

**Numerical Simulation Using Transition SST Model to Analyze Effects of Expanding
Manifold Angle and Jet Spacing for Submerged Liquid Jet Impingement**

by

Kayla Elizabeth Reid

A thesis submitted to the Graduate Faculty of
Auburn University
In partial fulfillment of the
Requirements for the Degree of
Master of Science

Auburn, Alabama
August 5, 2018

Keywords: jet impingement, staggered jet array, fountain region effects, spent flow management,
expanding manifold angle

Copyright 2018 by Kayla Elizabeth Reid

Approved by

Roy W. Knight, Chair, Associate Professor of Mechanical Engineering
Sushil H. Bhavnani, Professor of Mechanical Engineering
Daniel K. Harris, Associate Professor of Mechanical Engineering

Abstract

Increasing numbers of under-hood sensors and power electronics modules are becoming standard in both commercial and military vehicles. In order to function reliably, these technologies require a dedicated and dynamic cooling system, such as liquid jet impingement. In a jet array, the spent fluid from upstream jets interacts with the downstream jets degrading their performance. In this study, in order to counteract this effect, an expanding manifold, with larger area for flow downstream, was considered to allow the spent fluid from upstream jets to be diverted, reducing degradation of the heat transfer coefficients downstream. A numerical study of liquid jet impingement utilizing water as the working fluid was performed to examine the heat transfer rate in staggered jet arrays compared to inline jet arrays. The simulations performed examined manifold angles between 0 and 10 degrees, jet Reynolds numbers between 5600 and 14000, and pitches of 2.5, 3, 4.5, and 6 nozzle diameters. The simulations revealed details of the complicated interaction between the jets, their fountain regions and their crossflow in increasing the surface heat transfer coefficient and surface temperature homogeneity. The angled manifold systems had greater temperature uniformity and increased heat transfer coefficients compared to systems with constant area manifolds.

Acknowledgements

Thanks to the members of my advisory committee, Dr. Roy W. Knight, Dr. Sushil H. Bhavnani, and Dr. Daniel K. Harris, with special thanks to Dr. Knight and Dr. Bhavnani for being patient, supportive, and motivational throughout my classes and research process.

Thanks to Dr. John F. Maddox for his conception of this study and for his aid in troubleshooting initial struggles with numerical modeling.

Thanks to Michael Henry and Sriram Chandrasekaran for their constant support as academic peers and friends in our seemingly endless workload.

Most of all I would like to thank my four wonderful parents, sister, brother, and partner, Chris Daniel, for their endless love and encouragement without whom I would not have gotten through this process. Thank you for supporting me through my rants and tears; I could not have done this without you.

Table of Contents

Acknowledgements	iii
Table of Contents	iv
List of Figures.....	vii
List of Tables	ix
Nomenclature	x
Chapter 1: Introduction.....	1
1.1 Electronics Thermal Management	1
1.2 Single Impinging Jet Regions.....	1
1.2.1 Free Jet	2
1.2.2 Stagnation Region.....	2
1.2.3 Wall Jet Region.....	3
1.3 Array of Impinging Jet Regions	3
1.3.1 Fountain Regions	3
1.3.2 Spent Fluid	4
1.4 Numerical Modeling	4
1.4.1 Reynolds-Averaged Navier Stokes (RANS).....	5
1.4.2 k- ϵ Model	5
1.4.3 k- ω Model	6
1.4.4 v^2f Model.....	6
1.4.5 Hybrid Models	7

1.4.6	Alternate Numerical Models.....	7
1.5	Summary.....	8
Chapter 2:	Background	10
2.1	Optimum Geometric Considerations for Jet Arrays.....	10
2.1.1	Jet Height	10
2.1.2	Jet Spacing (Pitch)	11
2.2	Alternate Geometric Enhancements	11
2.3	Effect of Expanding Manifold Angle.....	12
2.4	Objective of Current Study.....	12
Chapter 3:	Numerical Simulation Setup	14
3.1	Geometry	14
3.2	Mesh	15
3.3	Fluent	17
3.3.1	Theory for Transition SST Model.....	17
3.3.2	Constants for Transition SST Model	18
3.4	Grid Independence.....	19
Chapter 4:	Simulation Results	23
4.1	Expanding Manifold Angle	24
4.2	Pitch.....	34
4.3	Comparison to Experiment.....	38
Chapter 5:	Conclusions.....	40

5.1	Suggestions for future work.....	41
Chapter 6:	References	43
Appendix A:	Simulation Setup Procedure using ANSYS Fluent 16.2 and Auburn HPCC. 48	
Appendix A.1	Build Geometry	48
Appendix A.2	Mesh Generation in Fluent.....	59
Appendix A.3	Fluent 16.2.....	65
Appendix A.4	HPCC Instructions	69
Appendix A.5	MATLAB Code for generating HTC images.....	76
Appendix B:	Transport Equations for Transition SST Model.....	77
Appendix B.1	Transport Equations from Two-Equation Models.....	77
Appendix B.2	Transport Equations for Intermittency and Transition Momentum Thickness	80
Appendix B.3	Separation-Induced Transition Correction	81
Appendix B.4	Empirical Correlations	81
Appendix C:	Collection of Raw HTC Contour Images.....	83

List of Figures

Figure 1.1: Flow Regions in a Single Jet [1]	2
Figure 1.2 Jet Regions for an Array of Impinging Jets [1]	3
Figure 1.3 Crossflow effect on array of impinging jets due to spent fluid interfering with core of downstream jet [1].....	4
Figure 3.1 2-D Cross section drawing of geometry showing jet spacing and applied symmetry conditions.....	14
Figure 3.2: 3-D Geometry showing applied fluid and heat flow.....	15
Figure 3.3 Completed Mesh for $\gamma = 10^\circ$ and $P^* = 6$	16
Figure 3.4: Surface profiles for grid independence study	21
Figure 4.1 Velocity Streamlines for $P^* = 4.5$, $\gamma = 5^\circ$, and $Re = 8400$	23
Figure 4.2: Surface heat transfer coefficient at $\gamma = 0^\circ$ and $Re_D = 11,200$.....	25
Figure 4.3: Surface heat transfer coefficient at $P^* = 6$ and $Re_D = 11,200$.....	27
Figure 4.4: Surface heat transfer coefficient at $P^* = 4.5$ and $Re_D = 11,200$.....	28
Figure 4.5: Surface heat transfer coefficient at $P^* = 3$ and $Re_D = 11,200$.....	29
Figure 4.6: Surface heat transfer coefficient of individual jet farthest downstream for $P^* =$ 4.5 and $Re_D = 11,200$	30
Figure 4.7: Average heat transfer coefficients across varying Reynolds number based on expanding manifold angle	31
Figure 4.8: Temperature Contours at $P^* = 3$, $Re = 11200$ for (Top) $\gamma = 0^\circ$ and (Bottom) $\gamma =$ 10°	32

Figure 4.9: Average θ varying expanding manifold angle	34
Figure 4.10: Surface heat transfer coefficient at $\gamma = 10^\circ$ and $Re_D = 11,200$.....	35
Figure 4.11: Effect of pitch on temperature homogeneity	36
Figure 4.12: Average θ for $P^* = 3$ varying expanding manifold angle	37
Figure 4.13: Comparison of experimental (left) and numerical (right) surface heat transfer coefficients at $Re = 8400$ (top) and $Re = 11200$ (bottom) for $P^* = 3$ [26].	38
Figure 4.14: Experimental (top) and Numerical (bottom) comparison of heat transfer coefficient based on pitch, angle, and jet Reynolds number	39
Figure A.2.6.1: Selecting Geometric Script in ANSYS Workbench.....	59
Figure A.2.6.2: Selecting Geometry File	59
Figure A.2.6.3: Components shown in Workbench Window	60
Figure A.2.6.4: Meshing Window in ANSYS	60
Figure A.2.6.5: Changing Material Properties in ANSYS Mesh.....	61
Figure A.2.6.6: Naming a selected Body/Element in Mesh	61
Figure A.2.6.7: Names of Faces in Mesh.....	62
Figure A.2.6.8: Face Sizing in Mesh.....	62
Figure A.2.6.9: Sizing bottom face of fluid in Mesh	63

List of Tables

Table 1-1 Comparison of CFD Turbulence models used for jet impingement analysis [8]...	9
Table 3-1: Constants used for Transition SST model	18
Table 3-2: Meshing parameters used for grid independence study	19
Table A.2.6-1: Specified Meshing Parameters	63

Nomenclature

Acronyms

CFD	Computational Fluid Dynamics
GUI	Graphical User Interface
HPCC	High Performance Computing Cluster
htc	heat transfer coefficient
RANS	Reynolds-Averaged Navier Stokes
SST	Shear Stress Transport
TUI	Text User Interface

English Letter Symbols

A	surface area, m^2
D_n	jet inner diameter, m
H	height of jet from plate
Nu	Nusselt number, hD_n/k
P	jet pitch
Pr	Prandtl number
Re	jet Reynolds number, $u_{mean}D_n/\nu$
T	temperature, K
h	local heat transfer coefficient, $W/(m^2 \cdot K)$
k	thermal conductivity, $W/(m \cdot K)$ turbulent kinetic energy, m^2/s^2

\dot{q}''	heat flux, W/m ²
u	velocity, m/s
x	coordinate along length of impingement surface, m
y	coordinate along width of impingement surface, m
z	coordinate normal to impingement surface, m

Greek symbols

ε	dissipation rate, m ² /s ³
θ	temperature difference between impingement surface and inlet, K
γ	angle of confining wall, ° intermittency
ρ	mass density, kg/m ³
ν	kinematic viscosity of fluid, kg/(m·s ²)
ω	specific dissipation rate, 1/s

Chapter 1: Introduction

1.1 Electronics Thermal Management

As power electronic modules have decreased in size, the heat fluxes generated have increased requiring improved thermal management techniques to remove the heat more efficiently than traditional air cooling methods. The most commonly used single-phase technique for heat removal is a heat sink which focuses on heat spreading utilizing conductive heat transfer through the base and fins, followed by convective heat transfer through the flow of coolant between the fins. A recent alternative to heat sinks has been the use of jet impingement which can have higher heat transfer coefficients when compared to the traditional heat sink method. Jet impingement has been shown to remove hot spots without the use of a heat spreader while requiring less pumping power to achieve sufficient cooling performance. In addition, jet arrays can be integrated into the existing coolant flow loop around modern power electronics modules without the added cost and complexity of creating a secondary flow loop dedicated to electronics cooling.

1.2 Single Impinging Jet Regions

An impinging jet discharges fluid onto a surface to achieve enhanced heat transfer coefficients. The stream of fluid has an increased velocity by forcing the fluid through an orifice or nozzle. The nozzle's geometry, including the shape, size, and angle to the surface, can affect the heat transfer characteristics of the jet. For this study, the jets will be submerged, signifying that the jet discharges fluid into a surrounding fluid of similar properties. An example of an unsubmerged jet would be water impinging onto a surface with air as the ambient fluid. The flow regions in a single jet are depicted in Figure 1.1 [1].

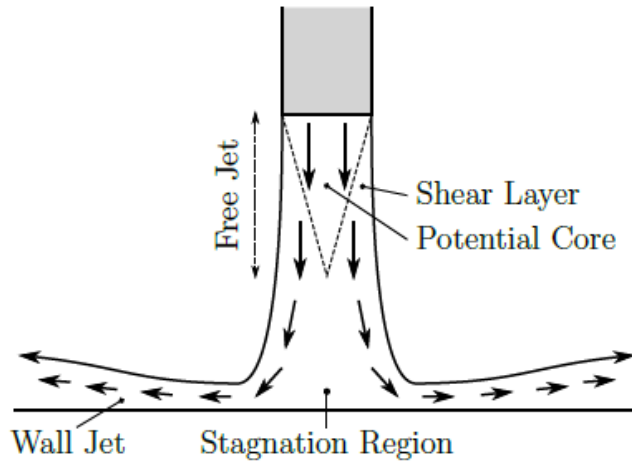


Figure 1.1: Flow Regions in a Single Jet [1]

1.2.1 Free Jet

For a single jet, the free jet region is the region below the nozzle and above the impinged surface that is unaffected by the impingement surface. This region begins as the fluid leaves the nozzle where a shearing layer forms due to the viscous effects between the impinging fluid from the jet and the surrounding fluid. At the nozzle exit, the potential core is formed, which is characterized as the region within a jet that has a uniform velocity profile. As the shearing layer thickness grows from the nozzle exit, the momentum exchange between the jet and the surrounding fluid increases causing the potential core to decrease in size, as seen in Figure 1.1.

1.2.2 Stagnation Region

Below the free jet region is the stagnation region where the fluid flow is influenced by the impingement surface. In this region, the flow direction changes from the normal direction (z) to flowing in parallel to the surface, the transverse directions (x and y). The point of the surface directly beneath the center of the jet flow is the stagnation point where the fluid has zero velocity. The highest heat transfer coefficients for a jet occur within the stagnation region.

1.2.3 Wall Jet Region

The wall jet region is where the fluid flows parallel to the surface. A boundary layer forms in the wall jet region due to the viscous effects of the surrounding fluid along with the no-slip condition. As the fluid flows farther away from the stagnation region, the boundary layer thickness increases and the heat transfer coefficient on the surface decreases.

1.3 Array of Impinging Jet Regions

Arrays of impinging jets are formed when multiple jets are used in close proximity. The flow regions for an array of jets are shown in Figure 1.2 [1]. If there is a small enough distance between neighboring jets, fountain regions are formed between jets.

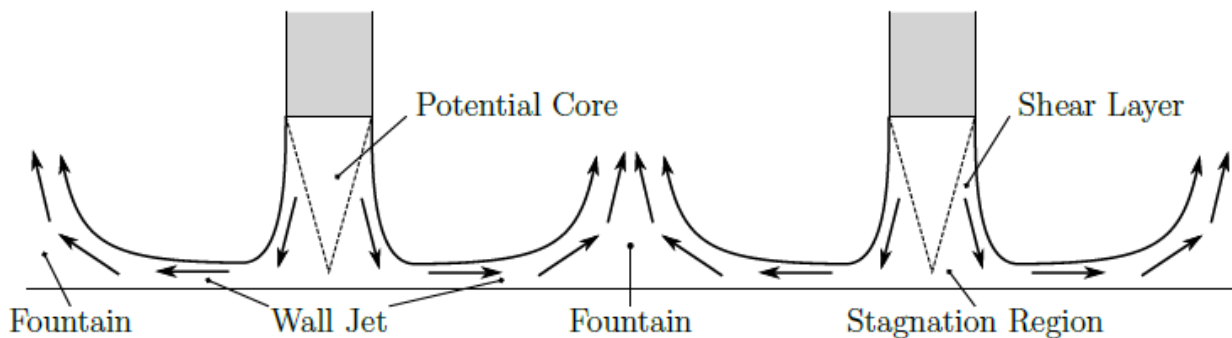


Figure 1.2 Jet Regions for an Array of Impinging Jets [1]

1.3.1 Fountain Regions

When wall jets from neighboring jet arrays collide, the fluid is forced upwards and away from the impingement surface. This inverted jet of fluid is called a fountain region. The fountain regions force the fluid away from the surface, creating areas of increased heat transfer resulting in higher average heat transfer coefficients on the impingement surface. The fluid that is forced upwards from a fountain region is known as spent fluid.

1.3.2 Spent Fluid

Although the fountain regions can be beneficial to heat transfer, the spent fluid forced upwards in a fountain region can also entrain back to the center of the impinging jet as shown in Figure 1.3 [1]. This entrainment can alter the flow of the impinging jets and degrade the heat transfer within the stagnation region. The degradation effects increase with the jets farther downstream as there is a larger volume of fluid to entrain within the downstream impinging jets.

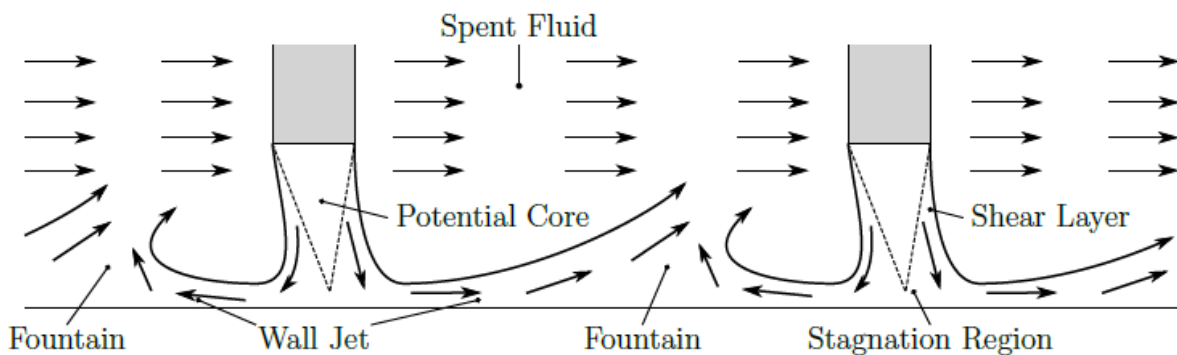


Figure 1.3 Crossflow effect on array of impinging jets due to spent fluid interfering with core of downstream jet [1]

In order to prevent the degradation of downstream jets in an array of jets, the spent fluid from the fountain regions needs to be effectively removed from the system. A few of the methods used to decrease the degradation of downstream jets include installing ducts or channels for the spent fluid to escape [2] [3], varying the jet diameters downstream, inciting swirling using helical inserts, and expanding the manifold angle. This study will focus on the use of an expanding manifold as the spent fluid management scheme.

1.4 Numerical Modeling

Using an appropriate numerical model to predict the flow in a jet impingement system is critical for designing a jet impingement system due to the amount of time and expense involved in experimental testing. Although there are purely analytical systems that exist for laminar jets, the

turbulent nature of a jet impingement system makes accurate analytical modeling difficult; the best turbulent models are only able to predict heat transfer and temperature surface conditions to within 20% of experimental jet arrays without the use of a supercomputer [4]. The models used to calculate the turbulent conditions in a jet impingement system must use time-averaged values and empirical correlations to more accurately depict the flow values.

1.4.1 Reynolds-Averaged Navier Stokes (RANS)

The RANS model decomposes all instantaneous variables into their time-averaged components resulting in time-averaged continuity and momentum equations as shown below.

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i) = 0 \quad (1.1)$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_i}(\rho u_i u_j) = \frac{\partial \rho}{\partial x_i} + \frac{\partial}{\partial x_j} \left[\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \delta_{ij} \frac{\partial u_k}{\partial x_k} \right) \right] + \frac{\partial}{\partial x_j}(-\rho \overline{u_i' u_j'}) \quad (1.2)$$

All considered models use the RANS equations shown above due to their ability to calculate time-averaged turbulence effects. The RANS models can either be calculated using a two-equation model with an eddy-viscosity to account for the Reynolds Stresses which assumes that the turbulent viscosity is an isotropic value or by using six independent, semi-empirical equations to solve for the Reynolds Stresses exactly while evaluating the turbulent viscosity as anisotropic [5].

1.4.2 k-ε Model

The k-ε model is the most widely used turbulent model due to its simplicity in solving a two-equation model while using the Eddy Viscosity approach to solve for the Reynolds stresses and its proven accuracy for a wide range of turbulent flows in the industry. Although the k-ε model is the most popular turbulent model used in the industry, it has been proven many times to be inaccurate in predicting impinging jet flow behavior [6], [7]. This can be attributed to the k-ε model's

insensitivity to adverse pressure gradients and boundary layer separation which is crucial for solving the shear layer at the impinging wall. Improvements upon the $k-\varepsilon$ model include the Realizable $k-\varepsilon$ model and using an enhanced wall treatment but these improvements have not enhanced the ability of the $k-\varepsilon$ model to predict the behavior near the surface for jet impingement models [5].

1.4.3 $k-\omega$ Model

The $k-\omega$ model provides improvements over the $k-\varepsilon$ model since the ω -equation can be integrated through the viscous sublayer allowing it to better predict adverse pressure gradient and boundary layer separation. The drawback is the inability of the $k-\omega$ model to predict freestream values resulting in the traditional $k-\omega$ model being unusable for turbulent modeling without enhancements to account for the conditions within the freestream flow. Due to inability of the $k-\omega$ model to predict freestream values, it is not used for jet impingement systems without the enhancements of either the Baseline (BSL) or shear-stress transport (SST) models [5].

1.4.4 v^2f Model

The v^2f model is similar to the $k-\varepsilon$ model but incorporates two additional equations to account for the anisotropic turbulent values and an improved velocity scale near the surface. The greatest benefit to the v^2f model is that it is valid all the way up to solid walls so it does not need to use wall functions. The v^2f model has proven to provide realistic models of turbulence including in the decelerating jet core which neither of the previously mentioned two-equation models are capable of computing. The major drawback to the v^2f model is the added computational time due to the refined grid required on the impingement surface. Despite the drawbacks, the v^2f model is regarded as one of the best choices for modeling jet impingement systems [8].

1.4.5 Hybrid Models

Although none of the two-equation turbulence models provide accurate results for jet impingement models, hybrid CFD models have been developed to utilize the best two-equation model approach for the varying regions: free jet, stagnation region, and wall jet. A transition equation is applied to smooth the boundaries between the different models used and two additional equations are added to for the intermittency and the transition onset criteria in terms of the momentum thickness Reynolds number. Menter [9] originally proposed a model that used the $k-\omega$ model within the boundary layer surface and the $k-\varepsilon$ model in the freestream flow called the shear stress transport (SST) model. Modifications made by Menter and Langry [4] added empirical correlations that could be modified by the user to control the transition onset momentum thickness Reynolds number equation. Maddox [1] and Esch and Menter [10] found that the Transition SST model predicted heat transfer rates within 5% of the v^2f model while using significantly fewer computation resources. Due to the inability of the two equation models to accurately depict jet impingement grid resolution should cover ~ 10 cells inside the boundary layer normal to the impingement wall to accurately resolve the shear layers in a turbulent model [5].

1.4.6 Alternate Numerical Models

Direct numerical simulation (DNS) and Large Eddy Simulation (LES) use high resolution grids to solve the full Navier-Stokes equations including the effects of the microscopic turbulent length scale. Due to the high computational load that can only be provided by supercomputers, DNS and LES were unable to be used for this study. Algebraic Stress Models (ASM) reduce computational costs by solving a set of algebraic equations rather than transport equations but requires advanced knowledge of the expected turbulent length and time values to calculate the turbulence terms within the algebraic equations. ASM could not be used since these values are unknown for the

staggered jet impingement models. Reynolds Stress Transport Models (RSTM) use anisotropic values as with the v^2f and Transition SST models by tracing all six components in the Reynolds stress tensor. However, the RSTM model were reported to have errors up to 100% depending on jet height and incorrectly predicted the secondary peak in heat transfer for jet arrays [11], [12].

