COUPLED INVISCID-VISCOUS SOLUTION METHODOLOGY FOR BOUNDED DOMAINS: APPLICATION TO DATA CENTER THERMAL MANAGEMENT

A Dissertation
Presented to
The Academic Faculty

by

Ethan E. Cruz

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Mechanical Engineering

Georgia Institute of Technology
December 2015

COUPLED INVISCID-VISCOUS SOLUTION METHODOLOGY FOR BOUNDED DOMAINS: APPLICATION TO DATA CENTER THERMAL MANAGEMENT

Approved by:

Dr. Yogendra K. Joshi, Advisor School of Mechanical Engineering *Georgia Institute of Technology*	Dr. Lakshmi N. Sankar School of Aerospace Engineering *Georgia Institute of Technology*
Dr. G. Paul Neitzel School of Mechanical Engineering *Georgia Institute of Technology*	Dr. Paul S. Krueger Department of Mechanical Engineering *Southern Methodist University*
Dr. Satish Kumar School of Mechanical Engineering *Georgia Institute of Technology*	Dr. Roger R. Schmidt Systems and Technology Group *International Business Machines*

Date Approved: August 19, 2015

Although to penetrate into the intimate mysteries of nature and thence to learn the true causes of phenomena is not allowed to us, nevertheless it can happen that a certain fictive hypothesis may suffice for explaining many phenomena.

-Leonard Euler (1707-1783)

To my wife, Ryan.
Without her love, patience, and understanding,
I would be lost forever.

ACKNOWLEDGEMENTS

I wish to thank my advisor, Professor Joshi, for his patience, encouragement, and guidance throughout the entire process. I am especially grateful for his permission to complete the work in absentia. I express my sincere gratitude to my PhD reading committee, Professors Neitzel, Kumar, Sankar, and Krueger, and Dr. Schmidt, for their time spent reviewing the dissertation, their support, and their helpful comments.

I would also like to thank my manager at IBM, Peter W. Kelly, for his support in pursuing my graduate education, and for his vital role in gaining the funding that allowed me to return to school full time. I also appreciate all of the encouragement of the late IBM Fellow, Dr. Dick Chu, who exhorted me to not just pursue this degree, but to "create something they will name after you." I wish to thank all of my colleagues at IBM for their patience and understanding. I would like to especially thank Dr. Vic Mahaney for picking up the slack that I invariably left as I finished the dissertation.

Although my residency at Georgia Tech was relatively short, I would like to acknowledge all of my professors and fellow students with whom I had the privilege of working. I would also like to thank the Woodruff School of Mechanical Engineering administrative staff, especially Glenda Johnson, who helped me with numerous issues that arose throughout my studies.

Finally, I would like to thank my wife, Ryan, my two children, Tino and Taryn, my parents, Eddie and Jean, my in-laws, Russ and Deb Miller, and my dear friends, Peter and Paulette Bonk, Jon and Lisa Ellingworth, and Dr. Gregory Durgin for all of their

love, patience, support, and encouragement throughout the years. Without them, my completion of this degree would never have been possible.

TABLE OF CONTENTS

Page

CHAPTER

LIST OF TABLES

Page

LIST OF FIGURES

LIST OF SYMBOLS

A	Area
c_p	Specific heat
C_μ	Empirical constant
d_{cell}	Length of the interface region in grid cells
d_{distance}	Length of the interface region in linear distance
e_a^{21}	Approximate discretization error
e_{ext}^{21}	Extrapolated relative discretization error
E	Sum of the internal and kinetic energy
g	Gravitational constant
h	Sensible enthalpy
i, j	Indices
I	Turbulent intensity
k	Thermal conductivity, or turbulent kinetic energy
K	Pressure loss coefficient
ℓ	Length scale
L	Characteristic length
n	Number of iterations of the partitioning algorithm
$\hat{n}$	Unit vector in normal direction
N	Maximum number of iterations of the partitioning algorithm
p	Static pressure
P	Percentile cutoff
r_{21}	Refinement factor from grid 2 to grid 1

R	Viscous region cell radius parameter, or residual
Ri	Richardson number
S	Volumetric heat sources
S_{MBF}	Modified body force source
t	Time
T	Static temperature
T_{ext}^{21}	Extrapolated temperature
u	Velocity
x	Cartesian coordinate
α	Under-relaxation factor
β	Thermal expansion coefficient
χ	Variable for determining viscous regions
ε	Turbulent dissipation
ε_{max}	Mass balance stopping criterion
ϕ	Variable, mass flux, or temperature
η	Ratio of the total number of viscous cells to the total number of cells
μ	Molecular viscosity
μ_t	Turbulent viscosity
ρ	Density
τ	Averaging interval
ω	Vorticity
ξ	Mass balance correction factor
$\forall$	Volume of the momentum source

LIST OF ABBREVIATIONS

ASHRAE	American Society of Heating, Refrigerating and Air Conditioning Engineers
CDU	Cooling Distribution Unit
CFD	Computational Fluid Dynamics
CFD/HT	Computational Fluid Dynamics and Heat Transfer
CIVSM	Coupled Inviscid-Viscous Solution Method
CRAC	Computer-Room Air-Conditioning
DCL	Data Center Laboratory
GCI	Grid Convergence Index
IT	Information Technology
MBF	Modified Body Force
PDU	Power Distribution Unit
PIV	Particle Image Velocimetry
PJ	Porous Jump
POD	Proper Orthogonal Decomposition
RANS	Reynolds-Averaged Navier-Stokes
RMS	Root Mean Squared
RSS	Root of the Sum of the Squares
SIMPLE	Semi-Implicit Method for Pressure Linked Equations
VSD	Variable Speed Drive

SUMMARY

Computational fluid dynamics and heat transfer (CFD/HT) models have been employed as the dominant technique for the design and optimization of both new and existing data centers. Inviscid modeling has shown great speed advantages over the full Navier-Stokes CFD/HT models (over 20 times faster), but is incapable of capturing the physics in the viscous regions of the domain. A coupled inviscid-viscous solution method (CIVSM) for bounded domains has been developed in order to increase both the solution speed and accuracy of CFD/HT models. The methodology consists of an iterative solution technique that divides the full domain into multiple regions consisting of at least one set of viscous, inviscid, and interface regions. The full steady, Reynolds-Averaged Navier-Stokes (RANS) equations with turbulence modeling are used to solve the viscous domain, while the inviscid domain is solved using the Euler equations. By combining the increased speed of the inviscid solver in the inviscid regions, along with the viscous solver's ability to capture the turbulent flow physics in the viscous regions, a faster and potentially more accurate solution can be obtained for bounded domains that contain inviscid regions which encompass more than half of the domain, such as data centers.

CHAPTER 1

INTRODUCTION

The pervasive trend of increasing heat flux and power dissipation of Information Technology (IT) equipment [1], has created a significant challenge for air-cooled data centers, which can contain as many as several thousand pieces of IT equipment. In order to maintain high reliability, this equipment must be supplied with adequate cooling air according to the manufacturers' specifications. As the power dissipation increases, so do the cooling requirements, necessitating higher IT equipment and cooling supply air-flow rates. This has led to complex flow patterns and temperature distributions within these data centers. In order to better understand these and to reduce mixing of hot and cold air streams, which degrades cooling efficiency, computational fluid dynamics and heat transfer (CFD/HT) modeling has been employed as the dominant technique for the design and optimization of both new and existing data centers. With as much as 1.5% of the world's and 2.2% of the United States' electrical power going to data centers, and roughly half of that used for cooling those data centers [2], optimizing the cooling systems of data centers for minimized power consumption has become an industry imperative.

1.1 Design of Data Centers

Data centers are large computer rooms which are used to house and secure IT equipment such as computational servers, storage devices, and networking equipment. Each piece of IT equipment within the data center must be supplied with adequate cooling resources, typically using air and water as the heat transfer fluids, in order to maintain high reliability of the IT equipment. The vast majority of IT equipment is air-cooled, although some IT equipment use water-cooling while still others employ dielectric fluids as the cooling fluid.

1.1.1 Air-Cooling Configurations

In air-cooled data centers, IT equipment is typically placed into hot and cold aisle layouts [3-5]. Most IT equipment racks employ front to back cooling which, when aligned in hot and cold aisles, helps separate the hot exhaust air from the cold inlet air [3]. Figure 1 shows an example of this arrangement for a raised floor design and how the cooling air from the perforated tiles in the cold aisle is somewhat isolated from the hot exhaust air exiting the back of the IT equipment racks in the hot aisle.

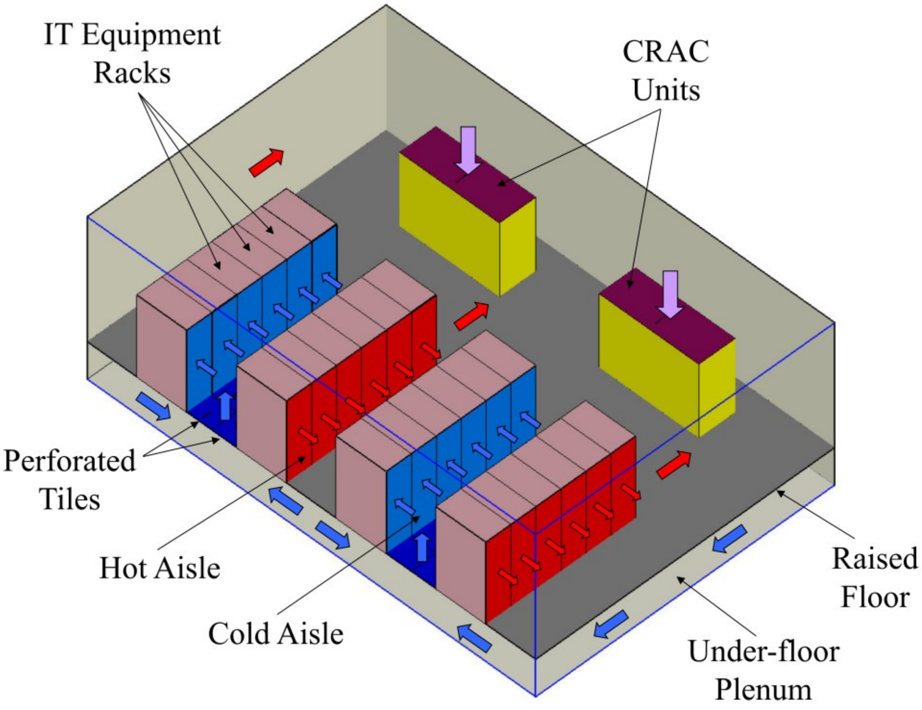


Figure 1. Hot and Cold Aisle Raised Floor Data Center Layout

Computer-room air-conditioning (CRAC) units provide cooling air to the cold aisles by extracting, chilling, and routing (preferably hot) air from the data center. In raised floor designs, as shown in figure 1, the CRAC units pressurize the under-floor plenum with chilled air that then exits perforated floor tiles in the cold aisles. Although less common, multiple other forms of both raised floor and non-raised floor designs also exist [3,4].

1.1.2 Air-Flow Velocities and Reynolds Numbers

IT equipment air-flow rates vary considerably based on their design, configuration, heat load, and air inlet temperatures. Typical air inlet velocities for IT equipment range from 0.5 to 1.0 m/s where the air-flow is evenly distributed across the face of the inlets, while exhaust velocities can reach as high as 25 m/s at the concentrated exhausts of air moving devices. When using the width of the inlet and exhaust as the characteristic length, these velocities correspond to Reynolds numbers on the order of 2×10^4 for IT equipment inlets, and as high as 2×10^5 for the concentrated exhausts.

Perforated-tile velocities, averaged over the entire tile area, are typically around 0.3 to 0.6 m/s with some tiles reaching as high as 1.3 m/s [5,6]. Using the perforated-tile width as the characteristic length, the Reynolds numbers are on the order of 1×10^4 to 5×10^4. Localized perforated-tile discharge velocities can reach 7.4 m/s [6], or Reynolds numbers of 5×10^3 when using the diameter of the perforation as the characteristic length. CRAC inlet (exhaust from the room) velocities typically range from 1.0 to 3.0 m/s and average about 2.2 m/s. This corresponds to Reynolds numbers on the order of 1×10^5 when using the width of the CRAC as the characteristic length.

1.1.3 Air Temperatures

IT equipment manufacturers specify inlet air temperatures for their products and generally follow the guidelines defined by the American Society of Heating, Refrigerating and Air Conditioning Engineers (ASHRAE) [4]. As a result, data center temperatures in the cold aisle are prescribed to a range of 10 to 35°C for most IT equipment (class A2) and up to a range of 5 to 45°C (class A4). A temperature difference of 10°C across both IT equipment and CRAC units is typical while some high powered IT equipment can reach differences as high as 20°C [3]. In practice temperatures within data centers range from a minimum of 10°C at the inlet of the IT equipment to a maximum of 55°C at the exhaust of the IT equipment.

1.1.4 Cooling Energy Efficiency

One of the easiest ways of cutting energy consumption used to cool data centers is reducing the amount of cold and hot air mixing. By allowing the cold and hot air to mix, CRAC units must sub-cool the air in order to supply adequately cool air to the IT equipment. The placement of the perforated tiles, CRAC units, and IT equipment is critical to optimizing the cooling of data centers for energy efficiency. CFD/HT and other modeling techniques have proven to be valuable tools in optimizing data center cooling energy efficiency.

1.2 Data Center Numerical Modeling

Numerical modeling of data centers allows for the prediction of air-flow patterns and the potential for cold and hot air mixing which leads to inefficiencies and increased power consumption for cooling the data center. Different methods have been employed including Reynolds-Averaged Navier-Stokes (RANS) CFD/HT, ad hoc methods, Proper Orthogonal Decomposition (POD), inviscid and potential flow methods, and various combinations of these methods.

1.2.1 Reynolds-Averaged Navier-Stokes

A number of researchers [3,5,7-16] have used the standard k-ε turbulence model to simulate data centers using different CFD/HT RANS solvers. It has become the most commonly used turbulence model in data center analysis [17], although it has neither proven to be the most accurate [13-15,17] nor the most computationally efficient [13-15]. Patel et al. [7] found an error ranging from 7 to 12% in temperature predictions for an overhead return, raised floor data center test lab.

Shrivastava et al. [8] compared a CFD/HT model to an actual 690 m^2 (7,400 ft^2) data center housing over 130 IT equipment racks and found a mean absolute rack inlet air temperature error of 4°C, with a standard deviation of 3.3°C. Schmidt et al. [9] also compared a CFD/HT model to measured temperature data from a data center. While the general trends in the IT equipment inlet air temperatures were reasonably predicted, there were deviations of more than 30°C between the measured and predicted values. Amemiya et al. [10], using a three-dimensional temperature mapping tool [18], found that the transient temperature fluctuations within the data center affected the temperature mapping results due to the differing time scales between those transient fluctuations and the time it took to completely map the data center. Temperature spikes of over 2°C were discovered at both the IT equipment and the chiller units, raising the uncertainty of the temperature measurements.

1.2.2 Reduced Order Modeling

CFD/HT modeling tools are quite useful in providing a high-level understanding of the flow patterns and temperature distributions in a data center. Simple parametric models can be built and run in a number of different layouts, with multiple case studies in a fraction of the time it takes to construct or rearrange a data center. Although increasing the potential accuracy, more elaborate and finer grid models require increased computational effort which can severely limit the number of runs, and subsequently limit the ability to find an optimized cooling design.

In order to increase computational efficiency, reduced order modeling techniques have been used such as ad hoc methods [19-22], POD [23], and various inviscid and potential flow methods [15,16,24,25]. In general, these methods increase the computational efficiency relative to traditional CFD/HT RANS modeling methods, but at the cost of modeling accuracy. Combinations of these methods are also being studied in order to increase modeling accuracy [26-30].

The reduced order models based on ad hoc methods rely on measurements in order to predict air temperatures at specific locations within a data center. These methods require the compilation of large amounts of experimental data in order to create a computational model. Although able to help predict how the specific data center will react to changes in certain parameters, these methods are not able to provide timely insight into the construction of new, or rearrangement of existing data centers [19-22].

POD based algorithms have proven quite effective in reducing the computational effort, along with providing reasonably accurate results at the rack scale, but have not been able to provide accurate results at the room scale [23]. In combination with ad hoc methods, a POD model has shown both increased computational speed and accuracy for a specific data center layout [26]. Although future improvements to this method could reduce or eliminate these limitations, the combined method is currently incapable of predicting the effects of rearranging existing data centers, or in optimizing the construction of new ones without new observations.

1.2.3 Inviscid Methods

Inviscid and potential flow reduced order modeling techniques are the most similar to the full Navier-Stokes CFD/HT models and in some cases are simply a subset of the same equations. The inviscid (Euler) equations are a subset of the full Navier-Stokes equations obtained through the elimination of the viscous terms. The potential flow models require the flow to be irrotational, which implies that the flow is also inviscid. Both of these simplified modeling techniques allow for significantly smaller grid counts and therefore faster solution times, but don’t necessarily capture all of the physics in a complex, turbulent flow field. For large rooms with high Reynolds-number flows such as those found in many data centers, it may be worth the trade-off of loss of accuracy for the guaranteed reduction in solution time.

Toulouse et al. [24] explored the use of a finite difference solver based on the potential flow equations. An experimental validation was performed on a modified version of this solver [25], which showed large deviations between predicted and measured temperatures. A later study [16] used the method of vortex superposition to modify the original model to account for the effects of buoyancy, which were thought to be the cause of the previous inaccuracies. This new model was then “tuned” using temperature data collected in the modeled data center. Three models were used to compare measured temperatures in a data center: the new “optimized” model with superimposed Rankine vortices, the original “basic” potential flow model, and a CFD/HT model using the k-ε turbulence model. The CFD/HT model produced the lowest overall temperature deviations to measured data, while the “optimized” model produced the lowest rack inlet temperature deviations. The low deviations at the inlet were expected for the “optimized” model since that was the location the model was “tuned” to produce the best results. While providing a significant reduction in computational effort, the “optimized” model may need to be “tuned” for different geometries of data centers in order to be able to limit the deviations.

1.2.4 Hybrid Methods

There are a few other examples of combining potential flow modeling with other techniques. Most notable are the combinations of potential flow and ad hoc methods [27-30] for the modeling of data centers. Hamann et al. [28] explored the zones that each CRAC creates within the data center using sensor data, along with a potential flow model. Das et al. [29] expanded on this potential flow modeling, incorporating sensor data to predict and optimize CRAC thermal zones in order to minimize power consumption. Lopez and Hamann [30] considered the number of ad hoc data points needed for an accurate potential flow model.

In order to evaluate the performance of helicopter rotors, Sankar et al. [31] and Berezin and Sankar [32] developed a closely coupled potential flow and full Navier-Stokes modeling technique. When compared with a full Navier-Stokes model, this technique reduced the computational effort by 40-50% without reducing the accuracy. However, the method relies heavily on the a priori knowledge of the shape and approximate location of the inviscid-viscous boundary for the flow around the rotor. Two regions are formed: a viscous region near the rotor that employs the full Navier-Stokes solver, and an inviscid region a distance away from the rotor that employs the potential flow solver. The two regions are solved iteratively, and are coupled via an interface “surface”, which provides the boundary conditions for the two regions, and allows them to interact with one another.

1.3 Preview of the Present Work

This research is primarily focused on reducing the solution time of data center fluid-dynamic models, while maintaining the accuracy of traditional RANS CFD/HT modeling techniques. Reductions in solution time are necessary for better optimization of data center layouts which ultimately lead to energy conservation and reduced operational costs.

Chapter 2 presents a comparison of traditional RANS CFD/HT models to experimental data from a single-rack data center test cell. The temperature field prediction capabilities of different turbulence models as well as the laminar and inviscid model are evaluated. Chapter 3 introduces a novel coupled inviscid-viscous solution method (CIVSM) for bounded domains. Both the RANS equations for the viscous domains and the Euler equations for the inviscid domains are presented. The chapter also includes an example of the method applied to the single-rack data center test cell from chapter 2. Chapter 4 performs a thermal verification of a modified version of the CIVSM modeling technique on the computational section of the multi-rack Data Center Laboratory (DCL) at Georgia Tech. It compares the IT equipment inlet temperature predictions from the inviscid solver, the CIVSM, and traditional RANS CFD/HT using the k-ε turbulence model with measured data from 13 powered racks and 71 pieces of IT equipment.

Chapter 5 verifies the flow predictions of the CIVSM, which employs one additional change to the updated version that was used in chapter 4. It compares flow measurements from particle image velocimetry (PIV) with flow predictions from the inviscid solver, the CIVSM, and traditional RANS CFD/HT using the k-ε turbulence model. Only one rack is powered for the flow measurements, which were collected on the experimental section of the multi-rack DCL at Georgia Tech. Finally, future work, recommendations and conclusions are outlined in chapter 6.

CHAPTER 2

MODELING OF A SINGLE-RACK DATA CENTER TEST CELL

The focus of this chapter is to analyze the use of an inviscid CFD/HT model and compare its performance to that of more conventional viscous CFD/HT employing different turbulence models. The performance of the various models is assessed partly on the root mean squared (RMS) difference in temperature between the models' temperature field and the data measured in a small, single-rack data center test cell. The other performance criterion is the computational effort used to obtain the predicted temperature and flow fields.

2.1 Experimental Setup of the Single-Rack Data Center Test Cell

A small, single-rack data center test cell was constructed at the IBM site in Poughkeepsie, New York in order to compare three-dimensional temperature data to the numerical results of CFD/HT [11]. Figure 2 shows a picture of the single-rack data center test cell, specifically looking at the area of interest near the single IT equipment rack. The test cell was reasonably isolated from the rest of the data center both above and below the raised floor. Sheets of plastic were used above the raised floor while cardboard was used underneath the raised floor in the supply air plenum in order to minimize air leakage to the surrounding data center floor. This limited the test area to a control volume of 84 m^2 (900 ft^2) floor area with a 3 m (10 ft) ceiling and 0.43 m (17 in.) under-floor plenum.

Figure 3 depicts a plan view while figure 4 displays an isometric view of the single rack data center test cell. The room contained four IT equipment simulators housed in one rack, one CRAC unit, and depending on the configuration one, two, or three 27% open perforated tiles located in front of the IT equipment simulator. Bypass flow was provided for by removing five tiles on the opposite side of the CRAC from the IT equipment simulator. This allowed for more realistic under floor static pressure and flow through the perforated tiles in front of the simulator.

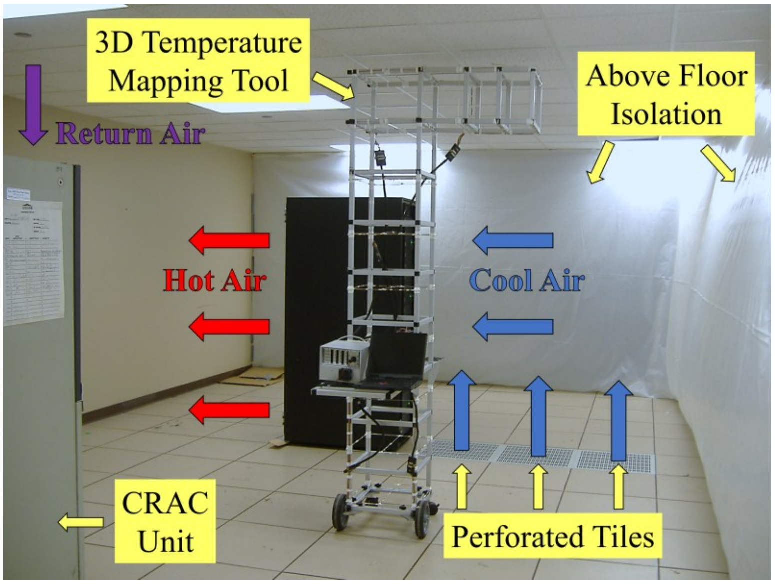


Figure 2. Photograph of the Region of Interest in the Single-Rack Data Center Test Cell

The CRAC unit was rated for 70 kW cooling capacity and provided 4.81 m^3/s (10,200 CFM) of cool air to the single-rack data center test cell. The IT equipment simulator was set to 11 kW or 23 kW and had a volumetric flow rate of 1.13 m^3/s (2,400 CFM). The perforated tiles provided a total of 1.12 m^3/s (2,380 CFM); 0.80 m^3/s (1,700 CFM); and 0.496 m^3/s (1050 CFM) of cooling air or roughly 99%, 71%, and 44% of the simulator volumetric flow rate for the three, two, and one tile cases, respectively.

