison of Cooling Methods: Sprays, Water Curtains, Laminar Tubes and Aspirated Sprays, 35th Mechanical Working and Steel Processing Conference. (1993) pp. 297-310.
- [6] R. M. Guo, and S. T. Hwang, Investigation of Strip Cooling Behavior in the Runout, *Journal OF Materials Processing and Manufacturing Science*, vol. 4 pp.339-351, 1996.
- [7] W.K. Soh, and W.Y.D. Yuen, Flow Visualization of the Boiling Heat Transfer at the Run-Out Table, 41 st Mechanical Working and Steel Processing Conference (1999) pp. 707-715.
- [8] Z. Liu, Experimental and Mathematical Modeling of Controlled Run-Out Table Cooling in a Hot Rolling Mill, Ph.D. Thesis, University of British Columbia, 2001.
- [9] Q. Meng, Experimental Study of Transient Cooling of a Hot Steel Plate by an Impinging Circular Jet, Master Thesis, University of British Columbia, 2002.
- [10] H. Robidou, H. Auracher, P. Gardin, and M. Lebouché, Controlled Cooling of a Hot Plate with a Water Jet, *Experimental Thermal and Fluid Science*, vol. 26 (2-4) pp.123-129, 2002.
- [11] K.V. Jondhale, Heat Transfer during Multiple Jet Impingement on the Top Surface of Hot Rolled Steel Strip, M.Sc. Thesis, University of British Columbia, 2007.
- [12] G. Tacke, H. Litzke, and E. Raquet, Investigation into the Efficiency of Cooling Systems for Wide-Strip Hot Rolling Mills and Computer-Aided Control of Strip Cooling, in: *Accelerated Cooling of Steel : Proceedings of a Symposium Sponsored by the Ferrous Metallurgy Committee of the Metallurgical Society of AIME and Held in Pittsburgh, Pennsylvania, August 19-21*, P. D. Southwick, PA: Warrendale, pp. 35-54, 1985.

- [13] D. H. Wolf, F. P. Incropera, and R. Viskanta, Jet Impingement Boiling, *Advances in Heat Transfer*, vol. 23 pp.1-132, 1993.
- [14] D. A. Zumbrennen, F. P. Incropera, and R. Viskanta, Method and Apparatus for Measuring Heat Transfer Distributions on Moving and Stationary Plates Cooled by a Planar Liquid Jet, *Experimental Thermal and Fluid Science*, vol. 3 pp.202-213, 1990.
- [15] T. Ochi, S. Nakanishi, M. Kaji, and S. Ishigai, Cooling of a Hot Plate with an Impinging Circular Jet, Multi-Phase Flow and Heat Transfer III, Proceedings of the third multi-phase heat and flow transfer symposium workshop Miami Beach, FL, USA (1983) pp. 671-681.
- [16] S. J. Chen, J. Kothari, and A. A. Tseng, Cooling of a Moving Plate with an Impinging Circular Water Jet, *Experimental Thermal and Fluid Science*, vol. 4 pp.343-353, 1991.
- [17] J. Stevens, and B. W. Webb, Local Heat Transfer Coefficients Under an Axisymmetric Single-Phase Liquid Jet, *Journal of Heat Transfer*, vol. 113 pp.71-78, 1991.
- [18] N. Hatta, and H. Osakabe, Numerical Modeling for Cooling Process of a Moving Hot Plate by a Laminar Water Curtain, *ISIJ International*, vol. 29 pp.919-925, 1989.
- [19] J. Filipovic, R. Viskanta, and F. P. Incropera, Thermal Behavior of a Moving Steel Strip Cooled by an Array of Planar Water Jets, *Steel Research*, vol. 63 pp.438-446, 1992.
- [20] D. H. Wolf, F. P. Incropera, and R. Viskanta, Local Jet Impingement Boiling Heat Transfer, *International Journal of Heat and Mass Transfer*, vol. 39 pp.1395-1406, 1996.
- [21] Z. Liu, and Q. Zhu, Prediction of Critical Heat Flux for Convective Boiling of Saturated Water Jet Impinging on the Stagnation Zone, *Journal of Heat Transfer*, vol. 125 pp.159-165, 2003.
- [22] J. Kokado, N. Hatta, H. Takuda, J. Harada, and N. Yasuhira, An Analysis of Film Boiling Phenomena of Subcooled Water Spreading Radially on a Hot Steel Plate, *Steel Research*, vol. 55 pp.113-118, 1984.
- [23] S. Ishigai, S. Nakanishi, and T. Ochi, Boiling Heat Transfer for Plane Water Jet Impinging on a Hot Surface, Proceeding of the sixth International Heat Transfer Conference Toronto, ON, Canada (1978).
- [24] S. Summerfield, and D. Fraser, A New Heat Transfer Correlation for Impinging Zone Heat Transfer on a Hot Steel Plate, *Canadian Metallurgical Quarterly*, vol. 45 (1) pp.69-78, 2006.
- [25] F. Xu, and M. S. Gadala, Investigation of Error Sources in Temperature Measurement using Thermocouples in Water Impingement Cooling, *Experimental Heat Transfer*, vol. 18 (3) pp.153-177, 2005.