1.5 Summary

Arrays of impinging jets have proven challenging to model due to the different flows within the various regions within the jet array. The two equation models, $k-\epsilon$ model and $k-\omega$ model, are unable to predict all of the regions within a jet array so they were not considered for numerical simulation. RSTM and ASM were unable to predict different regions within the jet array. While DNS and LES models have revealed information about the flow regions within a jet, both models are too computationally expensive for this study. Although the v^2f model is less computationally expensive than the DNS and LES models, the added computational expense from the Transition SST model to the v^2f model was shown to have less than a 5% improvement on predictions of surface heat transfer coefficients [1], [10]. Due to the reduced computational expense and ability to accurately predict surface heat transfer coefficients and temperatures for jet arrays when compared to other commercially available models, the Transition SST model was chosen as the numerical simulation model for this study. A comparison of the CFD turbulence models in analyzing jet impingement regions is shown in Table 1.1.

Table 1-1 Comparison of CFD Turbulence models used for jet impingement analysis [8]

Turbulence model	Computational cost (time required)	Impingement jet transfer coefficient prediction	Ability to predict secondary peak
$k-\varepsilon$	★★★★★ Low cost	★ Poor: Nu error of 15–60%	★ Poor
$k-\omega$	★★★★★ Low–moderate	★★ Poor–fair: anticipate Nu errors of at least 10–30%	★★ Fair: may have incorrect location or magnitude
Realizable $k-\varepsilon$ and other $k-\varepsilon$ variations	★★★★★ Low	★★ Poor–fair: expect Nu errors of at least 15–30%	★★ Poor–fair: may have incorrect location or magnitude
Algebraic stress model	★★★★★ Low	★★ Poor–fair: anticipate Nu errors of at least 10–30%	★ Poor
Reynolds stress model (full SMC)	★★ Moderate–high	★ Poor: anticipate Nu errors of 25–100%	★★ Fair: may have incorrect location or magnitude
Shear stress transport (SST), hybrid method	★★★ Low–moderate	★★★ Good: typical Nu errors of 20–40%	★★ Fair
v^2f	★★★★ Moderate	★★★★★ Excellent: anticipate Nu errors of 2–30%	★★★★★ Excellent
DNS/LES time-variant models	★ Extremely high (DNS available for low Re only)	★★★★★ Good–Excellent	★★★★★ Good–Excellent

★: undesirable model characteristics
★★★★★: excellent model characteristics

Chapter 2: Background

The traditional method of using forced or natural convection across heat sinks do not provide enough surface cooling for modern power electronics. Advanced cooling options for single-phase techniques that have shown the highest surface heat transfer coefficients are microchannels and jet impingement. Jet impingement has been shown to remove hot spots without the use of a heat spreader while requiring less pumping power to achieve sufficient cooling performance [13], [14]. These inline jet arrays effectively created more homogeneous temperatures across the surface when compared to heat sink arrays. When the target surface is smaller than 0.07 m by 0.07 m, microchannels have better performance than jet arrays [15], [16]. The cooling technique chosen for this study was jet impingement due to the size of the heated surface.

2.1 Optimum Geometric Considerations for Jet Arrays

Variables which determine the effectiveness of jet impingement include the height of the nozzles from the heated surface, spacing between the jets (pitch), additions of microchannels on the impingement surface, and spent fluid management schemes including an expanding manifold angle. To optimize a jet array, focus was placed on the uniformity of the heat transfer and temperature on the cooled surface.

2.1.1 Jet Height

Metzger, et al [17] determined that a height of 1.0D for staggered and inline arrays produced the strongest uniformity in the Nusselt number on the surface when compared to larger heights. J. Lee [18] agreed that the best heat transfer performance occurred between the lowest tested height values of 1.5D and 3.0D for an inline array. Based on these studies along with Maddox [1], a height of 1.0D is used for all modeling. Based on these conclusions, a constant jet height of 1.0D for all geometries was used for this study.

2.1.2 Jet Spacing (Pitch)

The optimum spacing between jets in both inline and staggered arrays has been debated by various researchers. Maddox [1] concluded that a pitch of $4.0D$ resulted in an increased average Nusselt number for an inline array. A study on circular and square nozzle shapes in inline arrays done by Attalla [19] agreed that an optimum spacing of $4.0D$ occurred for circular jets that was independent of the height of jets for heights ranging from $2.0D$ to $8.0D$. For a staggered jet array of five nozzles, San and Lai [20] determined that the optimum spacing was $6.0D$ for Reynolds values of 10,000 and 20,000 based upon an optimized stagnation Nusselt number (the Nusselt number located directly beneath the center of a jet). In a later publication, San [21] agreed with the optimum spacing but concluded that this optimized pitch value would not be valid on larger arrays of staggered jets since the degradation of jets further downstream was neglected. A later study on a microjet array by Michna [22] determined that the optimum spacing for a staggered array was between $1.8D$ and $3.6D$. The increased number of jets in this study accounted for the degradation of the jets downstream and had entrainment effects with the jets that were next to the walls of the device. Wae-Hayee [23] observed that jet degradation increases for a staggered jet array more than an inline jet array and that the jets that are farther downstream degraded more than those that are farther upstream.

2.2 Alternate Geometric Enhancements

As jets degrade downstream, the heat transfer is reduced and the temperature on the surface increases. Improvements to traditional inline jet arrays have been proposed in order to create a more uniform temperature and surface heat transfer. Rattner [2] used fluid extraction ports to achieve lower surface temperatures and improved temperature uniformity but this design created additional complexity and added cost with the configuration of the fluid extraction ports and the

routing of the fluid. Adding air induced jets proved to slightly increase the heat transfer for pitches of 6D and 8D [3]. Adding helical inserts to create a swirling jet had better performance in heat transfer uniformity for $H > 6D$ across pitch values from 2D to 10D for a single Reynolds value [24].

2.3 Effect of Expanding Manifold Angle

Creating an angled manifold to reduce the degradation of downstream jets has shown an improvement on temperature and heat transfer uniformity for inline arrays with $Re > 5000$ while inline arrays with $Re < 5000$ showed very little improvement with an expanding manifold [1], [25]. For the $Re < 5000$, the jets do not generate enough turbulence to create the need for the expanding manifold since the downstream jets are not degraded significantly. Arens, et al. [25] found in one optimization that variable jet diameters decreased the amount of spent fluid downstream, allowing fountain regions to be formed but removing the issue of too much spent fluid degrading the stagnation regions of the downstream jets.

2.4 Objective of Current Study

This study investigates the effects of an expanding manifold on staggered arrays and the effects of the spacing of staggered arrays as a function of the angle of the confining wall. Staggered jet arrays are shown to have more degradation of the heat transfer coefficient in the stagnation region in downstream jets than that of inline arrays and require improved performance on a cost effective spent fluid management form [23]. This study investigates pitch values of 2.25D, 3.0D, 4.5D, and 6.0D across Reynolds values of 5600, 8400, 11200, and 14000. The numerical simulations showed the strong fountain region effects created by a staggered array and provided more detailed images of the benefits the fountain region effects have on the homogeneity of temperature and heat transfer on the plate. The expanding manifold is shown to increase the uniformity of temperature and the

heat transfer on the surface of the impingement plate for all investigated values of pitch and Reynolds number.

Chapter 3: Numerical Simulation Setup

The numerical simulations used for this study were developed using ANSYS® Fluent, Academic Research, Version 16.2. The available text user interface (TUI) using Python coding was used to reduce the development time of the geometries and fluent code. The SST turbulence model was used for the fluent analysis. A detailed set of instructions on the development of the geometry, mesh, and Fluent code can be found in Appendix A.

3.1 Geometry

An initial design of a staggered jet array geometry was developed using the graphical user interface (GUI) in ANSYS® Workbench 16.2. To reduce computation time, symmetry conditions were applied to both sides of the staggered jet array as shown in Figure 3.1.

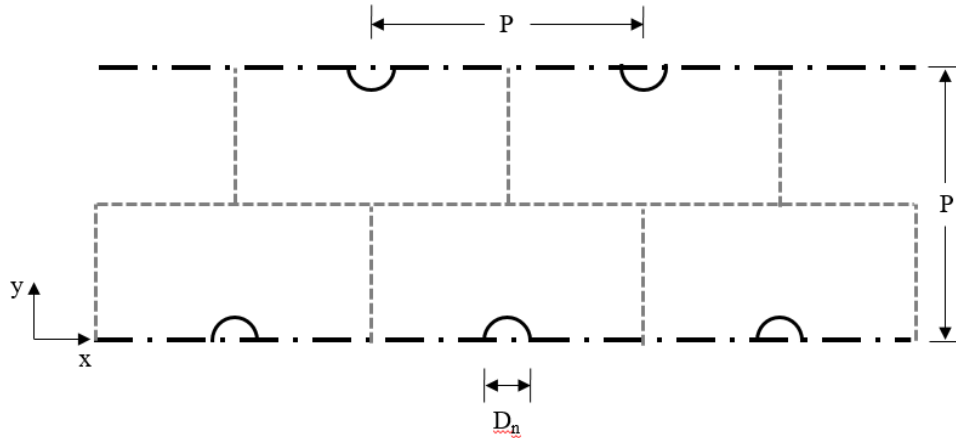


Figure 3.1 2-D Cross section drawing of geometry showing jet spacing and applied symmetry conditions

A plenum was included above the nozzle inlets to accurately depict the velocity profile within the nozzles since Maddox [1] found that imposing a velocity profile at the nozzle inlet resulted in an offset velocity profile within the nozzles. A copper block of 0.25 inch (0.03175 m) thickness, with a uniform heat flux imposed on the bottom of the block, was included below the impingement surface to account for the conductive heat spreading within the solid.

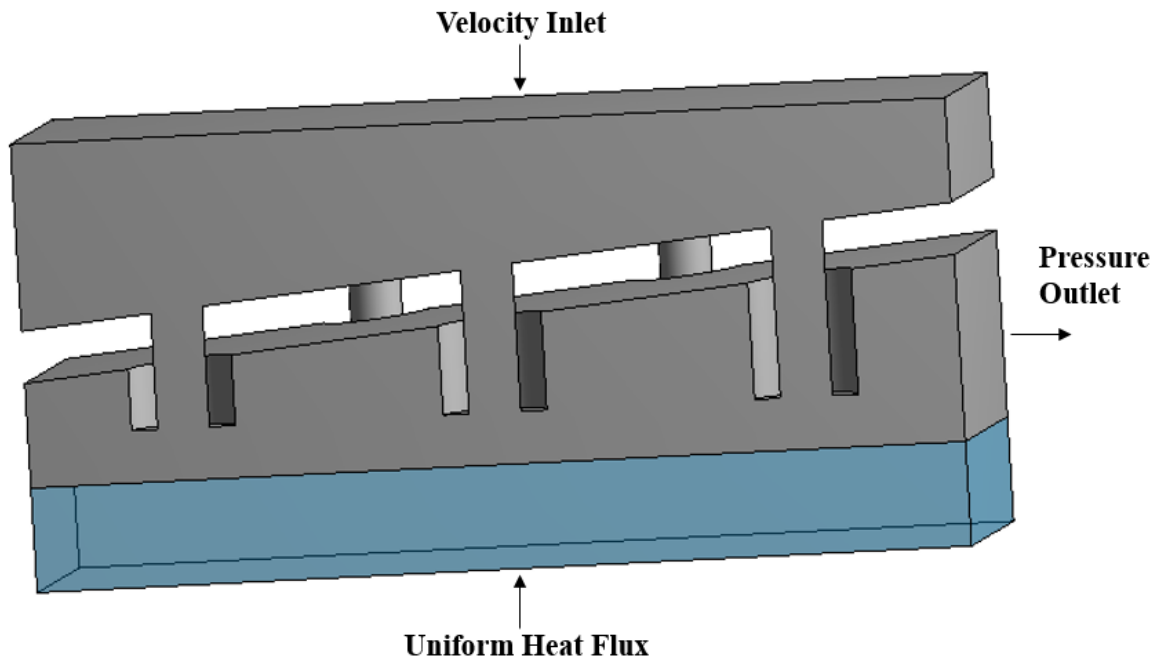


Figure 3.2: 3-D Geometry showing applied fluid and heat flow

3.2 Mesh

Each geometry was meshed using ANSYS® Workbench 16.2. The copper block used a coarse rectangular grid and was meshed independently of the fluid volume which used a fine mesh of tetrahedral elements. Each surface and block was named through the meshing GUI, which helped streamline the calculations using the TUI for Fluent. Inflation controls were used to refine the mesh near the impingent surface to resolve the boundary layer physics while reducing computation time by not utilizing a fine mesh throughout the volume as shown in Figure 3.3. A detailed description of the grid and inflation controls used throughout the mesh can be found in Appendix A.2.

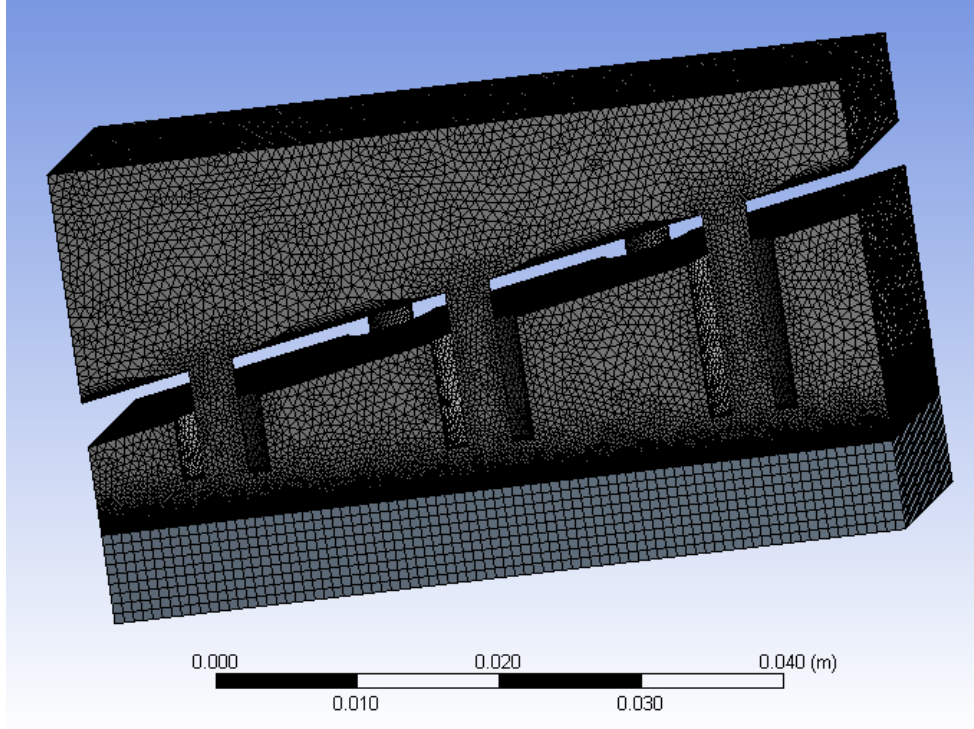


Figure 3.3 Completed Mesh for $\gamma = 10^\circ$ and $P^* = 6$

For jet impingement problems, Transition SST modeling requires a fine mesh along the stagnation and wall jet regions, which was the entire impingement surface for this model. The near wall grid must be fine enough so that the dimensionless wall distance of the first grid point is less than one,

$$y^+ = \frac{u_\tau y}{\nu} < 1 \quad (3.1)$$

where y is the distance to the nearest wall, ν is the kinematic viscosity, and u_τ is the shear velocity defined by:

$$u_\tau = \sqrt{\frac{\tau}{\rho}} \quad (3.2)$$

Since y^+ is dependent upon the cell Reynolds, it is computationally inefficient to predict the necessary grid spacing to maintain $y^+ < 1$ for impingement surfaces. To decrease computation time, a check for $y^+ < 1$ is used during the grid refinement analysis.

3.3 Fluent

ANSYS® Fluent was used to apply boundary conditions, set material properties, initialize the pressure and flow fields, select the turbulent model, and solve the resulting equations for the grid. Symmetry boundary conditions were applied to the sides that were normal to the y-axis of the geometry. This reduced the required computation time by reducing the size of the geometry and the resulting nodes that needed to be resolved within the mesh. A pressure outlet boundary condition was applied to the surface at the exit of the impingement region, a uniform heat flux boundary condition was applied to the bottom of the copper block, a no-slip conjugate heat transfer condition was applied at the interface between the fluid and the copper block, and all other surfaces were treated as an adiabatic wall. A uniform velocity inlet boundary condition was applied to the top of the plenum where the velocity was based on the jet Reynolds number and number of jets within the staggered array as shown in the MATLAB code in Appendix A.3.1.

For the initial grid independence study, the models were completed using the graphical user interface in Fluent. To reduce computation time for the other simulations in the study, the Samuel Ginn College of Engineering Virtual Symmetric Multiprocessing High Performance Computing Cluster (vSMP HPCC) which has no graphical user interfaces was used as shown in Appendix A.3.2. This study using the Transition SST model within ANSYS® Fluent as outlined in the next section.

3.3.1 Theory for Transition SST Model

The Transition SST model in ANSYS® Fluent expands upon the traditional SST $k-\omega$ transport model which is a hybrid turbulence model that uses the $k-\omega$ model near the wall and the $k-\varepsilon$ model in the far field [5]. The Transition SST model expands upon the traditional SST transport equations by implementing two additional equations to track the intermittency and transition onset

criteria using empirical correlations developed by Menter et al. [4]. The ANSYS® Fluent implementation of the Transition SST model is documented in the ANSYS® Academic Research, Release 16.2, Help System, Fluent Theory Guide is summarized in Appendix A.6.

3.3.2 Constants for Transition SST Model

The constants used in the transition SST model as defined by the ANSYS® Fluent Theory Guide [5] are defined below:

Table 3-1: Constants used for Transition SST model

Constant	Value for Transition SST Model
a_0	1/9
a_0^*	$\beta_i/3$
a_1	0.31
α_∞^*	1
$\beta_{i,1}$	0.075
$\beta_{i,2}$	0.0828
β_∞^*	0.09
C_{a1}	2
C_{a2}	0.06
C_{e1}	1
C_{e2}	50
C_{s1}	2
$c_{\theta t}$	0.03
$c_{\gamma 3}$	0.5
M_{t0}	0.25
$\sigma_{k,1}$	1.176
$\sigma_{k,2}$	1.0
$\sigma_{\omega,1}$	2.0
$\sigma_{\omega,2}$	1.168
$\sigma_{\theta t}$	2.0
σ_y	1.0
R_β	8
R_k	6
R_ω	2.95
ζ^*	1.5

3.4 Grid Independence

A grid independence study was conducted for a single geometry of a staggered jet array with an angle of 0° , a pitch of six jet diameters, a height of one jet diameter, and an average jet Reynolds number of 5,600. A coarse mesh was generated to solve the model then the mesh was refined and the model was solved again. The successive refinement of the mesh and model solution continued until grid independence was reached based on the heat flux and temperature gradient on the surface. Four comparisons were made on the impingement surface for grid independence: along the left side inline to the flow, along the left side inline to the flow, centerline of the geometry inline to the flow, and centerline of the geometry transverse to the flow. Due to the nature of the Transition SST model, an additional constraint of $y^+ < 1$ for the grid size on the surface as a grid independence check. The meshing parameters for the three finest meshes in the grid independence study are given in Table 3.1. For each of the meshes, the parameters were identical except for the sizing of the elements on the impingement surface since this is the area requiring the finest grid sizing to resolve the boundary layer conditions.

Table 3-2: Meshing parameters used for grid independence study

	Mesh ID		
	5.0e-05	3.5e-05	2.0e-05
Physics Preference	CFD	CFD	CFD
Solver Preference	Fluent	Fluent	Fluent
Relevance	99	99	99
Relevance Center	Fine	Fine	Fine
Smoothing	Medium	Medium	Medium
Transition	Slow	Slow	Slow
Span Angle Center	Fine	Fine	Fine
Minimum Edge Length (m)	1×10^{-6}	1×10^{-6}	1×10^{-6}
Maximum Inflation Layers	10	10	10
Inflation Growth Rate	1.2	1.2	1.2
Surface Element Sizing (m)	5.0×10^{-5}	3.5×10^{-5}	2.0×10^{-5}
Resulting Nodes	8.6×10^6	17.3×10^6	35.8×10^6

The solutions obtained using the Transition SST model with the meshes listed in Table 3.1 are shown in Figure 3.4. All of the rows indicate surface profiles with temperature difference, θ , in the top row, heat flux, q'' , in the center row, and y^+ in the bottom row. The left column shows the surface profile values in the streamwise direction of the flow along the centerline of the surface, the next column to the right shows the profiles where $y^* = 6$ along the center of the nozzles, the next column shows the profiles where $y^* = 0$ along the center of the nozzles, and the right column shows the profiles transverse to the direction of flow in the center of the modeled surface (starting underneath the center jet at $y^* = 0$). The dashed blue line represents the solution for the coarse grid, the red line represents the solution for the intermediate grid, and the green line represents the solution for the fine grid.