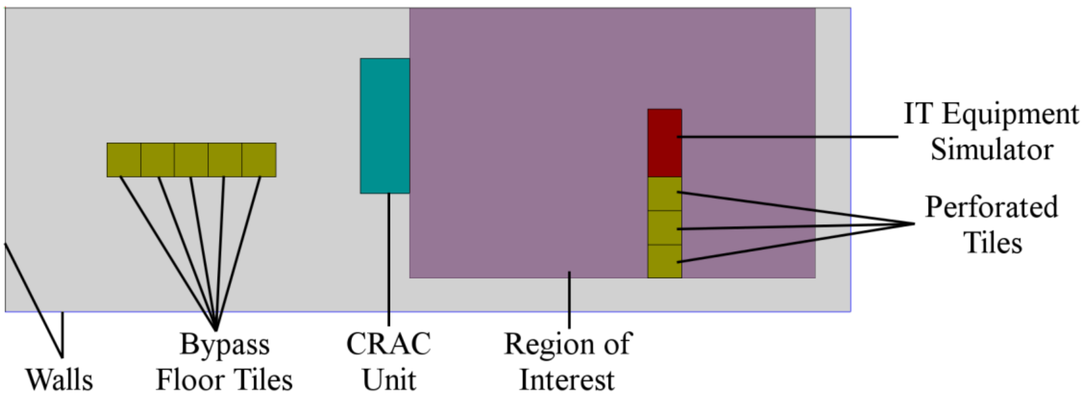


Figure 3. Plan View of the Single-Rack Data Center Test Cell

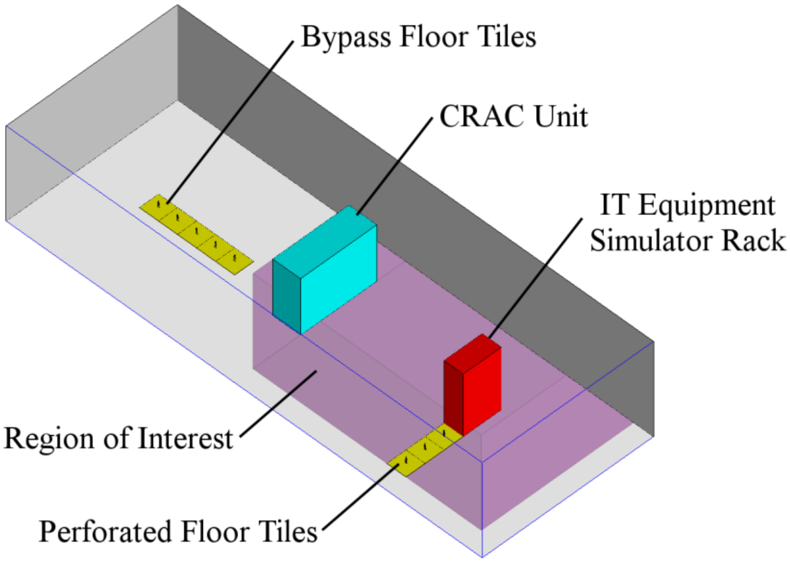


Figure 4. Isometric View of the Single-Rack Data Center Test Cell

A three-dimensional temperature mapping scheme was utilized to collect the experimental temperature data in a region near the IT equipment simulator. The region of interest, shown in figures 3 and 4, encompasses most of the area between the CRAC unit and bounding walls of the test cell on the IT equipment simulator side of the test cell. The three-dimensional temperature-mapping tool used to create the temperature map can be seen in figure 2 between the CRAC and the IT equipment simulator. The thermal time constant for the thermocouple temperature sensors was determined to be less than two seconds [11]. Nine thermocouple sensors were placed on each plane in a 3×3 equally spaced array, which gave a spatial resolution of 0.203 m (8 in.) in the plane, and 0.305 m (12 in.) out of the plane. Additional details on the single-rack data center test cell including the isolation scheme and measurement techniques can be found in [11] and more information on the three-dimensional mapping tool can be found in [18].

2.1.1 Experimental Uncertainty

A temperature measurement uncertainty of ±1°C was estimated for the individual thermocouples used in the three-dimensional temperature mapping tool [10,11,13,14,18]. However, Hamann et al. [18] found the maximum total measurement difference of the three-dimensional mapping tool to be less than 2.5°C (i.e. ±1.25°C). This difference included the air-flow disruption of the tool in a uniform hot air stream, combined with the ±1°C thermocouple temperature reading uncertainty mentioned above. This suggests a ±1.25°C total uncertainty for the three-dimensional temperature mapping tool.

Temperature fluctuations within the room of as much as 2.5°C (i.e. ±1.25°C) during the course of the three-dimensional temperature mapping were found. This variation is somewhat higher, but still consistent with the temperature spikes that were found in [10]. This gives a ±1.25°C total uncertainty for the room temperature measurement.

These temperature measurement uncertainties (i.e. the overall accuracy of the three-dimensional temperature mapping tool and the transients during the measurement process) along with positional uncertainty (i.e. placing the tool directly over the floor tile), measured power uncertainty (found to be within 4% for the 23kW cases [11], and within 10% for the 11 kW cases [14]), and boundary condition air-flow uncertainty combine to increase the overall measured temperature uncertainty. Using only the two ±1.25°C temperature uncertainties, combined by taking the root of the sum of the squares (the RSS combination method), nets an overall measured temperature uncertainty of ±1.8°C. This suggests that differences between experimental and numerical temperatures less than 1.8°C are within the experimental uncertainty of the measurements.

2.2 Numerical Model

Six CFD/HT models were generated using the commercially available ANSYS Fluent [33] software which employs the finite-volume method. The governing equations used in the analysis can be found in chapter 3.2. The six models were constructed to match the physical layout of the single-rack data center test cell with the three different floor tile arrangements. Each of the models consisted of six major components: a CRAC unit, an IT equipment rack, a set of perforated tiles in front of the IT equipment rack, bypass air-flow tiles, walls (including ceiling and floor), and an air-flow leakage source. Only the space above the raised floor was modeled, eliminating the added complexity of the under floor air plenum.

The inlet tile flow rates, both bypass and in front of the IT equipment rack, were modeled using the standard, porous jump model (see chapter 4.2.2 for more detail), and set to the measured values of the single-rack data center test cell as fixed, uniform air-flow boundary conditions. The IT equipment rack was modeled as a black box with either a 11 kW or 23 kW heat source with an air-flow of 1.13 m^3/s (2,400 CFM). The CRAC unit was also modeled as a black box exhausting 4.81 m^3/s (10,200 CFM) from the model as a second mass flow boundary condition. An air-flow leakage source was placed uniformly along the ceiling and provided the balance (i.e. the difference between the inlet tile flow rates and the CRAC exhaust flow rate) of the air-flow into the model. The air-flow leakage boundary condition was chosen based on the findings from [12] as providing the lowest RMS difference in the previous study of the same single-rack data center test cell. For the Spalart-Allmaras and all of the k-ε turbulence models, all of the domain inlets were set to have a turbulence intensity of 10% and a turbulent length scale of 0.061 m (2.4 in.), both of which were the default settings. No additional detail was added to the IT equipment rack since it was found in [12] to offer no benefit.

Figures 3 and 4 show the CFD/HT model domain, including the region of interest near the IT equipment rack where the temperature measurements were taken. The region contains 7,650 points providing the models' temperature monitor points, which are located in the same positions as the three-dimensional temperature mapping tools' data measurement thermocouples in the single-rack data center test cell. Both the temperature monitor point readings and experimental measurements were taken at nine heights in a 3×3 array for a total of 81 measurements per floor tile. An 8×12 array of 0.61 m × 0.61 m (2 ft × 2 ft) floor tiles, excluding the two tiles occupied by the IT equipment, provides the 7,650 data points. The lowest readings were at a height of 0.15 m (0.5 ft) and increased to 2.59 m (8.5 ft) in 0.305 m (1.0 ft) increments. The temperature monitor points are so densely packed that they would be indistinguishable if plotted in figure 3 or 4.

2.2.1 Flow Models

Nine viscous flow models were used, along with the inviscid flow model for the six different cases. The first two studies [13,14] focused on the nine viscous models which consisted of the laminar, zero-equation, Spalart-Allmaras, standard k-ε, realizable k-ε, enhanced k-ε, enhanced realizable k-ε, RNG k-ε, and enhanced RNG k-ε flow models. All of the turbulence models are of the RANS type, and all of the k-ε model variations add two transport equations to the solver, while the Spalart-Allmaras model adds one transport equation, and the zero-equation model, which as the name implies adds no additional transport equations. Due to the similarity in results of many of the turbulence models [13,14] only a subset of the turbulence models were included in the third study which also evaluated the inviscid model. Only the laminar, zero-equation, Spalart-Allmaras, and the standard k-ε model were retained [15]. A more detailed discussion on the different turbulence models can be found in Appendix A.

The Boussinesq approximation was implemented for all of the models. This allows buoyancy effects to be included in the momentum equations in the form of a body force while modeling the flow as being incompressible with constant density. The appropriate dimensionless number to determine if buoyancy effects in mixed convection can be neglected is the Richardson number, which is defined as:

$$Ri = \frac{g\beta \Delta T L}{u^2} \tag{1}$$

where g is the gravitational constant, β is the thermal expansion coefficient, ΔT is the temperature difference, L is the characteristic length, and u is the characteristic velocity. The Richardson number for this data center was calculated to be on the order of 0.4, which is greater than 0.1 and less than 10, meaning that both forced and natural convection effects are important. This value is typical of data centers where, in general, buoyancy effects should not be ignored.

2.2.2 Grid Studies

The six CFD/HT models consisted of much finer grids than were used in [11,12]. The total node count for the three grids were 248,148; 2,091,872; and 7,993,152 corresponding to a maximum individual control volume size of 1.33×10^{-3}, 1.25×10^{-4}, and 3.19×10^{-5} m^3 respectively, and a minimum control volume size of 6.20×10^{-4}, 1.04×10^{-4}, and 2.97×10^{-5} m^3 respectively. Grid studies were performed in [13,14] using the three different sizes to determine grid independence for each of the six CFD/HT models. Grid independence was declared if the maximum temperature difference between the different grid sizes is less than 1°C for the 846 points monitored in those studies. Based on the results from these studies, the laminar and zero-equation models were run using the 2,091,872 node grid, while the remaining turbulence models were run with the larger 7,993,152 node grid. A finer grid size was not analyzed based on the computational system limitations. The grid convergence index (GCI) for the two coarser grids are presented in Appendix B.

In order to see what affects a smaller grid would have on the performance of the models, a different approach to the grid study was performed for the third study [15]. Ten different grids were analyzed from one containing as few as 1,361 nodes all the way up to the same exact grid used in the previous analyses of 248,148 nodes. Each of the five flow models was run for each of the six test cases until the overall RMS difference between successive grids was below 1°C for the worst of the six cases. The GCI for the 3 tile, 23 kW cases are presented in Appendix B.

The inviscid flow model was the first to reach this newly defined criterion with a grid size of 17,061 nodes. This corresponded to a maximum control volume of 0.027 m^3. Interestingly all of the other (viscous) flow models met the criterion at 248,148 nodes which was the smallest of the three grids in the previous studies. These corresponded to a maximum control volume of 1.33×10^{-3} m^3. Also worth noting is that the inviscid flow model had a remarkably similar grid sizing to the coarser of two grids used in [24], while the finer grid was also quite similar in grid size to the grids used for the viscous flow models in this study. Even though the solution method is completely different, the fundamental equations in the potential-flow-based analysis are quite similar to those used in the inviscid model in this analysis.

2.3 Computational Effort

Each of the numerical models was run independently on the same IBM IntelliStation Z Pro [34] system with two dual-core Xeon 5160 CPUs (4 cores total) rated at 3.00 GHz with 16.0 GB of RAM in parallel mode on all four cores using the Microsoft Windows XP Professional x64 Edition operating system.

2.3.1 Convergence Criteria

For the first two studies, convergence was determined to be either when the solution residuals reached the default values of less than 0.001 for the flow equations and less than 10^{-7} for the energy equation, or when the residuals no longer changed for one third of the total iterations. Only the Spalart-Allmaras and laminar models did not completely converge to the default convergence values, but they did reach sufficiently small residuals as determined by the calculated iteration error (see Appendix B). In order to limit the computational effort for the third study, the computation of each model was terminated after 500 iterations, or when the solution residuals reached the default values of less than 0.001 for the flow equations and less than 10^{-7} for the energy equation. Only the zero-equation model met the residual convergence criteria values and did not compute all 500 iterations, except for in the 1 tile, 23 kW case. The worst case iteration error was calculated to be an order of magnitude less than the discretization error as recommended by Celik et al. [35]. The numerical error analyses are presented in Appendix B.

2.3.2 Results

Figures 5, 6, and 7 show the overall RMS difference between measured and predicted values for the temperature data points plotted against the solution time needed to either converge, or reach the stopping criteria for each of the flow models for the three different studies [13-15]. Each of the flow models is represented by a different basic symbol. For example, the laminar models are all shown as downward pointing triangles with different fills for the different perforated-tile cases.

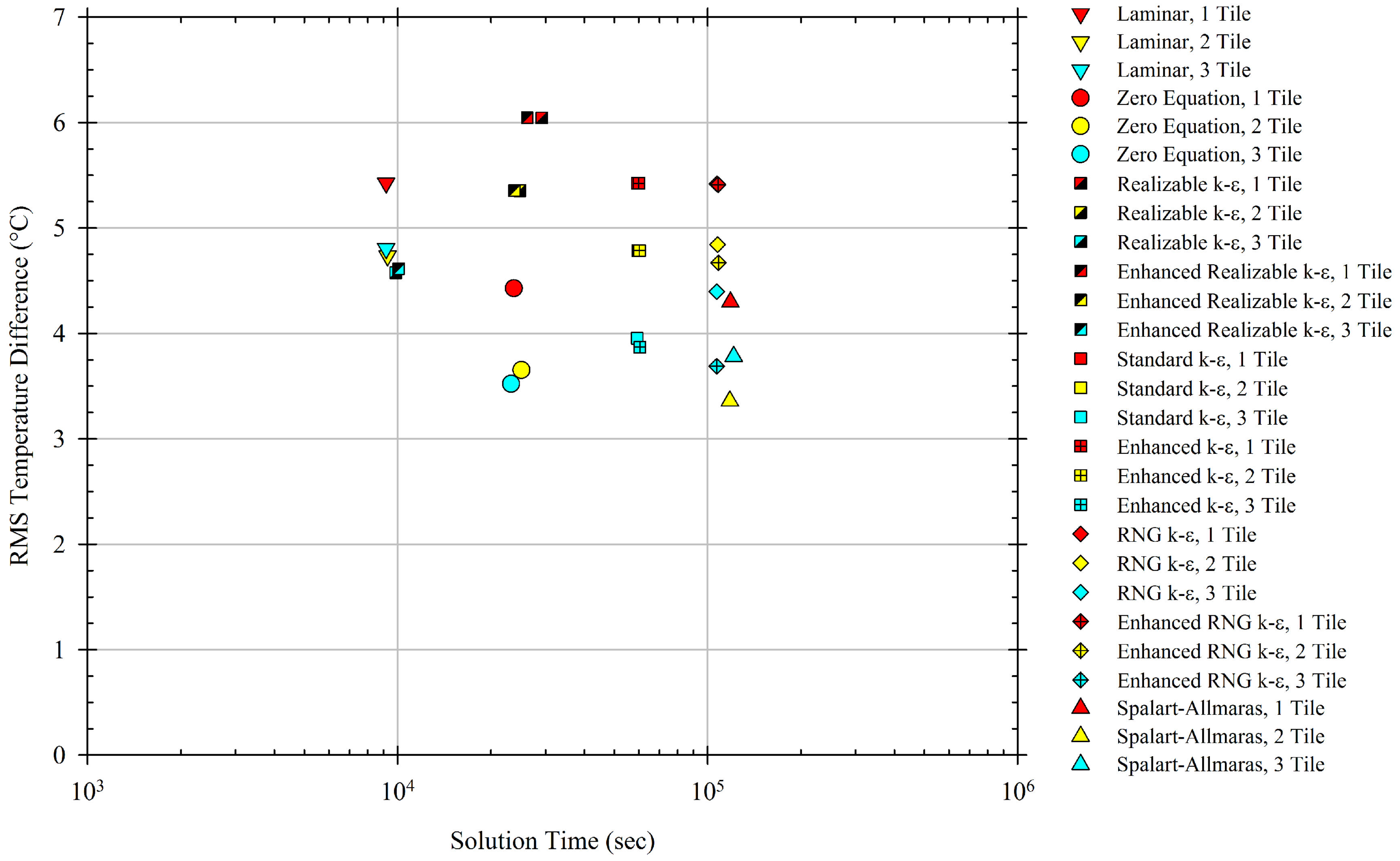


Figure 5. RMS Temperature Difference vs. Solution Time for the 23 kW Cases

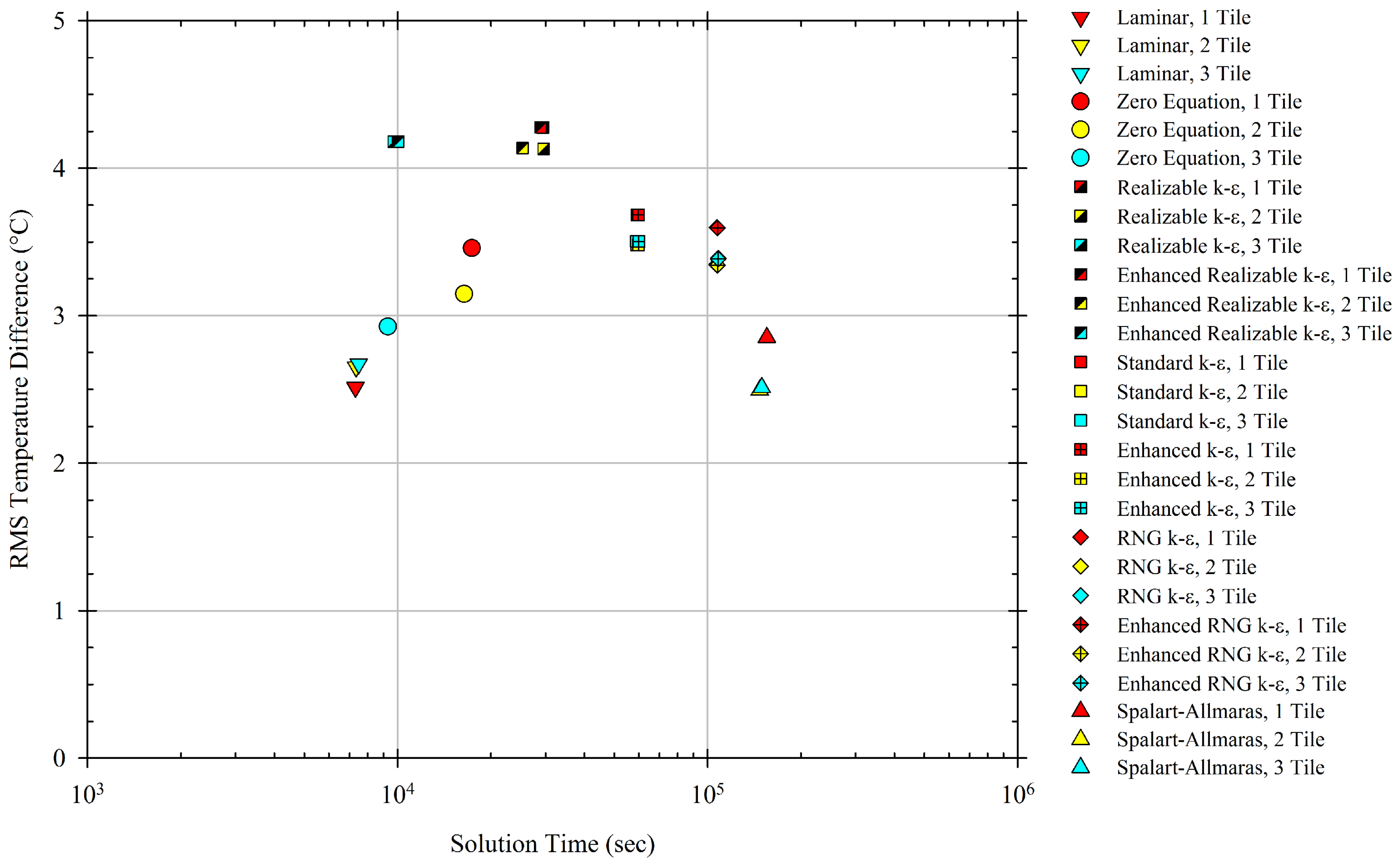


Figure 6. RMS Temperature Difference vs. Solution Time for the 11 kW Cases

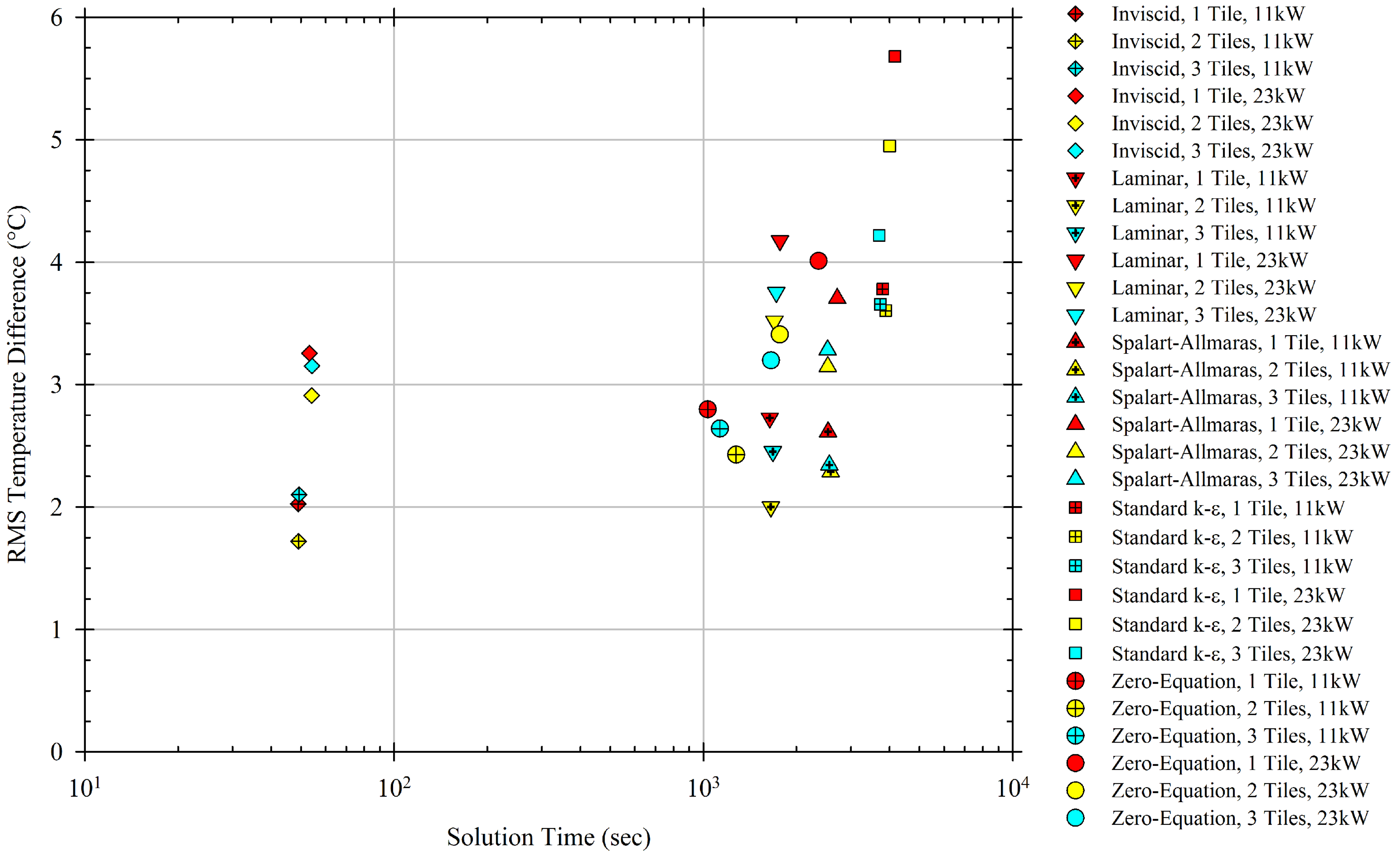


Figure 7. RMS Temperature Difference vs. Solution Time for the Inviscid Cases

Not surprisingly, the solution time for the inviscid model, seen in figure 7, was by far the shortest. With the least equations and complexity and by far the smallest grid, the inviscid model finished iterating in 1/33 the time it took for the next fastest model. The zero-equation model was the next fastest (depending on the case) followed closely by the laminar model, both of which have no additional transport equations to solve over the inviscid equations. The Spalart-Allmaras model with its one additional transport equation was next, followed by the two additional transport equation standard k-ε model. This follows what would be predicted, unlike the results seen in figures 5 and 6 which were due to the differing convergence criteria.