- [26] S.L. Summerfield, Boiling Heat Transfer for an Impinging Jet on a Hot Plate and the Development of a New Correlation, M.Sc. Thesis, University of Manitoba, 2004.
- [27] A.T. Hauksson, Experimental Study of Boiling Heat Transfer during Water Jet Impingement on a Hot Steel Plate, Master Thesis, University of British Columbia, 2001.
- [28] N. Chester, A Study of Boiling Heat Transfer on A Hot Steel Plate Cooled by an Inclined Circular Bottom Water Jet, M.Sc. Thesis, University of British Columbia, 2006.
- [29] P. Chan, Jet Impingement Boiling Heat Transfer at Low Coiling Temperatures, M.Sc. Thesis, University of British Columbia, 2007.
- [30] H. Maghzian, Simulation of Hydrodynamics of the Jet Impingement using Arbitrary Lagrangian Eulerian Formulation, Ph.D. Thesis, University of British Columbia, 2007.
- [31] G. Franco, Boiling Heat Transfer during Cooling of a Hot Moving Steel Plate by Multiple Top Jets, M.Sc. Thesis, University of British Columbia, 2008.
- [32] M.M. Seraj, Hydrodynamics of Long Circular Water Jets in Industrial Metal Cooling, Ph.D. Thesis, University of British Columbia, 2010.
- [33] W. Timm, K. Weinzierl, and A. Leipertz, Heat Transfer in Subcooled Jet Impingement Boiling at High Wall Temperatures, *International Journal of Heat and Mass Transfer*, vol. 46 (8) pp.1385-1393, 2003.
- [34] A. M. T. Omar, M. S. Hamed, and M. Shoukri, Modeling of Nucleate Boiling Heat Transfer Under an Impinging Free Jet, *International Journal of Heat and Mass Transfer*, vol. 52 (23-24) pp.5557-5566, 2009.
- [35] J.V. Beck, B. Blackwell, and C.R.S. Clair Jr, *Inverse Heat Conduction: Ill-Posed Problem*, New York: Wiley-Interscience Publication, 1985.
- [36] O.M. Alifanov, *Inverse Heat Transfer Problems*, Berlin ; New York: Springer-Verlag, 1995.
- [37] M.N. Ozisik, *Inverse Heat Transfer: Fundamentals and Applications*, New York: Taylor & Francis, 2000.
- [38] J.V. Beck, and K.J. Arnold, *Parameter Estimation in Engineering and Science*, New York: Wiley, 1977.
- [39] J. V. Beck, B. Blackwell, and A. Haji-Sheikh, Comparison of some Inverse Heat Conduction Methods using Experimental Data, *International Journal of Heat and Mass Transfer*, vol. 39 pp.3649-3657, 1996.