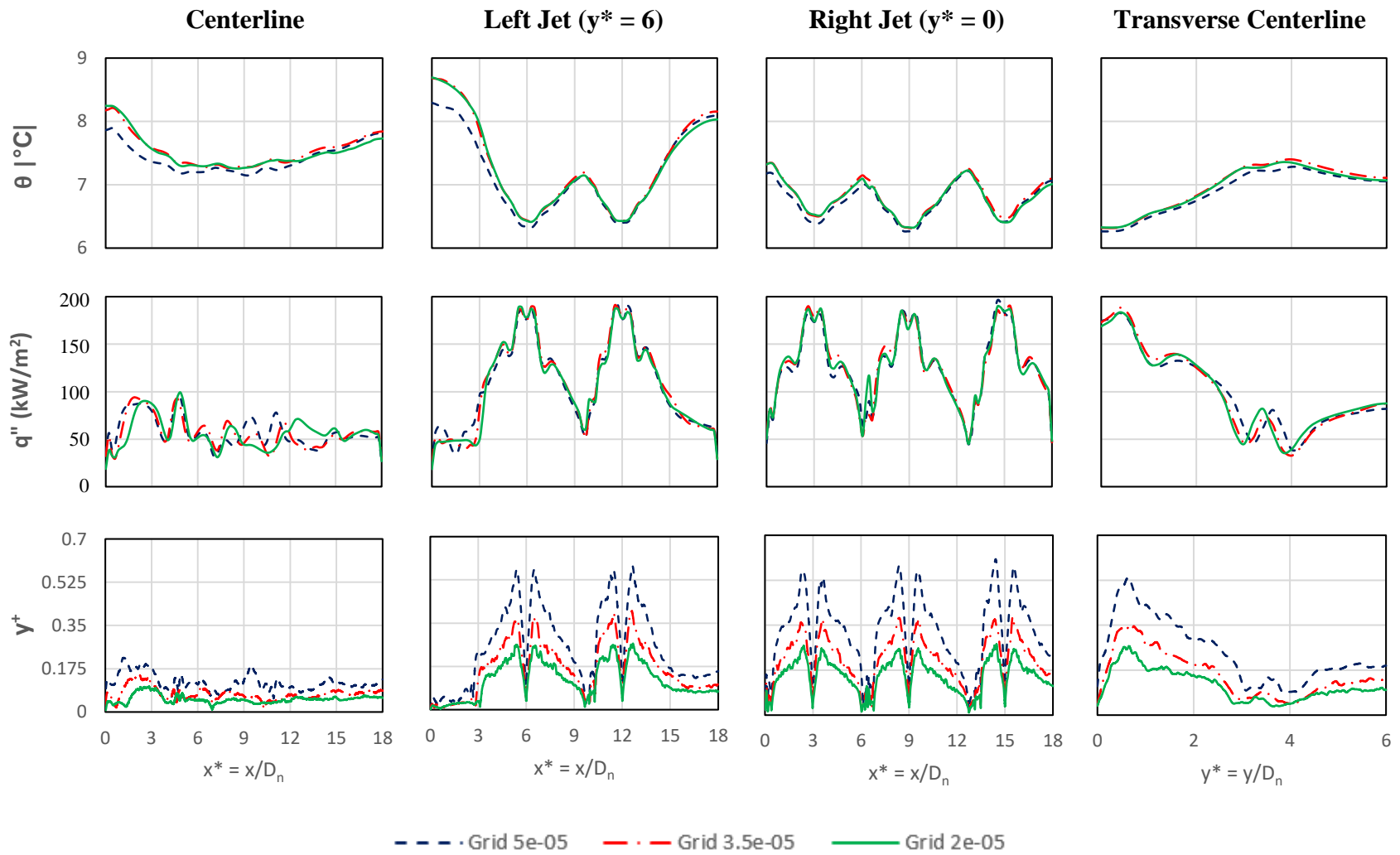


Figure 3.4: Surface profiles for grid independence study

The bottom row of graphs in Figure 3.4 shows the values of y^+ along the surface. Since all of the meshes maintain $y^+ < 1$ throughout the surface, all of the modeled meshes meet the minimum requirements for resolving the behavior in the boundary layers of the surface. The similarities in the profiles between all of the modeled meshes indicate comparable solutions especially with the less than 5% difference between the intermediate and fine grid sizes. The temperature and heat flux surface profiles result in a change of less than 0.5% of the average heat transfer coefficient between the intermediate and fine meshes. Therefore, the parameters used for the intermediate mesh with a surface element sizing of 3.5×10^{-5} m using the SST turbulence model were used for this study.

Chapter 4: Simulation Results

The final meshing parameters from the grid independence study were used to generate meshes with: expanding manifold angles of 0° , 5° , and 10° ; pitches of 2.25, 3, 4.5, and 6 jet diameters; and average jet Reynolds numbers of 5,600, 8,400, and 11,200.

To illustrate the flow patterns within the geometry, the velocity streamlines for a pitch of 4.5 jet diameters and an expanding manifold angle of 5° is shown in Figure 4.1. Since the simulation was solved for steady flow, the pathlines, streamlines, and streaklines coincided. The jet inlets are through the seven jets for the 4.5 pitch geometry with the highest velocities occurring within the jets farthest upstream. The fountain regions between the jets show the spent fluid being forced upwards and into the open area between the jets with the expanding manifold angle. The fluid decelerates as it moves further away from the jets with the largest mass of decelerated fluid

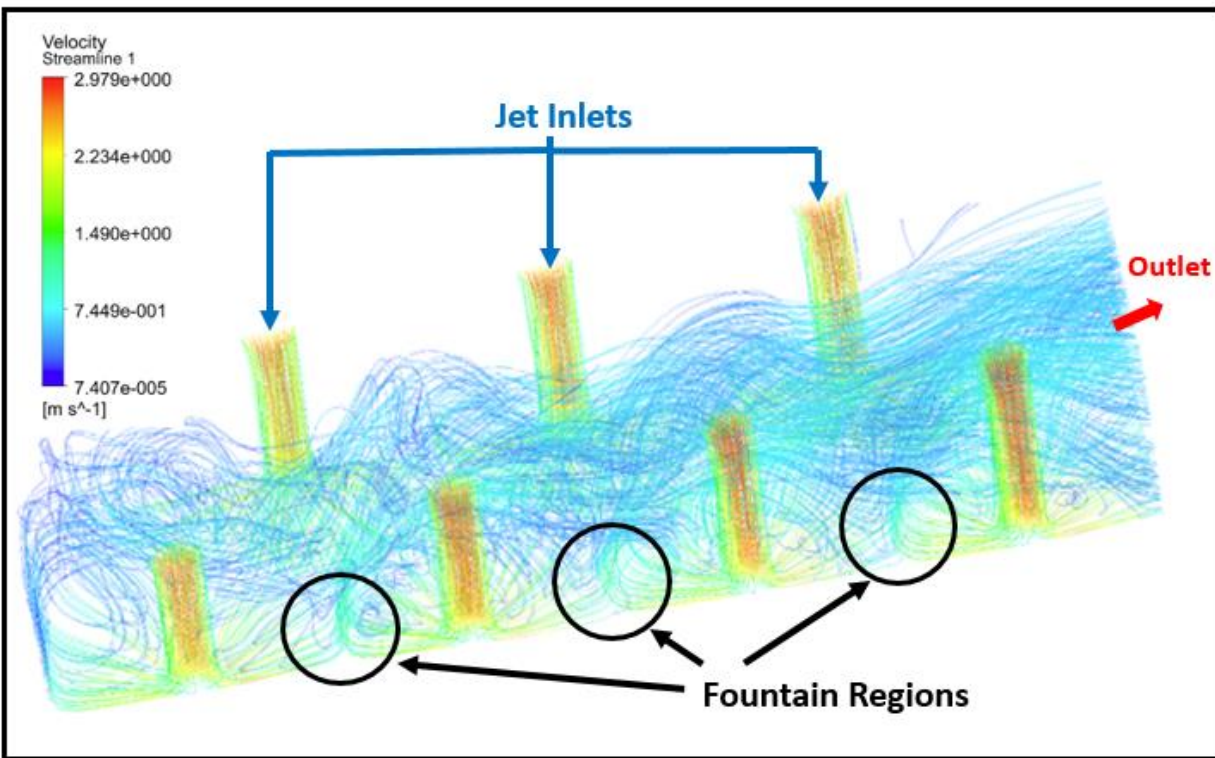


Figure 4.1 Velocity Streamlines for $P^* = 4.5$, $\gamma = 5^\circ$, and $Re = 8400$

concentrated in the volume farthest downstream created by the expanding manifold angle, where the outlet is located.

The number of jets within the geometry increased as the pitch decreased in order to compare the same length in the streamwise direction; thereby also comparing the same surface area for an infinite array of jets. These geometries utilized 5 jets for $P^* = 6$, 7 jets for $P^* = 4.5$, 9 jets for $P^* = 3$, and 11 jets for $P^* = 2.25$. As the pitch decreased, the geometry had a larger number of jets at the same average jet Reynolds number as for larger pitch values requiring an increased volume of fluid through the jets for the smaller pitch values. The increased volume of fluid through the jets results in higher surface heat transfer coefficients further downstream. In addition, the surface heat transfer coefficients increased as the jet Reynolds value increased.

4.1 Expanding Manifold Angle

By creating an angled outlet, the flow from the jets is encouraged to go to the lower pressure area in the center of the geometry between the two rows of jets instead of entraining within the high-pressure jet cores. Figure 4.2 shows the heat transfer coefficient going from a pitch of 2.25 jet diameters (a) to a pitch of 6 jet diameters (d) when the expanding manifold angle is not present in the geometry, i.e. where $\gamma = 0^\circ$. In this figure, the jets farthest downstream no longer have the expected circular shape in the stagnation region from the jet impinging on the surface due to the flow forcing the flow surrounding the jet core to be forced farther downstream. In addition, the high heat transfer that was seen in the fountain regions farthest upstream had dissipated downstream for all pitch values. These degradation effects were more prominent with decreased pitch values as the fountain regions entirely disappeared by a 2.25 jet diameter pitch. The geometry with a 2.25 pitch had the largest degradation effects due to the increase in fluid flow that was produced by the larger number of jets in the geometry.

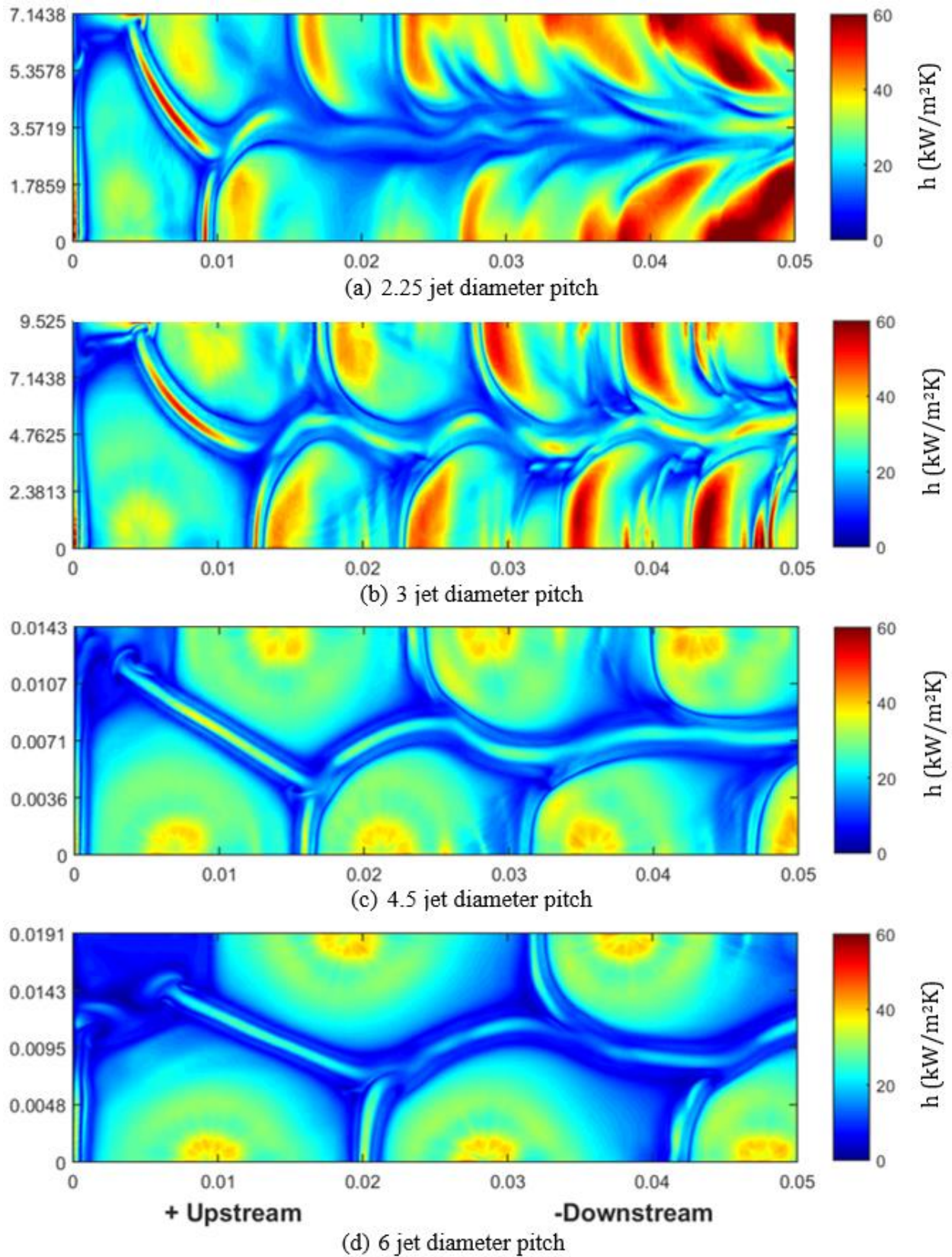
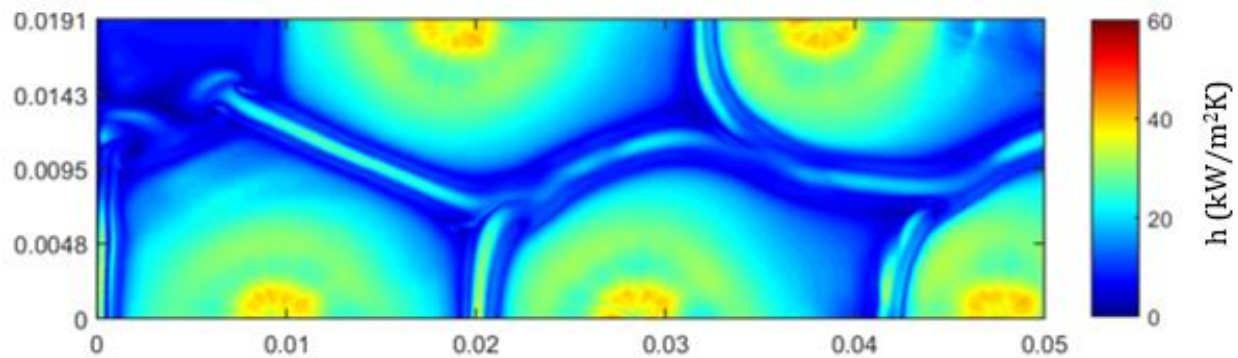


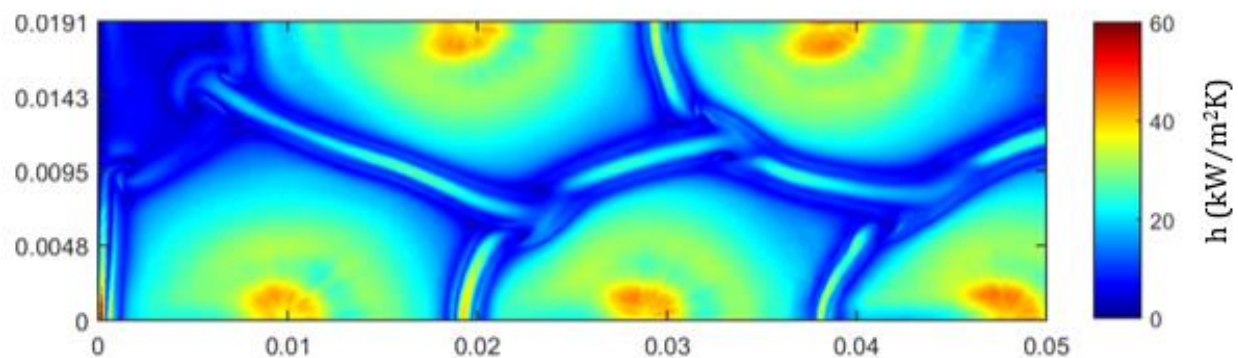
Figure 4.2: Surface heat transfer coefficient at $\gamma = 0^\circ$ and $Re_D = 11,200$

Increasing the expanding manifold angle results in decreased degradation in heat transfer coefficient in downstream jets as shown from Figure 4.3 to Figure 4.5 at a Reynolds value of 11,200. Figure 4.3 shows the results of increasing the expanding manifold angle from 0° to 10° while Figure 4.4 and Figure 4.5 show similar results for a pitch of 4.5 and 3, respectively. The stagnation regions are more rounded and have a larger area of effect at the expanding manifold angle of $\gamma = 10^\circ$ in Figure 4.5 resulting in higher average heat transfer coefficient than without an expanding manifold angle.

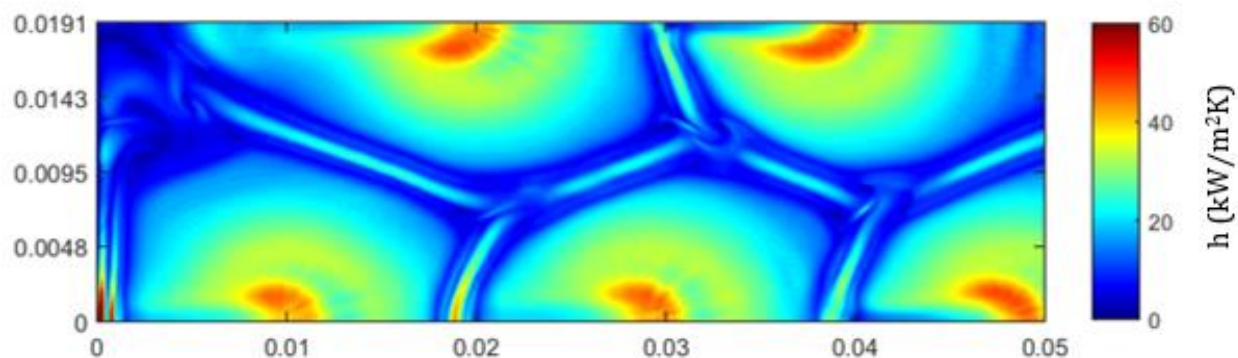
Fountain region effects were more pronounced between jets and have higher heat transfer coefficients as the expanding manifold angle increased for all pitch values. In Figure 4.3, the distinction between the fountain regions between an angle of 0° to 10° were not as distinct even for the highest tested Reynolds number due to the larger pitch of the jet array. As the pitch decreased in Figures 4.4 and 4.5, the fountain regions become more distinct from images (a) to (c) for both pitches. In Figure 4.4, the fountain regions have nearly dissipated in the farthest downstream jets in image (a) only reaching heat transfer coefficients of $20 \text{ kW/m}^2\text{K}$. With the expanding manifold angle at its highest value in Figure 4.4 (c), the fountain regions reach heat transfer coefficients of $56 \text{ kW/m}^2\text{K}$. In Figure 4.5, the fountain region along the center of the surface in the streamwise flow direction reach higher heat transfer coefficients and higher average heat transfer coefficients in Figure 4.5(c) than Figure 4.5(a) due to the expanding manifold angle. The distortions seen downstream in Figure 4.5(a) and (b) are due to the time-averaged turbulent model which does not depict the fountain regions as shifting which would occur in experiments. Similar trends were seen in lower Reynolds value that were modeled and shown in Appendix. As expected, the lower Reynolds values reduced the maximum and average heat transfer coefficients seen for all values of pitch and expanding manifold angle.



(a) $\gamma = 0^\circ$ confining wall

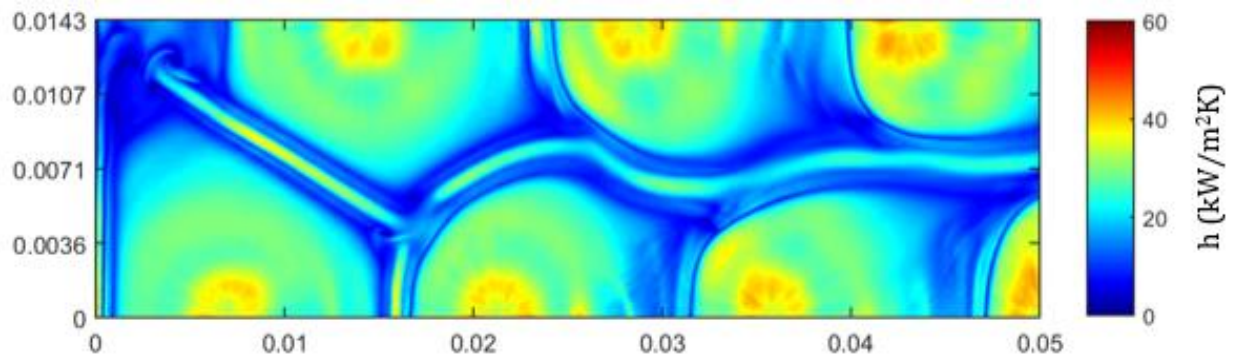


(b) $\gamma = 5^\circ$ confining wall

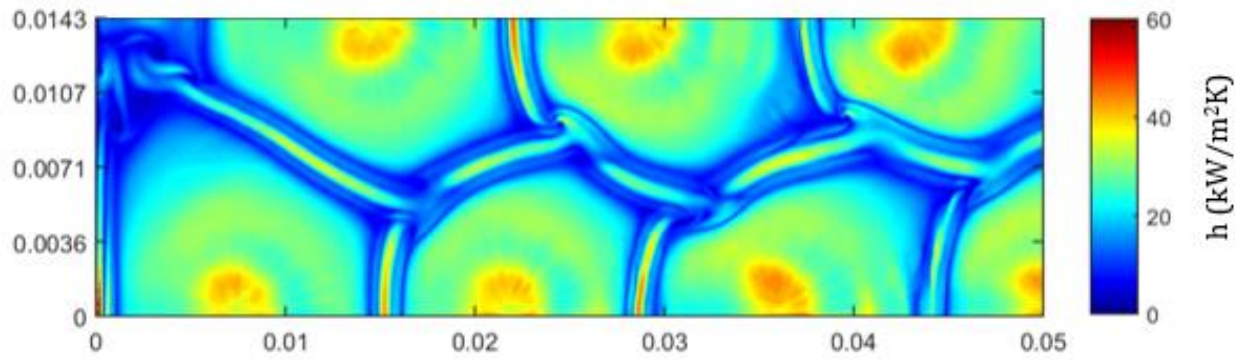


(c) $\gamma = 10^\circ$ confining wall

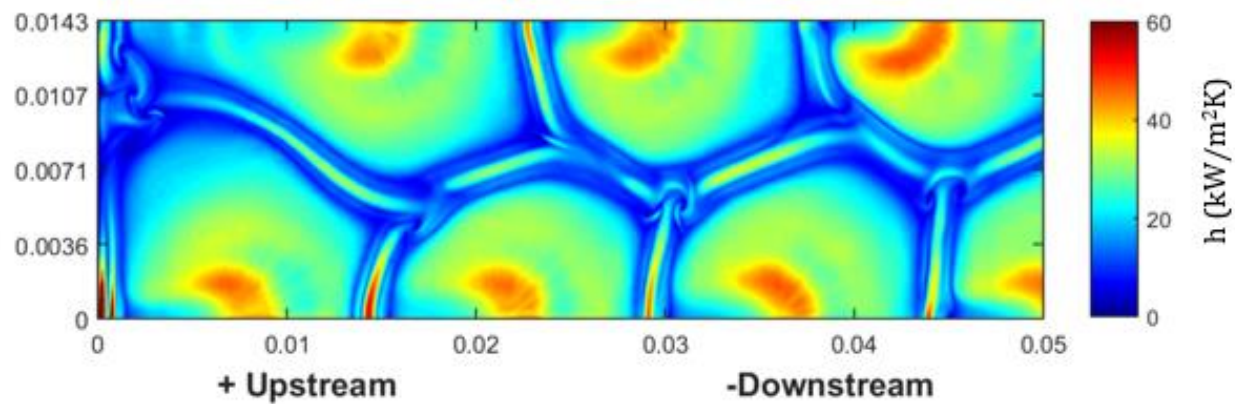
Figure 4.3: Surface heat transfer coefficient at $P^* = 6$ and $\text{ReD} = 11,200$