2.4 Experimental Data and Model Comparison

2.4.1 Overall Comparison

Not only did the inviscid flow model require the least computational effort, but it also produced the smallest RMS difference. Figure 7 shows a general trend in the data points that move up (higher RMS difference) and to the right (longer solution time). This goes against standard expectation of increasing model complexity (and with it solution time) providing higher accuracy (lower RMS difference). However, this is really only the case for models that properly capture the flow physics of the problem. In these cases it appears as though the simpler models do a better job of capturing the physics than the more complex models. Of course, this is a generalization for the RMS difference across the entire region of interest and does not take into consideration localized effects.

2.4.2 Localized Comparison

Figure 8 shows the measured temperature contours, the inviscid model prediction, and the absolute temperature difference between the measured data and the inviscid model prediction for all 7,650 data points along the nine heights for the 1 tile, 23 kW case. All of the contour planes show the region of interest in the same orientation as is

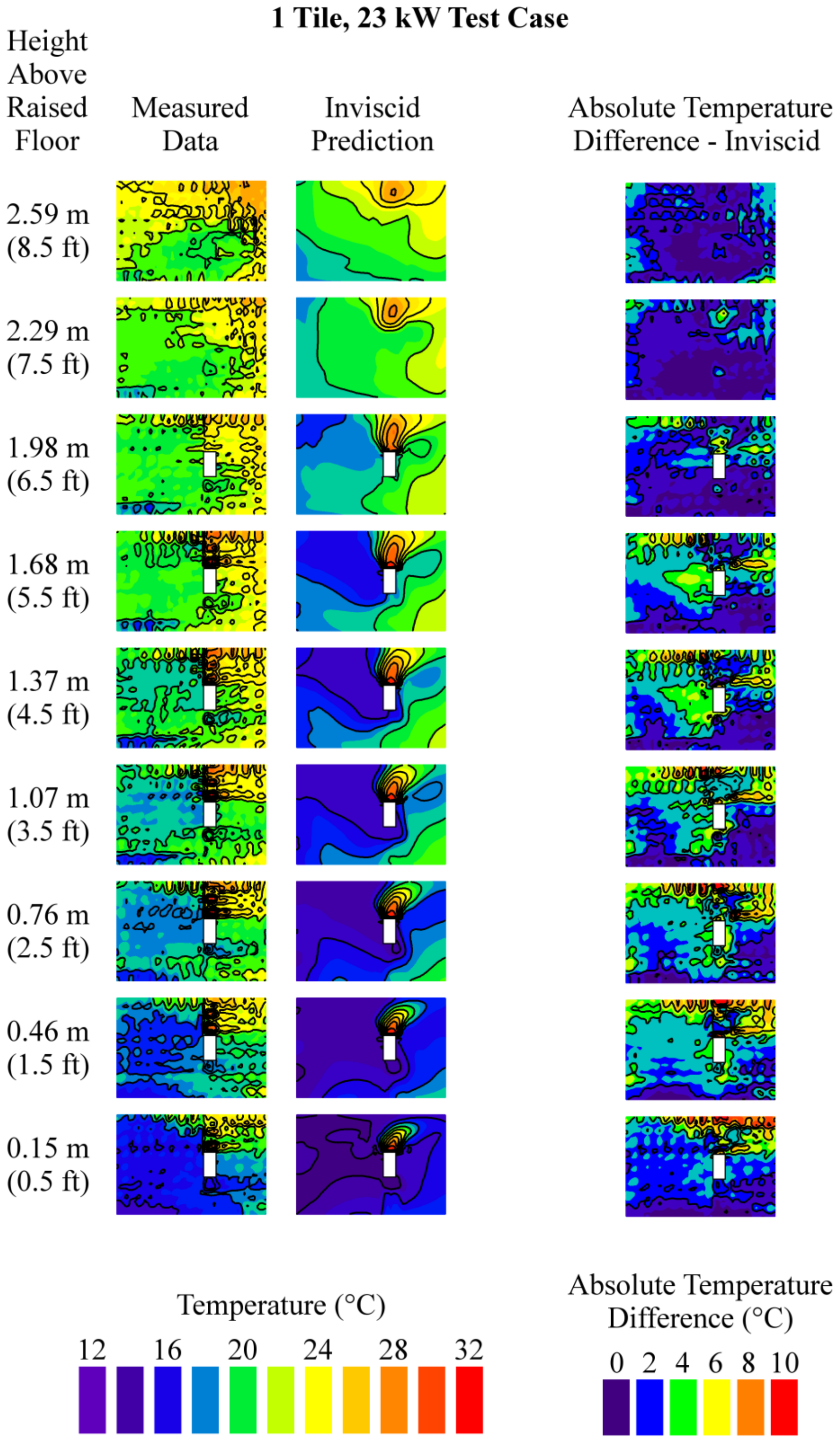


Figure 8. Temperature and Absolute Temperature Difference Contours within the Region of Interest of the Single-Rack Data Center Test Cell for the 1 Tile, 23 kW Case

shown in figure 3. The IT equipment rack is shown as blank space in the middle of the contours at the lower seven heights.

Figure 9 depicts the absolute temperature difference between the measured data and the five flow model predictions for the 1 tile, 23 kW case. It shows the results from only one of the six cases, but is representative of all of the cases since the other cases show the same basic trends. This figure allows for a visual analysis into where and to what degree the differences between the experimental data and the various numerical models occur. By studying this figure a better understanding of prediction trends of each model can be ascertained.

Overall, the inviscid model did the best job in predicting the temperature field, as can be seen in figure 7. This can also be seen in figure 9 with the relatively high level of cool colored (purple-blue-green) areas which correspond to 0°C to 4°C difference. Most of the high levels of deviation are located against the wall where the IT equipment is exhausting. This is most likely where viscous and turbulent effects are the highest and the inviscid model is incapable of capturing the physics of the mixing and entrainment in that region. The highest deviation was 11.7°C and was located against the wall at 0.46 m (1.5 ft). There are other large pockets that show moderate deviation, most notably between the IT equipment and the CRAC unit at the low to middle heights.

Overall, the laminar models did quite well for RMS difference (see figure 7), given the high-Reynolds-number flows within the model. The laminar model performed similarly to the inviscid model but with higher deviations especially near the corner opposite the CRAC towards the exhaust side of the IT equipment rack. The highest deviation was 12.5°C and was located against the wall at 0.46 m (1.5 ft).

Where the inviscid model performed the best is the exact place where the zero-equation model performed the worst (i.e. at the higher heights). Likewise in the worst performing areas of the inviscid model the zero-equation model performed the best of the viscous models in this study. For the lower and middle heights the zero-equation model

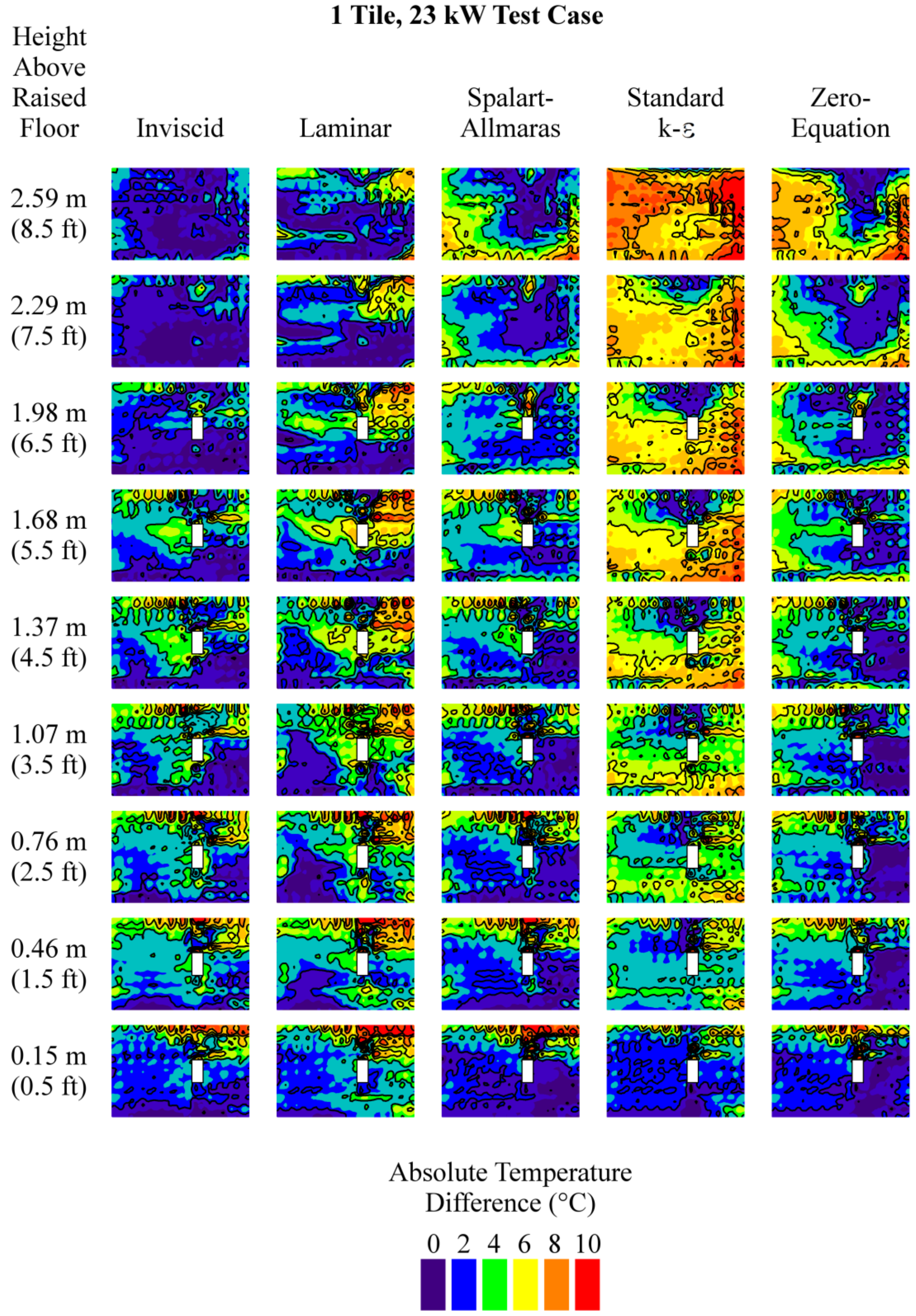


Figure 9. Absolute Temperature Difference Contours within the Region of Interest of the Single-Rack Data Center Test Cell for the 1 Tile, 23 kW Case

performed almost identical to inviscid model, but with less overall deviation. The highest deviation was 11.6°C and was located at the exhaust of the rack at 0.46 m (1.5 ft).

The Spalart-Allmaras turbulence modeling cases also performed similarly to the inviscid model. The high deviations were concentrated behind the IT equipment rack especially at the lower heights towards the CRAC side. The highest deviation was 12.3°C and was located against the wall at 0.46 m (1.5 ft). At higher elevations, higher deviation was found between the CRAC and the IT equipment rack.

Although the standard k-ε, produced the worst overall RMS difference, it did the best job of predicting the temperature field behind the IT equipment up to the highest two planes. Most of the deviation was found to be at the higher two heights, namely 2.29 m (7.5 ft) and 2.59 m (8.5 ft), and near the wall opposite the CRAC unit. The highest deviation was 12.6°C and was located near the wall opposite the CRAC at 2.59 m (8.5 ft). The inviscid model did remarkably well in those areas where the standard k-ε performed poorly.

2.5 Conclusions

It was found that the inviscid flow model provided both the best speed and accuracy for this application. It ran over thirty times as fast as the next fastest model while producing better RMS differences. The addition of the viscous terms and finer grid did not help the viscous models outperform the inviscid model in this study, however this may not always be the case.

Note that this very open data center is an ideal case for the inviscid model and it is not too surprising that it performed well. There are not many high velocity gradients to create turbulence or a lot of walls, obstacles, or impinging jets. In fact, as noted earlier, the worst performing areas are where turbulence is most likely located (i.e. where the IT equipment exhaust impinges on the wall). In a properly designed, densely populated data center with a hot-aisle, cold-aisle arrangement, there would certainly be a lot more turbulent mixing with jet impingement flows that the inviscid model may not be able to accurately model.

However, by combining the speed benefits of the inviscid model with the viscous mixing predictions of an appropriate turbulence model, there is potential to create an overall more accurate and faster modeling scheme for data center analysis. Such a method has been developed and is described in detail in the following chapter.

CHAPTER 3

COUPLED INVISCID-VISCOUS SOLUTION METHOD

A coupled inviscid-viscous solution method (CIVSM) for subsonic flow in a bounded domain was developed in an attempt to create a faster and more accurate solution technique than traditional RANS CFD/HT for data center analysis. The CIVSM presented here is based on the two patents of Cruz [36,37], which are applications of the Schwarz alternating method [38], and has been further developed, modified, and refined.

First, a high-level description of the method is presented. This is followed by a delineation of the governing equations for the different solution domains, along with guidelines on properly sizing the grid. The specific algorithms and procedures referred to in the high-level description are then illustrated more completely. These include a detailed explanation of the partitioning algorithm, boundary conditions and coupling procedure, mass and energy balance algorithms, various convergence criteria, and pressure-velocity coupling methods.

Once the CIVSM is fully described, results from a test case based on the single-rack data center test cell from chapter 2 are compared with traditional CFD/HT with multiple turbulence models. Further details on the values for the different parameters are enumerated and discussed. The results include a discussion of the partitioning algorithm, a grid study, the mass balance algorithm, modeling accuracy, and solution time.

3.1 Overall Solution Method

Figure 10 shows a flow chart of the general solution approach. After a CFD/HT model is created with the appropriate boundary conditions, it is solved using the inviscid equations on a coarse mesh. The second step is to re-run the model using the solution from the initial solve (the first step) as the initial solution with a more refined “medium”

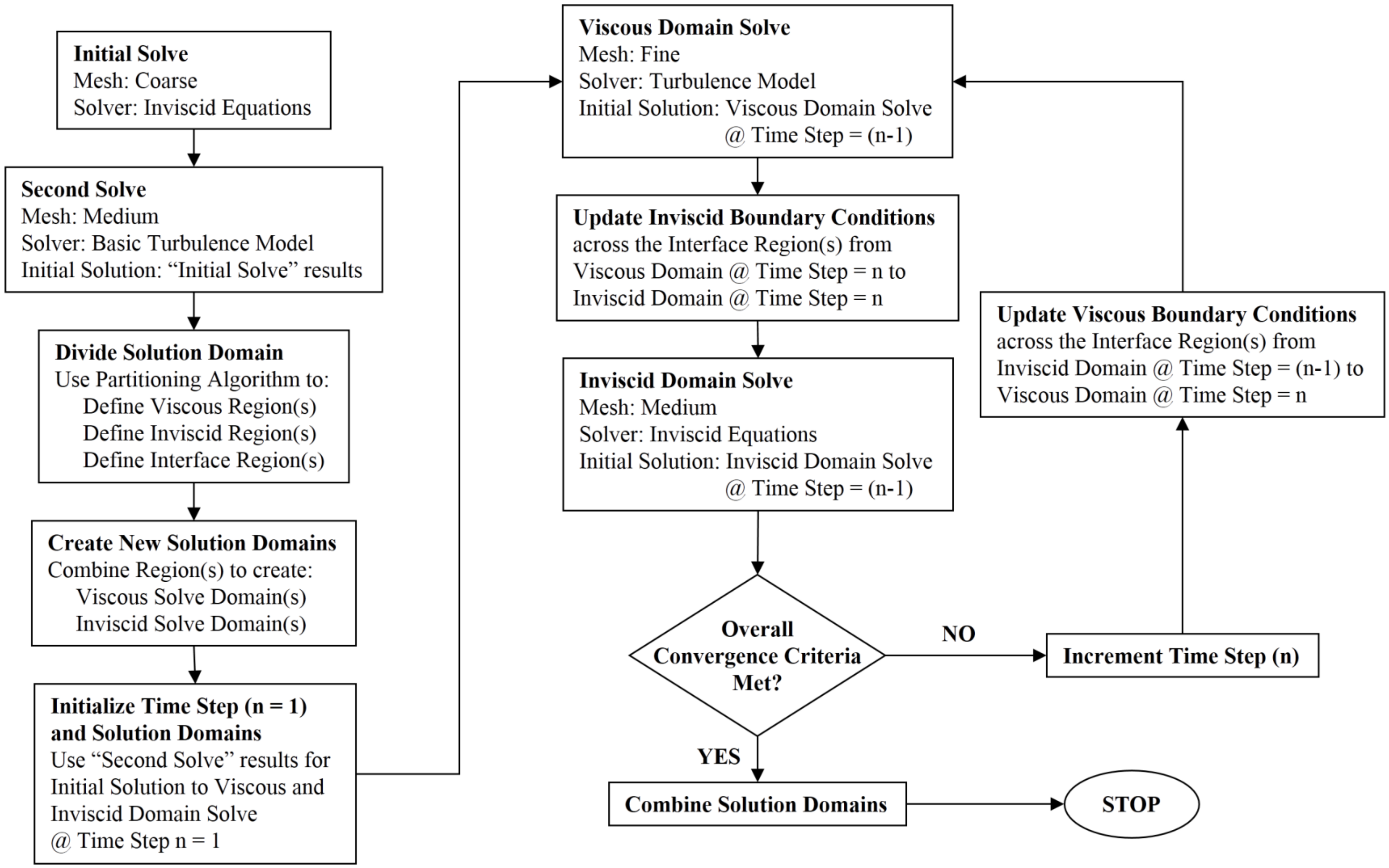


Figure 10. Overall Coupled Inviscid-Viscous Solution Method (CIVSM) Flow Chart

mesh and using a basic turbulence model, such as the zero-equation turbulence model [39,40].

The third step is the division of the entire solution domain into three regions: viscous, inviscid, and interface. A partitioning algorithm is used to determine the location of the viscous region, thereby defining the interface and inviscid regions. A more thorough explanation of this partitioning algorithm can be found in section 3.3, which is dedicated to this procedure and defines the different parameters such as the χ-value. Figure 11 depicts the isometric, top, left, and front views of the single-rack data center test cell described in chapter 2, and highlights the cells within the three viscous regions which were defined by the partitioning algorithm. Part A of figure 12 illustrates different steps in the partitioning algorithm on a cross-section of the domain, which is defined in figure 11 as the cut-plane which passes through the IT-equipment simulator rack and the three perforated floor tiles in front of the rack.

The fourth step is the formation of the two solution domains and further refinement of the mesh. The viscous region is combined with the interface region to create the viscous solution domain. In the same way, the inviscid region is combined with the interface region to create the inviscid solution domain. Part B of figure 12 demonstrates the formation of the two domains from the three regions for the same cross-section as shown in part A, and is defined in figure 11 as the cut-plane.

The fifth step is to initialize the iteration counter (n) and the newly formed solution domains. The iteration counter is set equal to one. The solution from the second solve is interpolated onto the newly refined mesh as the initial solution for both the viscous and inviscid solution domains.