- [40] G. Blanc, M. Raynaud, and T. H. Chau, A Guide for the use of the Function Specification Method for 2D Inverse Heat Conduction Problems, *Revue Generale De Thermique*, vol. 37 (1) pp.17-30, 1998.
- [41] N. Al-Khalidy, A General Space Marching Algorithm for the Solution of Two-Dimensional Boundary Inverse Heat Conduction Problems, *Numerical Heat Transfer, Part B*, vol. 34 pp.339-360, 1998.
- [42] R. Abou khachfe, and Y. Jarny, Determination of Heat Sources and Heat Transfer Coefficient for Two-Dimensional Heat flow–numerical and Experimental Study, *International Journal of Heat and Mass Transfer*, vol. 44 (7) pp.1309-1322, 2001.
- [43] C. H. Huang, and S. P. Wang, A Three-Dimensional Inverse Heat Conduction Problem in Estimating Surface Heat Flux by Conjugate Gradient Method, *International Journal of Heat and Mass Transfer*, vol. 42 (18) pp.3387-3403, 1999.
- [44] C. H. Huang, I. C. Yuan, and A. Herchang, A Three-Dimensional Inverse Problem in Imaging the Local Heat Transfer Coefficients for Plate Finned-Tube Heat Exchangers, *International Journal of Heat and Mass Transfer*, vol. 46 (19) pp.3629-3638, 2003.
- [45] P. C. Tuan, C. C. Ji, L. W. Fong, and W. T. Huang, An Input Estimation Approach to on-Line Two-Dimensional Inverse Heat Conduction Problems, *Numerical Heat Transfer, Part B: Fundamentals*, vol. 29 (3) pp.345-363, 1996.
- [46] S. K. Ali, M. S. Hamed, and M. F. Lightstone, A Modified Online Input Estimation Algorithm for Inverse Modeling of Steel Quenching, *Numerical Heat Transfer, Part B: Fundamentals*, vol. 57 (1) pp.1-29, 2010.
- [47] J. L. Battaglia, A Modal Approach to Solve Inverse Heat Conduction Problems, *Inverse Problems in Engineering*, vol. 10 (1) pp.41-63, 2002.
- [48] M. Girault, D. Petit, and E. Videcoq, The use of Model Reduction and Function Decomposition for Identifying Boundary Conditions of A Linear Thermal System, *Inverse Problems in Science and Engineering*, vol. 11 (5) pp.425-455, 2003.
- [49] E. Videcoq, and D. Petit, Model Reduction for the Resolution of Multidimensional Inverse Heat Conduction Problems, *International Journal of Heat and Mass Transfer*, vol. 44 (10) pp.1899-1911, 2001.
- [50] B. R. Bass, Application of the Finite Element Method to the Nonlinear Inverse Heat Conduction Problem using Beck's Second Method, *Journal of Engineering and Industry*, vol. 102 pp.168-176, 1980.
- [51] A. S. Osman, Investigation of Transient Heat Transfer Coefficients in Quenching Experiments, *Journal of Heat Transfer*, vol. 112 pp.843-848, 1190.

- [52] S. Kumagai, S. Suzuki, Y.R. Sano, and M. Kawazoe, Transient Cooling of Hot Slab by an Impinging Jet with Boiling Heat Transfer, *ASME/JSME Thermal Engineering Conference* (1995) pp. 347-352.
- [53] H. Louahia-Gualous, P. K. Panday, and E. A. Artioukhine, Inverse Determination of the Local Heat Transfer Coefficients for Nucleate Boiling on a Horizontal Pipe Cylinder, *Journal of Heat Transfer*, vol. 125 pp.1087-1095, 2003.
- [54] H. K. Kim, and S. I. Oh, Evaluation of Heat Transfer Coefficient during Heat Treatment by Inverse Analysis, *Journal of Material Processing Technology*, vol. 112 pp.157-165, 2001.
- [55] M. Pietrzyk, and J. G. Lenard, A Study of Heat Transfer during Flat Rolling, *International Journal of Numerical Methods in Engineering*, vol. 30 pp.1459-1469, 1990.
- [56] O. M. Alifanov, A. V. Nenarokomov, S. A. Budnik, V. V. Michailov, and V. M. Ydin, Identification of Thermal Properties of Materials with Applications for Spacecraft Structures, *Inverse Problems in Science and Engineering*, vol. 12 (5) pp.579-594, 2004.
- [57] M. S. Gadala, and F. Xu, An FE-Based Sequential Inverse Algorithm for Heat Flux Calculation during Impingement Water Cooling, *International Journal of Numerical Methods for Heat and Fluid Flow*, vol. 16 (3) pp.356-385, 2006.
- [58] M. Raudensky, K. A. Woodbury, J. Kral, and T. Brezina, Genetic Algorithm in Solution of Inverse Heat Conduction Problems, *Numerical Heat Transfer, Part B: Fundamentals*, vol. 28 (3) pp.293-306, 1995.
- [59] M. Silieti, E. Divo, and A. J. Kassab, An Inverse Boundary Element method/genetic Algorithm Based Approach for Retrieval of Multi-Dimensional Heat Transfer Coefficients within Film Cooling holes/slots, *Inverse Problems in Science and Engineering*, vol. 13 (1) pp.79-98, 2005.
- [60] C. L. Karr, I. Yakushin, and K. Nicolosi, Solving Inverse Initial-Value, Boundary-Value Problems Via Genetic Algorithm, *Engineering Applications of Artificial Intelligence*, vol. 13 (6) pp.625-633, 2000.
- [61] J. Krejsa, K. A. Woodbury, J. D. Ratliff, and M. Raudensky, Assessment of Strategies and Potential for Neural Networks in the IHCP, *Inverse Problems in Engineering*, vol. 7 (3) pp.197-213, 1999.
- [62] E. H. Shiguemori, J. D. S. Da Silva, and H. F. De Campos Velho, Estmiation of Initial Condition in Heat Conduction by Neural Networks, *Inverse Problems in Science and Engineering*, vol. 12 (3) pp.317-328, 2004.