(a) $\gamma = 0^\circ$ confining wall

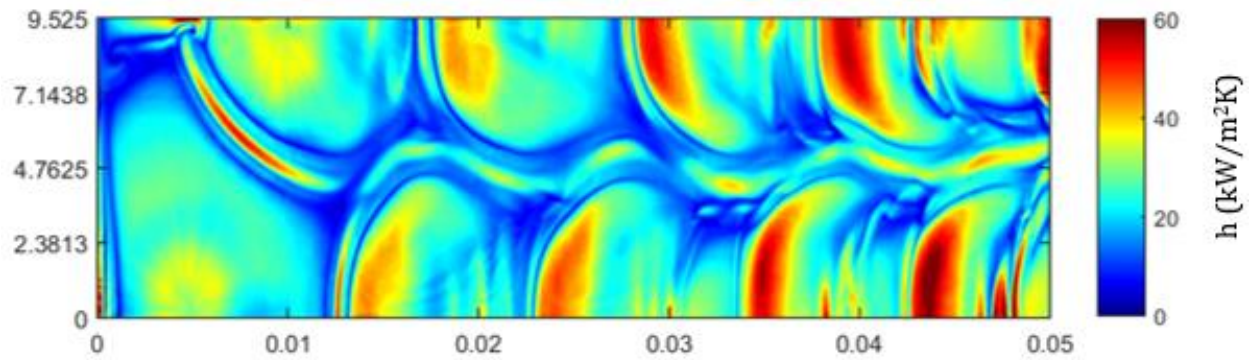


(b) $\gamma = 5^\circ$ confining wall

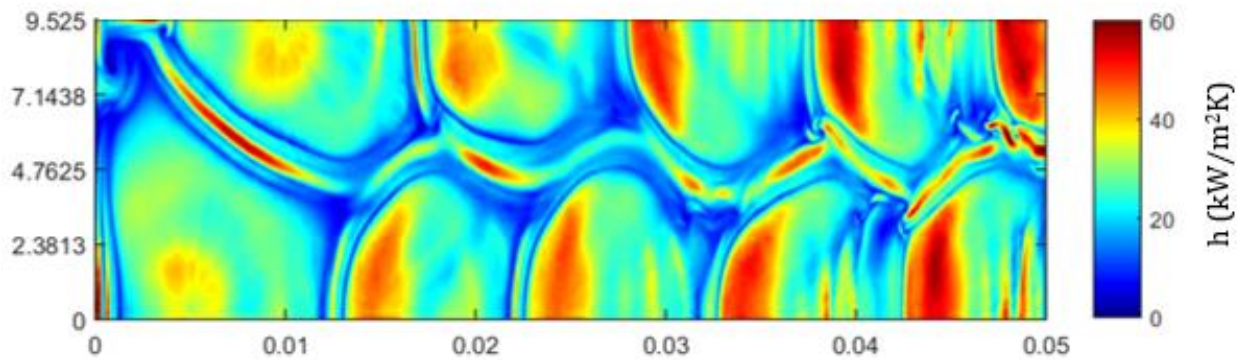


(c) $\gamma = 10^\circ$ confining wall

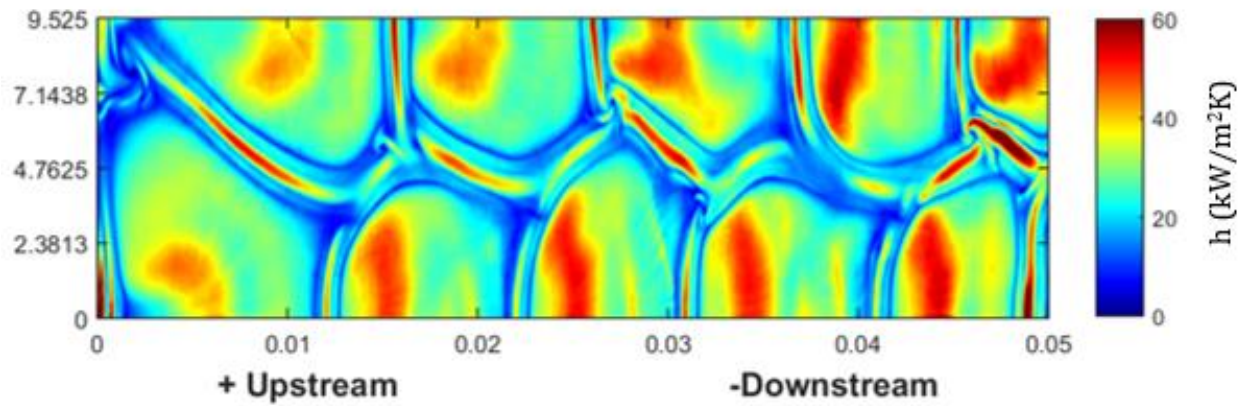
Figure 4.4: Surface heat transfer coefficient at $P^* = 4.5$ and $Re_D = 11,200$



(a) $\gamma = 0^\circ$ confining wall



(b) $\gamma = 5^\circ$ confining wall



(c) $\gamma = 10^\circ$ confining wall

Figure 4.5: Surface heat transfer coefficient at $P^* = 3$ and $Re_D = 11,200$

A comparison of the surface heat transfer coefficient in the jet farthest downstream, with the removal of the outlet effects, with 0° angle in the confining wall and with a pitch of 4.5 and a 10° angle in the confining wall is shown in Figure 4.6. The area of the stagnation region is larger in Figure 4.6(b) than in Figure 4.6(a) resulting in a higher average heat transfer coefficient. The stagnation region for the jet upstream of the jet in Figure 4.6(a) has devolved into the area surrounding the jet shown in Figure 4.6(a) due to the flow forcing the stagnation region to be pushed farther downstream. This degrades the heat transfer coefficient and the temperature of the surface beneath the jets further downstream. The fountain region is also more pronounced and has reaches a higher heat transfer coefficient for the higher expanding manifold angle geometry than for the geometry with an angle of 0° . The average heat transfer coefficient has increased from 22.5 $\text{kW/m}^2\text{K}$ in Figure 4.6(a) to 26 $\text{kW/m}^2\text{K}$ in Figure 4.6(b) which was a 15% increase in the average heat transfer coefficient solely due to the addition of an expanding manifold angle. Average heat transfer coefficients for the entire surface are shown in Figure 4.7.

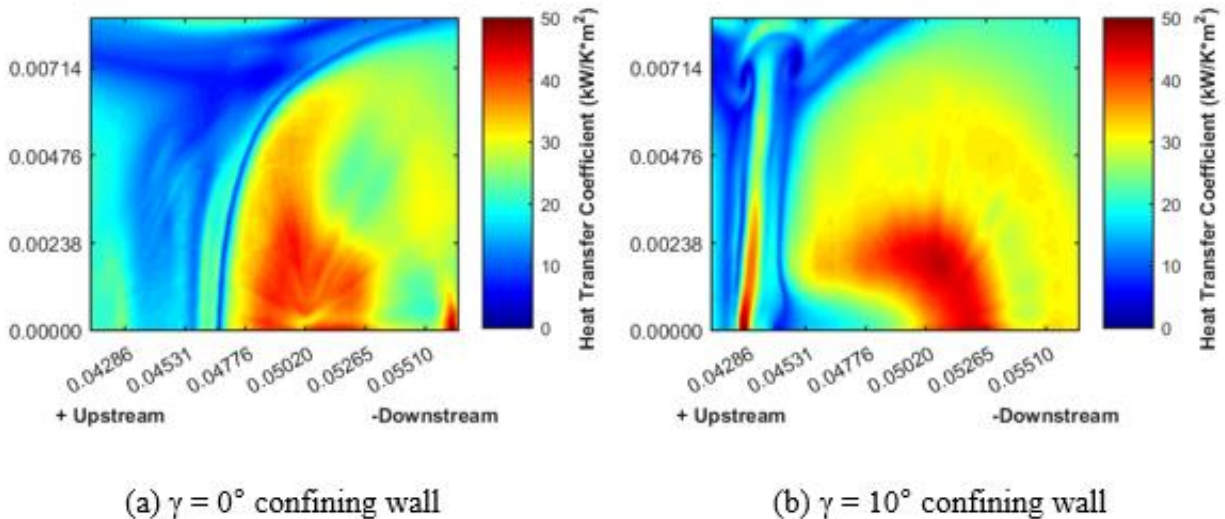
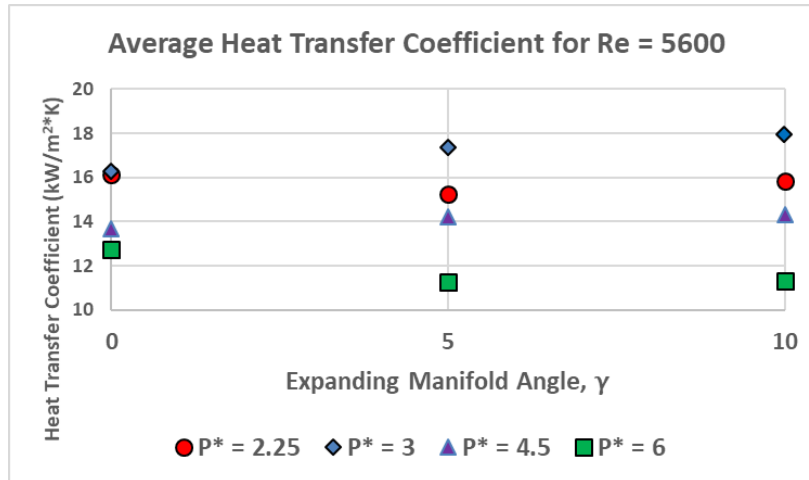
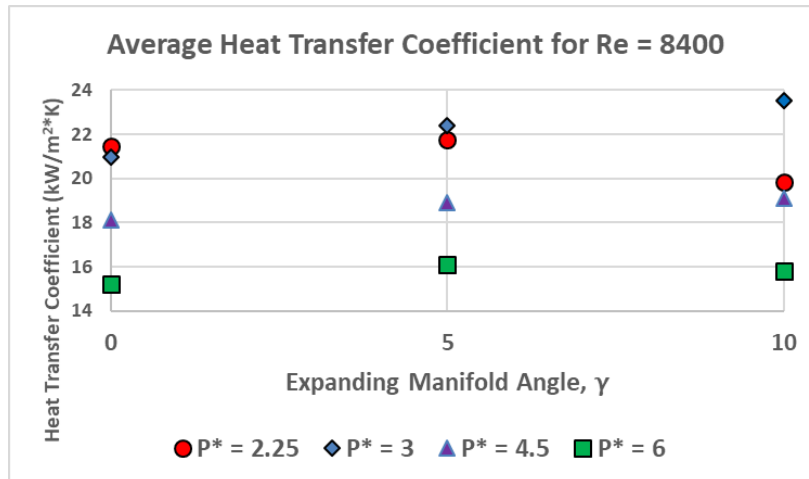


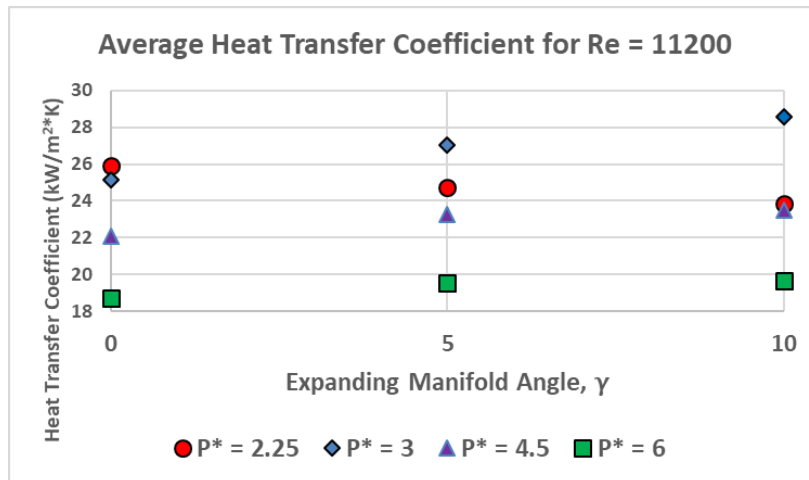
Figure 4.6: Surface heat transfer coefficient of individual jet farthest downstream for $P^* = 4.5$ and $ReD = 11,200$



(a) Average heat transfer coefficient for Re = 5600



(b) Average heat transfer coefficient for Re = 8400



(c) Average heat transfer coefficient for Re = 11200

Figure 4.7: Average heat transfer coefficients across varying Reynolds number based on expanding manifold angle

The increase in heat transfer coefficient between the fountain and stagnation regions results in decreased area-averaged surface temperatures. Figure 4.8 shows temperature contours between a 0° and 10° expanding manifold angle with different temperature scales. The surface with a 10° expanding manifold angle has a lower temperature scale than the 0° expanding manifold angle resulting in the highest temperature only reaching 303.7 K for the surface with $\gamma = 10^\circ$ which is a 17.8% improvement on temperature difference from the maximum temperature of 304.5 K for the surface with $\gamma = 0^\circ$. For the higher expanding manifold angle, the stagnation regions are more pronounced throughout the surface resulting in a lower average temperature. The stagnation regions upstream have reached cooler values for the higher expanding manifold angle and effect a larger region than for the $\gamma = 0^\circ$ surface. In addition, the stagnation region is more rounded for the surface with $\gamma = 10^\circ$ than the surface with $\gamma = 0^\circ$ and has less of a shift downstream. Overall, the surface with $\gamma = 10^\circ$ has a lower average temperature than the surface with $\gamma = 0^\circ$ due to the jets no longer entraining in flow downstream, reducing the degradation of heat transfer at the surface.

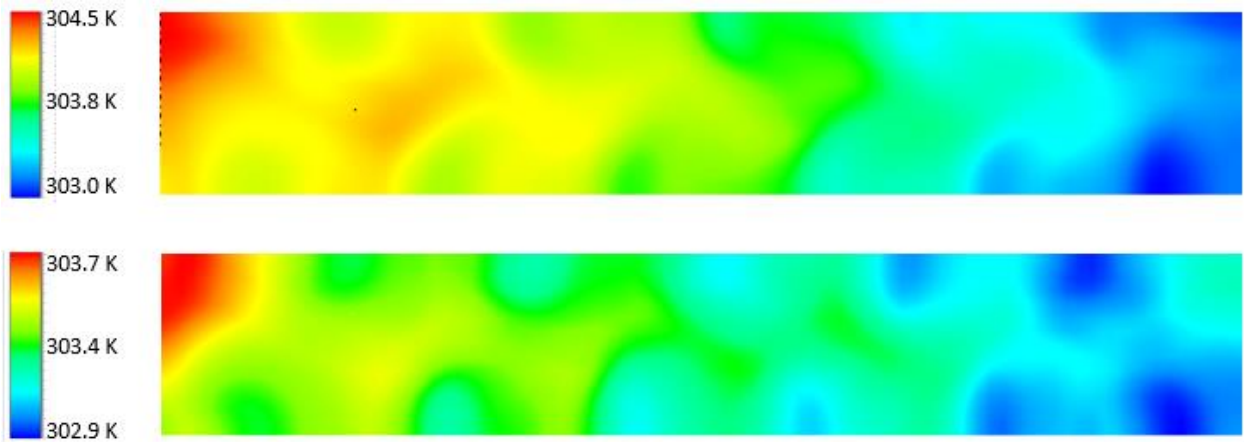
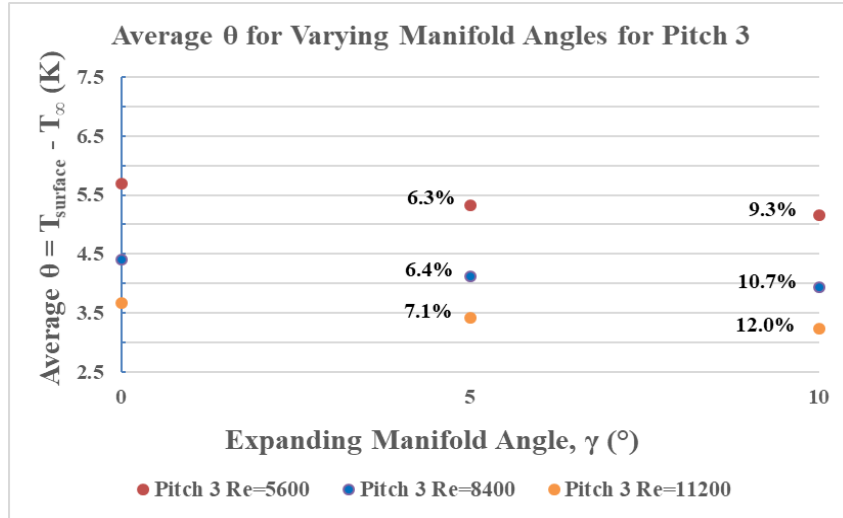


Figure 4.8: Temperature Contours at $P^* = 3$, $Re = 11200$ for (Top) $\gamma = 0^\circ$ and (Bottom) $\gamma = 10^\circ$

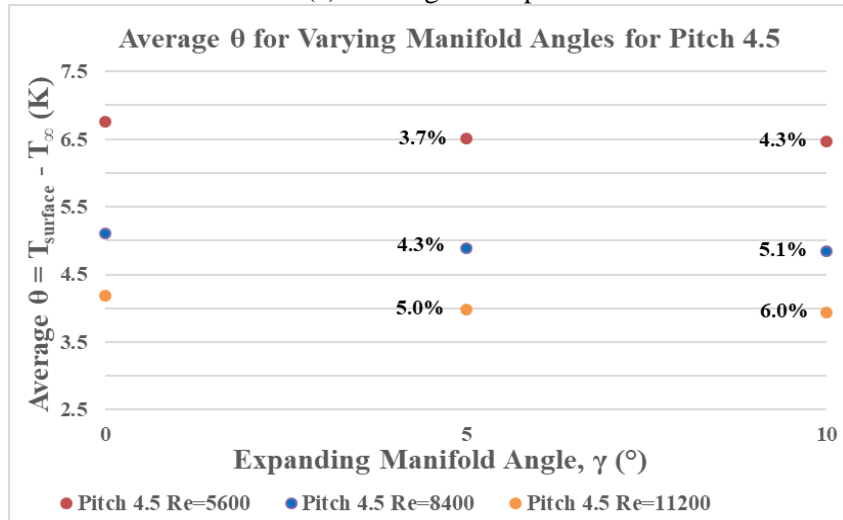
The average temperature differences for pitches of 4.5 and 3 with respect to the expanding manifold angle are shown in Figure 4.9. Average surface temperature differences dropped by 9.3% to 14.8% for all Reynolds values for a pitch of 3 when comparing the highest modeled expanding manifold angle of 10° to the geometry without an expanding manifold angle. The percent difference of area-averaged surface θ from $\gamma = 0^\circ$ to a higher value of γ is defined by Equation 4.1.

$$\% Diff = \frac{\theta_\gamma - \theta_{\gamma=0^\circ}}{\theta_{\gamma=0^\circ}}, \quad (4.1)$$

As pitch decreases from 4.5 to 3, the percent difference increased due to the increased amount of fluid from the larger number of nozzles in the pitch 3 geometry. This implies that with larger amounts of fluid to manage, the angled manifold had a greater effect on temperature uniformity. This could also be implied from the increased effect of expanding manifold angle seen on the average heat transfer coefficient up to a pitch of 2.25 as shown in Figure 4.7. The larger amount of fluid contributed to the effectiveness of the expanding manifold angle since the increased expanding manifold angle had a greater effect at higher Reynolds values than at lower Reynolds values shown between Figure 4.7(a) to Figure 4.7(c). Therefore, the improved heat transfer coefficient and temperature values from the use of the expanding manifold angle will decrease for lower flow rates and higher pitch values as seen by Arens [25]. The temperature also decreased as pitch decreased up to a pitch value of 2.25 as seen in section 4.2.



(a) Average θ for pitch 3



(b) Average θ for pitch 4.5

Figure 4.9: Average θ varying expanding manifold angle

4.2 Pitch

Staggered jet arrays are significantly affected by the distance between the jets, which was defined as the horizontal and vertical distance between jets. As the pitch decreases, there is a larger number of jets in a smaller area, causing jet degradation for downstream jets to occur more rapidly for smaller pitch values than for larger pitch values. Without an expanding manifold angle, downstream jet degradation increased as pitch decreased for all simulated Reynolds values as seen in Figure 4.2. The effects of pitch at the highest modeled manifold angle of 10° are shown from a pitch of 2.25 in Figure 4.9(a) to a pitch of 6 in Figure 4.9(d).