An iterative routine is then used to solve the two domains and pass the boundary conditions back and forth. However, a mass balancing correction algorithm is needed in order to preserve conservation of mass within each of the solution domains. This is primarily due to the non-conformal mesh and the way in which the boundary conditions

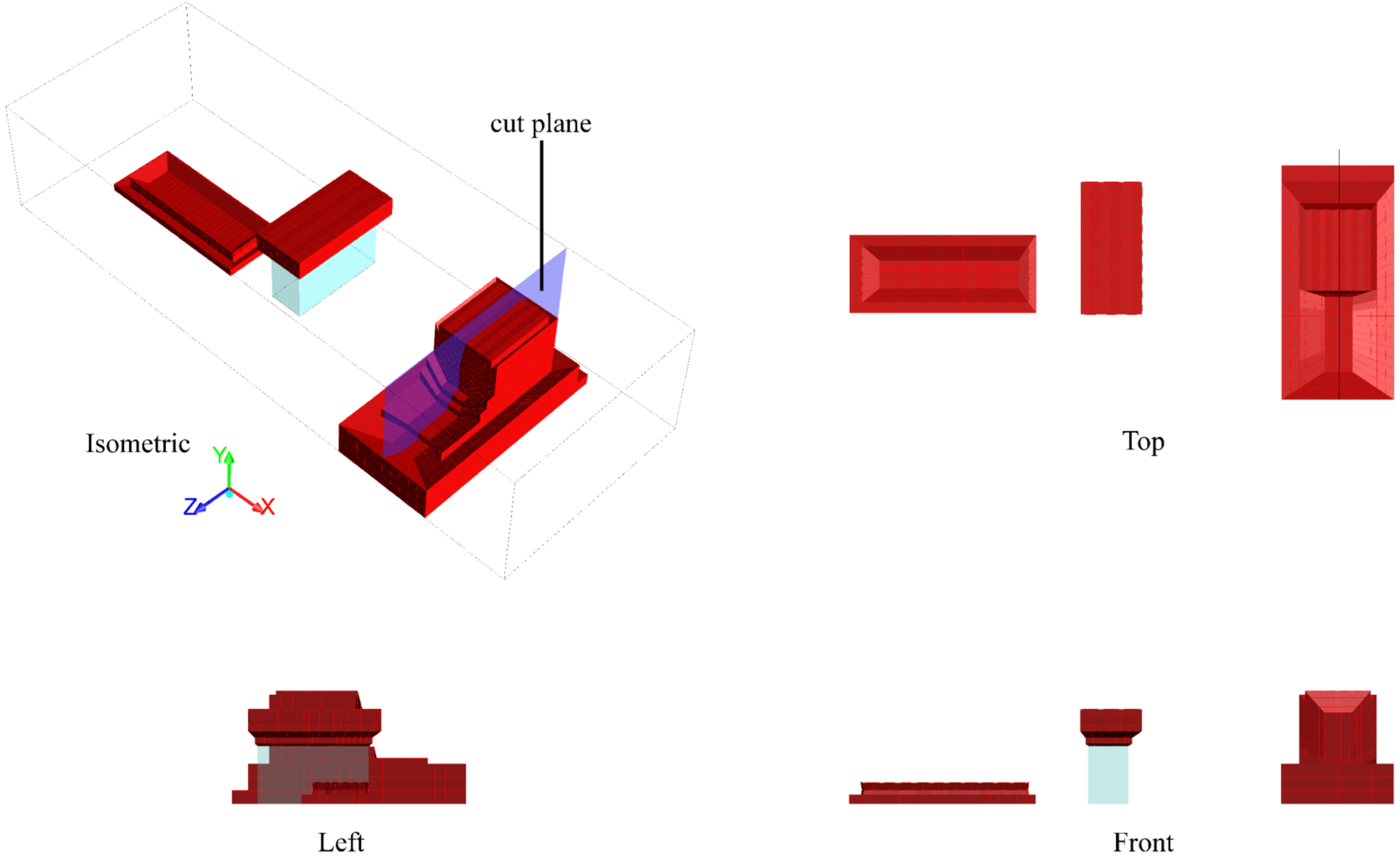
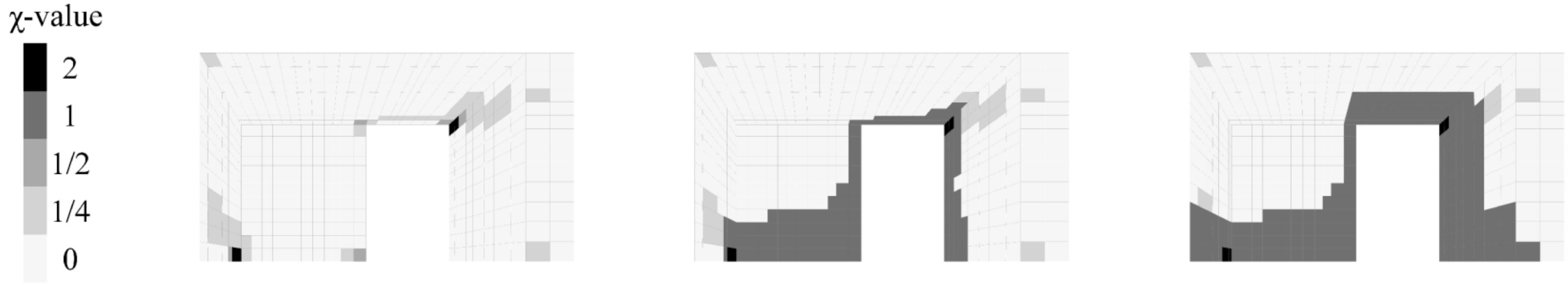
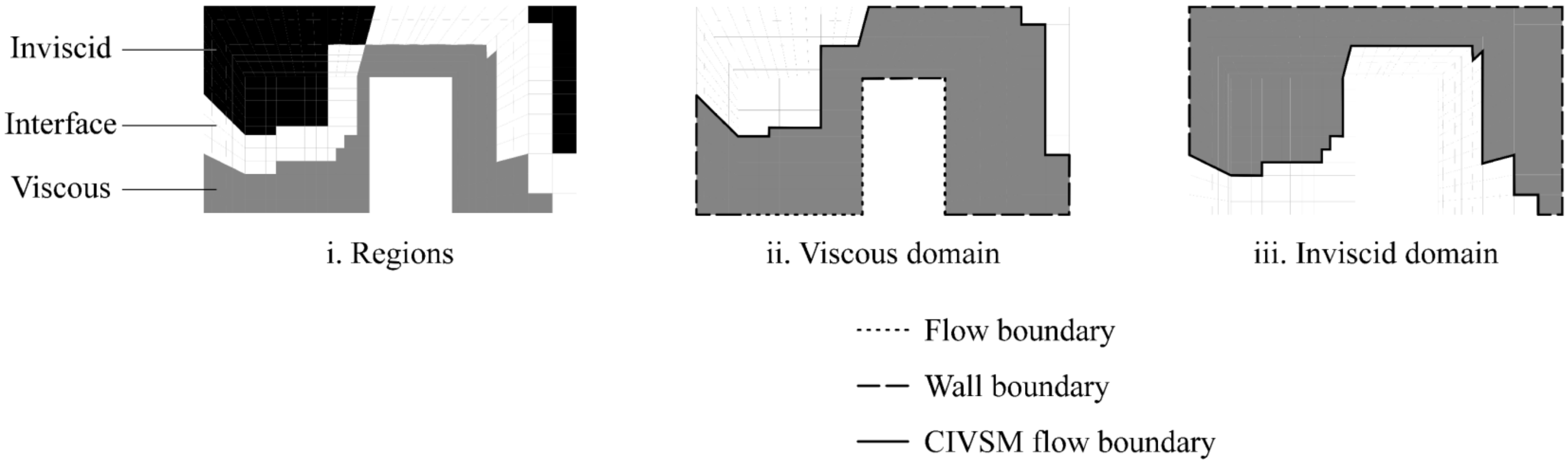


Figure 11. Viscous Regions of the Single-Rack Data center Test Cell



i. Initial values ii. Intermediate step iii. Final values

A. χ-values from the partitioning algorithm, right-view of cut plane



i. Regions ii. Viscous domain iii. Inviscid domain

B. Regions, domains, and boundary conditions, right-view of cut plane

Figure 12. Partitioning Algorithm χ-values and Region and Domain Definition for the Single-Rack Data center Test Cell

are specified. An in-depth discussion of the mass balance algorithm and a description of the resulting boundary conditions can be found in section 3.5, which is dedicated to the algorithm.

The first of the steps in the iterative routine is to solve the viscous domain using the chosen turbulence model. Once the domain has been solved, the boundary conditions can be updated for the inviscid domain. Having already been initialized with the solution from the second solve, the inviscid domain is then solved.

The next step is to check the iteration counter (n) and overall convergence criteria. If the convergence criteria or maximum number of iterations have not been met then the iteration counter (n) is incremented. Then the viscous domain is initialized using the solution from iteration (n-1) and the boundary conditions of the viscous domain are updated from the interior of the inviscid domain. Similarly the inviscid domain is initialized with the previous inviscid domain solution (iteration n-1), and updated boundary conditions are passed from the interior of the viscous domain. This iterative procedure is repeated until the maximum number of iterations, or the convergence criteria are met.

Finally the viscous and inviscid domains are combined to form the entire solution domain. This is accomplished by patching together the results from the two domains into a single solution. The solution from the viscous domain is used for the interface region since it has the finer mesh, and the turbulent data (e.g. turbulent viscosity, turbulent kinetic energy, etc.) that are not calculated within the inviscid domain. In this way the final CIVSM solution is obtained.

This methodology allows for the two solution domains to be non-contiguous, thereby allowing for the ability to apply both different grid sizes and, in the case of the viscous domain, different turbulence models to the multiple contiguous zones. The ability to apply different grid sizes allows for finer control over the final solution accuracy for the different zones. The refined grids in each of the domains are not required to be multiples of the original grid, but can be re-grided independently of one another. The parameters used to re-grid the domains can be set either a priori, or a posteriori, but have been programmed a priori in all cases presented here.

In the viscous domain, different turbulence models can be employed in order to extract the best overall solution for the entire domain. Different turbulence models may provide better accuracy in different zones than a single model across all of the zones. For optimized results, different turbulence models may require finer or allow for coarser mesh which can be accommodated for with this method. A coarser mesh will help yield lower computational cost and faster solution speed.

3.2 Governing Equations

The two types of solution domains are solved using different sets of equations. The viscous domains are solved using the RANS equations while the inviscid domains are solved using the Euler equations. Compared with the Euler equations, the RANS equations require additional relations or equations (turbulence modeling) in order to close the set of equations due to the Reynolds stresses, which arise from the Reynolds decomposition.

3.2.1 Viscous Domains

The viscous domains in the CIVSM are solved using the RANS equations and an appropriate turbulence model for each of the domains. The RANS equations decompose each of the variables of the instantaneous Navier-Stokes equations into the sum of the mean (time-averaged) and fluctuating parts of that variable:

$$\phi_j(x_i,t)=\bar{\phi}_j(x_i)+\phi'_j(x_i,t), \tag{2}$$

where $\bar{\phi}(x_i)$ is the mean (time-averaged) value defined by:

$$\bar{\phi}_j(x_i) = \lim_{\tau \to \infty} \frac{1}{\tau} \int_0^{\tau} \phi_j(x_i, t)\, dt, \qquad (3)$$

where x_i (i=1,2,3) or (x,y,z) are the Cartesian coordinates, j=1,2,3 when representing the Cartesian components of a vector quantity or j can be dropped for scalar quantities of ϕ, t is time, and τ is the averaging interval.

Using the Reynolds decomposition of the variables for a homogeneous, incompressible fluid (i.e. $d\rho = 0$) with gravitational body force, and neglecting the pressure work, the kinetic energy terms, and the dissipative heating terms due to viscous stresses, the averaged continuity, momentum, and energy equations can be written in Cartesian tensor notation as:

$$\frac{\partial}{\partial x_i}(\bar{u}_i) = 0 \qquad (4)$$

$$\rho \frac{\partial}{\partial t}(\bar{u}_i) + \rho \frac{\partial}{\partial x_j}(\bar{u}_i \bar{u}_j) = -\frac{\partial}{\partial x_i}(\bar{p}) + \frac{\partial}{\partial x_j}\left\{ \mu \left[\frac{\partial}{\partial x_j}(\bar{u}_i) + \frac{\partial}{\partial x_i}(\bar{u}_j) \right] - \rho \overline{u_i' u_j'} \right\} + \rho g_i \qquad (5)$$

$$\frac{\partial}{\partial t}(\bar{T}) + \frac{\partial}{\partial x_i}(\bar{u}_i \bar{T}) = \frac{\partial}{\partial x_j}\left[\frac{k}{\rho c_p} \frac{\partial}{\partial x_j}(\bar{T}) \right] + S - \frac{\partial}{\partial x_i}\left(\overline{u_i' T'}\right) \qquad (6)$$

where ρ is the density, $\bar{u}_i$ the mean (time averaged) velocity vector from the Reynolds decomposition, u_i' the fluctuating velocity vector from the Reynolds decomposition, p the static pressure, μ the molecular viscosity, ρg_i the gravitational body force per unit volume, T the static temperature, k the thermal conductivity, c_p the specific heat, and S the volumetric heat sources.

The Boussinesq approximation states that differences in the density of a fluid are small enough to be neglected in all but the buoyancy term, ρg_i, which can then be rewritten as $(\rho_0 + \Delta\rho)g_i$, where $\Delta\rho = \rho - \rho_0$ is the density variation, and ρ_0 is the reference density. The buoyancy term due to density variation $\Delta\rho g_i = (\rho - \rho_0)g_i$ can be rewritten in terms of the thermal expansion coefficient β as:

$$(\rho - \rho_0)g_i = -\rho_0\beta(T - T_0)g_i \tag{7}$$

where T_0 is the reference temperature that corresponds with the reference density, ρ_0. By substituting equation (7) into equation (5) the momentum equation becomes:

$$\begin{aligned}\rho_0 \frac{\partial}{\partial t}(\bar{u}_i) + \rho_0 \frac{\partial}{\partial x_j}(\bar{u}_i\bar{u}_j) = -\frac{\partial}{\partial x_i}(\bar{p}) + \frac{\partial}{\partial x_j}\left\{\mu\left[\frac{\partial}{\partial x_j}(\bar{u}_i) + \frac{\partial}{\partial x_i}(\bar{u}_j)\right] - \rho_0\overline{u_i'u_j'}\right\} \\ + \rho_0 g_i\left[1 - \beta(T - T_0)\right]\end{aligned} \tag{8}$$

The Reynolds decomposition introduces additional terms, $\rho\overline{u_i'u_j'}$ and $\overline{u_i'T'}$, in the momentum and the energy equations, respectively, as compared with the instantaneous Navier-Stokes equations. The first terms, $\rho\overline{u_i'u_j'}$, are referred to as the Reynolds stresses. The Reynolds stresses along with the additional temperature transport terms require additional relations or models in order to close the set of equations.

A number of different turbulence models have been proposed and used for many different applications, but only the zero-equation turbulence model [39,40], the one-equation Spalart-Allmaras model [41,42], and variations of the two-equation k-ε models [43-47] have been used with and compared with the CIVSM.

One problem with using these turbulence models, especially the k-ε models, is that they are typically developed for fully turbulent conditions and do not adequately account for transition from laminar to turbulent flow [48]. This has been shown to be the case for data centers where complex flow patterns and large changes in velocities are present throughout the domain [3,5,7-16]. It was also noted that in the locations where the k-ε model performed the worst in terms of predicting the temperature field, the inviscid model performed better than any of the viscous models [15]. This is due to the k-ε model over-predicting the viscous effects and dissipating the heat instead of advecting it further downstream as is done by the inviscid model.

3.2.2 Inviscid Domains

The inviscid domains within the CIVSM are solved using the Euler equations, and for a homogeneous, incompressible fluid (i.e. $d\rho = 0$) using the Boussinesq approximation with gravitational body force, the continuity, momentum, and energy equations can be written in Cartesian tensor notation as:

$$\frac{\partial}{\partial x_i}(u_i) = 0 \tag{9}$$

$$\rho_0 \frac{\partial}{\partial t}(u_i) + \rho_0 \frac{\partial}{\partial x_j}(u_i u_j) = -\frac{\partial}{\partial x_i}(p) + \rho_0 g_i [1 - \beta(T - T_0)] \tag{10}$$

$$\rho \frac{\partial}{\partial t}(E) + \frac{\partial}{\partial x_i}[u_i(E + p)] = S \tag{11}$$

where E is the sum of the internal and kinetic energy:

$$E = h - \frac{p}{\rho} + \frac{u^2}{2} \tag{12}$$

where h is the sensible enthalpy defined by:

$$h = \int c_p dT \tag{13}$$

By eliminating the viscous terms from the Navier-Stokes equations, the inviscid equations help to improve the speed of the solution in two ways: by reducing the computational expense per iteration, and allowing for coarser mesh. For the same size mesh the inviscid model runs 30% faster than the fastest of viscous models [15]. However, the largest gains in reducing solution time come from using a coarser mesh.

3.2.3 Grid Sizing

In order to benefit from each of the solution techniques' advantages in the different domains, the selection of the grid size should be taken into account and much finer grids should be used in the viscous domains than in the inviscid domain. The viscous domains should have fine enough grid to adequately model the turbulent structures within the domain. Likewise, the inviscid domain's mesh should be fine enough to adequately resolve the velocity and temperature fields, while not being so fine as to severely impact the speed of the solution.

3.3 Partitioning Algorithm

One of the most critical steps in the process is defining the location of the different regions and subsequently the solution domains. An overall flow chart of the algorithm is shown in figure 13. Of the many choices of criteria available, vorticity magnitude was chosen as the critical value to determine the location of the different regions. The two main reasons for choosing vorticity magnitude are that it is easily calculated based on the flow field regardless of the flow model used, and both viscous effects and turbulence will be most prevalent where there is significant vorticity.

After performing the second solve step using a basic turbulence model, the vorticity magnitude is calculated and sorted for all of the cells in the domain. Three cutoff values ($\omega_{cutoff1} < \omega_{cutoff2} < \omega_{cutoff3}$) are then calculated based on three decreasing

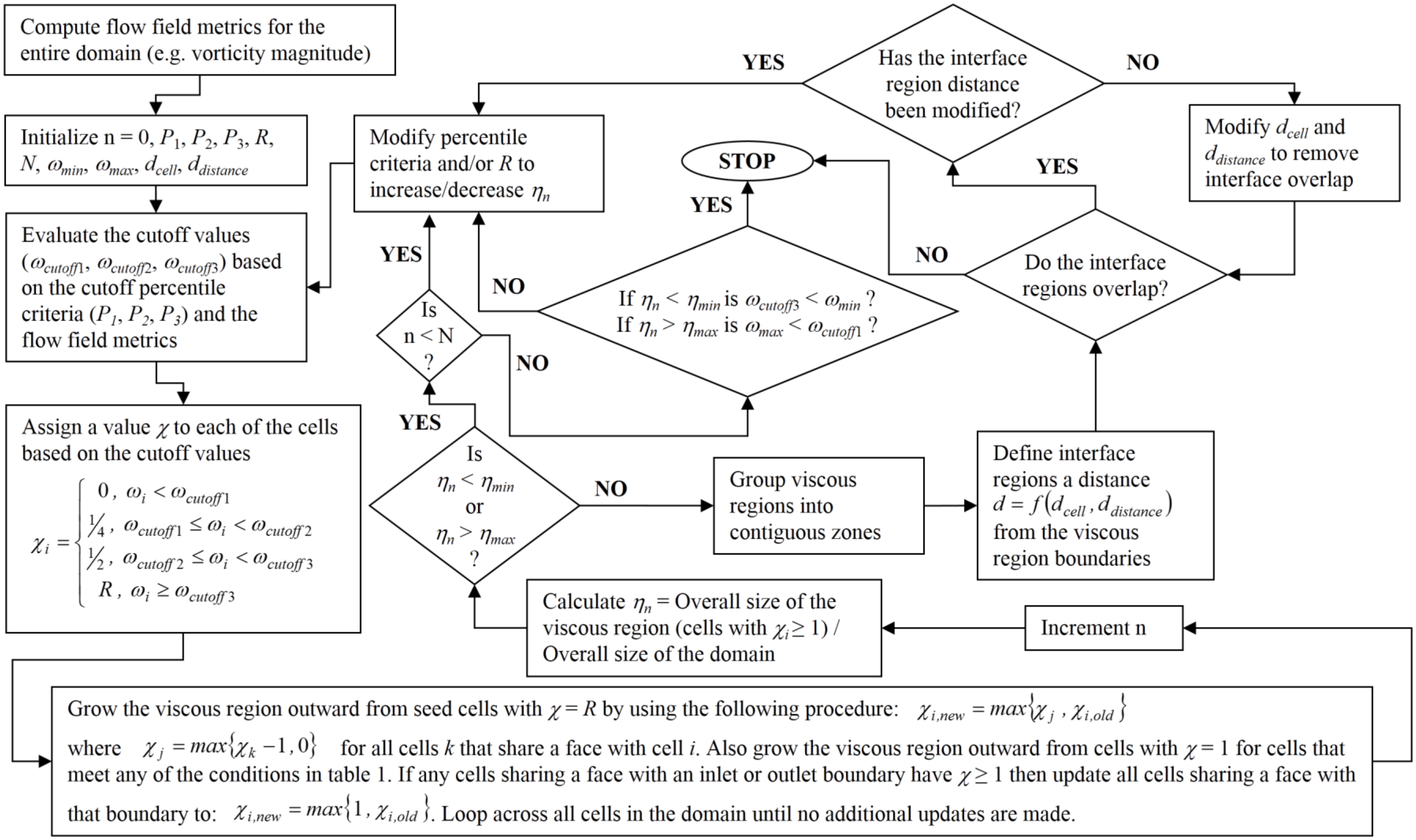


Figure 13. Partitioning Algorithm Flow Chart

percentiles ($P_1 > P_2 > P_3$) of vorticity magnitude within the domain. All of the cells within the domain are marked with a value (χ_i) based on the vorticity magnitude (ω_i) for the *i*th cell as follows:

$$\chi_i = \begin{cases} 0, & \omega_i < \omega_{cutoff1} \\ \frac{1}{4}, & \omega_{cutoff1} \le \omega_i < \omega_{cutoff2} \\ \frac{1}{2}, & \omega_{cutoff2} \le \omega_i < \omega_{cutoff3} \\ R, & \omega_i \ge \omega_{cutoff3} \end{cases} \tag{14}$$

where *R* is the viscous region cell radius parameter, which is defined as a natural number of cells. After the viscous region growth has completed, only the cells with $\chi \geq 1$ are included in the viscous regions.

The cells with the highest vorticity magnitude within the domain ($\omega_i \geq \omega_{\text{cutoff3}}$), called "seed" cells, are used to grow the viscous region. All cells bordering the seed cells within the radius *R* are included in the viscous region. This is accomplished by recursively changing the χ-value of the cells sharing a face with the seed cell to *R-1* and then having those cells act as seed cells by changing the χ-value of the cells sharing a face with these new seed cells to *R-2* and continuing the procedure until the χ-value is equal to 1. The highest χ-value for a cell is always retained and the χ-value is never lowered. For example, if two neighboring cells are seed cells, then they both would have $\chi=R$ rather than having one of the two change the other to the lower χ-value of *R-1*. In this way the viscous regions grow outwardly in a "sphere" with radius *R* from the centers located at the seed cells.

The viscous regions are also expanded by another growth function that is based on each cell's χ-value, its neighboring cells' χ-value, and whether or not the cell is adjacent to a boundary. Cells that meet any of the conditions shown in table 1 are updated to a χ-value of 1. If any cells adjacent to an inlet or outlet boundary have a χ-value that is greater than or equal to 1, then all cells adjacent to that boundary are changed to χ=1. This ensures that all of the boundary conditions are either entirely in the viscous or the inviscid regions and not split among the two regions. The viscous growth procedures continuously update the χ-value for all of the cells in the domain until no new cells are updated. An example of the initial and final χ-values, as well as an intermediate step, is demonstrated for a cross section of the single-rack data center test cell in figure 11.

Table 1. Conditions necessary to change a cell to χ=1

Condition	χ_i	**Minimum number of adjacent cells with χ=1**	**Cell is adjacent to a boundary**
1	½	1	-
2	¼	2	-
3	¼	1	Yes
4	0	3	-
5	0	2	Yes

After the viscous regions have finished growing, the ratio (η_n) of the overall size of the viscous region to the overall size of the domain is then compared with the minimum (η_{min}) and maximum (η_{max}) allowable ratios. If the viscous regions comprise too little or too much of the domain then the parameters used in this algorithm ($\omega_{cutoff1}$, $\omega_{cutoff2}$, $\omega_{cutoff3}$, R) are automatically modified in order to expand or shrink the viscous regions and the viscous region growth algorithm is repeated.

In order to take advantage of the benefits the CIVSM offers, the method needs to ensure there is a large enough inviscid domain to help reduce computational effort as compared with a fully viscous model and a large enough viscous domain to add any detail that would be lost in a fully inviscid model. The CIVSM should only be used if clear distinctions between inviscid and viscous domains exist. If after N iterations ($n \geq N$) of modifying the parameters ($\omega_{cutoff1}$, $\omega_{cutoff2}$, $\omega_{cutoff3}$, R) the ratio (η_n) of the overall size of the viscous region to the overall size of the domain is less than the minimum ($\eta_n < \eta_{min}$) or greater than the maximum ($\eta_{max} < \eta_n$) then the values of the cutoff parameters are compared with their limits ($\omega_{cutoff3} < \omega_{min}$ for $\eta_n < \eta_{min}$, or $\omega_{max} < \omega_{cutoff1}$ for $\eta_{max} < \eta_n$). If the parameters exceed their limits then the model should be run as either a fully inviscid ($\omega_{cutoff3} < \omega_{min}$) or fully viscous ($\omega_{max} < \omega_{cutoff1}$) and not using the CIVSM. If the parameters do not exceed their limits then the parameters are further modified until either the overall size of the viscous region to the overall size of the domain is within the limits and the CIVSM can continue to the next step or the parameters exceed their limits and the model should be run as either fully inviscid or fully viscous.