- [63] S. Lecoeuche, G. Mercere, and S. Lalot, Evaluating Time-Dependent Heat Fluxes using Artificial Neural Networks, *Inverse Problems in Science and Engineering*, vol. 14 (2) pp.97-109, 2006.
- [64] W. Aquino, and J. C. Brigham, Self-Learning Finite Elements for Inverse Estimation of Thermal Constitutive Models, *International Journal of Heat and Mass Transfer*, vol. 49 (15-16) pp.2466-2478, 2006.
- [65] S. Roudbari, Self Adaptive Finite Element Analysis, Cornell University, 2006.
- [66] S. K. Kim, and W. I. Lee, Solution of Inverse Heat Conduction Problems using Maximum Entropy Method, *International Journal of Heat and Mass Transfer*, vol. 45 (2) pp.381-391, 2002.
- [67] Ostrowski, Z., R. A. Bialstrokecki, and A. J. Kassab. "Solving Inverse Heat Conduction Problems using Trained POD-RBF Network Inverse Method." *Inverse Problems in Science and Engineering* (2007).
- [68] R. Eberhart, and J. Kennedy, A New Optimizer using Particle Swarm Theory, Proceedings of the Sixth International Symposium on Micro Machine and Human Science (1995) pp. 39-43.
- [69] R. Hassan, B. Cohanin, O. de Weck, and G. Venter, A Comparison of Particle Swarm Optimization and the Genetic Algorithm, Proceedings of the 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference (2005).
- [70] K. H. Lee, S. W. Baek, and K. W. Kim, Inverse Radiation Analysis using Repulsive Particle Swarm Optimization Algorithm, *International Journal of Heat and Mass Transfer*, vol. 51 pp.2772-2783, 2008.
- [71] J. Filipovic, R. Viskanta, F. P. Incropera, and T. A. Veslocki, Cooling of a Moving Steel Strip by an Array of Round Jets, *Steel Research*, vol. 65 (12) pp.541-547, 1994.
- [72] B. W. Webb, and C. F. Ma, Single-Phase Liquid Jet Impingement Heat Transfer, *Advances in Heat Transfer*, vol. 26 (2) pp.105-217, 1995.
- [73] D. T. Vader, F. P. Incropera, and R. Viskanta, A Method for Measuring Steady Local Heat Transfer to an Impinging Liquid Jet, *Experimental Thermal and Fluid Science*, vol. 4 pp.1-11, 1991.
- [74] D. T. Vader, F. P. Incropera, and R. Viskanta, Convective Nucleate Boiling on a Heated Surface Cooled by an Impinging, Planar Jet of Water, *Journal of Heat Transfer*, vol. 114 pp.152-160, 1992.