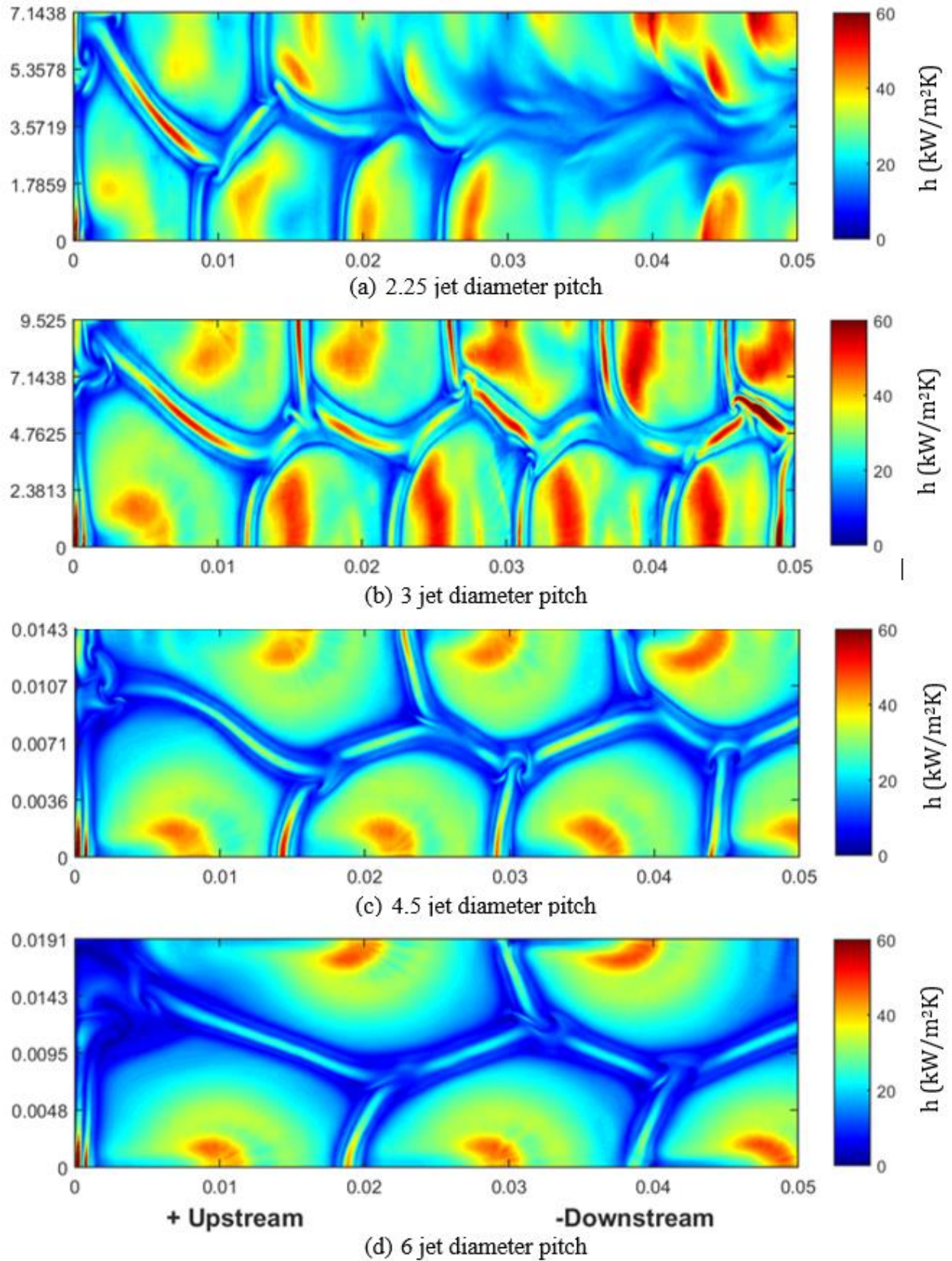


Figure 4.10: Surface heat transfer coefficient at $\gamma = 10^\circ$ and $Re_D = 11,200$

Decrease in pitch resulted in increased heat transfer coefficients in fountain regions as well as increased average surface heat transfer coefficients. Figure 4.10 shows the effect of pitch on the minimum, maximum, and average temperature differences over the surface where the temperature difference is defined in Equation 4.2 below:

$$\theta_{avg} = T_{surf,avg} - T_{\infty}, \quad (4.2)$$

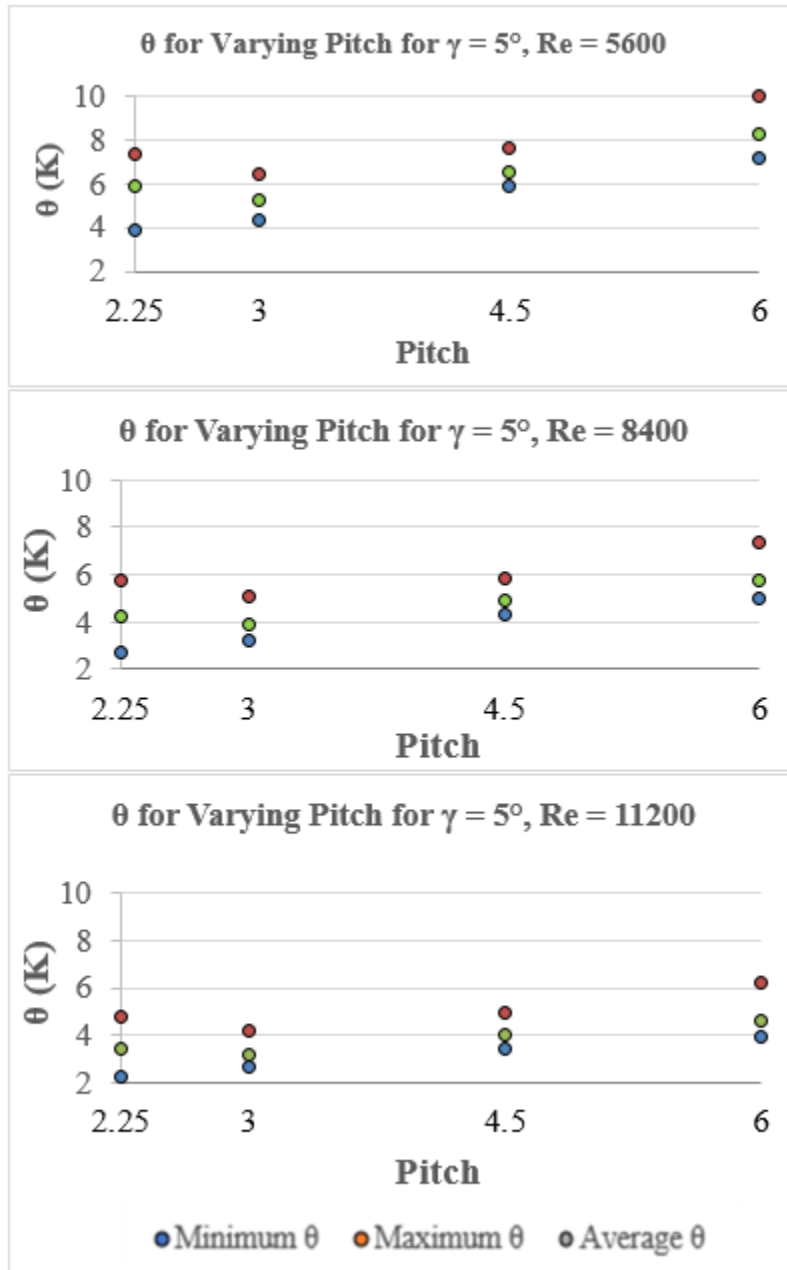


Figure 4.11: Effect of pitch on temperature homogeneity

Using the expanding manifold angle with the smaller pitch values resulted in the highest obtainable average heat transfer coefficients, highest heat transfer coefficients in the fountain regions, largest stagnation regions effects, and increase in highest heat transfer coefficient within the stagnation region. A decrease in pitch resulted in increased uniformity of temperature and heat transfer coefficient when used with expanding manifold angle that was independent of Reynolds number. The maximum surface temperature difference decreased by 11.4% - 15.3% for a pitch of 3 for all Reynolds values at the highest expanding manifold angle. Since a pitch of 3 had shown the highest average heat transfer coefficients, an expanding manifold angle of 15° was tested but showed less than 0.5% improvement on temperature difference and surface heat transfer coefficient when compared to the values for an expanding manifold angle of 10°.

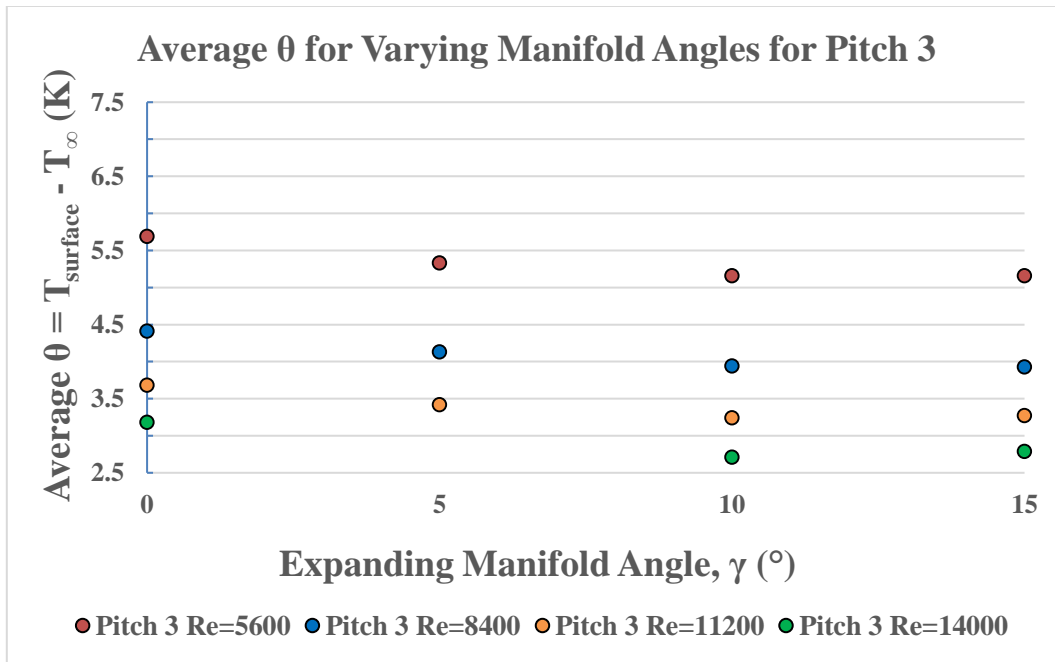


Figure 4.12: Average θ for $P^* = 3$ varying expanding manifold angle

4.3 Comparison to Experiment

An experimental study into jet impingement for a staggered jet array with an expanding area manifold was conducted concurrently but separately from this numerical study by Henry [26]. The model employs two symmetric boundary conditions implying an infinite array of staggered jets in the transverse direction with no wall effects while the experimental array is confined to a staggered array of seven jets with walls that are close enough to cause jet flow to be altered by hitting the side walls. The resolution of the experimental model only contains 36 data points on the surface compared to millions of surface points used in the numerical model. This results in the experimental setup being unable to identify the fountain region effects between the jets resulting in lower average heat transfer coefficients and higher temperatures when compared to the numerical simulation as can be seen in Figure 4.12 below.

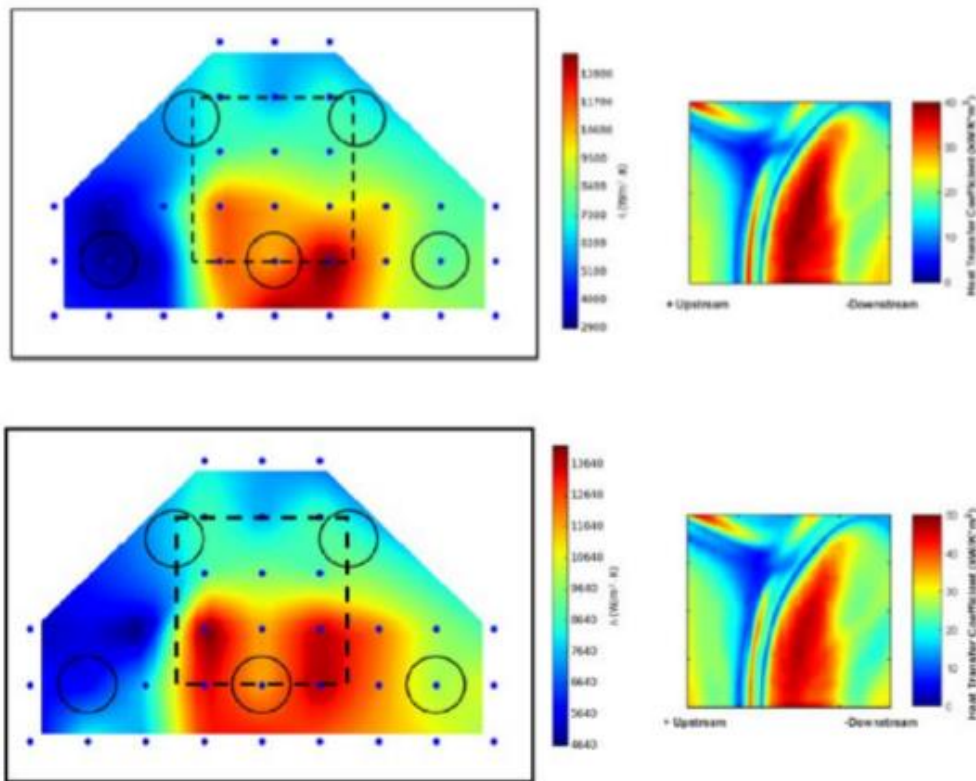


Figure 4.13: Comparison of experimental (left) and numerical (right) surface heat transfer coefficients at $Re = 8400$ (top) and $Re = 11200$ (bottom) for $P^* = 3$ [26].

The dotted line in the experimental images delineates the approximate location of the area represented in the numerical plots depicted in Figure 4.12. The numerical uses a constant value for the kinematic viscosity of water; therefore the average jet Reynolds value is based upon this value, which accounts for some of the differences between the numerical and experimental results. As seen by Maddox [1], the average surface heat transfer coefficient and temperature rises are roughly 2-3 times larger in the numerical simulation than in the experimental setup resulting from the Transition SST modeling limitations and the limited data points within the experiment. Both the experiment and numerical results show trends suggesting thermal improvements with increasing Reynolds numbers, increasing expanding manifold angle, and decreasing pitch seen in Figure 4.13.

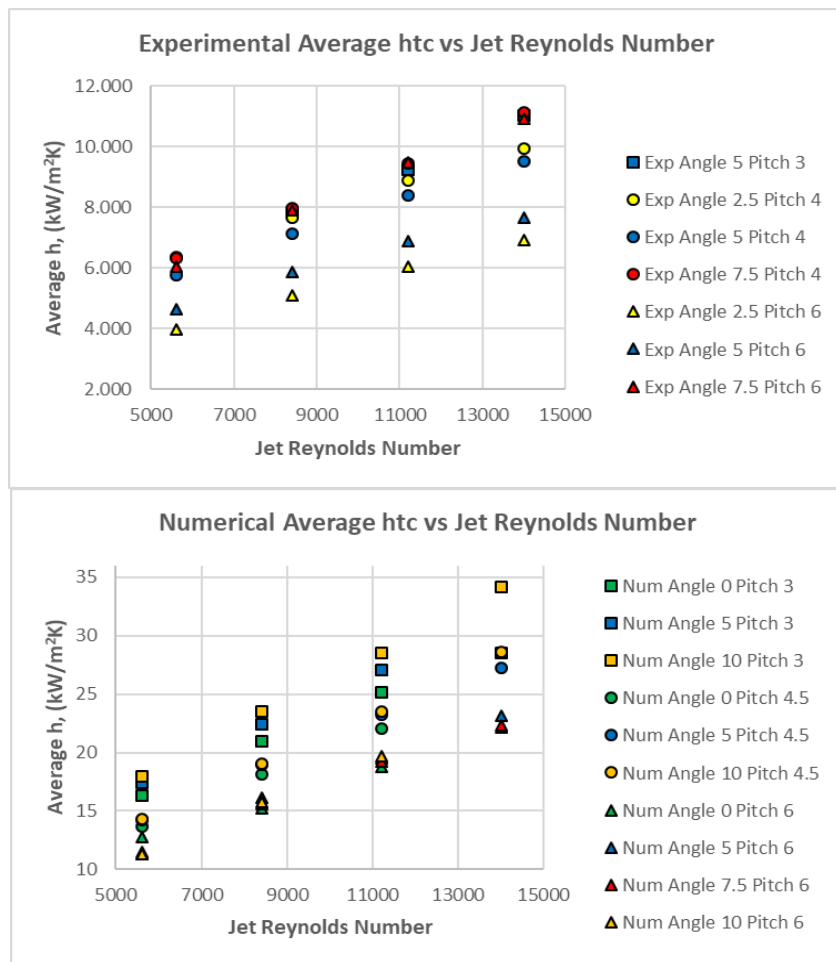


Figure 4.14: Experimental (top) and Numerical (bottom) comparison of heat transfer coefficient based on pitch, angle, and jet Reynolds number

Chapter 5: Conclusions

Liquid jet impingement can operate at relatively high volumetric flow rates with low pressure drops to create more homogeneous temperatures across a surface than traditionally passive cooling techniques. A disadvantage to using liquid jet impingement is the spent fluid from upstream jets entraining in the flow of downstream jets causing degradation in the heat transfer coefficient on the surface resulting in temperature non-uniformity. This study used an expanding manifold angle to manage the spent fluid crossflow in order to alleviate the degradation of downstream jets.

A numerical model was implemented to determine the effects of the expanding manifold angle, jet pitch, and Reynolds number on the surface temperature and heat transfer coefficient for a staggered jet array with water as the working fluid. The Transition SST model in ANSYS® Fluent was selected for the study due to the reduced computational cost and its reported ability to predict surface heat transfer coefficients within 20% of experimental studies. The mesh was refined to ensure that the solution was grid independent while maintaining the requirement of $y^+ < 1$ across the grid points on the surface to resolve the boundary conditions. The resulting mesh had approximately 17.3 million nodes when modeled with a jet Reynolds number of 5,600 for five nozzles.

This study simulated models varying between 5 and 11 jets based on the spacing between the jets with symmetric boundary conditions that implied an infinite array of jets to reduce the computation time. Observed trends included heat transfer increasing both with increasing manifold angle and with decreasing pitch up to a pitch of 2.25. The highest average surface heat transfer coefficient of 34.1 kW/m²K and lowest average temperature rise of 2.7 K were observed for a pitch of 3, angle of 10°, and jet Reynolds number of 14,000. The pitch of 3 had the highest average

surface heat transfer coefficients for all modeled Reynolds values and expanding manifold angles occurring within the fountain regions between the jets.

The expected increase in thermal performance with increasing manifold angle was confirmed by this numerical study. This study showed improvement on the surface heat transfer coefficient and homogeneity in the surface temperature down to a pitch of 2.25 for all tested Reynolds values concluding that the expanding manifold angle is an effective method for spent flow management. The effectiveness of the expanding manifold angle is dependent upon the volume of fluid reaching the heated surface or the jet Reynolds number and the spacing between the jets.

5.1 Suggestions for future work

Since the Transition SST model used for the numerical simulation has only shown predictions of heat transfer coefficients to within 20% of experiments, modifications to the numerical model could be made by adjusting constants within the model. Future studies would need to adjust constants so as not to match a single jet impingement case to ensure that the constants can match multiple jet impingement situations. For future simulations, the constant pressure outlet should be placed farther in the streamwise direction to allow laminar flow to begin to take effect and reduce simulation errors at the outlet. Other proposed modifications that could improve thermal performance include:

- Rounded/curved jet inlets to reduce the pressure drop at the nozzle inlet
- Modeling higher angles of expanding manifold angles
- Modeling a larger number of expanding manifold angles for optimization
- Optimization of angle based on Reynolds number
- Optimization of angle based on pitch
- Varying nozzle inlet diameters

Modeling varying nozzle internal diameters at same jet Reynolds values would allow future studies to determine if the thermal trends seen in this study are dependent upon the nozzle diameter. The current study has not determined optimum values for the expanding manifold angle or spacing between jets that are dependent upon the jet Reynolds number. Correlations between the jet Reynolds number, fluid Prandtl number, angle of expanding manifold, and jet spacing could be developed to determine improved experimental setups by modeling a larger range of pitch, angles, and Reynolds values.

Chapter 6: References

- [1] J. F. Maddox, "Liquid jet impingement with spent flow management for power electronics cooling," Auburn University, Auburn, 2015.
- [2] A. S. Rattner, "General characterization of jet impingement array heat sinks with interspersed fluid extraction ports for uniform heat-flux cooling," *Journal of Heat Transfer*, vol. 139, 2017.
- [3] K. Yeranee, M. Wae-Hayee, I. Piya, Y. Rao and C. Nuntadusit, "The study of flow and heat transfer characteristics of impinging jet array mounting air-induced duct," *IOP: Conf. Series: Materials and Science and Engineering*, vol. 243, 2017.
- [4] F. R. Menter, R. B. Langtry, S. R. Likki, Y. B. Suzen, P. B. Huang and S. Volker, "A correlation based transition model using local variables Part 1 - Model Formulation," *Journal of Turbomachinery*, vol. 128, no. 3, pp. 413-422, 2006.
- [5] *ANSYS Academic Research Mechanical, Release 16.2, Help System*, ANSYS Inc.
- [6] K. Heyerichs and A. Pollard, "Heat transfer in separated and impinging turbulent flows," *International Journal of Heat and Mass Transfer*, vol. 39, no. 12, pp. 2385-2400, 1996.
- [7] T. Craft, T. Launder and K. Suga, "Development and application of a cubic eddy-viscosity model of turbulence," *International Journal of Heat and Fluid Flow*, vol. 17, no. 2, pp. 108-115, 1996.

- [8] N. Zuckerman and N. Lior, "Impingement heat transfer: correlations and numerical modeling," *Journal of Heat Transfer*, vol. 127, no. 5, pp. 544-552, 2005.
- [9] F. R. Menter, "Zonal two equation k-w turbulence models for aerodynamic flows," in *The 24th Fluid Dynamics Conference*, 1993.
- [10] T. Esch and F. Menter, "Heat transfer predictions based on two-equation turbulence models with advanced wall treatment," *Turbulence, Heat, and Mass Transfer*, vol. 4, pp. 633-640, 2003.
- [11] T. Craft, L. Graham and B. E. Launder, "Impinging jet studies for turbulent model assessmentii. An examination of the performance of four turbulence models.," *International Journal of Heat and Mass Transfer*, vol. 36, no. 10, pp. 2685-2697, 1993.
- [12] Y. Shi, M. Ray and A. Mujumdar, "Computational study of impingement heat transfer under a turbulent slot jet," *Industrial & Engineering Chemistry Research*, vol. 41, no. 18, pp. 4643-4651, 2002.
- [13] J. Jorg, S. Taraborrelli, G. Sarriegui, R. Kneer, R. W. De Doncker and W. Rohlf, "Direct Single Impinging Jet Cooling of a MOSFET Power Electronics Module," *IEEE Transactions on Power Electronics*, 2017.
- [14] J. Jorg, S. Taraborrelli, E. Sabelberg, R. Kneer, R. De Doncker and W. Rohlf, "Hot Spot Removal in Power Electronics by means of direct liquid cooling," *16th IEEE Itherm Conference*, 2017.

- [15] D.-Y. Lee and K. Vafai, "Comparative analysis of jet impingement and microchannel cooling for high heat flux applications," *International Journal of Heat and Mass Transfer*, vol. 42, no. 9, pp. 1555-1568, 1999.
- [16] A. J. Robinson, "A thermal-hydraulic comparison of liquid microchannel and impinging liquid jet array heat sinks for high-power electronics cooling," *Components and Packaging Technologies, IEEE Transactions on*, vol. 32, no. 2, pp. 347-357, 2009.
- [17] D. E. Metzger, L. W. Florschuetz, D. I. Takeuchi, R. D. Behee and R. A. Berry, "Heat Transfer characteristics for inline and staggered arrays of circular jets with crossflow of spent air," *ASME Trans Journal of Heat Transfer*, vol. 101, pp. 526-531, 1979.
- [18] J. Lee and et al, "Crossflows from jet array impingement cooling: hold spacing, target plate distance, Reynolds number effects," *International Journal of Thermal Sciences*, vol. 88, pp. 7-18, 2015.
- [19] M. Attalla, H. M. Maghrabie, A. Qayyum, A. G. Al-Hasnawi and E. Specht, "Influence of the nozzle shape on heat transfer uniformity for in-line array of impinging air jets," *Applied Thermal Engineering*, vol. 120, pp. 160-169, 2017.
- [20] J. Y. San and M. D. Lai, "Optimum jet-to-jet spacing of heat transfer for staggered arrays of impinging air jets," *International Journal of Heat and Mass Transfer*, vol. 44, pp. 3997-4007, 2001.
- [21] J. Y. San, Y. M. Tsou and Z. C. Chen, "Impingement heat transfer of staggered arrays of air jets confined to a channel," *International Journal of Heat and Mass Transfer*, vol. 50, pp. 3718-3728, 2007.