Once the overall size of the viscous region to the overall size of the domain falls within the minimum and maximum limits ($\eta_{min} < \eta_n < \eta_{max}$) the viscous regions are grouped into contiguous zones. This is accomplished by first looping through the domain and, for every cell with $\chi \geq 1$, changing the cells' χ-value to an increasing integer value starting with $R+1$. After all of the cells within the viscous regions (previously $\chi \geq 1$) are set to $\chi \geq R+1$, and no two values are the same, then the algorithm loops over all of the faces for every cell in the viscous regions ($\chi \geq R+1$), and updates the cells' χ-value to the maximum value of its own and its nearest neighbors χ-value. The loop is repeated until no new updates are performed. Once this has occurred each of the contiguous zones can be determined by identifying regions of constant $\chi > R+1$.

After the contiguous viscous regions are defined, then the interface regions are defined based on a specified distance and/or cell count from each of the contiguous viscous regions. The interface regions grow outwardly from the viscous regions into the inviscid regions and will be incorporated into both the viscous and inviscid domains. A check must be performed to ensure there is no overlap in the interface regions emanating from the different viscous regions. If there is an overlap then the interface regions can be reduced in size to try to eliminate the overlap. If the overlap remains after the interface size was reduced, then the cutoff parameters are modified to either try to reduce the viscous regions so that there is a large enough gap between the viscous regions for the interface region to not overlap, or they are modified to include the gap between the offending viscous regions to create a new, larger contiguous viscous region. After the interface region is defined and there is no overlap between the different regions then partitioning algorithm is complete and the viscous and inviscid domains can be created.

The partitioning algorithm has been implemented such that it automatically modifies the percentiles and/or the radius R for partitions as needed. Different starting parameters can lead to slightly different partitions, but the partitions are typically not significantly different from each other as to affect the overall solution. The partitioning algorithm has been designed to require minimal input or experience from the user, and when using the default values, has consistently produced good, repeatable results.

3.4 Boundary Conditions and Coupling Procedure

Before the individual viscous and inviscid domains can be solved, they first need to have their boundary conditions defined. The boundary conditions for the viscous domains come from the interior of the inviscid domain, and vice versa. The intersection of the inviscid and interface regions within the inviscid domain forms the outer boundary of the viscous domain. Likewise the intersection of the viscous and interface regions within the viscous domain forms the outer boundary of the inviscid domain. Part B of figure 12 depicts the locations of the boundaries for the two domains in a cross-section of the single-rack data center test cell.

The wall boundary conditions do not change within the viscous and inviscid regions. However, the wall boundary conditions do change in the interface region depending on whether the boundary is being solved in the viscous or inviscid domain. The viscous domains have no-slip boundary conditions for walls while the inviscid domains have no penetration boundary conditions at the walls.

Due to the elliptic nature of the equations for subsonic, steady flow, the boundary conditions must specify (Dirichlet) the dependent variables (velocity and temperature), the derivatives of the dependent variables (Neumann), or a combination of the two over the entire boundary of the domain, as well as specify the pressure at one location within the flow domain, and if using a purely Neumann set of boundary conditions, then the velocity and temperature must also be specified at a location within the flow domain, in order to create a well-posed problem. Given that the flow patterns within the viscous and inviscid domains are not necessarily aligned with the boundaries, as defined by the partitioning algorithm, it is nearly impossible to distinguish traditional inlet and outlet boundaries. Therefore the methodology does not allow for applying boundary conditions that are sufficiently upstream or downstream of the area of interest to use zero gradient boundary conditions.

In order to satisfy the flow boundary condition requirements the velocity components and temperature are directly specified (Dirichlet-type) across all of the flow boundaries. When running an incompressible, unsteady case, time-marching from a fully specified initial condition is all that is needed in addition to the above boundary conditions in order to well-pose the hyperbolic nature of the inviscid domain and the parabolic nature of the viscous domain.

Grid differences between the domains in the form of non-conformal mesh require an interpolation scheme in order to apply the boundary conditions. The CIVSM employs the inverse distance interpolation scheme to apply the boundary conditions across the domains. This scheme applies a weighted average to the profile data that is inversely proportional to the distance between the cell face centers. The scheme effectively smooths out the many data points when going from a dense grid to a coarse grid by applying the most weight to the nearest faces. A similar result is obtained when the coarse data points are applied to the dense grid; not all of the fine cell faces near the edges of coarse cell faces will have the same value. In this way the boundary conditions are directly specified across the domains.

However, by directly specifying the velocity components at all of the boundaries, and thereby the mass flux for an incompressible flow of constant density (using the Boussinesq approximation), satisfying continuity must also be addressed when applying the boundary conditions. This leads to the necessity of a mass balance algorithm during the specification of the velocity boundary conditions in order to ensure continuity for the entire domain. Similarly, directly defining the temperature at all of the boundaries would over-specify the problem and constrain the solution.

3.5 Mass and Energy Balance Algorithm

A novel mass balance algorithm has been developed in order to satisfy continuity through the boundaries of the domain. The algorithm was developed in order to try to minimize the change in mass flux through each face area of the flow boundaries between the domains. It also has to account for the non-conformal mesh between the different domains. Mass flux specified at other boundary conditions are not affected by the algorithm.

An iterative method is employed as outlined in the flow chart of figure 14. After initializing the iteration counter and the initial correction factor, ξ_0, the net mass flux and then the total mass flux through all of the boundaries of the domain are calculated. The mass imbalance ratio is then calculated by dividing the net mass flux by the total mass flux. The absolute value of the mass imbalance ratio is then compared with the stopping criterion, ε_{max}. The stopping criterion, ε_{max}, should at least be small enough to ensure stability of the solution and ideally would be on the order of the discretization error.

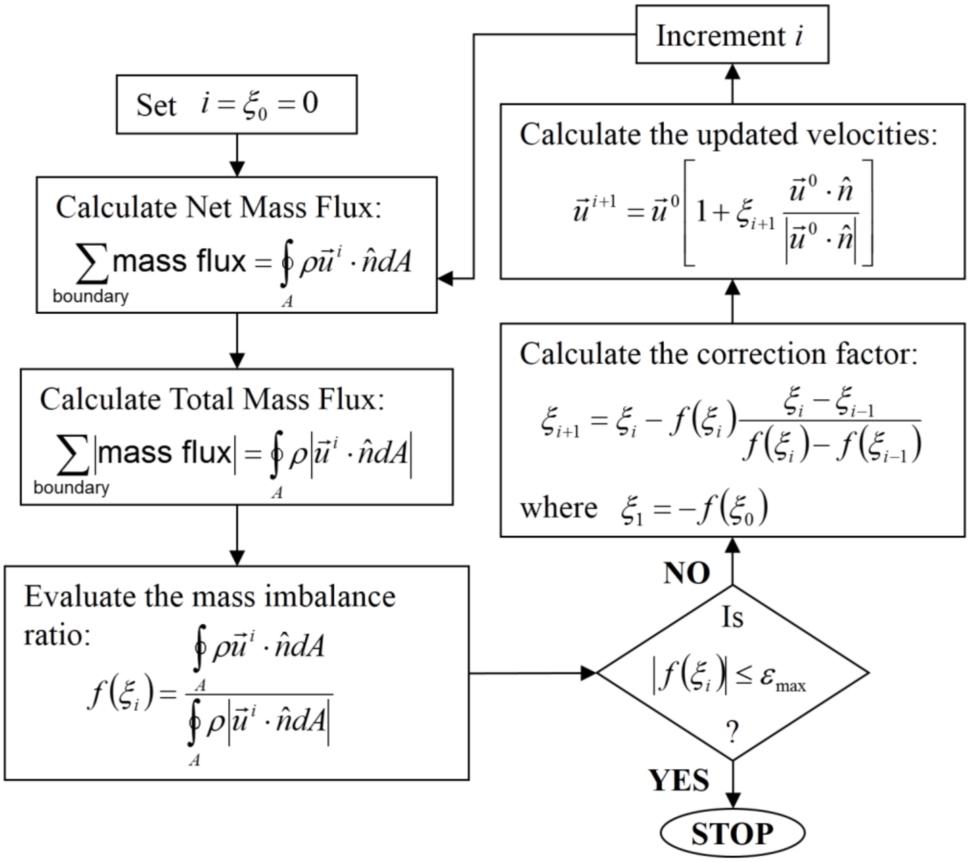


Figure 14. Mass Balance Algorithm Flow Chart

If the mass imbalance is not less than or equal to the stopping criterion then the correction factor is calculated. For the first iteration the correction factor is simply the negation of the imbalance ratio. Subsequent correction factors are calculated using the secant method, which provides superlinear convergence, requires minimal storage, and does not necessitate any additional mass imbalance evaluations [49].

The correction factor is then used to update the magnitude of the velocities at the flow boundaries of the interface regions. The algorithm always modifies the magnitude of the velocities in the direction that decreases the net mass flux through the domain. The magnitude of the velocities is either increased or decreased depending on the whether the net mass flux through the boundary is positive or negative and whether the flow direction of the individual velocity is into or out of the domain.

For example, if the net mass flux is into the domain and a particular boundary velocity is also into the domain, then that velocity is reduced in magnitude. However if a particular boundary velocity is out of the domain, then that velocity is increased in magnitude. In this way the overall change in the velocities along the interface regions is minimized and any other boundary conditions are not affected.

After the velocities are updated, the iteration counter is incremented and the algorithm is repeated. The iterative process continues until the mass imbalance ratio meets the stopping criterion. In practice the algorithm only takes three to five iterations to converge to the stopping criterion.

Although a simple scaling of the velocities simplifies the calculation considerably and produces a near-exact solution in a single iteration, it can also lead to divergence of the solution. Divergence occurs when the velocity vectors are altered by a high correction factor on the first outer viscous- iteration of the CIVSM which has to account for the non-conformal mesh between the different domains. The subsequent correction factors ultimately amplify the initial velocity vector modification until the correction factors become so high that the solution diverges. By always minimizing the change to the velocity vectors using the mass balance algorithm outlined in figure 14, the CIVSM avoids this divergence mechanism of overcorrecting the velocity vectors.

Unlike the mass balance algorithm for the continuity equation, no additional algorithm is required in order to alter the boundary conditions to satisfy the energy equation. The way the energy equation is satisfied is by using an upwind discretization scheme that ignores the temperature boundary conditions where the flow is exiting the domain. This is done in order to not over-specify the problem and constrain the solution. No specific algorithm is necessary for the energy equation beyond removing the strict enforcement of the temperature specification for flow exiting the domain, which is fulfilled in an upwind discretization scheme.

3.6 Pressure-Velocity Coupling Method

Although the Semi-Implicit Method for Pressure Linked Equations (SIMPLE) pressure-correction algorithm provides [50] good convergence for the viscous and inviscid full-domain solutions, it can produce oscillations and in many cases divergence of the solution for the CIVSM when applied to the individual viscous and inviscid domains. The oscillations and divergence are due to interactions of the pressure correction within the SIMPLE algorithm with the mass balance algorithm. This may be corrected by significantly reducing the under-relaxation factors, but that would greatly decrease the rate of convergence and subsequently extend the overall solution time.

However by using the coupled pressure-velocity calculation method, the oscillations noted when using the SIMPLE algorithm can be eliminated. The coupled method [51,52] simultaneously calculates the velocity and pressure field which comes at the cost of requiring roughly twice the amount of memory and computational time per outer-iteration as the SIMPLE algorithm. However, the coupled pressure-velocity method is quite robust and in practice converges in less than half of the number of outer-iterations as the SIMPLE algorithm (if the SIMPLE algorithm does converge). The net result when switching to the coupled pressure-velocity method is increased solution stability at the cost of about twice the memory usage with nearly identical solution time.

Switching from the SIMPLE algorithm to the coupled method provides no additional benefits to the full domain viscous or inviscid models. The results for the conventional CFD/HT results are nearly identical for the SIMPLE algorithm and the coupled method in terms of a final solution and solution time. However the coupled pressure-velocity method required twice the memory usage and does not add stability to the solution for the conventional CFD/HT models.

In summary, the coupled method is required for the individual viscous and inviscid domains using the CIVSM in order to avoid divergence of the solution due to the interaction with the mass balance algorithm. Either the coupled or the SIMPLE pressure-velocity coupling methods can be used for domains not running the mass balance algorithm with nearly identical results. However the coupled method does require roughly twice the memory usage, which is why the SIMPLE algorithm is typically favored.

3.7 Convergence Criteria

Each of the solving steps in the CIVSM (see figure 10) requires a set of convergence criteria before proceeding to the next step. Beginning with the initial solve step, the overall convergence is not critical since its function is mainly to ensure that the solution is properly started such that the residuals begin to converge, and the solution allows the second solve step to properly converge. With complicated flows it is often necessary to start the flow solver off using the inviscid equations. Therefore once the residuals start to decrease monotonically, the initial solve step can be stopped and the solution can be applied to the second solve step.

The second solve step should use typical convergence criteria since the solution to this step is used to divide the domain using the partitioning algorithm. The partitioning algorithm should have a good representation of the flow, especially the vorticity field, in order to properly divide the domain into the viscous and inviscid domains. Typical values for convergence are a reduction in the unscaled residuals of three orders of magnitude, steady values of critical monitor points or regions, or unscaled residuals compared with an appropriate value such as inlet conditions.

In order to achieve the highest computational efficiency, the viscous and inviscid domain solve steps require a fine balance between over-solving and under-solving their respective equations. Since the overall solution is based on a combination of the two domains and each domain affects the other, over-solving each domain can needlessly lengthen the overall solution time. However, under-solving each domain may lead to a poor update to the boundary conditions for the other domains which could lead to an increased number of outer inviscid-viscous domain solve iterations or even divergence of the solution.

To ensure adequate convergence within each solution domain, an additional set of convergence criteria are defined that monitor the change in the values of the other domains' boundary conditions. The set of convergence criteria are based on the residuals of the mass-weighted average of the velocity and temperature values through the other domains' boundary. Convergence of a particular domain occurs when either the overall typical convergence criteria are met, or if the velocity and temperature boundary conditions for the other domains have stopped changing. This prevents the algorithm from over-solving when the boundary conditions for the other domains have already converged.

The last set of convergence criteria are the overall convergence criteria for the CIVSM. Overall convergence occurs once all of the individual viscous domains have met the outer-convergence criteria or a predefined number of outer-iterations have been completed. Once all of the components of the updated boundary conditions stop changing between successive outer-iterations for an individual viscous domain, that viscous domain has reached its outer-convergence criteria. This is done in much the same manner that the individual domains are considered converged when the residuals for the boundary conditions for the other domain have reached their convergence criteria. The scaled residuals for the outer-iteration convergence are defined as:

$$R_{\phi} = \frac{\sum_{j} \left| \phi_{n,j} - \phi_{n-1,j} \right|}{\sum_{j} \left| \phi_{n,j} \right|} \tag{15}$$

where $\phi_{n,j}$ is the mass flux for one of the three Cartesian coordinates or the temperature for the *n*th outer-iteration and the *j*th face of the boundary.

After an individual viscous domain has met the outer-convergence criteria as described above, it is no longer solved for in each successive outer-iteration of the CIVSM and no further boundary condition updates are made to the inviscid domains. Once all of the viscous domains have converged or a specified number of outer-iterations has been completed then the iterative process is concluded and the individual solution domains are combined into a final solution.

3.8 Results and Discussion

The CIVSM was applied to the single-rack data center test cell numerical model described in detail in chapter 2. The commercially available finite-volume, CFD/HT code ANSYS Fluent [33] was used to model the physical layout of the single-rack data center test cell for the three perforated-tile, 23 kW case. The same IBM IntelliStation Z Pro [34] system was used for all of the reported solution times in this dissertation and allow for direct comparison of computational effort.

A new grid study was performed in order to determine the appropriate grid sizes for the different steps of the method. Also an extensive design of experiments was performed in order to determine the appropriate values of the newly defined parameters in the CIVSM.

3.8.1 Parameter Definition

A number of previously undefined parameters are required in order to execute the CIVSM. An optimization algorithm in the form of an extensive design of experiments was performed which led to the formation of an optimized set of parameters which produce consistent results. A list of the input parameters that were used as well as the model calculated values can be found in table 2.

Table 2. List of parameters used in the CIVSM

Model Inputs			
Parameter	**Description**	**Value**	**Unit**
ω_{min}	Vorticity magnitude minimum	1.0	s^{-1}
ω_{max}	Vorticity magnitude maximum	5.0	s^{-1}
P_1	Percentile 1 (vorticity magnitude 1)	10%	
P_2	Percentile 2 (vorticity magnitude 2)	2.5%	
P_3	Percentile 3 (vorticity magnitude 3)	1.0%	
R	Radius of viscous region	2	Cells
η_{min}	Minimum ratio (marked/total)	5%	
η_{max}	Maximum ratio (marked/total)	35%	
N	Maximum number of iterations	10	
	Interface thickness	3	Cells
ε_{max}	Mass balance stopping criterion	$10^{-9}+10^{-6}*\lvert\xi_1\rvert$	
Model Outputs			
Parameter	**Description**	**Value**	**Unit**
$\omega_{cutoff1}$	Vorticity magnitude 1	1.35	s^{-1}
$\omega_{cutoff2}$	Vorticity magnitude 2	3.0	s^{-1}
$\omega_{cutoff3}$	Vorticity magnitude 3	4.65	s^{-1}
η_{final}	Final ratio (marked/total)	22.7%	

The first set of parameters that needed to be defined was for the operation of the partitioning algorithm. The three vorticity magnitude cutoff values ($\omega_{cutoff1} < \omega_{cutoff2} < \omega_{cutoff3}$) were calculated to be 1.35, 3.00, and 4.65 s^{-1} based on the decreasing percentiles ($P_1 > P_2 > P_3$) of vorticity magnitude within the domain of 10, 2.5, and 1%, respectively. In other words, 90% of the cells within the domain had a vorticity magnitude less than 1.35 s^{-1}, 97.5% were less than 3.00 s^{-1}, and 99% were less than 4.65 s^{-1}. By setting $R = 2$, the partitioning algorithm produced three viscous regions that comprised 22.7% of the cells within the domain, which is between the limits defined as 5% and 35% for the minimum and maximum, respectively.

The overall partitioning algorithm is quite fast and takes less than two seconds to run through to convergence for the single-rack data center test cell. In general, it takes about one second for every 25,000 cells in the grid per partitioning algorithm iteration on the IBM IntelliStation Z Pro [34]. So for a 100,000 cell grid that requires three iterations to converge, it would take about 12 seconds. For another data center case where the model was determined to be fully turbulent, the partitioning algorithm ran for the equivalent of 10 outer-iterations of the flow solver. Since that model was stopped after 500 outer-iterations of the flow solver, the attempt to solve that case using the CIVSM method added less than 3% to the total solution time.

The partitioning algorithm created three viscous regions. As can be seen in figure 11, the three viscous regions were located at the major flow inlets and exits of the domain. Region one surrounded the IT equipment rack and the floor tiles in front of the rack. Region two was comprised of the bypass floor tiles while region three encompassed the entrance to the CRAC unit.

Although increasing the interface thickness increases the number of cells that are solved in both the inviscid and viscous domains, it was found to also decrease the number of outer-iterations required for overall convergence. This is due in part to oscillations that develop in the boundary conditions from the iterative method. A larger interface thickness helps to dampen the effects of these oscillations in the boundary conditions by creating a buffer between the two boundaries. This dampening effect helps to speed convergence by requiring fewer outer iterations of the CIVSM.

However, too large of an interface thickness can create interferences between multiple viscous regions that must be resolved in the partitioning algorithm. The partitioning algorithm automatically modifies the necessary parameters to eliminate these interferences. It does this by first trying to reduce the interface thickness, and then if unsuccessful, the algorithm modifies the cutoff criteria to either incorporate more of the domain into the viscous regions to eliminate the gaps between the regions, or reduce the size of the viscous regions to create larger gaps which can accommodate the interface thickness.

Excessively large grid counts for the viscous and inviscid domains, which lead to increased solution time, can also occur due to too large of an interface thickness. A balance must be made between increasing the total number of grid cells in the overall model, which increases individual domain solution time, and effectively reducing the number of outer iterations.

An interface thickness of three grid cells provided the fastest overall solution times and did not hinder the formation of the different domains during partitioning. An interface thickness of two grid cells reduced outer inviscid-viscous iteration solution time, but significantly increased the number of outer inviscid-viscous iterations required for overall convergence. Increasing to four grid cells for the interface thickness did not reduce the number of outer-iterations enough to overcome the increase in solution time per outer-iteration. The partitioning algorithm also needed to make modifications to the different default parameters when trying to specify a four grid cell interface thickness and required three iterations before creating a good partition.

Once the inviscid and viscous domains were defined, the next sets of parameters that need to be defined were the convergence criteria for the two domain types. Increasingly more stringent sets of convergence criteria were employed for both domain types in order to try to prevent over- or under-solving each domain. This helped reduce the number of inner and outer-iterations thereby decreasing solution time.

By solving the viscous and inviscid domains' equations for the first few outer inviscid-viscous iterations more loosely, the overall solution converged more quickly by not over-solving during the inner iterations. The later outer inviscid-viscous iterations however required more stringent convergence criteria from the inner iterations so that a more accurate solution could be obtained.

3.8.2 Grid Study

A new grid study was performed for the CIVSM. A coarse grid with 3998 nodes, 3218 cells, and a maximum control volume of 0.4138 m^3 was used as the base grid for the study. Five grids, which are outlined in table 3, were built from the base grid with different levels of refinement. The coarsest of the five grids, grid #1, used the base grid for the inviscid domain and one level of refinement for the viscous domain. Each level of refinement cuts each cell in half for each dimension of the grid. This effectively increases the number of cells by a factor of eight for the three dimensional grid.

Table 3. Details of the Various Grids Used in the Grid Study

	Grid #				
	1	2	3	4	5
Level of refinement					
Inviscid	0	0	1	1	2
Viscous	1	2	2	3	3
× factor					
Inviscid	1	1	8	8	64
Viscous	8	64	64	512	512
Number of cells					
Inviscid	2648	2648	19888	19888	159104
Region	1601	1601	14634	14634	117072
Viscous	12936	103488	88880	711040	711040
Region	4560	36480	46848	374784	374784
Total	15584	106136	108768	730928	870144
Max Cell Size (m^3)					
Inviscid	4.14×10^{-1}	4.14×10^{-1}	5.17×10^{-2}	5.17×10^{-2}	6.47×10^{-3}
Viscous	2.27×10^{-2}	3.08×10^{-3}	6.47×10^{-3}	8.08×10^{-4}	8.08×10^{-4}
Volume (m^3)					
Inviscid	231.895				
Region	191.541				
Viscous	57.922				
Region	17.568				
Interface Region	40.354				
Total Domain	249.463				

The results presented here are from grid #3 since it was found to be grid independent in the same way that the previous results from the standard CFD/HT models used in [15] that is Grid #3 showed less than a 1.0°C RMS difference between both grid #4 and grid #5. The maximum control volume for the inviscid domain was 0.0517 m^3 which is a little less than twice the size of the maximum control volume of 0.027 m^3 used in the standard CFD/HT inviscid grid. Similarly the maximum control volume for the viscous domain was 6.47×10^{-3} m^3 which is a little less than five times the size of the maximum control volume of 1.33×10^{-3} m^3 used in the standard CFD/HT viscous grids. Even though the grid was somewhat larger for both the inviscid and viscous domains as compared with their standard CFD/HT counterparts, the overall solution did not change significantly with further grid refinement.

3.8.3 Mass Balance Algorithm

The stopping criterion used for the mass balance algorithm was:

$$\varepsilon_{max} = 10^{-9} + 10^{-6} * \left|\xi_1\right|. \tag{16}$$

This criterion consists of the base value of 10^{-9} and a reduction of the initial mass imbalance by six orders of magnitude. If the mass imbalance is very low to begin with, then the base value allows the algorithm to converge. Otherwise the stopping criterion requires a significant reduction in mass imbalance.