- [75] H. Robidou, H. Auracher, P. Gardin, M. Lebouché, and L. Bogdanić, Local Heat Transfer from a Hot Plate to a Water Jet, *Heat and Mass Transfer*, vol. 39 (10) pp.861-867, 2003.
- [76] F. Xu, and M. S. Gadala, Heat Transfer Behavior in the Impingement Zone Under Circular Water Jet, *International Journal of Heat and Mass Transfer*, vol. 49 (21-22) pp.3785-3799, 2006.
- [77] S. Ishigai, S. Nakanishi, M. Mizuno, and T. Imamura, Heat Transfer of the Impinging Round Water Jet in the Interference Zone of Film Flow Along the Wall, *Bulletin of the JSME*, vol. 20 (139) pp.85-92, 1977.
- [78] R. K. Sakhuja, F. S. Lazgin, and M. J. Owen, Boiling Heat Transfer with Arrays of Impinging Jets, *ASME Paper 80-HT-47* pp.47-61, 1981.
- [79] M. Monde, H. Kusuda, and H. Uehara, Burnout Heat Flux in Saturated Forced Convection Boiling with Two Or More Impinging Jets, *Heat Transfer-Japanese Research*, vol. 9 (3) pp.18-31, 1982.
- [80] S. Slayzak, R. Viskanta, and F. Incropera, Effects of Interaction between Adjacent Free Surface Planar Jets on Local Heat Transfer from the Impingement Surface, *International Journal of Heat and Mass Transfer*, vol. 37 (2) pp.269-282, 1994.
- [81] S. J. Slayzak, R. Viskanta, and F. P. Incropera, Effects of Interactions between Adjoining Rows of Circular, Free-Surface Jets on Local Heat Transfer from the Impingement Surface, *Journal of Heat Transfer*, vol. 116 (1) pp.88-95, 1994.
- [82] Y. Haraguchi, and M. Hariki, Analysis of Heat Transfer of Laminar Cooling for Uniform Temperature Control in Hot Strip Mill, 2nd International Conference on Modelling of Metal Rolling Processes (1996) pp. 606-611.
- [83] K. V. Jondhale, M. A. Wells, M. Militzer, and V. Prodanovic, Heat Transfer during Multiple Jet Impingement on the Top Surface of Hot Rolled Steel Strip, *Steel Research International*, vol. 79 (12) pp.938-946, 2008.
- [84] M. Gradeck, A. Kouachi, M. Lebouché, F. Volle, D. Maillet, and J. L. Borean, Boiling Curves in Relation to Quenching of a High Temperature Moving Surface with Liquid Jet Impingement, *International Journal of Heat and Mass Transfer*, vol. 52 (5-6) pp.1094-1104, 2009.
- [85] Z. Zhou, Flatness Control of Hot Rolled Steel Strip during Cooling on the Run-Out Table, Ph.D. Thesis, School of Physics and Material Engineering, Monash University, Australia, 2003.
- [86] H. Yoshida, Analysis of Flatness of Hot Rolled Steel Strip After Cooling, *Transactions of the Iron and Steel Institute of Japan*, vol. 24 (3) pp.212-220, 1984.

- [87] N. Nakata, and H. Yoshida, Influence of Cooling Uniformity at Run-Out Table on Strip Flatness, Recent Advances in Heat Transfer and Micro-Structure Modelling for Metal Processing, Proceedings of the 1995ASME International Engineering Congress and Exposition, San Francisco, USA, Society of Mechanical Engineers, Materials Division, ASME, MD (1995) pp. 67-77.
- [88] H. N. Han, J. K. Lee, H. J. Kim, and Y. S. Jin, A Model for Deformation, Temperature and Phase Transformation Behavior of Steels on Run-Out Table in Hot Strip Mill, *Journal of Materials Processing Technology*, vol. 128 (1-3) pp.216-225, 2002.
- [89] Z. Zhou, P. F. Thomson, Y. C. Lam, and D. D. W. Yuen, Numerical Analysis of Residual Stress in Hot-Rolled Steel Strip on the Run-Out Table, *Journal of Materials Processing Technology*, vol. 132 (1-3) pp.184-197, 2003.
- [90] Z. Zhou, Y. C. Lam, P. F. Thomson, and D. D. W. Yuen, Numerical Analysis of the Flatness of Thin, Rolled Steel Strip on the Runout Table, *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 221 (2) pp.241-254, 2007.
- [91] S. Vakili, and M. S. Gadala, Boiling Heat Transfer of Multiple Impinging Round Jets on a Hot Moving Plate, *Submitted to Heat Transfer Engineering* 2010.
- [92] C. G. Sun, H. N. Han, J. K. Lee, Y. S. Jin, and S. M. Hwang, A Finite Element Model for the Prediction of Thermal and Metallurgical Behavior of Strip on Run-Out-Table in Hot Rolling, *ISIJ International*, vol. 42 (4) pp.392-400, 2002.
- [93] M. Miyake, J. Too, and I. Samarasekera, Mathematical Modelling of Hot Rolling of Steel Strip with Microstructure Analysis, Simulation of materials processing: theory, methods and applications: proceedings of the 7th International Conference on Numerical Methods in Industrial Forming Processes--NUMIFORM 2001, Toyohashi, Japan, 18-20 June 2001 (2001) pp. 275.
- [94] A. Kumar, C. McCulloch, E. Hawbolt, and I. Samarasekera, Modelling Thermal and Microstructural Evolution on Runout Table of Hot Strip Mill, *Materials Science and Technology*, vol. 7 (4) pp.360-368, 1991.
- [95] "ANSYS Home Page." Available from <http://www.ansys.com>.
- [96] J. V. Beck, and D. A. Murio, Combined Function Specification-Regularization Procedure for Solution of Inverse Heat Conduction Problem, *AIAA Journal*, vol. 24 (1) pp.180-185, 1986.
- [97] M. H. Attia, and L. Kops, Distortion in Thermal Field Around Inserted Thermocouples in Experimental Interfacial Studies, *Journal of Engineering for Industry*, vol. 108 pp.241-246, 1986.