- [22] G. J. Michna, E. A. Browne, Y. Peles and M. K. Jensen, "The effect of area ratio on microjet array heat transfer," *International Journal of Heat and Mass Transfer*, vol. 54, pp. 1782-1790, 2011.
- [23] M. Wae-Hayee, P. Tekasakul and C. Nuntadusit, "Influence of nozzle arrangement on flow and heat transfer characteristics of arrays of circular impinging jets," *Songklanakarin Journal of Science and Technology*, vol. 35, pp. 203-212, 2013.
- [24] A. Ianiro and G. Cardone, "Heat transfer rate and uniformity in multichannel swirling impinging jets," *Applied Thermal Engineering*, vol. 49, pp. 89-98, 2012.
- [25] N. R. Arens, M. P. Morem, J. Doom and G. J. Michna, "Reducing crossflow effects in arrays of impinging jets," *ASME Summer Heat Transfer Conference*, 2017.
- [26] M. A. Henry, K. E. Reid, S. H. Bhavnani, R. W. Knight, J. F. Maddox and W. D. Brannon, "Staggered and in-line submerged jet arrays for power electronics using variable discharge manifolds - Part I - Experimental," *17th IEEE Itherm Conference*, no. P123, 2018.
- [27] R. B. Langtry and F. R. Menter, "Correlation-based transition modeling for unstructured parallelized computational fluid dynamics codes," *AIAA Journal*, vol. 47, no. 12, pp. 2894-2906, 2009.
- [28] K. E. Reid, M. A. Henry, R. W. Knight, S. H. Bhavnani and J. F. Maddox, "Staggered and in-line submerged liquid jet arrays for power electronics using variable discharge manifolds: Part II - Numerical," *17th IEEE Itherm Conference*, no. P192, 2018.

Appendices

Appendix A: Simulation Setup Procedure using ANSYS Fluent 16.2 and Auburn HPCC

Appendix A.1 Build Geometry

```
geom_id = 'A5_P3_H1_Ho2_Hpo2_NR6_NL5'

#### Begin Geometry ####

try:
    # Connect to an existing geometry system
    geom_system = GetSystem(Name="Geom")
except:
    ## Create a geometry system
    geom_template = GetTemplate(TemplateName="Geometry")
    geom_system = geom_template.CreateSystem()

geom_geom_container = geom_system.GetContainer(ComponentName="Geometry")
geom_system.DisplayText = geom_id
geom_geom_container.Edit()
geom_geom_container.SendCommand( Command = """"

A = 5;
PD = 3;
HD = 1;
HoD = 2;
HpoD = 2;
N = 6;

in2m = 0.0254;
length = 3*in2m;
D = 0.125*in2m;
OD = 2*D;
Wth = 0.125*in2m;

H = HD*D;
Ho = HoD*D;
if (H == Ho) {
    Ho = Ho+1e-6;
}

Hpo = HpoD*D;
pitch = PD*D;
angle = A;
angle_rad = angle * Math.PI/180
tan_ang = Math.tan(angle_rad)
block_thickness = 0.25*in2m;
```

```

x_len = N*pitch;
y_len = pitch;
z_len = Ho + Wth + Hpo + tan_ang*x_len;

//Clean the session before running
ag.m.ClearAllErrors();
ag.m.NewSession(true);

function drawSurfaceSketch (p)
{
//Plane
p.Plane = agb.PlaneFromPlane(XYPlane);
//Sketch
p.sketch1 = p.Plane.NewSketch();
p.sketch1.Name = "fluidBaseSketch";
//Edges
with (p.sketch1)
{
p.Ln1 = Line(0, -y_len, x_len, -y_len);
p.Ln2 = Line(x_len, -y_len, x_len, y_len);
p.Ln3 = Line(x_len, y_len, 0, y_len);
p.Ln4 = Line(0, y_len, 0, -y_len);
}
return p;
} //End drawSurfaceSketch function

function drawLowerConfiningWallSketch (p)
{
//Plane
// (i,j,k) is normal vector to confining wall
x = 0;
y = 0;
z = Ho;
i = -tan_ang;
j = 0;
k = 1;
p.Plane = agb.PlaneFromCoord(x,y,z,i,j,k);
//Sketch
p.sketch1 = p.Plane.NewSketch();
p.sketch1.Name = "LowerConfiningWallSketch";
//Edges
cut_len = 1.5*x_len;
dx = cut_len;
dy = y_len;
with (p.sketch1)
{
p.Ln1 = Line(0, -dy, dx, -dy);
p.Ln2 = Line(dx, -dy, dx, dy);
}
}

```

```

    p.Ln3 = Line(dx, dy, 0, dy);
    p.Ln4 = Line(0, dy, 0, -dy);
}
return p;
} //End drawLowerConfiningWallSketch function

```

```

function drawUpperConfiningWallSketch (p)
{
//Plane
// (i,j,k) is normal vector to confining wall
x = 0;
y = 0;
z = Ho + Wth;
i = -tan_ang;
j = 0;
k = 1;
p.Plane = agb.PlaneFromCoord(x,y,z,i,j,k);
//Sketch
p.sketch1 = p.Plane.NewSketch();
p.sketch1.Name = "UpperConfiningWallSketch";
//Edges
cut_len = 1.5*x_len;
dx = cut_len;
dy = y_len
with (p.sketch1)
{
    p.Ln1 = Line(0, -dy, dx, -dy);
    p.Ln2 = Line(dx, -dy, dx, dy);
    p.Ln3 = Line(dx, dy, 0, dy);
    p.Ln4 = Line(0, dy, 0, -dy);
}
return p;
} //End drawUpperConfiningWallSketch function

```

```

function drawRightNozzleOutletSketch (p){
//Plane
p.Plane = agb.PlaneFromPlane(XYPlane);
p.Plane.AddTransform(agc.XformZOffset, H);
p.Plane.Name = "Right Nozzle Outlet Plane"
p.sketch1 = p.Plane.NewSketch();
p.sketch1.Name = "Right Nozzle Outlet Sketch";
//Edges
with (p.sketch1)
{
    for (i=0; i<N; i++){
        Circle(pitch*(i+0.5), 0, D/2);
        Circle(pitch*(i+0.5), 0, OD/2);
    }
}
}

```

```

    }
    return p;
}

function drawRightNozzleInteriorSketch (p){
    //Plane
    p.Plane = agb.PlaneFromPlane(XYPlane);
    p.Plane.AddTransform(agc.XformZOffset, H);
    p.Plane.Name = "Right Nozzle Interior Outlet Plane"
    p.sketch1 = p.Plane.NewSketch();
    p.sketch1.Name = "Right Nozzle Interior Sketch";
    with (p.sketch1)
    {
        for (i=0; i<N; i++){
            Circle(pitch*(i+0.5), 0, D/2);
        }
    }
    return p;
}

```

```

function drawLeftNozzleOutletSketch (p){
    //Plane
    p.Plane = agb.PlaneFromPlane(XYPlane);
    p.Plane.AddTransform(agc.XformZOffset, H);
    p.Plane.Name = "Left Nozzle Outlet Plane"
    p.sketch1 = p.Plane.NewSketch();
    p.sketch1.Name = "Left Nozzle Outlet Sketch";
    //Edges
    with (p.sketch1)
    {
        for (i=0; i<N-1; i++){
            Circle(pitch*(i+1), y_len, D/2);
            Circle(pitch*(i+1), y_len, OD/2);
        }
    }
    return p;
}

```

```

function drawLeftNozzleInteriorSketch (p){
    //Plane
    p.Plane = agb.PlaneFromPlane(XYPlane);
    p.Plane.AddTransform(agc.XformZOffset, H);
    p.Plane.Name = "Left Nozzle Interior Outlet Plane"
    p.sketch1 = p.Plane.NewSketch();
    p.sketch1.Name = "Left Nozzle Interior Sketch";
    with (p.sketch1)
    {
        for (i=0; i<N-1; i++){

```

```

        Circle(pitch*(i+1), y_len, D/2);
    }
}
return p;
}

function drawInletTrimSketch (p)
{
    // (i,j,k) is normal vector to the plane
    x = 0;
    y = 0;
    z = 0;
    i = -1;
    j = 0;
    k = 0;
    p.Plane = agb.PlaneFromCoord(x,y,z,i,j,k);
    p.Plane.Name = "Inlet Trim Plane"
    //Sketch
    p.sketch1 = p.Plane.NewSketch();
    p.sketch1.Name = "inletTrimSketch";
    //Edges
    dx = 2*z_len;
    dy = 2*y_len;
    with (p.sketch1)
    {
        p.Ln1 = Line(-dx, -dy, dx, -dy);
        p.Ln2 = Line(dx, -dy, dx, dy);
        p.Ln3 = Line(dx, dy, -dx, dy);
        p.Ln4 = Line(-dx, dy, -dx, -dy);
    }
    return p;
} //End drawInletTrimSketch function

function drawOutletTrimSketch (p)
{
    // (i,j,k) is normal vector to the plane
    x = x_len;
    y = 0;
    z = z_len;
    i = 1;
    j = 0;
    k = 0;
    p.Plane = agb.PlaneFromCoord(x,y,z,i,j,k);
    p.Plane.Name = "Outlet Trim Plane"
    //Sketch
    p.sketch1 = p.Plane.NewSketch();
    p.sketch1.Name = "outletTrimSketch";
    //Edges

```

```

dx = 2*z_len;
dy = 2*y_len;
with (p.sketch1)
{
  p.Ln1 = Line(-dx, -dy, dx, -dy);
  p.Ln2 = Line(dx, -dy, dx, dy);
  p.Ln3 = Line(dx, dy, -dx, dy);
  p.Ln4 = Line(-dx, dy, -dx, -dy);
}
return p;
} //End drawOutletTrimSketch function

```

```

function drawPlenumTrimSketch (p)
{
  // (i,j,k) is normal vector to the plane
  x = x_len/2;
  y = 0;
  z = z_len;
  i = 0;
  j = 0;
  k = 1;
  p.Plane = agb.PlaneFromCoord(x,y,z,i,j,k);
  p.Plane.Name = "Plenum Trim Plane"
  //Sketch
  p.sketch1 = p.Plane.NewSketch();
  p.sketch1.Name = "plenumTrimSketch";
  //Edges
  dx = 2*x_len;
  dy = 2*y_len;
  with (p.sketch1)
  {
    p.Ln1 = Line(-dx, -dy, dx, -dy);
    p.Ln2 = Line(dx, -dy, dx, dy);
    p.Ln3 = Line(dx, dy, -dx, dy);
    p.Ln4 = Line(-dx, dy, -dx, -dy);
  }
  return p;
} //End drawPlenumTrimSketch function

```

```

function drawCenterLineTrimSketch (p)
{
  // (i,j,k) is normal vector to the plane
  x = x_len/2;
  y = 0;
  z = 0;
  i = 0;

```



```

j = -1;
k = 0;
p.Plane = agb.PlaneFromCoord(x,y,z,i,j,k);
p.Plane.Name = "Center Line Plane"
//Sketch
p.sketch1 = p.Plane.NewSketch();
p.sketch1.Name = "centerLineTrimSketch";
//Edges
dx = 2*x_len;
dy = 2*z_len;
with (p.sketch1)
{
  p.Ln1 = Line(-dx, -dy, dx, -dy);
  p.Ln2 = Line(dx, -dy, dx, dy);
  p.Ln3 = Line(dx, dy, -dx, dy);
  p.Ln4 = Line(-dx, dy, -dx, -dy);
}
return p;
} //End drawCenterLineTrimSketch function

function drawLeftCenterLineTrimSketch (p)
{
  // (i,j,k) is normal vector to the plane
  x = x_len/2;
  y = y_len;
  z = 0;
  i = 0;
  j = -1;
  k = 0;
  p.Plane = agb.PlaneFromCoord(x,y,z,i,j,k);
  p.Plane.Name = "Left Center Line Plane"
  //Sketch
  p.sketch1 = p.Plane.NewSketch();
  p.sketch1.Name = "LeftCenterLineTrimSketch";
  //Edges
  dx = 2*x_len;
  dy = 2*z_len;
  with (p.sketch1)
  {
    p.Ln1 = Line(-dx, -dy, dx, -dy);
    p.Ln2 = Line(dx, -dy, dx, dy);
    p.Ln3 = Line(dx, dy, -dx, dy);
    p.Ln4 = Line(-dx, dy, -dx, -dy);
  }
  return p;
} //End drawLeftCenterLineTrimSketch function

function drawBlockSketch (p)

```

```

{
// (i,j,k) is normal vector to the plane
x = 0;
y = 0;
z = 0;
i = 0;
j = 1;
k = 0;
p.Plane = agb.PlaneFromCoord(x,y,z,i,j,k);
p.Plane.Name = "Block Profile Plane"
//Sketch
p.sketch1 = p.Plane.NewSketch();
p.sketch1.Name = "BlockProfileSketch";
//Edges
x1 = 0;
y1 = 0;
x2 = x_len;
y2 = block_thickness;
with (p.sketch1)
{
  p.Ln1 = Line(x1, y1, x2, y1);
  p.Ln2 = Line(x2, y1, x2, y2);
  p.Ln3 = Line(x2, y2, x1, y2);
  p.Ln4 = Line(x1, y2, x1, y1);
}
return p;
} //End drawBlockSketch function

// Call Functions to draw sketches
// Fluid Surface
var XYPlane = agb.GetXYPlane();
agb.SetActivePlane(XYPlane);
var SurfacePlaneSketches = drawSurfaceSketch (new Object());
// Right Nozzle outlet
var RightNozzleOutletObj = drawRightNozzleOutletSketch (new Object());
// Right Nozzle interior
var RightNozzleInteriorObj = drawRightNozzleInteriorSketch (new Object());
// Left Nozzle outlet
var LeftNozzleOutletObj = drawLeftNozzleOutletSketch (new Object());
// Left Nozzle interior
var LeftNozzleInteriorObj = drawLeftNozzleInteriorSketch (new Object());
// Confining Wall
var LowerConfiningWallObj = drawLowerConfiningWallSketch (new Object());
var UpperConfiningWallObj = drawUpperConfiningWallSketch (new Object());
// Trim extra material
var InletTrimObj = drawInletTrimSketch (new Object());
var OutletTrimObj = drawOutletTrimSketch (new Object());

```

```

var PlenumTrimObj = drawPlenumTrimSketch (new Object());
var CenterLineTrimObj = drawCenterLineTrimSketch (new Object());
var LeftCenterLineTrimObj = drawLeftCenterLineTrimSketch (new Object());
// Copper Block
var BlockObj = drawBlockSketch (new Object());

agb.regen();

// Create solid using Extrude
var FluidExtrude = agb.Extrude(agc.Add, SurfacePlaneSketches.sketch1, agc.DirNormal,
agc.ExtentFixed,z_len,0,0,0,0,0);

// Remove Right Nozzle wall
var RightNozzleWall = agb.Extrude(agc.Cut, RightNozzleOutletObj.sketch1, agc.DirNormal,
agc.ExtentToNext, 0, 0, 0, 0, 0, 0);

// Remove Left Nozzle wall
var LeftNozzleWall = agb.Extrude(agc.Cut, LeftNozzleOutletObj.sketch1, agc.DirNormal,
agc.ExtentToNext, 0, 0, 0, 0, 0, 0);

// Remove fluid above confining wall
var LowerConfiningWall = agb.Extrude(agc.Cut, LowerConfiningWallObj.sketch1, agc.DirNormal,
agc.ExtentToNext, 0, 0, 0, 0, 0, 0);

// Add fluid above upper confining wall
var UpperConfiningWall = agb.Extrude(agc.Add, UpperConfiningWallObj.sketch1, agc.DirNormal,
agc.ExtentFixed, z_len,0,0,0,0,0);

// Fill in missing fluid inside right nozzle
var InsideRightNozzleFluid = agb.Extrude(agc.Add, RightNozzleInteriorObj.sketch1,
agc.DirNormal,agc.ExtentFixed, z_len,0,0,0,0,0);

// Fill in missing fluid inside Left nozzle
var InsideLeftNozzleFluid = agb.Extrude(agc.Add, LeftNozzleInteriorObj.sketch1,
agc.DirNormal,agc.ExtentFixed, z_len,0,0,0,0,0);

// Trim Inlet
var InletTrim = agb.Extrude(agc.Cut, InletTrimObj.sketch1, agc.DirNormal,agc.ExtentFixed, 5*z_len,
0,0,0,0,0);

// Trim Outlet
var OutletTrim = agb.Extrude(agc.Cut, OutletTrimObj.sketch1, agc.DirNormal,agc.ExtentFixed, 5*z_len,
0,0,0,0,0);

// Trim Plenum
var PlenumTrim = agb.Extrude(agc.Cut, PlenumTrimObj.sketch1, agc.DirNormal,agc.ExtentFixed,
5*z_len, 0,0,0,0,0);

```

```

// Trim from Center Line
var CenterLineTrim = agb.Extrude(agc.Cut, CenterLineTrimObj.sketch1, agc.DirNormal,agc.ExtentFixed,
10*z_len, 0,0,0,0,0);

// Trim from Left Center Line
var LeftCenterLineTrim = agb.Extrude(agc.Cut, LeftCenterLineTrimObj.sketch1,
agc.DirReversed,agc.ExtentFixed, 10*z_len, 0,0,0,0,0);

// Create copper block
var BlockExtrude = agb.Extrude(agc.Frozen, BlockObj.sketch1, agc.DirNormal,
agc.ExtentFixed,y_len,0,0,0,0,0);

//Finish
agb.Regen(); //To insure model validity
ag.gui.ZoomFit();
//End DM JScript
""")

geom_geom_container.Exit()
#### End Geometry ####

#### Add Mesh template and open it for editing ####
mesh_template = GetTemplate(TemplateName="Mesh")
geom_system = GetSystem(Name="Geom")

try:
    # Connect to existing mesh system
    mesh_system = GetSystem(Name="SYS")
except:
    # Create a mesh system
    mesh_system = mesh_template.CreateSystem(
        Position="Right",
        RelativeTo=geom_system)

geom_geom_component = geom_system.GetComponent(Name="Geometry")
mesh_geom_component = mesh_system.GetComponent(Name="Geometry")
mesh_geom_component.ReplaceWithShare(
    TargetSystem=mesh_system,
    ComponentToShare=geom_geom_component,
    SourceSystem=geom_system)
mesh_mesh_component = mesh_system.GetComponent(Name="Mesh")
mesh_mesh_component.Refresh()
# mesh_mesh_containter = mesh_system.GetContainer(Name="Mesh")
# mesh_mesh_container.Edit()
# The mesh will tool will open an the following operations need to
# performed manually
# * Create named selections for

```

```
# - inlet
# - outlet
# - heated-surface
# - inflation-surfaces
# * Change fluid region to type "fluid"
# * Rename fluid region to "fluid"
# * Rename the solid region to "block"
# * Change "Physics Preference" to "CFD"
# * Set "Relevance Center" to "Fine"
# * Set "Use Automatic Inflation" to "All Faces in Chosen Named Selection"
# * Set "Named Selection" to "inflation-surfaces"
# * Set "Maximum Layers" to "10"
# * Exit the Meshing tool
# * Update the mesh
#### End Mesh ####

# Local Variables:
# mode: python
# End:
```

Appendix A.2 Mesh Generation in Fluent

Initialize ANSYS Workbench 16.2. To select a starting geometry for the mesh file elect File -> Scripting -> Run Script File as shown in Figure A.2.1 below.

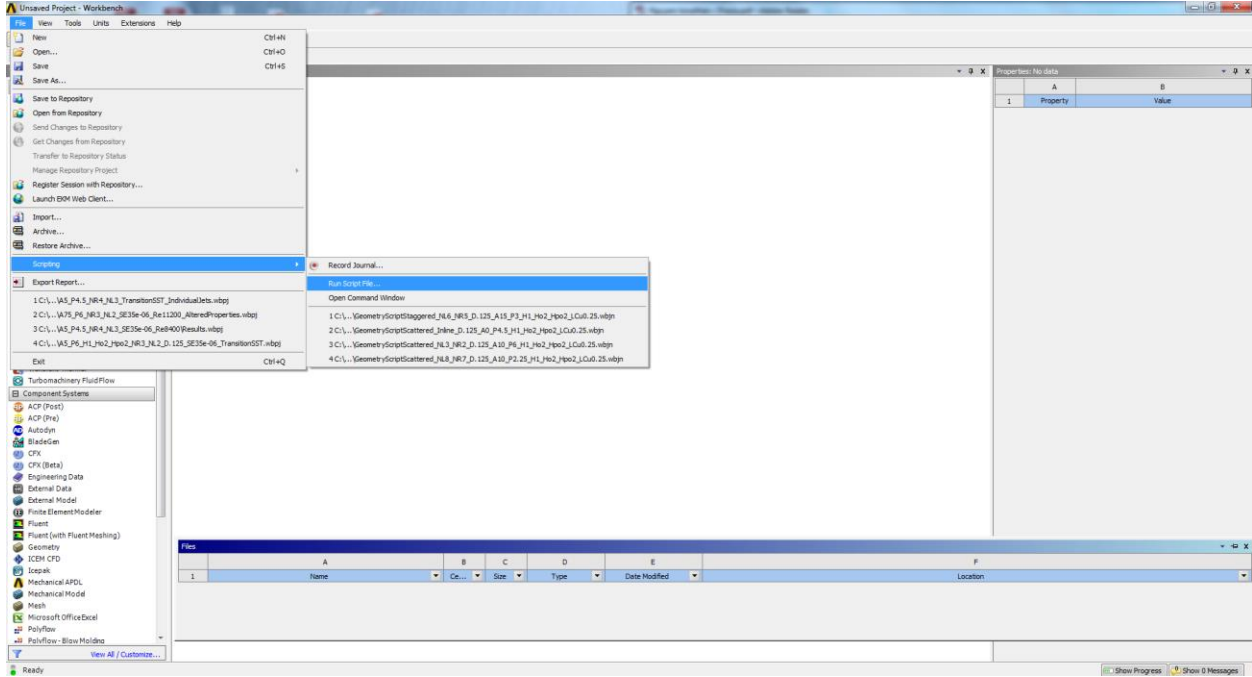


Figure A.2.6.1: Selecting Geometric Script in ANSYS Workbench

Select the desired geometric script as shown in Figure A.2.2 below.