In practice the algorithm only took three to five iterations to converge to the stopping criterion. The mass balance correction factors were typically less than 5% for the viscous domains and less than 2% for the inviscid domains. That means that the magnitudes of the velocity vectors were modified by the mass balance algorithm on the order of 2-5% in order to satisfy global continuity. Not only does the mass balance algorithm have to account for the different flow models in the different domains, but also the non-conformal mesh between the different domains.

3.8.4 RMS Difference vs. Solution Time

Figure 15 shows the overall RMS difference between the measured and predicted values for the 7,650 temperature data points plotted against the solution time needed to converge for each of the traditional flow models and the CIVSM for five increasingly more stringent convergence criteria. The traditional models used the default values of less than 0.001 for the flow equations and less than 10^{-7} for the energy equation for convergence or 500 iterations, which are the same conditions that were used in [15]. Except for the zero-equation model, none of the traditional models converged before reaching the maximum 500 iterations. The convergence criteria for the CIVSM models can be found in table 4. The turbulence model and the overall flow convergence criteria used for each of the CIVSM runs are shown in the legend of figure 15.

The inviscid model returned the fastest solution and also the lowest deviation of all of the traditional CFD/HT models. The next seven fastest solutions were from the CIVSM with either the zero-equation or the standard k-ε turbulence model and different convergence criteria. These solutions not only ran faster than all of the traditional viscous CFD/HT models, but also produced lower overall RMS deviation than all of the traditional CFD/HT models. The solution to the CIVSM model running the standard k-ε turbulence model with the strictest convergence criteria reduced the overall RMS deviation even further to 2.63°C, more than 0.5°C less than the inviscid solution.

The traditional CFD/HT viscous models solved in the order of their complexity. The zero-equation and the laminar models solved in about the same amount of time, while the more complex Spalart-Allmaras (one-additional-transport equation) and standard k-ε (two-additional-transport equations) models took significantly longer. The solution to the CIVSM model running the standard k-ε turbulence model with the strictest convergence criteria finished iterating slightly faster than the time required for the Spalart-Allmaras model to complete 500 iterations. However, the solution to the CIVSM

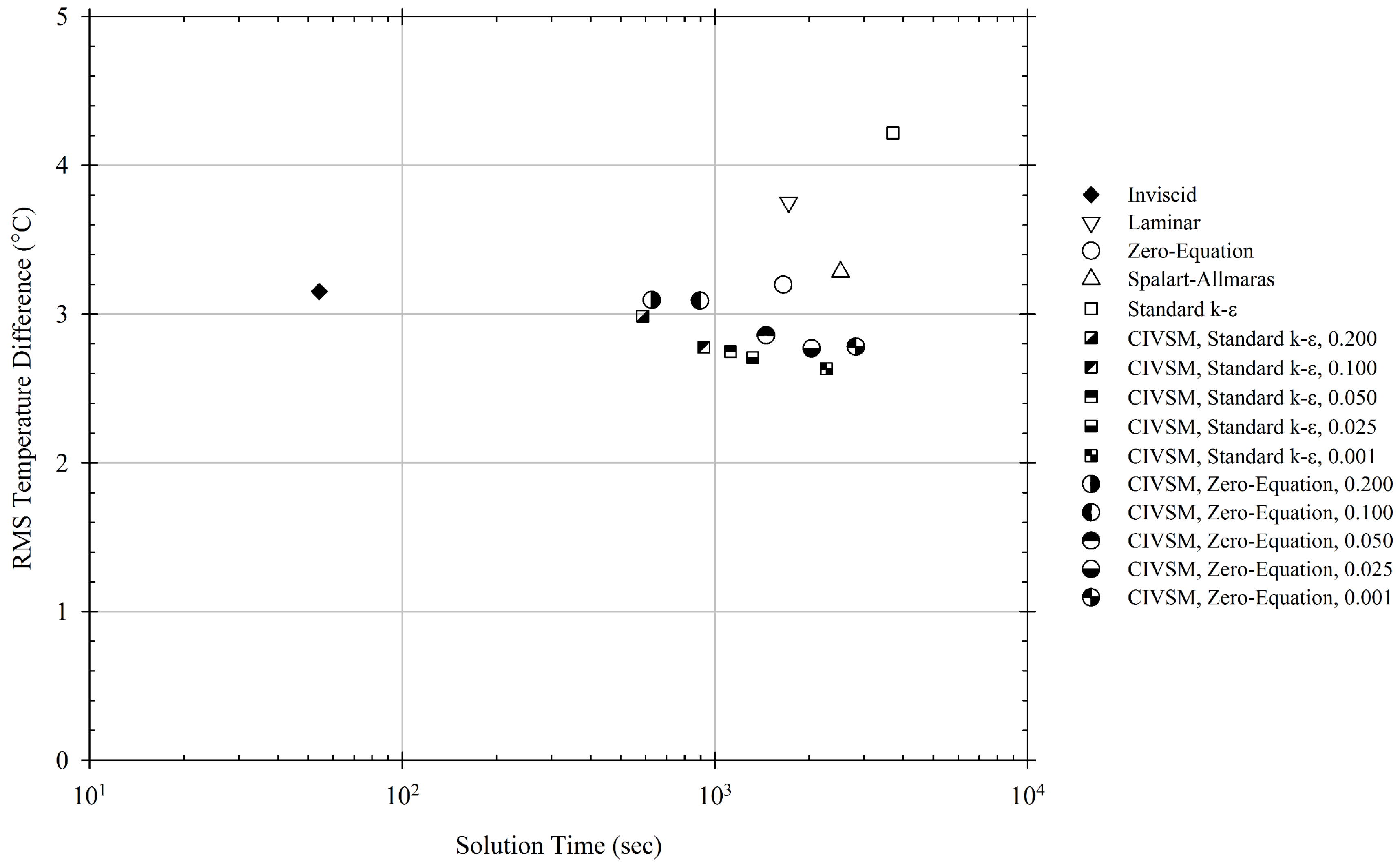


Figure 15. RMS Temperature Difference vs. Solution Time for the CIVSM and CFD/HT Cases

Table 4. Convergence Criteria Used for the CIVSM Models

Solution Step	**Parameter**	**Value**		
Initial Solve	Continuity	10^{-3}		
	Velocity	10^{-3}		
	Temperature	10^{-6}		
	Iterations	10		
	Model	Inviscid		
	Coupling Method	SIMPLE		
Second Solve	Continuity	10^{-3}		
	Velocity	10^{-3}		
	Temperature	10^{-6}		
	Iterations	100		
	Model	Zero-Equation		
	Coupling Method	SIMPLE		
Viscous Domain Solve	Continuity	10^{-3}	8×10^{-4}	5×10^{-4}
	Velocity	10^{-3}	8×10^{-4}	5×10^{-4}
	Temperature	10^{-6}	10^{-6}	10^{-6}
	Iterations	20	30	40
	Velocity BC monitor	5×10^{-3}	3×10^{-3}	10^{-3}
	Temperature BC monitor	5×10^{-6}	3×10^{-6}	10^{-6}
	Model	Standard k-ε or Zero-Equation		
	Coupling Method	Coupled		
Inviscid Domain Solve	Continuity	2×10^{-3}	10^{-3}	8×10^{-4}
	Velocity	2×10^{-3}	10^{-3}	8×10^{-4}
	Temperature	2×10^{-5}	10^{-5}	5×10^{-6}
	Iterations	40	60	80
	Velocity BC monitor	1.5×10^{-2}	9×10^{-3}	3×10^{-3}
	Temperature BC monitor	5×10^{-5}	10^{-5}	5×10^{-6}
	Model	Inviscid		
	Coupling Method	Coupled		
Overall Convergence	Overall Continuity	0.2, 0.1, 0.05. 0.025, or 10^{-2}		
	Overall Velocity	0.2, 0.1, 0.05. 0.025, or 10^{-2}		
	Overall Temperature	5×10^{-4}		
	Outer-Iterations	2	4	30

model running the zero-equation turbulence model with the strictest convergence criteria finished iterating slightly slower than the Spalart-Allmaras model completed 500 iterations.

All of the models employing the CIVSM spent roughly 38% of the time solving the inviscid domain, 23% of the time solving the viscous domains, and 4% of the time performing all of the operations prior to solving the two domain types. This means that roughly 35% of the time was spent performing operations other than solving or preparing the domains. The time not spent solving is mainly comprised of saving the solution, closing and opening the next domain, and then calculating and setting the boundary conditions. By further optimizing the code to minimize the dwell time between solving operations through parallelization, the overall solution time for running the CIVSM could be significantly reduced without changing the solution.

As the convergence criteria were tightened for the CIVSM, the overall RMS deviation fell. The modeling technique employing the standard k-ε turbulence model consistently produced the lowest overall RMS deviation of all of the models while solving faster than all but the inviscid model (except with the most stringent convergence criteria). However, the overall RMS deviation does not take into consideration any localized effects.

3.8.5 Temperature Difference Contours

Figure 16 shows the measured temperature contours, the predicted temperature contours from the CIVSM using the standard k-ε turbulence model with the most stringent convergence criteria, the absolute temperature difference between the measured data and the CIVSM using the standard k-ε turbulence model with the most stringent convergence criteria, and the location of the viscous, interface, and inviscid regions. Figure 17 depicts the absolute temperature difference between the measured data and the five traditional CFD/HT flow model predictions. The temperature and absolute

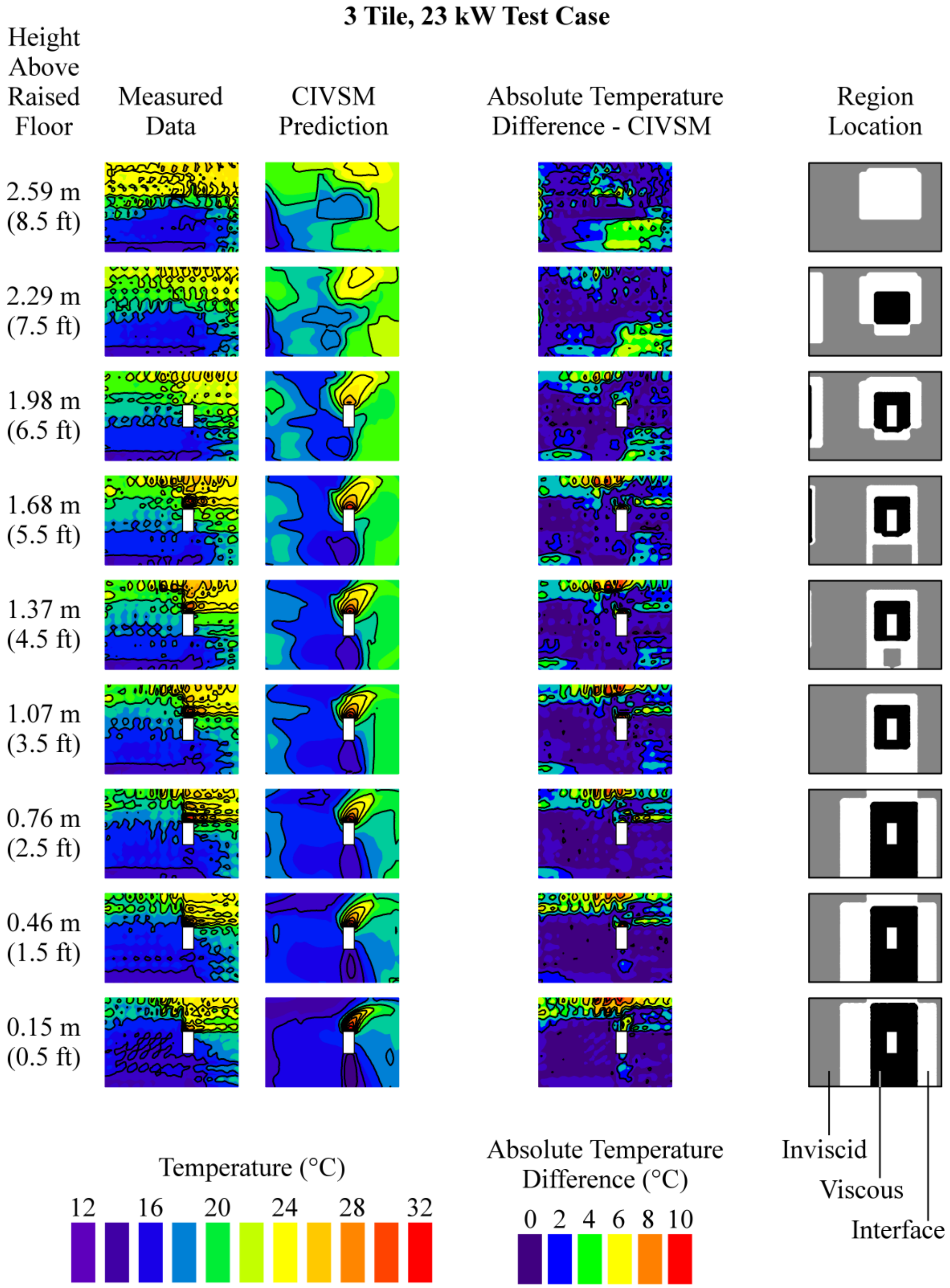


Figure 16. Temperature Contours, Absolute Temperature Difference Contours, and CIVSM Region Locations at the Nine Elevations within the Region of Interest of the Single-Rack Data Center Test Cell for the 3 Tile, 23 kW Case

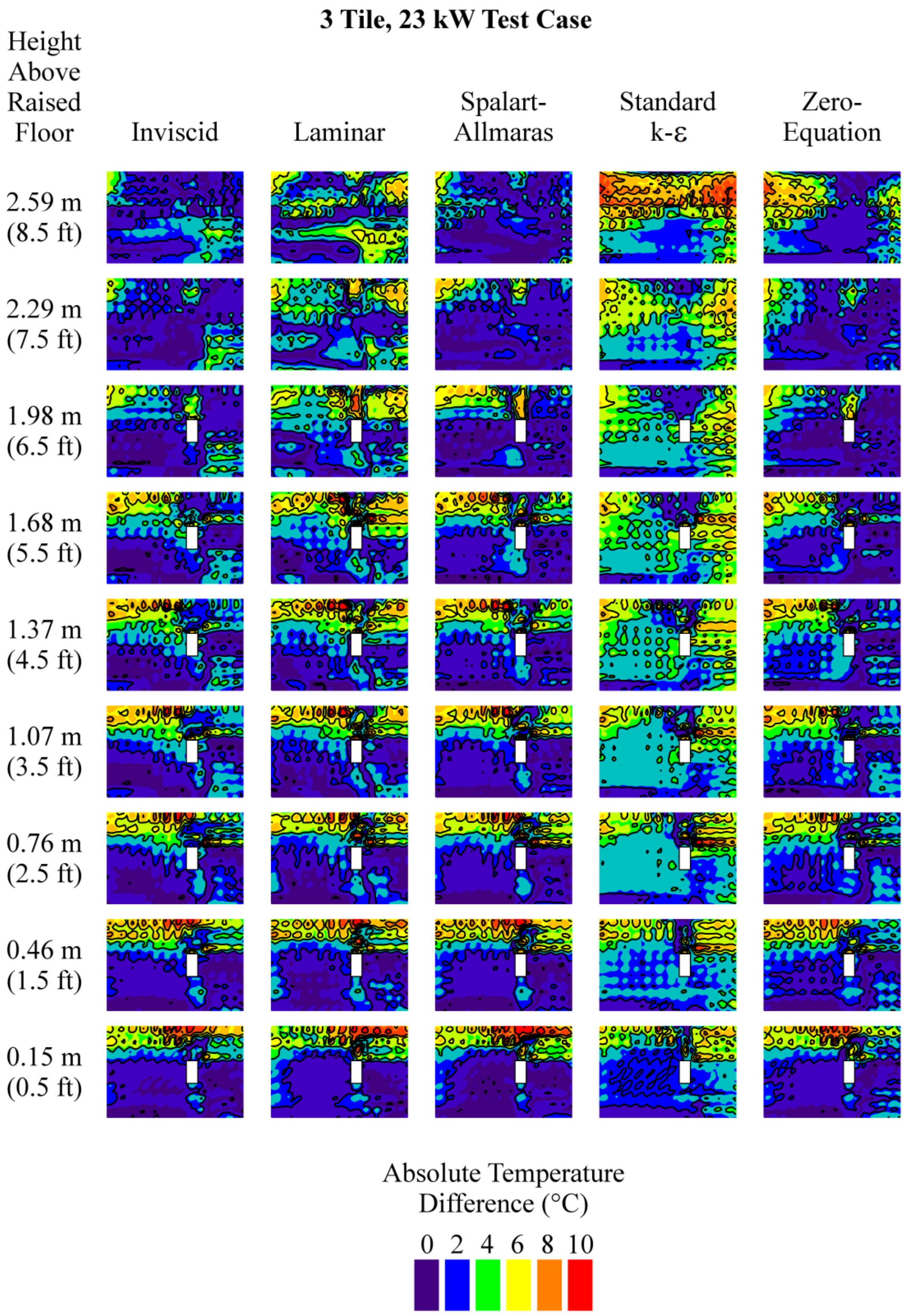


Figure 17. Absolute Temperature Difference Contours within the Region of Interest of the Single-Rack Data Center Test Cell for the 3 Tile, 23 kW Case

temperature difference contours plot all 7,650 data points along the nine heights for both figures 16 and 17. All of the contour planes show the region of interest in the same orientation as is shown in figure 3. The IT equipment rack is shown as blank space in the middle of the contours at the lower seven heights.

The CIVSM does a good job of predicting the temperature field of the region of interest in the single-rack data center test cell. At a high level, the deviations from the measured data can be seen in the contours that plot the absolute temperature differences and general comparisons between the effectiveness of the different models predictions can be made.

When comparing the different models' absolute temperature difference contours it is plain to see that all of the models had a significant amount of deviation at the lower and mid-elevations near the wall where the IT equipment rack was exhausting. The model that performed the best in this region was the standard k-ε model. However the standard k-ε model was the worst performing model nearly everywhere else. In fact, where the best traditional CFD/HT model, the inviscid model, performed the worst is where the standard k-ε model performed the best. It can also be seen that the zero-equation model was the next best model in the regions of highest deviation of the inviscid model. Although the standard k-ε and the zero-equation models do a better job of predicting the temperature field in the areas of highest deviation of the inviscid model, the locations of the viscous and interface regions in the CIVSM model are not necessarily in those same areas. Figure 16 shows the location of the inviscid, viscous, and interface regions, as defined by the partitioning algorithm. When comparing the different viscous models in the viscous and interface regions, one would expect that when running the CIVSM with the standard k-ε and the zero-equation models would produce the best results. Not only did that turn out to be the case, but also the laminar and Spalart- Allmaras viscous models produced worse results when run in CIVSM than as a traditional CFD/HT model.

The CIVSM model using the standard k-ε turbulence model not only produced the lowest overall RMS difference, but also improved on the inviscid model's localized deviation at nearly every elevation. The only exceptions are at the highest two elevations near the front of the rack. At these elevations a small patch of low deviation in the inviscid model seems to have grown both in size and deviation. This is due to the viscous domain passing higher temperature air (due to mixing) to the pure convection inviscid domain. The inviscid domain then advects this higher temperature air back to the CRAC unit rather than mixing with the cooler air exiting the perforated floor tiles. All of the traditional models do a better job of predicting the temperatures in this area due to either mixing with the cooler air from the perforated floor tiles or, as in the case of the inviscid model, simply advecting cooler air back to the CRAC unit.

Other than the area of higher deviation described above, the CIVSM does just as good, if not better, of a job predicting the temperature field. Visually analyzing the local deviations affirms the lower overall RMS deviation of the CIVSM compared with the traditional CFD/HT models. The CIVSM has fewer and smaller areas of higher deviation than any of the traditional CFD/HT models.

3.9 Summary

A novel CIVSM for bounded domains was developed that has the potential to produce more accurate temperature field predictions for data centers with similar solution times as compared with traditional CFD/HT. By combining the increased speed potential and more accurate results of the inviscid solver in the inviscid domain along with the viscous solver's ability to capture the turbulent flow physics in the viscous domain, a faster and overall more accurate solution can be obtained for bounded domains that contain large inviscid regions such as data centers. There is room for significant improvements in overall solution time by reducing the dwell time between solving operations in the current implementation of the CIVSM algorithm.

While external flow CFD/HT modeling has taken advantage of a coupled inviscid-viscous solution technique, the problem of defining the inviscid-viscous boundary is straightforward. This new method uses a partitioning algorithm which determines where this interface should be applied based on the flow physics not known a priori. Details of the overall modeling technique, the inviscid-viscous partitioning algorithm, and the required mass-balance algorithm were presented.

This new method also allows for the use of multiple turbulence models for the different viscous domains defined within the overall domain. Multiple grid sizes for the different solution domains allow for more control in the solution accuracy and computational efficiency based on the turbulence models being used in the different viscous domains and the inviscid solver's less stringent grid requirement.

The CIVSM was applied to a small, single-rack data center test cell and the results were compared with traditional CFD/HT. When using the standard k-ε and zero-equation turbulence models, the new modeling technique produced both lower overall RMS deviation and, for lower levels of convergence, faster solution times as compared with traditional viscous CFD/HT models.

Additional verification studies are presented in the following two chapters. Both temperature and flow predictions are compared with both experimental data from the Data Center Laboratory at Georgia Tech as well as traditional CFD/HT predictions.

CHAPTER 4

TEMPERATURE VERIFICATION OF THE CIVSM IN A MULTI-RACK DATA CENTER

In this chapter predictions from the CIVSM, which is described in detail in chapter 3, are compared with experimental temperature measurements from the Data Center Laboratory (DCL) at the Georgia Institute of Technology, as well as predictions from traditional CFD/HT. The flow models are quantitatively compared with measured inlet air temperatures of the IT equipment. The models are also evaluated on the amount of computational effort they required to obtain the predictions.

This research utilizes measured data from [53] in order to verify the performance of the CIVSM. It focuses on the temperature inlets to the IT equipment in the DCL using an open aisle and a closed aisle arrangement. Due to the difficulties in determining the locations of the air leakage between the contained cold aisle and the rest of the data center, only the open aisle arrangement has been modeled. The air-flow supplied to the room through the perforated tiles was varied by turning on or off one of the two CRAC units in the laboratory. This provided two cases of experimental data which were then compared with traditional CFD/HT models using two perforated-tile modeling techniques: a porous jump model, and a modified body force model. Details of the tile modeling techniques are provided in section 4.2.2.

4.1 Experimental Setup of the Data Center Laboratory

The DCL is split into two 56 m^2 (600 ft^2) rooms which both have a 2.64 m (8 ft 8 in.) ceiling. One was used for the temperature verification test cases presented here, and the other was used for the flow verification test cases delineated in chapter 5. The layouts are nearly identical except for most notably the IT equipment rack placement and the cold aisle containment geometry. Figure 18 shows an isometric view of the numerical model which illustrates the layout of the DCL at Georgia Tech, which was used for the temperature measurements.