- [98] M. H. Attia, and L. Kops, Distortion in Thermal Field Around Inserted Thermocouples in Experimental Interfacial Studies--Part II: Effect of the Heat Flow through the Thermocouple. *Journal of Engineering for Industry*, vol. 110 (1) pp.7-14, 1988.
- [99] M. H. Attia, and L. Kops, Distortion in the Thermal Field Around Inserted Thermocouples in Experimental Interfacial Studies. III: Experimental and Numerical Verification, *Journal of Engineering for Industry*, vol. 115 (4) pp.444-449, 1993.
- [100] M. H. Attia, A. Cameron, and L. Kops, Distortion in Thermal Field Around Inserted Thermocouples in Experimental Interfacial Studies. Part 4: End Effect, *Journal of Manufacturing Science and Engineering*, vol. 124 (1) pp.135-145, 2002.
- [101] F. Xu, On the Application of Intrinsic Thermocouples in Temperature Measurements in Water Jet Cooling, *Numerical Heat Transfer, Part A: Applications*, vol. 51 (4) pp.343-362, 2007.
- [102] K. A. Woodbury, and S. K. Thakur, Redundant Data and Future Times in the Inverse Heat Conduction Problem, *Inverse Problems in Science and Engineering*, vol. 2 (4) pp.319-333, 1996.
- [103] K. Okamoto, and B. Q. Li, Optimal Numerical Methods for Choosing an Optimal Regularization Parameter, *Numerical Heat Transfer, Part B: Fundamentals*, vol. 51 (6) pp.515-533, 2007.
- [104] J. V. Beck, and D. A. Murio, Combined Function Specification-Regularization Procedure for Solution of Inverse Heat Conduction Problem, *AIAA Journal*, vol. 24 (1) pp.180-185, 1986.
- [105] J. Kaipio, and E. Somersalo, *Statistical and Computational Inverse Problems*, 2006.
- [106] L. Gosselin, M. Tye-Gingras, and F. Mathieu-Potvin, Review of Utilization of Genetic Algorithms in Heat Transfer Problems, *International Journal of Heat and Mass Transfer*, vol. 52 (9-10) pp.2169-2188, 2009.
- [107] L. Davis, *Handbook of Genetic Algorithms*, Thomson Publishing Group, 1991.
- [108] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1989.
- [109] M. Clerc, *Particle Swarm Optimization*, ISTE, 2006.
- [110] M.R. Alrasheed, C.W. de Silva, and M.S. Gadala, Evolutionary Optimization in the Design of a Heat Sink, in: *Mechatronic Systems: Devices, Design, Control, Operation and Monitoring*, C. W. de Silva, Taylor & Francis, 2008.

- [111] F.O. Karray, and C.W. de Silva, *Soft Computing and Intelligent System Design - Theory, Tools, and Applications*, New York: Addison Wesley, 2004.
- [112] S. Vakili, and M. S. Gadala, Effectiveness and Efficiency of Particle Swarm Optimization Technique in Inverse Heat Conduction Analysis, *Numerical Heat Transfer, Part B: Fundamentals*, vol. 56 (2) pp.119-141, 2009.
- [113] S.K.S. Fan, and J.M. Chang, A Modified Particle Swarm Optimizer using an Adaptive Dynamic Weight Scheme, in: *Digital Human Modeling*, V. G. Duffy, Springer Berlin / Heidelberg, pp. 56-65, 2007.
- [114] J. Kennedy, R.C. Eberhart, and Y. Shi, *Swarm Intelligence*, Morgan Kaufmann, 2001.
- [115] M.R. Alrasheed, C.W. de Silva, and M.S. Gadala, A Modified Particle Swarm Optimization Scheme and its Application in Electronic Heat Sink Design, ASME/Pacific Rim Technical Conference and Exhibition on Packaging and Integration of Electronic and Photonic Systems, MEMS, and NEMS (2007).
- [116] O. Urfalioglu, Robust Estimation of Camera Rotation, Translation and Focal Length at High Outlier Rates, The First Canadian Conference on Computer and Robot Vision (2004) pp. 464-471.
- [117] P. C. Fourie, and A. A. Groenwold, The Particle Swarm Optimization Algorithm in Size and Shape Optimization, *Structural and Multidisciplinary Optimization*, vol. 23 (4) pp.259-267, 2002.
- [118] G.E.P. Box, J.S. Hunter, and W.G. Hunter, *Statistics for Experimenters: Design, Innovation, and Discovery*, 2nd ed., New York: Wiley, 2005.
- [119] R. Jin, W. Chen, and T. W. Simpson, Comparative Studies of Metamodelling Techniques Under Multiple Modelling Criteria, *Structural and Multidisciplinary Optimization*, vol. 23 (1) pp.1-13, 2001.
- [120] A. Forrester, A. Sobester, and A. Keane, *Engineering Design Via Surrogate Modelling: A Practical Guide*, Chichester, UK: Wiley, 2008.
- [121] Y. Jin, A Comprehensive Survey of Fitness Approximation in Evolutionary Computation, *Soft Computing*, vol. 9 (1) pp.3-12, 2005.
- [122] D. Gorissen, L. De Tommasi, K. Crombecq, and T. Dhaene, Sequential Modeling of a Low Noise Amplifier with Neural Networks and Active Learning, *Neural Computing & Applications*, vol. 18 (5) pp.485-494, 2009.
- [123] Y. Jin, M. Olhofer, and B. Sendhoff, A Framework for Evolutionary Optimization with Approximate Fitness Functions, *IEEE Transactions on Evolutionary Computation*, vol. 6 (5) pp.481-494, 2002.