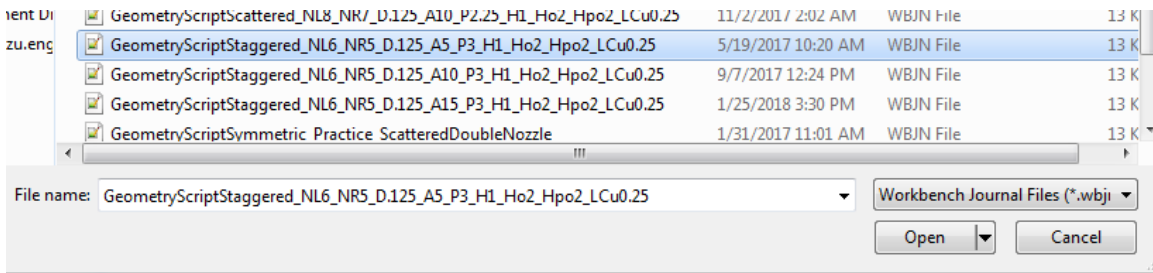


Figure A.2.6.2: Selecting Geometry File

Once the geometry has finished building, the following components shown in Figure A.2.3 will appear in the Workbench window.

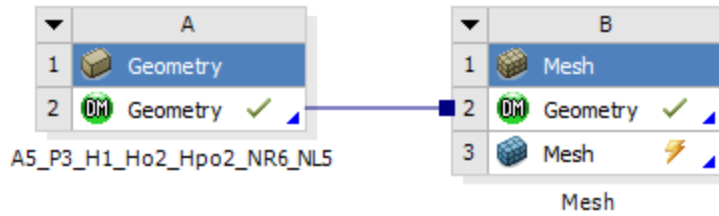



Figure A.2.6.3: Components shown in Workbench Window

Right click and select Edit on the Mesh line to open the ANSYS Meshing. An image similar to the one shown in Figure A.2.4 will appear. If the geometry does not appear immediately, select the Zoom to Fit (F7) tool to  zoom in on the geometry.

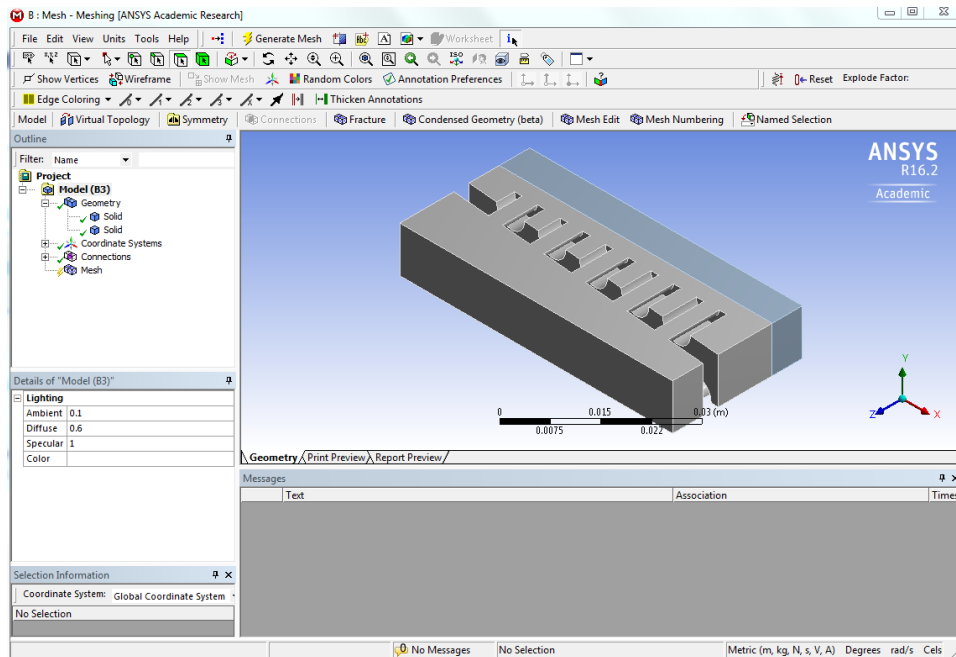


Figure A.2.6.4: Meshing Window in ANSYS

Select the Solid body that contains the fluid flow under the Geometry tab. The selected body will appear as highlighted in Figure A.2.5. Change the material in Fluid/Solid from Defined by Geometry to Fluid. Repeat the process for the solid copper block changing the material in Fluid/Solid from Defined by Geometry to Solid.

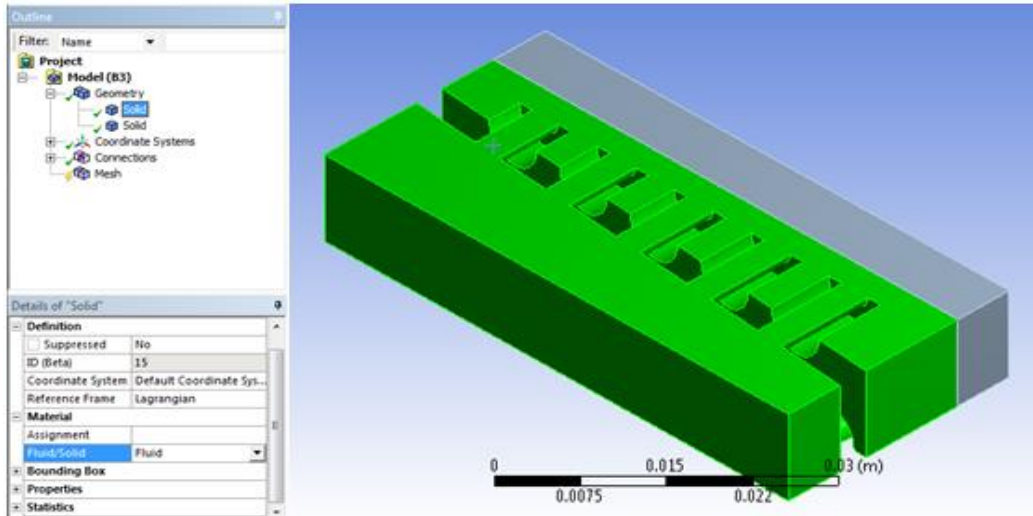



Figure A.2.6.5: Changing Material Properties in ANSYS Mesh

After defining the bodies of the Geometry, each of the bodies and required faces need to be named to use with the Fluent python script. Use the select Body/Element tool and  select the fluid body. Right click on the selected body and select Create Named Selection as shown in Figure A.2.6. Name the selection fluid. Repeat the process for the solid body with the name solid.

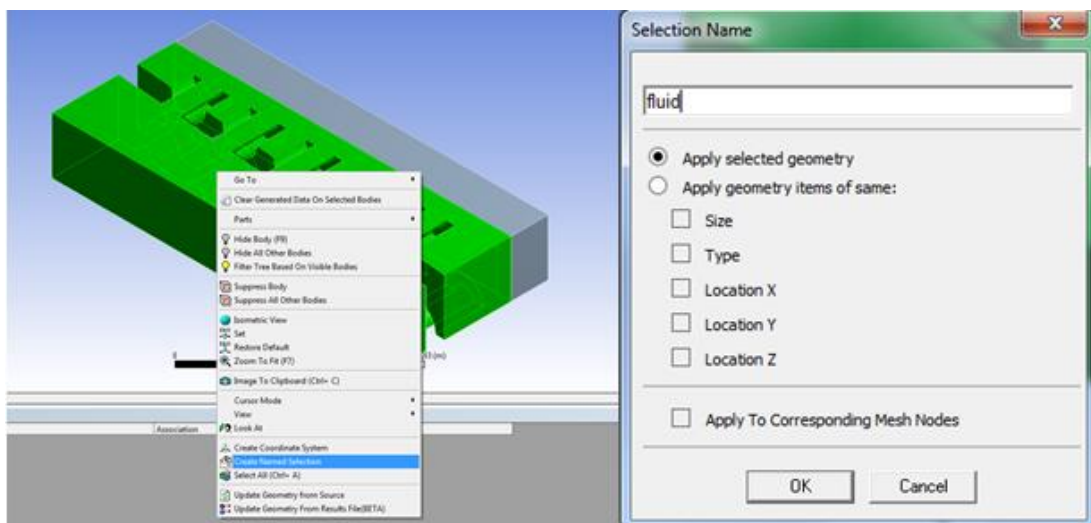


Figure A.2.6.6: Naming a selected Body/Element in Mesh

Use the select Face tool and select each of the required faces shown in Figure A.2.7 and name them using the names given below the image. Use Ctrl to select more than one surface to create one

named face. Right click on the solid body and select Hide Body in order to select the two surface in the face named inflationsurfaces.

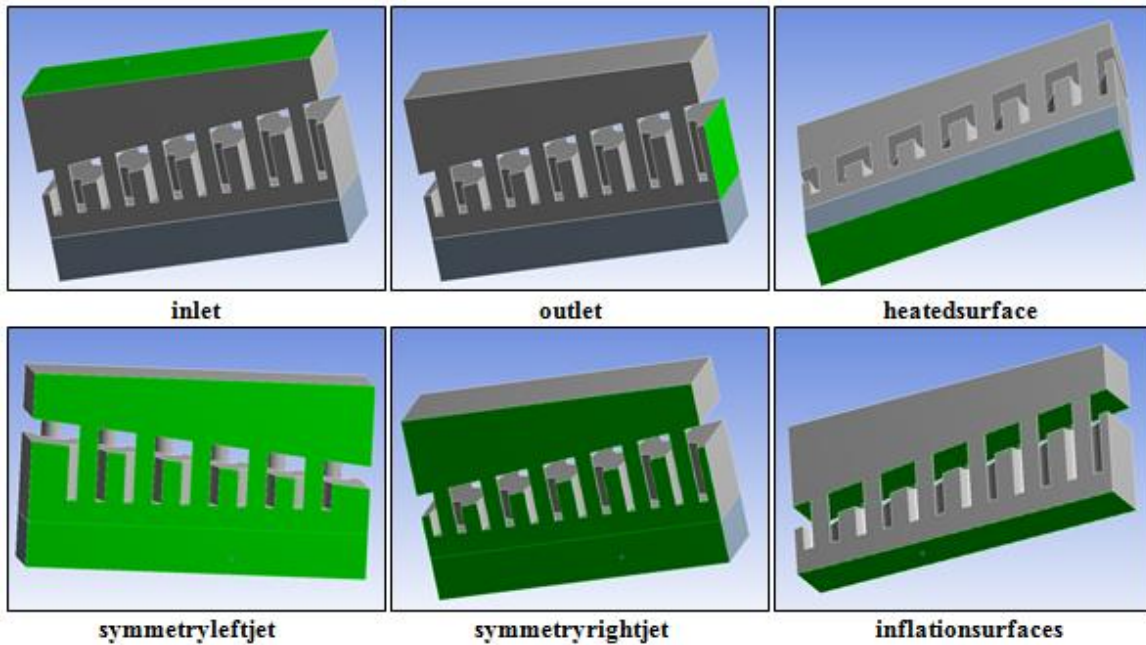


Figure A.2.6.7: Names of Faces in Mesh

Select Mesh Control -> Sizing as show in Figure A.2.8.

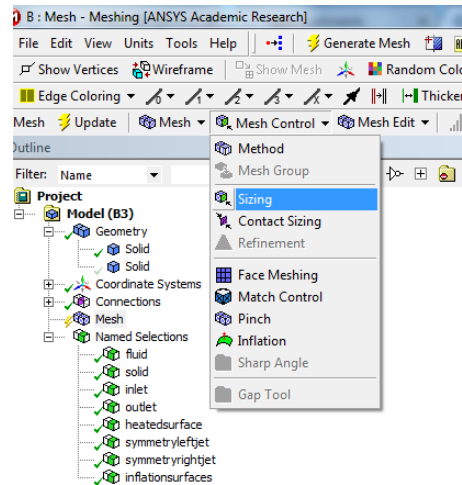


Figure A.2.6.8: Face Sizing in Mesh

Select the face at the bottom of the fluid as the geometry for sizing as shown in Figure A.2.9.

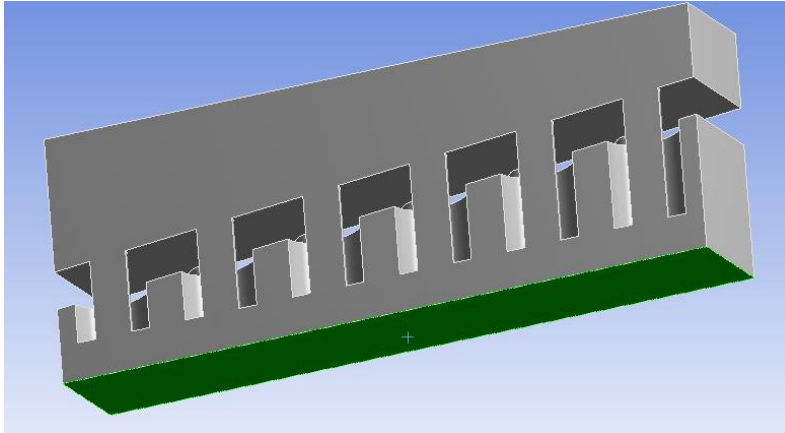


Figure A.2.6.9: Sizing bottom face of fluid in Mesh

Change the Type to Element Size and specify that the Element Size is $3.5e-05$ m.

Select the Mesh and change the specified values in Table A.2.1.

Table A.2.6-1: Specified Meshing Parameters

Defaults	Physics Preference	CFD
	Solver Preference	Fluent
	Relevance	99
Sizing	Relevance Center	Fine
	Smoothing	Medium
	Transition	Slow
	Span Angle Center	Fine
Inflation	Use Automatic inflation	All Faces in Chosen Named Selection
	Named Selection	inflationsurfaces
	Maximum Inflation Layers	10
	Inflation Growth Rate	1.2

Right click on Mesh and select Generate Mesh. Once the mesh has finished generating, select File

-> Export and save the .msh file. For the above example, the mesh file name was

A5_P3_H1_Ho2_Hpo2_NR6_NL5_D.125_SE35e-06.msh to specify the angle, pitch, heights, number of jets on the right and left sides, diameter of jet, and surface element sizing used.

Determine y^+

Appendix A.3 Fluent 16.2

Appendix A.3.1 MATLAB code for determining the appropriate inlet velocity.

```
% Determining Required Inlet Velocity for Re = 5600

clear all
clc

Re = 14000; % Reynolds number inside jet
% Values for comparison: 5600, 8400, 11200, 14000
nu = 7.7227232143e-07; %m^2/s
pitch = 6; % Desired pitch

% Jet properties
D = 0.003175; % diameter of nozzle in m
NR = 3; % number of jets on right side
NL = 2; % number of jets on left side
N = NR + NL; % total number of jets
Area_jets = 0.5*(pi*(D^2)/4)*N; % Area of jets in m^2
u_jets = (Re*nu)/D; % velocity inside jets in m/s
V = Area_jets*u_jets; % Volumetric Flow Rate in m^3/s

% Inlet area
P = pitch*D; % pitch
xlength = P*NR; % x length of rectangular inlet in m
ylength = P; % y length of rectangular inlet in m
% Note: ylength = P for staggered array and P/2 for inline array
Area_inlet = xlength*ylength; % Area of inlet m^2
u_inlet = V/Area_inlet; % velocity through inlet

fprintf('length in x direction %12.10f m. \n',xlength)
fprintf('length in y direction %12.10f m. \n\n',ylength)
fprintf('Area of inlet %8.6f m^2. \n\n',Area_inlet)

fprintf('The required inlet velocity is %15.13f m/s.\n\n',u_inlet)

nu_fluent = 1.005e-06; % fluent nu value for water at 300 K m^2/s
Re_fluent = (D*u_jets)/nu_fluent; % Reynolds value according to fluent

fprintf('Comparison of Reynolds values \n')
fprintf('Dr. Maddox \t Re = %6.0f \n',Re)
fprintf('Fluent \t\t Re = %6.0f \n',Re_fluent)
```

Appendix A.3.2 Code for creating fluent path in ANSYS Fluent 16.2.

```
;; Start batch mode
/file/set-batch-options no yes no
;; Start log
/file/start-transcript A0_P3_NR6_NL5_D125_Re5600_TransitionSST.log
/file/read A0_P3_H1_Ho2_Hpo2_NR6_NL5_D.125_SE35e-06.msh
```

```

;; Define surfaces for visualization
/surface/plane-point-n RightInline 0 0 0 0 1 0
/surface/plane-point-n LeftInline 0 0.009525 0 0 1 0
/surface/plane-point-n Transverse 0 0 0 1 0 0
/surface/plane-point-n Surface 0 0 0 0 0 1
/surface/plane-point-n Nozzle 0 0 0.003175 0 0 1

;; Turn on energy equation
/define/models/energy yes no no no no

;; Turn on Transition-SST turbulence model
/define/models/viscous/transition-sst yes

;; Copy materials from Fluent Database
/define/materials/copy solid copper
/define/materials/copy fluid water
;; Define materials for regions
/define/boundary-conditions/fluid fluid yes water no no no no 0 no 0
no 0 no 0 no 0 no 1 no no no no no
/define/boundary-conditions/solid solid yes copper no no no no 0 no 0
no 0 no 0 no 0 no 1 no no
;; Define inlet velocity
/define/boundary-conditions/zone-type inlet velocity-inlet
/define/boundary-conditions/velocity-inlet inlet no no yes yes no
0.1089616266966 no 0 no 300 no no yes no 1 5 10
;; Define heat flux
/define/boundary-conditions/wall heatedsurface 0 no 0 yes copper yes
heat-flux no 92554.0284141 no no 1
;; Define symmetric boundaries
/define/boundary-conditions/zone-type symmetryleftjet-solid symmetry
/define/boundary-conditions/zone-type symmetryrightjet-solid symmetry
;; Set auto-save interval
/file/auto-save/data-frequency 1000
;; Set number of iterations to keep
/file/auto-save/retain-most-recent-files yes
/file/auto-save/max-files 3
;; Set whether or not to save the case files with each set of data
files
/file/auto-save/case-frequency each-time
;; Set max number of iterations
/solve/set/number-of-iterations 750
;; Set how often to report results
/solve/set/reporting-interval 1
;; Initialize the solution
/solve/initialize/hyb-initialization
;; Set convergence criteria
/solve/monitors/residual/convergence-criteria 1e-5 1e-5 1e-5 1e-5 1e-6
1e-5 1e-5

;; Set monitors

```

```

/solve/monitors/surface/set-monitor surface-temp-avg "Area-Weighted
Average" temperature surface () no yes yes "monitor-surface-temp-
avg.txt" 1
/solve/monitors/surface/set-monitor surface-temp-max "Vertex Maximum"
temperature surface () no yes yes "monitor-surface-temp-max.txt" 1
/solve/monitors/surface/set-monitor surface-temp-min "Vertex Minimum"
temperature surface () no yes yes "monitor-surface-temp-min.txt" 1
/solve/monitors/surface/set-monitor surface-htc-avg "Area-Weighted
Average" heat-transfer-coef surface () no yes yes "monitor-surface-
htc-avg.txt" 1
/solve/monitors/surface/set-monitor surface-htc-max "Vertex Maximum"
heat-transfer-coef surface () no yes yes "monitor-surface-htc-max.txt"
1
/solve/monitors/surface/set-monitor surface-htc-min "Vertex Minimum"
heat-transfer-coef surface () no yes yes "monitor-surface-htc-min.txt"
1

```

```

;; Export residuals every 10 iterations
/solve/execute-commands/add-edit com1 10 "iteration" "/disp save-pic
A5_P3_H1_Ho2_Hpo2_NR6_NL5_D125_Re5600_TransitionSST-residuals.png"

```

```

;; temporarily turn off questions in the TUI
/file/set-batch-options no yes yes no
;; execute the solution for the number of iterations
/solve/iterate 3000
/plot/residuals
/disp save-pic A5_P3_H1_Ho2_Hpo2_NR6_NL5_D125_Re5600_TransitionSST-
residuals.png
;; turn question back on in the TUI
/file/set-batch-options yes

```

```

;; Save Surface data
/file/export/ascii
A5_P3_H1_Ho2_Hpo2_NR6_NL5_D125_Re5600_TransitionSST-Surface_Data.csv
Surface

```

```

yes
pressure
temperature
heat-flux
heat-transfer-coef
nusselt-number
stanton-number
()
no

```

```

;; Save RightInline data
/file/export/ascii

```

```

A5_P3_H1_Ho2_Hpo2_NR6_NL5_D125_Re5600_TransitionSST-
RightInline_Data.csv
Inline

yes
pressure
temperature
heat-flux
heat-transfer-coef
nusselt-number
stanton-number
()
no

;; Save LeftInline data
/file/export/ascii
A5_P3_H1_Ho2_Hpo2_NR6_NL5_D125_Re5600_TransitionSST-
LeftInline_Data.csv
Inline

yes
pressure
temperature
heat-flux
heat-transfer-coef
nusselt-number
stanton-number
()
no

;; Save Nozzle data
/file/export/ascii
A5_P3_H1_Ho2_Hpo2_NR6_NL5_D125_Re5600_TransitionSST-Nozzle_Data.csv
Nozzle

yes
pressure
temperature
heat-flux
heat-transfer-coef
nusselt-number
stanton-number
()
no

/file/write-case-data A5_P3_NR6_NL5_D125_Re5600_TransitionSST.cas.gz

/exit

```

Appendix A.4 HPCC Instructions

http://www.eng.auburn.edu/ens/hpcc/software_fluent13.0.html

Appendix A.4.1 Hopper Cluster Setup and Login

A basic user's guide to using the Cluster can be found here: [Auburn University Hopper HPCC User's Guide](https://wp.auburn.edu/hpc/?document=auburn-university-hopper-hpcc-users-guide) <https://wp.auburn.edu/hpc/?document=auburn-university-hopper-hpcc-users-guide>

Gain access to the Hopper Cluster using the Request an Account section found from the hyperlink above. Select your sponsored professor from the drop-down menu.

Email Shannon Price (pricesw@auburn.edu) for any additional questions regarding the Hopper Cluster that are not addressed in these instructions.

Open Secure CRT using auburn id and password

Type: ssh ker0017@hopper.auburn.edu

Use your auburn_user_id@hopper.auburn.edu in the highlighted area as shown in example above
Enter Auburn password, the following image should show indicating that you have access to the hopper cluster


```
welcome to...
HOPPER
Good afternoon, ker0017.
It is 3:42 PM on Wednesday, November 02, 2016.
On the Auburn campus it is 82°F, Partly Sunny
3382 of 4092 processors in use by local jobs (82.65%)
-----
[ker0017@hopper-login] (~) [15:42]:
```

Type `cd fluent_test`

Highlighted portion indicates the folder name which you created in WinSCP to run fluent case.

Can create as many files as needed on the WinSCP but note that these will be deleted after 30 days so remember to copy needed solutions to computer.

Type `qsub run_fluent.sh`

This `run_fluent.sh` is shown at the end of this report. It will need to be edited depending on the case that needs to be run.