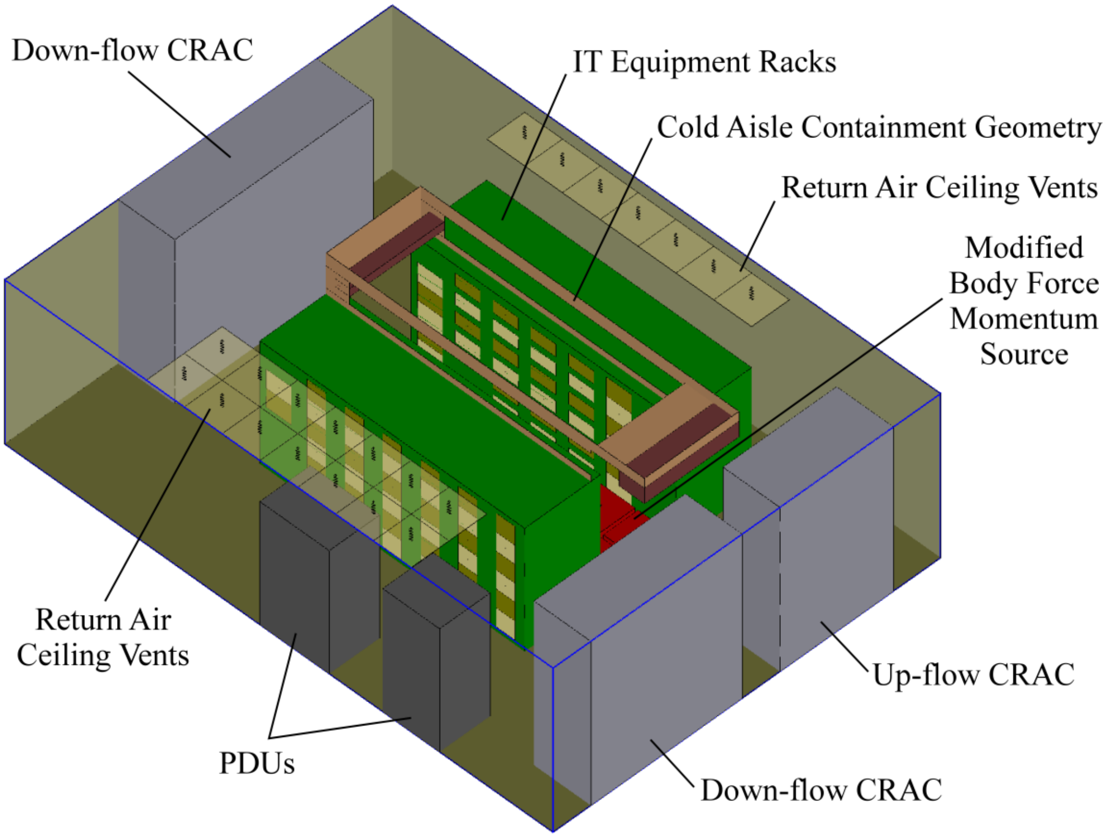


Figure 18. Isometric View of the Data Center Lab at Georgia Tech

The half of the DCL used for the temperature verification test cases consists of a single cold aisle with 14 perforated tiles, 14 racks of IT equipment, three CRAC units, and two power distribution units (PDUs). The racks are divided into two rows of seven racks each on either side of the cold aisle. Each rack has a generic 0.61 m × 0.61 m (2 ft × 2 ft) perforated tile located in front of it which supplies the cooling air to the IT equipment. Of the three CRAC units, two are down-flow units and one is an up-flow unit. Only the down-flow units were used in these tests, while the up-flow unit was blocked to prevent recirculation.

The IT equipment is housed in standard 42 U (1 U = 0.0445 m) racks. Ten of the racks contain IBM Blade Centers and are fitted with rear-door heat exchangers that partially cool the hot exhaust air before it returns to the data center. One rack each are populated with standard 1 U servers, networking equipment, and storage equipment. The remaining rack is empty but incorporates blanking panels to prevent recirculation of hot aisle air into the cold aisle.

Inlet and exhaust air temperatures along with anemometer measurements were taken for every 7 U section of the racks that housed IT equipment. These data were then used to calculate the air-flow rate and heat load for each of the racks. On average each populated 7 U section produced 860 W of bulk heating and 0.11 m^3/s (230 CFM) of air-flow for a total heat load of 61 kW and total rack air-flow of 7.8 m^3/s (16,000 CFM).

The perforated-tile air-flow rates were also measured for one down-flow CRAC operational, and for both down-flow CRACs operational. With one CRAC on, the total tile air-flow rate provided 79% of the total rack air-flow rate. With both CRAC units running the tiles supplied 121% of the total rack air-flow.

Further details on the DCL boundary conditions and measurement techniques can be found in [53].

4.2 Numerical Models

Four CFD/HT models were built using the commercially available finite-volume method solver, ANSYS Fluent [33]. The models were constructed to match the physical layout of the DCL. Each of the models consisted of ten major components: three CRAC units, two PDUs, two rows of IT equipment racks, an aisle of perforated tiles in front of the IT equipment racks, ceiling return air vents, and walls (including ceiling and floor). Only the space above the raised floor was modeled, eliminating the added complexity of the under floor air plenum.

The modeled inlet tile flow rates in front of the IT equipment racks were set to their measured values from the DCL as fixed air-flow boundary conditions. The IT equipment racks were modeled as a black boxes with their measured powers and flow rates. The hot-air return-vents were set as pressure outlet boundary conditions and balanced the inlet air from the perforated tiles.

4.2.1 Flow Modeling

Only the inviscid, zero-equation, and standard k-ε flow models were used on the DCL numerical models. The traditional CFD/HT models used either the inviscid model, or the standard k-ε model. The CIVSM models used the inviscid and standard k-ε models during the iterative scheme, and the zero-equation as the basic turbulence model before dividing the domain.

For the standard k-ε turbulence model the turbulent intensity and length scale were set to 5% and 2.54 mm (0.1 in.) for all of the domain inlets respectively. However these values were set to 2.5% turbulent intensity and 61 mm (2.4 in.) at all of the interface boundary conditions during the viscous domain solve steps in the CIVSM iterative process. This was done to lower the amount of turbulence entering the viscous domain from the inviscid domain. Since the inviscid domain does not calculate these quantities and measured flow data are unavailable, they were set based on engineering judgment from the recommendations of [51,52]. The turbulent intensity is defined as:

$$I \equiv \frac{u'_{RMS}}{\bar{u}} \tag{17}$$

where I is the turbulent intensity, u'_{RMS} is the RMS of the velocity vector fluctuations, and $\bar{u}$ the mean (time averaged) velocity vector from the Reynolds decomposition. Values of 1% or less are considered low while values of 10% or greater are considered high. Without flow measurements the default value is 5%, which is considered medium intensity, and is too high for flow entering the domain which is also modeled as inviscid. The length scale is generally taken to be about an order of magnitude less than the relevant length scale of the problem [54]. Therefore the length scale was chosen as 1/10th of the length of a floor tile.

The values for turbulence intensity and length scale were confirmed by inspecting the calculated values from the full CFD/HT models using the k-ε turbulence model in the locations of the interface boundaries from the CIVSM models. The following equation [51,52] was used to estimate the length scale at these locations:

$$\ell = C_{\mu}^{3/4} \frac{k^{3/2}}{\varepsilon} \tag{18}$$

where ℓ is the length scale, C_{μ} is the empirical constant defined by the turbulence model (0.9 for the standard k-ε model), k is the turbulent kinetic energy, and ε is the turbulent dissipation.

4.2.2 Tile Modeling

Two different tile modeling techniques were evaluated: the porous jump model, and the modified body force model. To better model the air-flow through a perforated tile without having to resolve the fine scale details, Arghode, et al. [55-57] proposed using a modified body force modeling technique. This modeling technique produces similar results to the fine grid predictions, but allows for much coarser mesh than a detailed perforated-tile model.

When not modeling the underfloor air-flow in a data center, the porous jump model essentially becomes a uniform, prescribed inlet flow to the model. This is typically what is employed above raised-floor data center models, and was used for all of the single-rack data center test cell cases. However, this type of model does not take into consideration the momentum changes which occur when the air passes through the perforations of the floor tile.

When passing through a perforated tile, air must travel through a significant restriction through the perforations and then expand into the data center at the exit. This rapid acceleration creates a large momentum difference between the air exiting the pores of the tile and the surrounding air. This momentum difference is neglected when using the porous jump model, but is simulated by adding the modified body force.

The modified body force model is a volumetric momentum source that is placed above the perforated tile to simulate the acceleration of the air through the tile perforations. The modified body force source strength, S_{MBF} , can be calculated using the following equation:

$$S_{MBF} = \frac{\rho A_{tile} u_{inlet}^{2}}{\forall}\left((K+1)^{1/2}-1\right). \tag{19}$$

where ρ is the density, A_{tile} is the total area of the tile, u_{inlet} is the inlet velocity, $\forall$ is the volume of the momentum source, and K is the pressure loss coefficient across the tile. A good estimate for the height of the momentum source was determined to be 12 pore diameters [56]. This distance was chosen as a midpoint between the axial distance of significant jet decay of a free jet (18 diameters) and a symmetric array of jets (8 diameters) [56].

The modified body force modeling method for the perforated tiles has been applied to both the CIVSM and traditional CFD/HT models.

4.2.3 Grid Studies

Three grid studies were performed on the numerical models: one each for the CIVSM model, the inviscid CFD/HT model, and the standard k-ε CFD/HT model. The modified body force perforated-tile models were used for the grid studies since they produced the highest velocities and gradients. The 1 CRAC models were chosen because they produced more mixing in the cold aisle where the comparisons were made.

A similar approach was taken for the grid study of the CIVSM models as was done for the single-rack data center test cell [58]. The base grid for the models consisted of 68,488 cells and the largest control volumes were 0.0336 m^3. The intermediate grid refined the viscous domains to a maximum control volume of 2.36×10^{-3} m^3. These are somewhat smaller than the maximum control volumes in the single-rack data center test cell, which were 0.0517 m^3 for the inviscid domain, and 6.47×10^{-3} m^3 for the viscous domains. The grid convergence index (GCI) for the three CIVSM grids is presented in Appendix B.

For the traditional CFD/HT grid studies, the same base grids from the CIVSM models were used and refined in the same manner as well as a nearly uniform grid similar to the one used in [53]. The nearly uniform grid consisted of cells with a maximum linear dimension of 0.508 m and maximum control volume of 1.31×10^{-4} m^3. It was found that for the inviscid model none of the four grids would converge, however after 1500 iterations they all produced similar temperature predictions. For this reason the results presented are from the coarsest of the four grids and were all run for 1500 iterations. The GCI for the two coarsest grids is presented in Appendix B.

For the standard k-ε models, the two coarser grids used for the CIVSM would not converge. The finest of the three grids used for the CIVSM did converge, but contained nearly four times the cells as the nearly uniform grid and produced nearly identical results. Thus, the results are presented from the nearly uniform grid for the standard k-ε models using traditional CFD/HT. The GCI for the standard k-ε model is also presented in Appendix B.

4.3 CIVSM Refinements

The CIVSM modeling technique was further modified from the implementation used on the single-rack data center test cell. The biggest alteration to the method was the inclusion of under-relaxation factors to the application of the boundary conditions. The other modification was to not allow any thin flow blockages (i.e. plates) to bound the viscous or inviscid region. Thin flow blockages were allowed to cross the domains, however they were not allowed to form the boundary of a domain.

4.3.1 Boundary Condition Under-Relaxation

Oscillations in any iterative solution technique have the potential to lead to delayed convergence or even divergence of the solution. Under-relaxation is used to dampen these oscillations and allow the iterative technique to converge more quickly. Large oscillations in the application of the boundary conditions of the CIVSM can lead to divergence of the solution especially for the first couple of iterations. Although this was noted for a couple of the cases when running the CIVSM for the single-rack data center test cell with one grid cell in the interface region, an easy solution was to increase the interface region thickness to dampen out the oscillations.

However, the complex flow patterns of the DCL led to large oscillations in the boundary conditions which significantly lengthened convergence, and in certain cases even led to divergence. This was observed using the default three grid cell thick interface region developed from the single-rack data center test cell results. Therefore under-relaxation was applied to the boundary conditions in order to dampen these oscillations and ensure faster convergence of the solution.

Under-relaxation was applied by using the standard linear model:

$$\phi_{i+1} = \phi_i + \alpha(\phi_{calculated} - \phi_i) \tag{20}$$

where ϕ_{i+1} is the new boundary condition value, ϕ_i is the old boundary condition value, $\phi_{calculated}$ is the calculated boundary condition value, and α is the under-relaxation factor.

4.3.2 Partitioning Algorithm and Thin Flow Blockages

The partitioning algorithm was modified to prevent thin flow blockages from bounding the domain. It was discovered that when the cold aisle containment geometry (see figure 18), which is constructed of thin flow blockages (i.e. plates), bounded either of the domains that the CFD/HT modeling software would not know how to treat the boundary condition. This is a known issue and the software will issue warnings if a model is originally built in this manner, however the check is not in place when dividing the domain as is automatically performed by the CIVSM. Therefore the partitioning algorithm was modified to perform this check and modify the partition to either enclose the thin flow blockage or exclude it from the boundary of the domain.

4.4 Results and Discussion

All of the numerical models were run independently on the same IBM IntelliStation Z Pro [34] system with two dual-core Xeon 5160 CPUs (4 cores total) rated at 3.00 GHz with 16.0 GB of RAM. This is the same machine used for the single-rack data center test cell cases.

4.4.1 CIVSM Parameter Definition

The same parameters from the single-rack data center test cell, which are located in table 2, were used for the DCL models. The only additional parameters that needed to be specified were for the under-relaxation values. A few different sets of values were tested, but the set that gave the most consistent convergence results were 0.2 for the pressure, 0.3 for the velocity components, and 0.5 for the temperature. Slowly increasing these to higher values after the first few iterations may lead to even faster convergence, but only a few sets of constant values were explored.

4.4.2 Partitioning Algorithm

For all of the cases the partitioning algorithm converged in either one or two iterations depending on the case. The maximum amount of time spent running the partitioning algorithm was less than seven seconds.

For all of the cases, the partitioning algorithm created one viscous region that included all of the IT equipment inlets and exhausts and all of the perforated tiles. An example of the location of the viscous region from the 1 CRAC, modified body force case can be seen in figure 19. All of the viscous partitions were very similar and consisted of approximately 15% of the cells in the domain.

4.4.3 Convergence Criteria

For the traditional CFD/HT models, convergence was attained when the solution residuals dropped to values of less than 10^{-4} for the flow equations and less than 10^{-7} for the energy equation or 1500 iterations. These are the same values that were used in [53].

Only three changes were made to convergence criteria as compared with those used in the single-rack data center test cell (see table 4). The first was that the number of iterations for the second solve was increased to 300 to ensure convergence prior to running the partitioning algorithm. The second was that only the k-ε turbulence model was used in these cases. Lastly the overall convergence criteria were lowered.

Three sets of overall convergence criteria were used for the CIVSM to see the impacts on solution time and RMS difference. In order to limit the computational effort, only the three loosest overall convergence criteria were used from the single-rack data center test cell case. The most relaxed of these were 0.2 for the pressure and flow residuals and 0.001 for the temperature residuals. The middle of the three used 0.1 for the

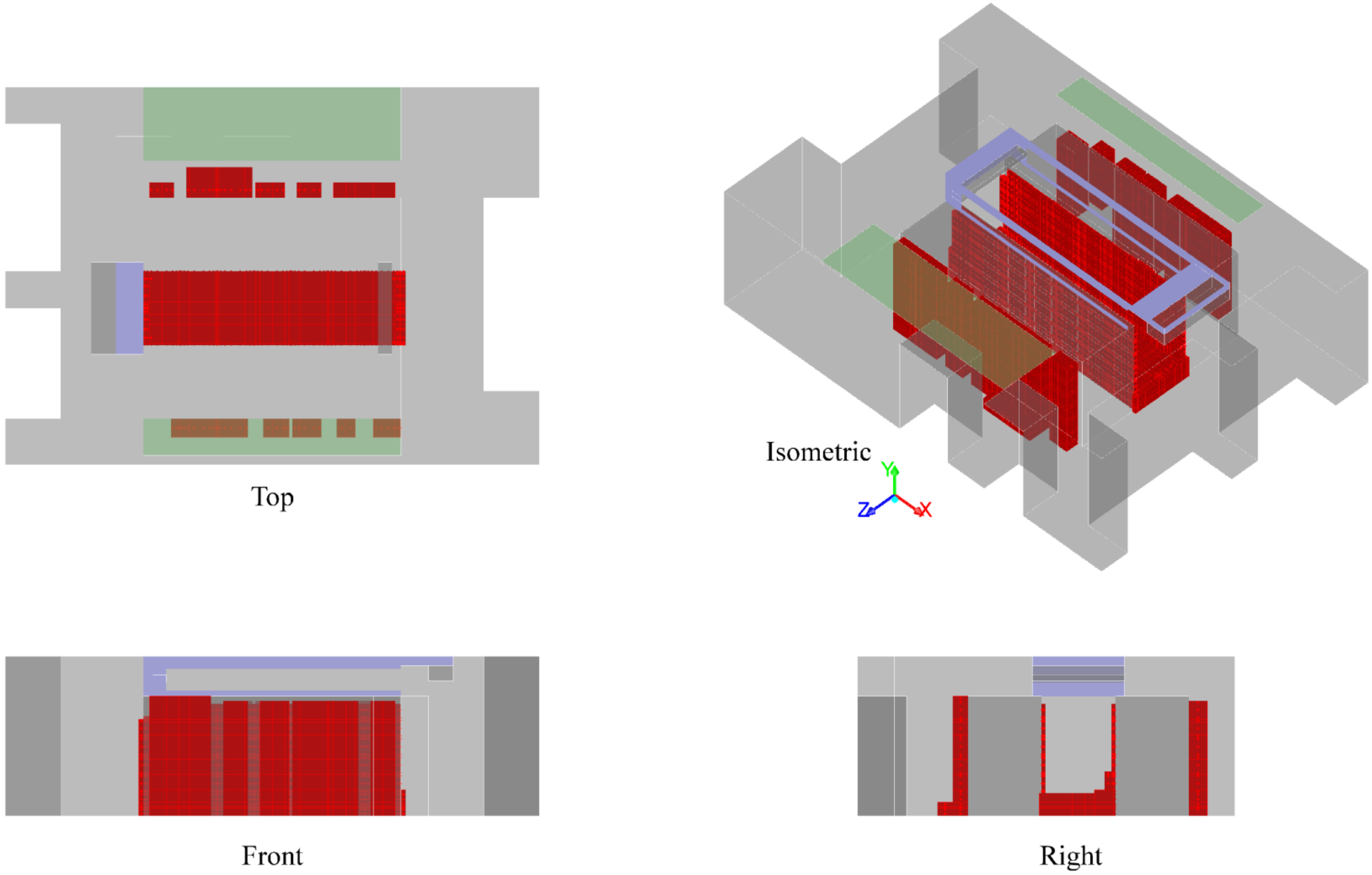


Figure 19. Viscous Regions of the Data Center Lab at Georgia Tech for the 1 CRAC, MBF Case

pressure and flow residuals and 5×10^{-4} for the temperature residual, while the tightest increased the pressure and flow residuals to 0.05 leaving the temperature residual at 5×10^{-4}.

4.4.4 RMS Difference vs. Solution Time

Figure 20 shows the overall RMS difference between measured and predicted values for the mass-averaged inlet temperatures into the 71 pieces of IT equipment housed in the 14 racks plotted against the solution time for all of the models. Each of the viscous model types is marked with a single shape. All of the inviscid models are marked with diamonds, while the standard k-ε models are marked with squares. Similarly, the CIVSM models are all marked with squares of different shading for the different convergence criteria used. For instance, the most stringent convergence criteria for the CIVSM models are filled in the top half of the square. Note that the 2 CRAC, porous jump case met both the 0.1 and the 0.05 convergence criteria on the same iteration and is therefore only shown as meeting the tightest convergence criteria.

The inviscid models, having the coarsest grids and simplest equations, were the first to reach their stopping criteria. The CIVSM models were the next fastest, followed by the standard k-ε models. All of the CIVSM models solved at least three times as fast as their standard k-ε CFD/HT counterpart, with the lowest convergence criteria CIVSM models solving from nine to 35 times faster.

Although the inviscid models solved in the least amount of time, for these cases they also produced the highest RMS temperature differences. Only the 2 CRAC, porous jump model produced RMS temperature differences within 10% of the CIVSM and k-ε models. Not only did the CIVSM models run significantly faster than the standard k-ε models, they also produced similar RMS temperature differences. In fact, only the 2 CRAC, modified body force model with the tightest convergence criteria did not outperform the corresponding standard k-ε CFD/HT model in terms of RMS temperature

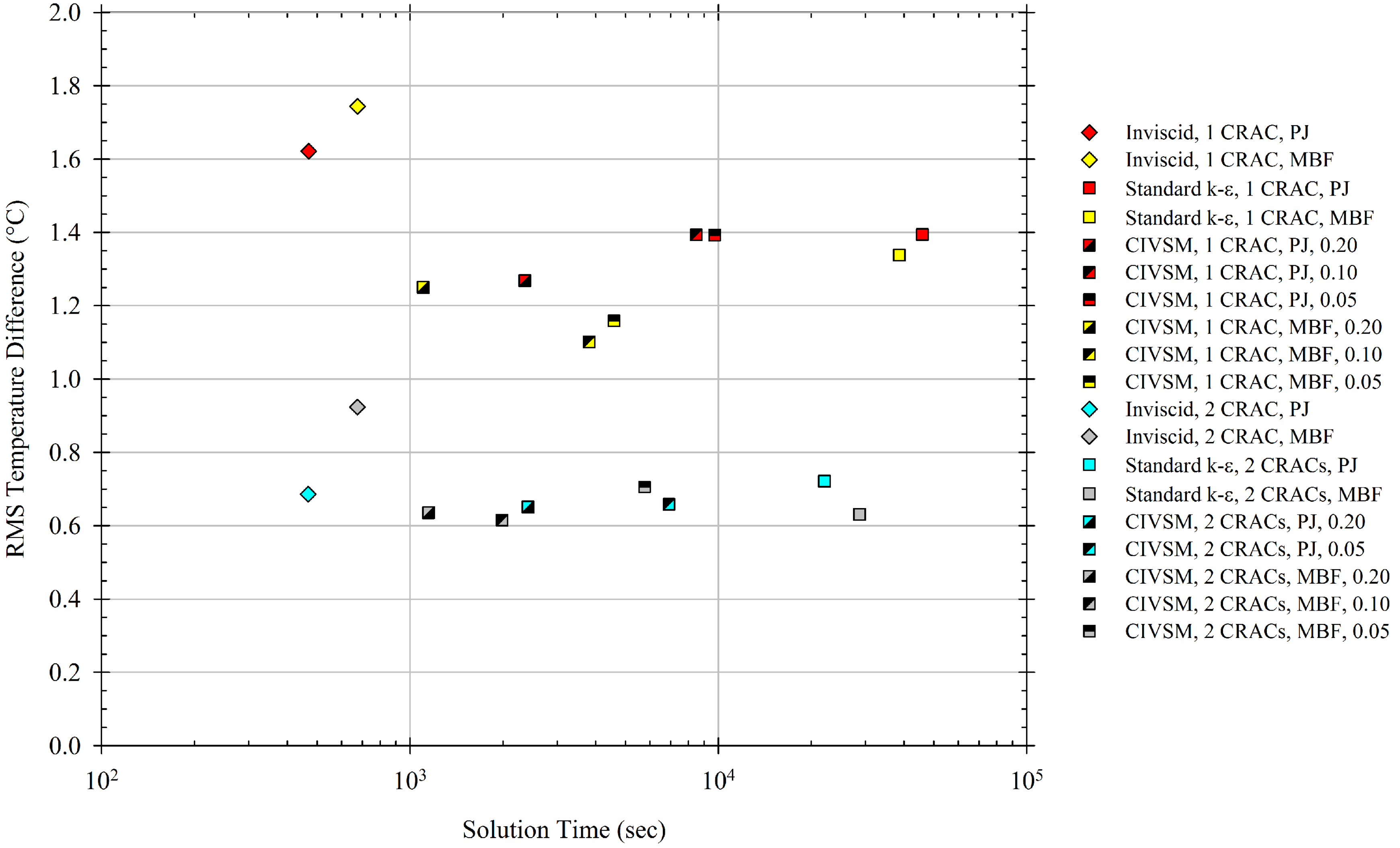


Figure 20. RMS Temperature Difference vs. Solution Time for the DCL at Georgia Tech

difference.