- [124] J.E. Dennis, and V. Torczon, Managing Approximation Models in Optimization, in: *Multidisciplinary Design Optimization: State-of-the-Art*, N. Alexandrov and M. Hussani. pp. 330–347, 1997.
- [125] C. Praveen, and R. DuVigneau, Low Cost PSO using Metamodels and Inexact Pre-Evaluation: Application to Aerodynamic Shape Design, *Computer Methods in Applied Mechanics and Engineering*, vol. 198 (9-12) pp.1087-1096, 2009.
- [126] B. Blackwell, and J. V. Beck, A Technique for Uncertainty Analysis for Inverse Heat Conduction Problems, *International Journal of Heat and Mass Transfer*, vol. 53 (4) pp.753-759, 2010.
- [127] R. J. Moffat, Describing the Uncertainties in Experimental Results, *Experimental Thermal and Fluid Science*, vol. 1 (1) pp.3-17, 1988.
- [128] B.F. Blackwell, and K.J. Dowding, Sensitivity and Uncertainty Analysis for Thermal Problems, Proceedings of the 4th Intern. Conf. on Inverse Problems in Engineering (4icipe), Rio de Janeiro, Brazil (2002).
- [129] G.E. Totten, M.H. Howes, and T. Inoue, *Handbook of Residual Stress and Deformation of Steel*, ASM International, 2002.
- [130] I. I. Boyadjiev, P. F. Thomson, and Y. C. Lam, Computation of the Diffusional Transformation of Continuously Cooled Austenite for Predicting the Coefficient of Thermal Expansion in the Numerical Analysis of Thermal Stress, *ISIJ International*, vol. 36 (11) pp.1413-1419, 1996.
- [131] M. Militzer, R. Pandi, and E. B. Hawbolt, Ferrite Nucleation and Growth during Continuous Cooling, *Metallurgical and Materials Transactions A*, vol. 27 (6) pp.1547-1556, 1996.
- [132] D. Liu, F. Fazeli, and M. Militzer, Modeling of Microstructure Evolution during Hot Strip Rolling of Dual Phase Steels, *ISIJ International*, vol. 47 (12) pp.1789-1798, 2007.
- [133] T. A. Kop, J. Sietsma, and S. Van Der Zwaag, Dilatometric Analysis of Phase Transformations in Hypo-Eutectoid Steels, *Journal of Materials Science*, vol. 36 (2) pp.519-526, 2001.