Appendix A.4.2 Useful terminology on Secure CRT (Linux operated)

`cd ->` returns to home directory

`cd folder ->` Takes to folder or directory specified

`ls ->` lists names of files in the directory that you are looking at

`showq ->` shows all jobs that are currently being run

On showq, it is useful to find your job number which will be next to your username and job number, as shown below. For my username, ker0017, the job number is 26944. This number can be used to cancel or view the progress of the job.

showq -u **username** -> shows jobs that are currently being run by a specific user.

Use Auburn username.

canceljob **26944** -> cancels job number highlighted.

```
active jobs-----
JOBID          USERNAME      STATE  PROCS   REMAINING          STARTTIME
26845          ezm0048      Running  20     4:40:16   wed Nov  2 14:30:54
26846          ezm0048      Running  20     4:40:32   wed Nov  2 14:31:10
26943          ezm0048      Running  20     5:42:35   Thu Nov  3 09:33:13
26856          ballacp      Running 100     6:28:13   wed Nov  2 16:18:51
26503          ezm0048      Running  20     9:38:44   Mon Oct 31 19:29:22
26766          ballacp      Running  66     9:52:27   wed Nov  2 13:43:05
26938          nza0031      Running 200    19:21:41   Thu Nov  3 09:12:19
26576          szs0118      Running   1     21:07:04   Tue Nov  1 08:57:42
26944          ker0017      Running  20     1:05:54:19 Thu Nov  3 09:44:57
26565          ira0002      Running  20     2:01:38:37 Tue Nov  1 07:29:15
26936          nak0007      Running   1     2:02:00:10 Thu Nov  3 08:50:48
25779          mak0037      Running  16     3:02:04:04 Thu Oct 27 11:54:42
```

Appendix A.4.3 Logging into Hopper via WinSCP

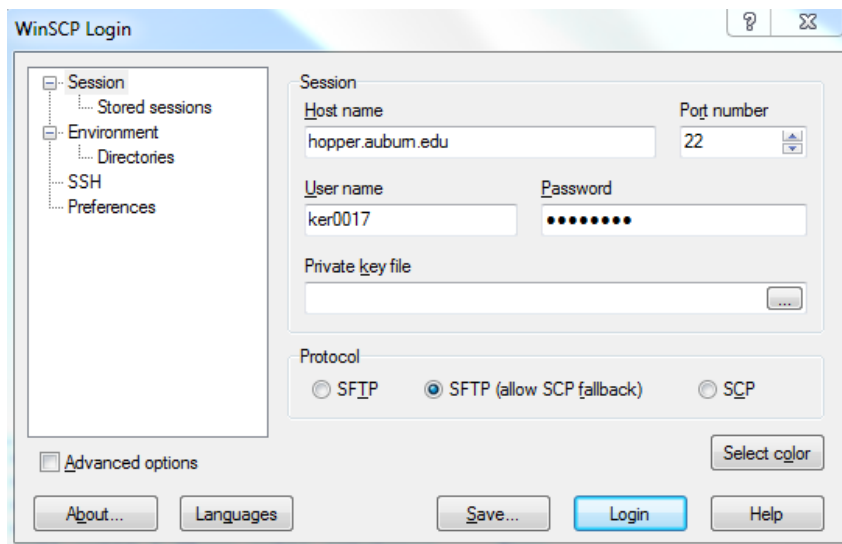
In addition, login to the hopper via WinSCP to view the files that will be added to the cluster

Open WinSCP

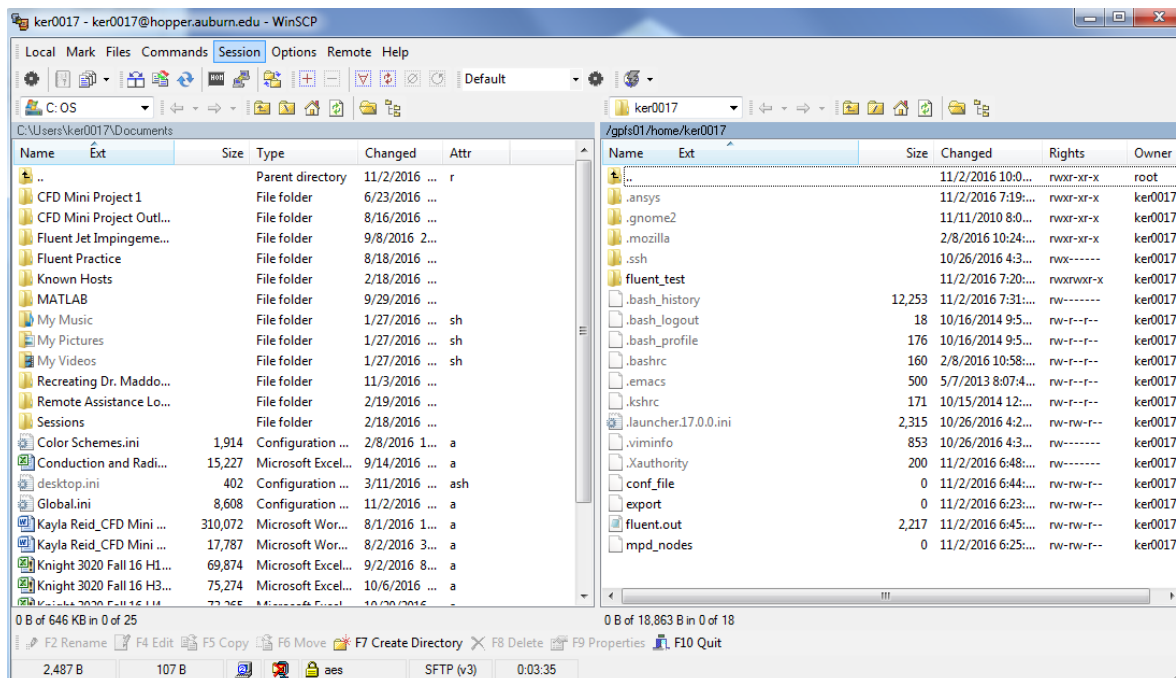
Hostname: hopper.auburn.edu

Auburn username and password

Port number should be 22 (for Engineering Department)



This will bring you to the following page shown in Figure X. The left side indicates your computer files and the right side indicates the hopper directory. You can add, edit, and remove files from this location.



Appendix A.4.4 Example of run_fluent.sh file

The original file can be found at [run_fluent Example](#)

Note that most of the code following a # is comments for understanding the code.

```
#!/bin/sh
#name the program, your default output error file are Test_fluent.oJobid
#and Test_fluent.eJobid
#PBS -N Test_fluent
#following 2 lines ensures that you'll be notified by email when your job is done
#PBS -M ker0017@tigermail.auburn.edu
#PBS -m abe
#you are asking for 4 node 8 processor each, 32 processors as a total for 30hrs
#after 30 hours your job will be killed
#PBS -l nodes=4:ppn=8,walltime=30:00:00
```

```

#your directory path can be obtained by pwd
#PBS -d /home/ker0017/fluant_test
#loading variables do not change this
export PATH=/tools/licensed/ansys-17.0/v170/fluant/bin:$PATH
export FLUENT_ARCH=lnamd64
export NO_LOCAL=1
#writing mpd_nodes to boot mpd in these nodes and conf_file to select processors
#`sort -u $PBS_NODEFILE > mpd_nodes`
#assigning nhosts variable to number of nodes
#nhosts=`cat mpd_nodes | wc -l`
#generating conf_file to select processors
`sort $PBS_NODEFILE > conf_file`
proc=`cat conf_file | wc -l`
#printing initial timestamp
date > fluent.out
#printing which host performed computation
/bin/hostname >> fluent.out
#booting mpd on the selected nodes
#mpdboot -n $nhosts -v -f mpd_nodes
#if you want to use intel as communication port the use -mpi=intel, also uncomment mpdboot and
mpdallexit, by default you can use hp and leave it unchanged
#executing fluent with 64 processor providing mpich communication connector
#and conf_file as config file and i=source_file >> writing_to_output_file
fluent 3ddp -g -t$proc -mpi=pcmpi -cnf=conf_file -i 3J_D0_SE75e-06_Re5000-sol-SST.jou >>
fluent.out
#end time stamp
date >> fluent.out
#stop mpd
#mpdallexit

```

The highlighted portions in the above code indicate text that needs to be changed.

```
#PBS -N Test_fluent
```

This text designates the name of the output error file if your code has an error. It can be changed to match the name of each of your files but this is an optional change.

```
#PBS -M ker0017@tigermail.auburn.edu
```

Replace this text with your Auburn user id.

```
#you are asking for 4 node 8 processor each, 32 processors as a total for 30hrs
```

```
#after 30 hours your job will be killed
```

```
#PBS -l nodes=4:ppn=8,walltime=30:00:00
```

Designates number of processors used for computational power as well as the maximum number of hours used on the cluster until the job is terminated. These are optional changes depending on the computational requirements of the program.

```
#PBS -d /home/ker0017/fluent_test
```

Designates location within Hopper directory where run files are stored. The folder name must match the folder in which the files being run are located. All files, including mesh and fluent (.jou) files, must be stored in the same folder.

```
3J_D0_SE75e-06_Re5600-sol-SST.jou
```

Designates the name of the Fluent file that you are running. Be sure that the mesh file used in the fluent file is located within the same folder in the Hopper directory.

Appendix A.5 MATLAB Code for generating HTC images

```
clear all
clc
clf

% Heat Transfer Coefficient Contour Plot

x = xlsread('Contour_A5_Re8400','A5_P3_Re8400','A2:A468005');
y = xlsread('Contour_A5_Re8400','A5_P3_Re8400','B2:B468005');
z = xlsread('Contour_A5_Re8400','A5_P3_Re8400','F2:F468005');
fprintf('finished reading Excel document\n')

xmin = min(x); xmax = max(x); dx = xmax - xmin;
ymin = min(y); ymax = max(y); dy = ymax - ymin;
zmin = min(z); zmax = max(z);
steps = 500;
nx = round(steps*(dx/dy));
ny = steps;

xd = linspace(xmin,xmax,nx);
yd = linspace(ymin,ymax,ny);
[xi,yi] = meshgrid(xd,yd);
fprintf('finished meshgrid of xi and yi\n')
zi = griddata(x,y,z,xi,yi,'cubic');
fprintf('Finished making zi\n')
%%
[C,h] = contourf(xi,yi,zi./1000,256);
w = h.LineStyle;
h.LineStyle = 'none';
title('Heat Transfer Coefficient Contour for Angle = 5 Pitch = 3, Re =
8400','FontSize',12)
xlabel(' + Upstream           -
Downstream','FontSize',14,'FontWeight','bold')
axis([xmin 0.05 ymin ymax])
xlabels = linspace(xmin,0.05,10);
ylabels = linspace(ymin,ymax,5);
set(gca,'XTick',xlabels)
set(gca,'YTick',ylabels)
set(gca,'FontSize',10)
colormap('jet')
caxis([0 40])
c = colorbar;
c.Label.String = 'Heat Transfer Coefficient (kW/K*m^2)  ';
c.Label.FontSize = 11;
c.Label.FontWeight = 'bold';

fprintf('Finished making plot\n')
fig = gcf;
fig.PaperUnits = 'inches';
fig.PaperPosition = [0 0 8 2.5];
saveas(gcf,'HTC_A5_P3_R8400_Contour_lim50','png')
fprintf('Finished saving plot to a png\n')
```

Appendix B: Transport Equations for Transition SST Model

Appendix B.1 Transport Equations from Two-Equation Models

The transport equations for the modified turbulent kinetic energy, k , and the specific dissipation rate, ω , are:

$$\frac{\partial}{\partial t}(\rho k) + \frac{\partial}{\partial x_i}(\rho k u_i) = \frac{\partial}{\partial x_j} \left(\Gamma_k \frac{\partial k}{\partial x_j} \right) + G_k^* - Y_k^* + S_k, \quad (B.1)$$

$$\frac{\partial}{\partial t}(\rho \omega) + \frac{\partial}{\partial x_i}(\rho \omega u_i) = \frac{\partial}{\partial x_j} \left(\Gamma_\omega \frac{\partial \omega}{\partial x_j} \right) + G_\omega - Y_\omega + D_\omega + S_\omega, \quad (B.2)$$

Where \tilde{G}_k represents the modified production of turbulent kinetic energy, G_ω represents the production of ω , Γ_k and Γ_ω represent the effective diffusivities of k and ω , Y_k and Y_ω represent the dissipation of k and ω , D_ω represents the cross-diffusion term, and S_k and S_ω are user-defined source terms.

Appendix B.1.1 Effective Diffusivity

The effective diffusivities, Γ_k and Γ_ω , are given by:

$$\Gamma_k = \mu + \frac{\mu_t}{\sigma_k}, \quad (B.3)$$

$$\Gamma_\omega = \mu + \frac{\mu_t}{\sigma_\omega}, \quad (B.4)$$

where the turbulent viscosity, μ_t combines the k and ω as shown:

$$\mu_t = \frac{\rho k}{\omega} \frac{1}{\max \left[\frac{1}{\alpha^*}, \frac{SF_2}{a_1 \omega} \right]}, \quad (B.5)$$

where S is the strain rate magnitude

$$\alpha^* = \alpha_\infty^* \left(\frac{\alpha_0^* + (Re_t/R_k)}{1 + (Re_t/R_k)} \right), \quad (B.6)$$

$$\alpha_0^* = \frac{\beta_i}{3}, \quad (B.7)$$

$$\beta_i = F_1 \beta_{i,1} + (1 - F_1) \beta_{i,2}, \quad (B.8)$$

$$Re_t = \frac{\rho k}{\mu \omega}, \quad (B.9)$$

$$F_2 = \tanh(\phi_2^2), \quad (B.10)$$

$$\phi_2 = \max \left[2 \frac{\sqrt{k}}{0.09 \omega y}, \frac{500 \mu}{\rho y^2 \omega} \right], \quad (B.11)$$

$$\sigma_k = \frac{1}{F_1 / \sigma_{k,1} + (1 - F_1) / \sigma_{k,2}}, \quad (B.12)$$

$$\sigma_\omega = \frac{1}{F_1 / \sigma_{\omega,1} + (1 - F_1) / \sigma_{\omega,2}}, \quad (B.13)$$

The blending function, F_1 is given by:

$$F_1 = \tanh(\phi_1^4), \quad (B.14)$$

$$\phi_1 = \min \left[\max \left(\frac{\sqrt{k}}{0.09 \omega y}, \frac{500 \mu}{\rho y^2 \omega} \right), \frac{4 \rho k}{\sigma_{\omega,2} D_\omega^+ y^2} \right], \quad (B.15)$$

$$D_\omega^+ = \max \left[2 \rho \frac{1}{\sigma_{\omega,2} \omega} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j}, 10^{-10} \right], \quad (B.16)$$

Where y is the distance to the next surface and D_ω^+ is the positive portion of the cross diffusion term defined in section B.1.4.

Appendix B.1.2 Production terms of k and ω

The modified production of k is given as:

$$G_k^* = \gamma_{eff} \tilde{G}_k, \quad (B.17)$$

The production term for k , \tilde{G}_k , is given by:

$$\tilde{G}_k = \min(G_k, 10 \rho \beta^* k \omega), \quad (B.18)$$

$$G_k = \mu_t S^2, \quad (B.19)$$

$$\beta^* = \beta_\infty^* \left(\frac{4/15 + (Re_t / R_\beta)^4}{1 + (Re_t / R_\beta)^4} \right), \quad (B.20)$$

The production term for ω , G_ω , is given by:

$$G_\omega = \frac{\alpha \alpha^*}{\nu_t} G_k, \quad (B.21)$$

$$\alpha = \frac{\alpha_\infty}{\alpha^*} \left(\frac{\alpha_0 + (Re_t/R_\omega)}{1 + (Re_t/R_\omega)} \right), \quad (B.22)$$

$$\alpha_\infty = F_1 \alpha_{\infty,1} + (1 - F_1) \alpha_{\infty,2}, \quad (B.23)$$

$$\alpha_{\infty,1} = \frac{\beta_{i,1}}{\beta_\infty^*} - \frac{\kappa^2}{\sigma_{\omega,1} \sqrt{\beta_\infty^*}}, \quad (B.24)$$

$$\alpha_{\infty,2} = \frac{\beta_{i,2}}{\beta_\infty^*} - \frac{\kappa^2}{\sigma_{\omega,2} \sqrt{\beta_\infty^*}}, \quad (B.25)$$

Appendix B.1.3 Dissipation terms of k and ω

The modified k dissipation term, Y_k^* is

$$Y_k^* = \min[\max(\gamma_{eff}, 0.1), 1.0] Y_k, \quad (B.26)$$

The dissipation terms, Y_k and Y_ω , are given by:

$$Y_k = \rho \beta^* k \omega, \quad (B.27)$$

$$Y_\omega = \rho \beta \omega^2, \quad (B.28)$$

$$\beta = \beta_i \left[1 - \frac{\beta^*}{\beta_i} \zeta^* F(M_t) \right], \quad (B.29)$$

Where the compressibility function, $F(M_t)$ is given by:

$$F(M_t) = \begin{cases} 0 & M_t \leq M_{t0} \\ M_t^2 - M_{t0}^2 & M_t > M_{t0} \end{cases}, \quad (B.30)$$

$$M_t^2 \equiv \frac{2k}{a^2}, \quad (B.31)$$

$$a = \sqrt{\gamma RT}, \quad (B.32)$$

Appendix B.1.4 Cross-Diffusion Modification

The k - ω model and the k - ε model are blended using the cross-diffusion term, defined as:

$$D_\omega = 2(1 - F_1) \rho \frac{1}{\omega \sigma_{\omega,2}} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j}, \quad (B.33)$$

Appendix B.2 Transport Equations for Intermittency and Transition Momentum

Thickness

The transport equation for the intermittency, γ is defined as:

$$\frac{\partial(\rho\gamma)}{\partial t} + \frac{\partial(\rho U_j \gamma)}{\partial x_j} = P_{\gamma 1} - E_{\gamma 1} + P_{\gamma 2} - E_{\gamma 2} + \frac{\partial}{\partial x_j} \left[\left(\mu + \frac{\mu_t}{\sigma_\gamma} \right) \frac{\partial \gamma}{\partial x_j} \right], \quad (B.34)$$

with the transition and destruction/relaminarization sources defined as:

$$P_{\gamma 1} = C_{a1} F_{length} \rho S [\gamma F_{onset}]^{c_{\gamma 3}}, \quad (B.35)$$

$$E_{\gamma 1} = C_{e1} P_{\gamma 1} \gamma, \quad (B.36)$$

$$P_{\gamma 2} = C_{a2} \rho \Omega \gamma F_{turb}, \quad (B.37)$$

$$E_{\gamma 2} = C_{e2} P_{\gamma 2} \gamma, \quad (B.38)$$

where F_{length} is an empirical correlation defined in section 3.3.5, Ω is the vorticity magnitude, and the transition onset is controlled by:

$$F_{onset} = \max(F_{onset2} - F_{onset3}, 0), \quad (B.39)$$

$$F_{onset1} = \frac{Re_\nu}{2193 Re_{\theta c}}, \quad (B.40)$$

$$F_{onset2} = \min(\max(F_{onset1}, F_{onset1}^4), 2.0) \quad (B.41)$$

$$F_{onset3} = \max\left(1 - \left(\frac{R_T}{25}\right)^3, 0\right), \quad (B.42)$$

$$F_{turb} = e^{-\left(\frac{R_T}{4}\right)^4}, \quad (B.43)$$

$$Re_\nu = \frac{\rho y^2 S}{\mu}, \quad (B.44)$$

$$R_T = \frac{\rho k}{\mu \omega}, \quad (B.45)$$

The transport equation for the transition momentum thickness Reynolds number, $\widetilde{Re}_{\theta t}$ is:

$$\frac{\partial(\rho \widetilde{Re}_{\theta t})}{\partial t} + \frac{\partial(\rho U_j \widetilde{Re}_{\theta t})}{\partial x_j} = P_{\theta t} + \frac{\partial}{\partial x_j} \left[\sigma_{\theta t} (\mu + \mu_t) \frac{\partial \widetilde{Re}_{\theta t}}{\partial x_j} \right], \quad (B.46)$$

with the source term defined as:

$$P_{\theta t} = c_{\theta t} \frac{\rho}{t} (Re_{\theta t} - \widetilde{Re}_{\theta t}) (1 - F_{\theta t}), \quad (B.47)$$

$$t = \frac{500\mu}{\rho U^2}, \quad (B.48)$$

$$F_{\theta t} = \min \left(\max \left(F_{wake} e^{(-\frac{y}{\delta})^4}, 1 - \left(\frac{\gamma - 1/50}{1 - 1/50} \right)^2 \right), 1.0 \right), \quad (B.49)$$

$$F_{wake} = e^{-\left(\frac{Re_{\omega}}{1E+5}\right)^2}, \quad (B.50)$$

$$Re_{\omega} = \frac{\rho \omega y^2}{\mu}, \quad (B.51)$$

$$\delta = \frac{500y}{U} \delta_{BL}, \quad (B.52)$$

$$\delta_{BL} = 7.5\theta_{BL}, \quad (B.53)$$

$$\theta_{BL} = \frac{\widetilde{Re}_{\theta t} \mu}{\rho U}, \quad (B.54)$$

Appendix B.3 Separation-Induced Transition Correction

The modification for separation-induced transition is:

$$\gamma_{eff} = \max(\gamma, \gamma_{sep}), \quad (B.55)$$

$$\gamma_{sep} = \min \left(C_{s1} \max \left[\left(\frac{Re_{\nu}}{3235 Re_{\theta c}} \right) - 1.0 \right] F_{reattach}, 2 \right) F_{\theta t}, \quad (B.56)$$

$$F_{reattach} = e^{-\left(\frac{R_T}{20}\right)^4}, \quad (B.57)$$

Appendix B.4 Empirical Correlations

The Transition SST model contains three empirical correlations: the transition onset as observed in experiments, $Re_{\theta t}$, the length of the transition zone, F_{length} , and $Re_{\theta c}$ is the point where the model is activated to match both $Re_{\theta t}$ and F_{length} . These empirical correlations are defined by

Langty and Menter [27] as shown:

$$Re_{\theta t} = f(Tu, \lambda_{\theta}), \quad (B.58)$$

$$F_{length} = f(\widetilde{Re}_{\theta t}), \quad (B.59)$$

$$Re_{\theta c} = f(\widetilde{Re}_{\theta t}), \quad (B.60)$$

with the local turbulent intensity, Tu , defined as:

$$Tu = \frac{100}{U} \sqrt{\frac{2}{3}k}, \quad (B.61)$$

where k is the turbulent kinetic energy.

The Thwaites' pressure gradient coefficient is defined as:

$$\lambda = (\theta^2/\nu) \frac{dU}{ds}, \quad (B.62)$$

Appendix C: Collection of Raw HTC Contour Images