## Appendix A. Definition of Terms in the Finite Element Heat Conduction Formulation

**Table A. 1. Definition of the Terms in the FE General Heat Conduction Equation**

$[C] = \int_V \rho c [N]^T [N] dV$	Thermal capacity matrix
$[K_c] = \int_V [B]^T [K] [B] dV$	Thermal conductivity matrix
$[K_h] = \int_S h [N^s]^T [N^s] dS$	Thermal conductivity matrix due to convection B.C.
$[K_r] = \int_S \kappa [N^s]^T [N^s] dS$	Thermal conductivity matrix due to radiation B.C.
$\{Q\}^b = \int_V q^b [N]^T dV$	Heat flux vector due to internal heat generation
$\{Q\}^s = \int_S q^s [N^s]^T dS$	Heat flux vector due to input surface flux
$\{Q\}^h = \int_S h T_f [N^s]^T dS$	Heat flux vector due to convection B.C.
$\{Q\}^r = \int_S \kappa T_r [N^s]^T dS$	Heat flux vector due to radiation B.C.
$\{T\}, \{\dot{T}\}$	Vector of global nodal temperatures and temperature gradients, respectively.
$\kappa = \varepsilon \sigma (T_r^2 + T_{sr}^2)(T_r + T_{sr})$	Equivalent heat transfer coefficient due to radiation
$\left\{ \frac{\partial T}{\partial \mathbf{x}} \right\} = \frac{\partial}{\partial \mathbf{x}} (N_i \bar{T}_i) = [B] \{\bar{T}\}^e$	Partial derivative of temperature
$[K] = \begin{bmatrix} k_x & 0 & 0 \\ & k_y & 0 \\ sym & & k_z \end{bmatrix}$	Element conductivity matrix
$[N] = [N_1 \quad N_2 \quad \dots \quad N_n]$	Approximation or shape function , $n$ is the number of nodes per element

**Table A. 2. Definition of the Terms in the Heat Transfer Equation**

Term	Name	Expression
${}^{t+\Delta t}\hat{\mathbf{Q}}_h^{(i-1)}$	Nodal heat flux contribution due to convection B.C, nonlinear and transient effects	${}^{t+\Delta t}\hat{\mathbf{Q}}_h^{(i-1)} = \int_{S_h} {}^{t+\Delta t}h^{(i-1)} \mathbf{N}^{S^T} \mathbf{N}^S \left( {}^{t+\Delta t}\mathbf{T}_f - {}^t\mathbf{T}^{(i-1)} \right) \cdot dS$ <p>Note: <math>\int_{S_h} {}^{t+\Delta t}h^{(i-1)} \mathbf{N}^{S^T} \mathbf{N}^S \cdot dS = {}^{t+\Delta t}\mathbf{K}_h^{(i-1)}</math></p>
${}^{t+\Delta t}\hat{\mathbf{Q}}_r^{(i-1)}$	Nodal heat flux contribution due to radiation B.C., nonlinear and transient effects	${}^{t+\Delta t}\hat{\mathbf{Q}}_r^{(i-1)} = \int_{S_h} {}^{t+\Delta t}\kappa^{(i-1)} \mathbf{N}^{S^T} \mathbf{N}^S \left( {}^{t+\Delta t}\mathbf{T}_r - {}^t\mathbf{T}^{(i-1)} \right) \cdot dS$ <p>Note: <math>\int_{S_h} {}^{t+\Delta t}\kappa^{(i-1)} \mathbf{N}^{S^T} \mathbf{N}^S \cdot dS = {}^{t+\Delta t}\mathbf{K}_r^{(i-1)}</math></p>
${}^{t+\Delta t}\hat{\mathbf{Q}}_c^{(i-1)}$	Nodal heat flux contribution due to conductivity, nonlinear and transient effects	${}^{t+\Delta t}\hat{\mathbf{Q}}_c^{(i-1)} = \int_V \mathbf{B}^T {}^{t+\Delta t}\mathbf{K}^{(i-1)} \mathbf{B} {}^{t+\Delta t}\mathbf{T}^{(i-1)} \cdot dV$ <p>Note: <math>\int_V \mathbf{B}^T {}^{t+\Delta t}\mathbf{K}^{(i-1)} \mathbf{B} dV = {}^{t+\Delta t}\mathbf{K}^{(i-1)}</math></p>
${}^{t+\Delta t}\hat{\mathbf{q}}_c^{(i-1)}$	Nodal heat flux contribution due to thermal capacity, nonlinear and transient effects	${}^{t+\Delta t}\hat{\mathbf{q}}_c^{(i-1)} = \int_V ({}^{t+\Delta t}\rho c)^{(i-1)} \mathbf{N}^T \mathbf{N} \cdot \left[ \left( {}^{t+\Delta t}\mathbf{T}^{(i-1)} - {}^t\mathbf{T} \right) / \Delta t \right] \cdot dV$ $\equiv {}^{t+\Delta t}\mathbf{C}^{(i-1)} \left[ {}^{t+\Delta t}\mathbf{T}^{(i-1)} - {}^t\mathbf{T} \right] / \Delta t$