It is also interesting to note that for all of the viscous models, except for the 2 CRAC, modified body force model solved using the CIVSM with the tightest convergence criteria, the modified body force models outperformed the porous jump models in terms of RMS temperature difference. In terms of solution time, only the 2 CRAC CFD/HT model did not solve faster using the modified body force model than the porous jump model.

However, the opposite trend was found for the inviscid models, where the porous jump model produced both lower RMS difference and solution time compared with the modified body force model. This is most likely due to the inviscid model's lack of a means for mixing two flow streams of different temperatures together, and only allows for the increased momentum of the cold air from the perforated tiles in the modified body force models to advect further towards the top of the racks towards the center of the aisle. Thus the modified body force model will, relative to the porous jump model, predict lower temperatures for the IT equipment at the tops of the central racks and predict higher temperatures for the IT equipment at the racks on the ends of the rows where warm air from the room must be "entrained". This phenomena was verified for both of the cases run for this study.

4.5 Summary

The CIVSM was modified to include under-relaxation factors for the boundary conditions between the inviscid and viscous domains. The partitioning algorithm was also updated to eliminate poorly defined boundary conditions caused by plates at the boundaries of the inviscid and viscous domains. These alterations to the CIVSM were then applied to the DCL at Georgia Tech, a compact, multi-rack data center. Temperature predictions from the CIVSM, as well as inviscid and standard k-ε CFD/HT models, were compared with measured data. Two perforated-tile modeling techniques were employed for each of the flow models. Computational effort for all of the models were also evaluated.

Although the inviscid models solved faster than the CIVSM models, the CIVSM models were much more consistent in their temperature predictions. Overall the CIVSM does a good job of replicating the standard k-ε CFD/HT model temperature predictions while significantly reducing solution time. The CVISM outperformed the standard k-ε CFD/HT in terms of solution time by a minimum factor of three, while maintaining the same level of predictive accuracy in terms of RMS difference.

The modified body force tile modeling technique, in conjunction with the CIVSM, performed quite well, and led to reduced solution times and similar RMS difference temperature predictions as compared with the porous jump modeling technique. This technique better simulates the effects of entrainment the air experiences after it flows through the perforated tiles for both the CIVSM and the standard k-ε CFD/HT models as measured by the reduced RMS difference.

However, the inviscid model does not show the same reductions in RMS difference. The inviscid model’s inability to properly entrain the warm air from the room into the cold air supplied by the perforated tiles is suspected to be the reason for the decrease in modeling accuracy of the modified body force model compared with the porous jump model.

Even though the CIVSM has done a good job of reducing solution time and replicating the temperature predictions of traditional CFD/HT in a multi-rack data center, verification of the flow predictions has not yet been demonstrated. Chapter 5 presents comparisons of the CIVSM predictions to PIV measurements from the DCL at Georgia Tech, as well as to traditional CFD/HT predictions.

CHAPTER 5

FLOW VERIFICATION OF THE CIVSM IN A MULTI-RACK DATA CENTER

In this chapter predictions from the CIVSM, which is described in detail in chapter 3, are compared with experimental flow measurements from the DCL at the Georgia Institute of Technology as well as predictions from traditional CFD. The flow models' predictions are qualitatively compared with measured air-flow data taken using particle image velocimetry (PIV). The flow models are also compared with each other on the basis of the computational effort used to obtain the predictions.

This research utilizes previously published, measured data from [59] in order to verify the performance of the CIVSM. The study [59] focused on the air-flow exiting a perforated tile and entering an IT equipment simulator. PIV measurements were taken for seven different cases by varying the perforated-tile design, air-flow rate, and IT equipment simulator rack placement. Due to the similarity of the air-flow patterns only three of the seven cases were modeled for this comparison. They represent the largest changes in perforated-tile design, and subsequently, air-flow patterns. Both the porous jump and modified body force modeling techniques were implemented for the perforated-tile flow in the CIVSM and traditional CFD models. Details of the perforated-tile modeling techniques are provided in section 4.2.2.

5.1 Experimental Setup of the Data Center Laboratory

The DCL is split into two 56 m^2 (600 ft^2) rooms which both have a 2.64 m (8 ft 8 in.) ceiling. One was used for the flow verification test cases presented here, and the other was used for the temperature verification test cases described in chapter 4. The layouts are nearly identical except for most notably the IT equipment rack placement and the cold aisle containment geometry. Figure 21 shows an isometric view of the domain simulated numerically, which illustrates the layout of the half of the DCL at Georgia Tech used for the flow measurements.

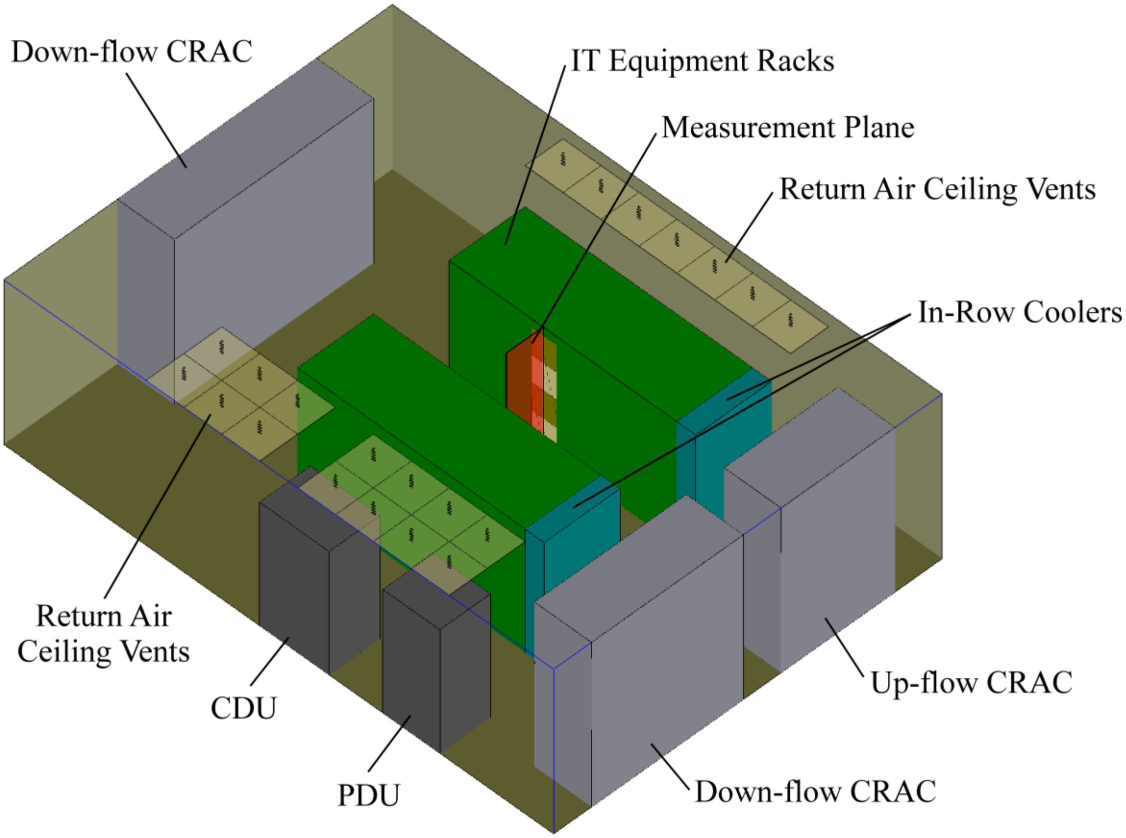


Figure 21. Isometric View of the Data Center Lab at Georgia Tech

The DCL contained one perforated tile, 12 IT equipment racks, three CRAC units, two in-row-coolers, one PDU, and one cooling distribution unit (CDU). The 12 racks were split evenly in front of two rows of tiles. Only one of the tiles in the cold aisle was perforated and it was located directly in front of the only rack that was functional during the tests. That rack housed four-10 U IT equipment simulators, and was only powered for flow since no thermal measurements were made. The two in-row-coolers were not powered on and neither were two of the CRAC units. Since only one perforated tile was open during the testing, bypass air was directed through the down-flow CRAC unit that was powered off. This allowed for reasonable air-flow rates through the lone perforated tile. The CRAC unit that was powered on incorporated a variable speed drive (VSD) which allowed the researchers to provide constant air-flow through the various perforated-tile designs [59].

The four IT equipment simulators were housed in a standard 42 U rack adjacent to the perforated tile. The same measurement techniques were used to determine the rack and tile flow rates as were used in [53]. The air-flow rates for the rack and perforated tile were matched using the simulators' speed control dials and the CRAC's VSD. Both were set to provide a measured flow rate of 0.56 m^3/s (1177 CFM).

Although the air-flow rates were held constant for the three different perforated-tile cases, the discharge velocities increased by over 30% due to the change in open area. The three perforated-tile cases studied here correspond to the test cases #1, 5, and 6 from [59] and are referred to as the full tile (case 1), the framed tile (case 2), and the blocked tile (case 3), respectively. The full tile (case 1) acted as the base tile for the modifications in [59], and consisted of a tile with 40% open tile porosity, and 6.35 mm (0.25 in.) pores on 9.53 mm (0.375 in.) pore-pore center spacing. The full tile was then "framed" with a 38.1 mm (1.5 in.) edge blockage around the perimeter of the tile to produce the framed tile (case 2). The third case evaluated here was the blocked tile, which corresponds to the full tile being fully blocked along one edge by a distance of 0.1524 m (6 in.).

The velocity field was measured using PIV in a plane across the centerline of the perforated tile starting at the cold aisle center, and ending at the IT equipment rack front face, and spanning the height of the rack. The plane is shown in figure 21. The PIV system was able to cover the width of the tile, but needed to be adjusted vertically in order to stitch together the entire plane. The average velocity field was created from 30 image pairs at each vertical location and the spatial resolution was determined to be approximately 7 mm. The standard error of the velocity measurements was quite small for the flow near the perforated tile and gradually increased as the distance from the perforated tile increased. This indicates that the fluctuations in the velocity measurements increased as the distance from the perforated tile increased, but they were still relatively small even at the highest elevations. Additional information on the experimental setup can be found in [59].

5.2 Numerical Models

Six CFD models were built using the commercially available finite-volume method solver, ANSYS Fluent [33]. All six of the models were constructed to match the physical layout of the DCL. Each of the models consisted of twelve major components: three CRAC units, one PDU, one CDU, two rows of IT equipment racks, two in-row coolers, one perforated tile in front of the IT equipment simulator rack, ceiling return air vents, and walls (including ceiling and floor). Only the space above the raised floor was modeled, eliminating the added complexity of the under floor air plenum.

The modeled inlet tile flow rate in front of the IT equipment rack was set to its measured value from the DCL as a fixed air-flow boundary conditions. The IT equipment server simulators were modeled as a black boxes with their measured flow rates. The hot-air return-vents were set as pressure outlet boundary conditions, and balanced the inlet air from the one perforated tile.

5.2.1 Flow Modeling

Only the inviscid and standard k-ε flow models were used in the DCL numerical models built for flow verification. As with the temperature verification models of chapter

4, the traditional CFD models used either the inviscid model, or the standard k-ε model. However, the CIVSM models used only the standard k-ε turbulence model for both the iterative scheme, and the turbulence model before dividing the domain. This is different from what was done for all of the previous implementations of the CIVSM which used the zero-equation model for dividing the domain. Further discussion of the reasons for this change can be found in section 5.3.1.

The turbulent intensity and length scale were set to 5% and 2.54 mm (0.1 in.) for all of the domain inlets, respectively, for the standard k-ε models. These values were lowered to 2.5% for the turbulent intensity and 61 mm (2.4 in.) at all of the interface boundary conditions for the viscous domain solve steps in the CIVSM iterative process. This was done to lower the amount of turbulence entering the viscous domain from the inviscid domain. Additional details on the selection of these boundary conditions can be found in section 4.2.1.

5.2.2 Tile Modeling

The same two tile modeling techniques described in section 4.2.2 were analyzed for this study: the porous jump model, and the modified body force model. A modified body force modeling technique was proposed by Arghode, et al. [55-57] in order to better model the air-flow through a perforated tile without having to resolve the fine scale details. This modeling technique produces similar results to the fine grid predictions, but allows for much coarser mesh than a detailed perforated-tile model. Further details of both tile modeling methods can be found in section 4.2.2.

The height of the momentum source for the modified body force model is dependent on the pore size, and a good estimate was determined to be 12 pore diameters [56,57]. However the previous implementations of the modified body force modeling technique all used perforated tiles with pore sizes at least twice the size of the pores in this study, and subsequently twice the momentum source height [55-57]. Given that the momentum source height is outside the range of previous implementations, some unexpected consequences may arise.

5.2.3 Grid Studies

Three grid studies were performed on the numerical models: one each for the CIVSM model, the inviscid CFD model, and the standard k-ε CFD model. The blocked tile (case 3), modified body force flow tile models were used for the grid studies since they produced the highest velocities and gradients. The same approach was taken for the flow verification models as was done for the temperature verification models (see section 4.2.3) with the same results.

Since the flow verification models only have one perforated tile, one modified body force source, and no cold aisle containment geometry, the base grid could be more evenly distributed across the DCL than for the temperature verification models described in chapter 4. The base grids ranged from 58,999 to 67,292 cells for the different tile configurations. Even though the grid counts for the flow verification models were about the same as the temperature verification models' grid counts, the maximum control volumes were significantly smaller. The flow verification models all had maximum control volumes of 5.05×10^{-3} m^3 for the inviscid domains and 6.06×10^{-4} m^3 for the viscous domains. These were an order of magnitude smaller than the maximum grid sizes used in the single-rack data center test cell. As with the temperature verification models, further grid refinements of the CIVSM did not alter the flow field predictions along the measurement plane for this case.

For the traditional CFD grid studies, the same base grids from the CIVSM models were used, and refined in the same manner, as well as a nearly uniform grid similar to the one used in [53]. The nearly uniform grid consisted of cells with a maximum linear dimension of 0.508 m and maximum control volume of 1.31×10^{-4} m^3.

Similar to the temperature verification models, the inviscid model would not converge for any of the four grids. However, after 1500 iterations the predicted flow field was nearly the same for all of the grids, and the residuals were near convergence and had reached their minimum values. Therefore the results are shown for the coarsest of the four grids which all were run for 1500 iterations.

As in the temperature verification models for the standard k-ε models, it was found that the two coarsest grids used for the CIVSM would not converge. Also the finest of the three grids used for the CIVSM did converge, but contained significantly more cells and increased solution time as compared to the nearly uniform grid. Thus, just as in the temperature verification models, the results are presented from the nearly uniform grid for the standard k-ε models using traditional CFD. Additional details on the grid study and the grid convergence index can be found in Appendix B.

5.3 Results and Discussion

All of the numerical models were run independently on the same IBM IntelliStation Z Pro [34] system with two dual-core Xeon 5160 CPUs (4 cores total) rated at 3.00 GHz with 16.0 GB of RAM. This is the same machine used for both the single-rack data center test cell cases and the temperature verification cases.

5.3.1 CIVSM Parameter Definition

Except for the initial turbulence model, the same parameters from the temperature verification models, which are delineated in section 4.4.1, were used for the flow verification models. The initial turbulence model was changed to match the final turbulence model (i.e. the standard k-ε turbulence model) due to instabilities in the first two iterations of the iterative portion of the CIVSM which would lead to divergence.

It was found that the zero-equation turbulence model did a poor job of initializing the solution for the CIVSM when using the standard k-ε turbulence model as the solver in the viscous domain. All attempts to reduce the instabilities by means of reducing the various under-relaxation factors only delayed the onset of divergence. The only method that prevented divergence was to initialize both the viscous and inviscid domains with the standard k-ε turbulence model's solution.

The easiest and most straightforward way to implement this change was to use the standard k-ε turbulence model as the solver for the second solve step for the overall CIVSM. This meant the standard k-ε turbulence model's solution, rather than use the zero-equation model's solution, was used as the basis for defining the viscous and inviscid domains using the partitioning algorithm. For all six cases there was no difference in domain definition when switching from the zero-equation model to the standard k-ε model.

The only drawback that accompanied this change was the increased solution time for the second solve step. The standard k-ε model required an additional 70 to 430 seconds of computation for the second solve step compared with the zero-equation model. However without the standard k-ε turbulence model used for the initial solution, the CIVSM diverged.

5.3.2 Partitioning Algorithm

For all of the cases the partitioning algorithm converged in either one or two iterations depending on the case. The maximum amount of time spent running the partitioning algorithm was less than six seconds.

The flow verification test cases all consisted of one viscous region that covered the inlet and exhaust planes of the IT equipment simulator and the perforated tile. As in the temperature verification test cases, the partitioning algorithm defined very similar viscous regions for the flow models. In these cases about 10% of the domain was included in the viscous region.

Figure 22 depicts the region locations, for all six numerical models, along the measurement plane, which is shown in figure 21. Each of the plots in figure 22 shows the center plane of the perforated tile directly in front of the IT equipment simulator. The origin of the plots are vertically located at top of the perforated tile, and horizontally located at the center of the cold aisle. The IT equipment simulator is on rightmost edge of all of the plots, up to 1.98 m (78 in.). Therefore the air enters the data center at a height of zero, and flows upwards and to the right towards the IT equipment simulator.

As can be seen in figure 22, the viscous regions were all slightly different for each perforated-tile case and flow model type. The partitioning algorithm created larger viscous regions for all of the modified body force (MBF) models than for the corresponding porous jump (PJ) models. This is to be expected since the modified body force models increase the vertical momentum which would be carried further downstream, thereby increasing the height above the perforated tile where viscous effects would be important.

5.3.3 Convergence Criteria

For the traditional CFD models, convergence was attained when the solution residuals dropped to values of less than 10^{-4} for the flow equations, or the solution was stopped after reaching 1500 iterations. These are the same values that were used in the temperature verification test cases (see section 4.4.3).

Three changes were made to the convergence criteria as compared with those used in the temperature verification test cases (see section 4.4.3). The first, and largest change, was to use the standard k-ε turbulence model for the second solve step solver. The second was that the number of iterations for the second solve was increased to 500 to ensure either convergence, or low level residuals prior to running the partitioning algorithm. Thirdly, only the middle of the three overall convergence criteria were used, i.e. 0.10 for the flow variables.

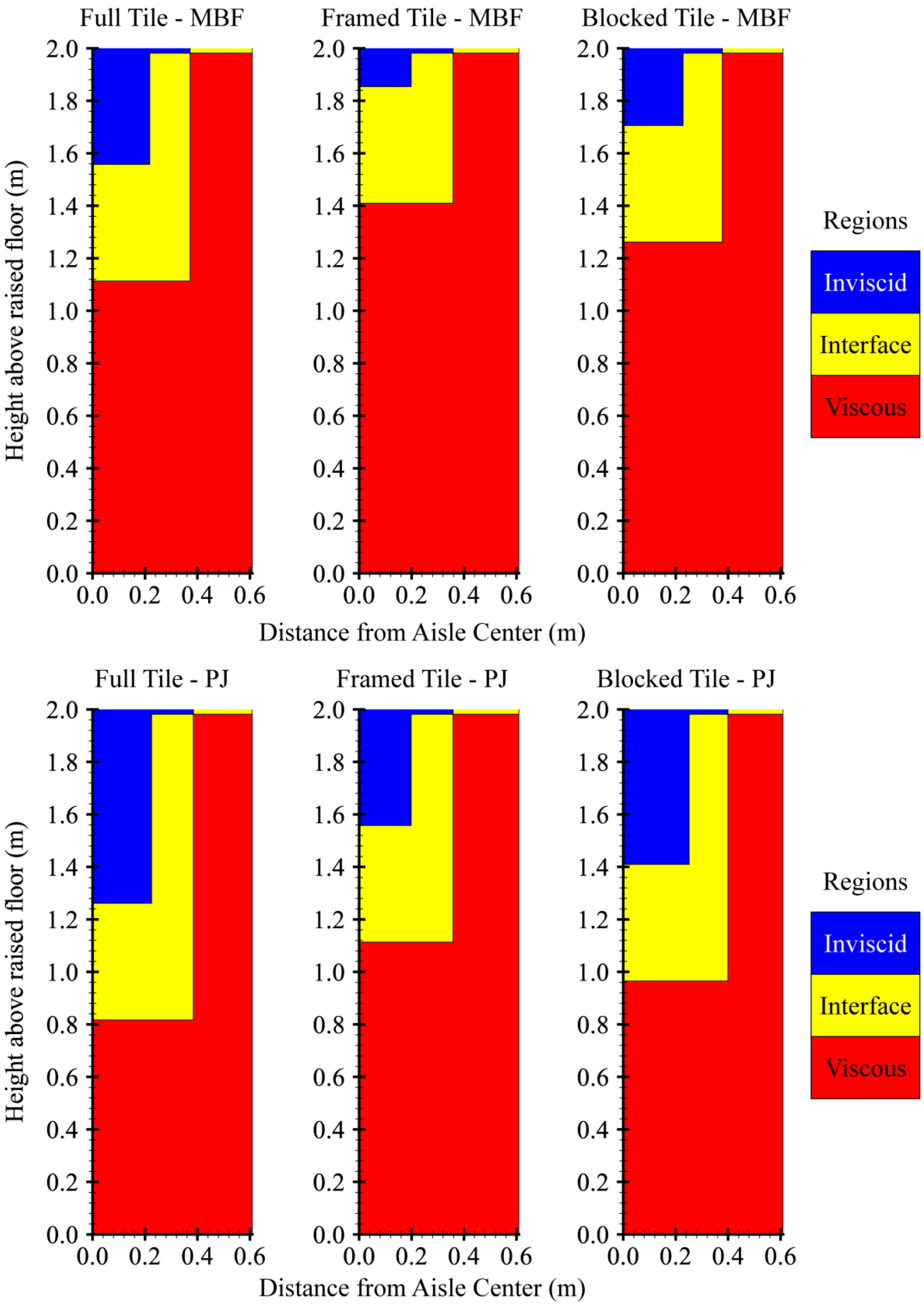


Figure 22. Region Locations for the Flow Verification Models

5.3.4 Flow Results

Figures 23-28 show the measured and calculated streamtrace plots for the three flow cases and the two perforated-tile flow models along the measurement plane. Streamtraces are defined as lines that are everywhere parallel to the velocity vector field. This is essentially the same definition as a streamline, but for these cases only two dimensional flow field data are available since the PIV measurements can only be taken within a plane. Therefore, the plots technically do not show streamlines, rather two-dimensional projections of the velocity field onto a plane. Thus, the lines will be referred to as streamtraces.

Each of the six figures shows the measured data plotted as streamtraces on the left, the inviscid model prediction second from the left, the CIVSM prediction second from the right, and the standard k-ε model on the right. The streamtraces are colored by the local velocity magnitude and the scale can be found on the far right of the figures. The CIVSM predictions also show the boundaries of the viscous, interface, and inviscid regions.

Figures 23 and 24 show the full tile cases for the modified body force, and the porous jump perforated-tile models, respectively. The measured air-flow velocities leaving the tile are somewhat evenly distributed and only about a third of the air-flow appears to enter the IT equipment simulator. The CIVSM models closely approximate the CFD counterparts, and all under-predict the amount of air leaving the left side of the plot, which is towards the other row of racks, away from the IT equipment simulator. The CIVSM model predictions appear to almost average the viscous and inviscid CFD model predictions for both perforated-tile flow models.

The modified body force perforated-tile models all severely over-predicted the vertical velocities. This may be due to the location of the measurement plane and that there is only a single perforated tile supplying the flow for the entire data center. The modified body force model enhances entrainment near the edges of the tiles, but this can

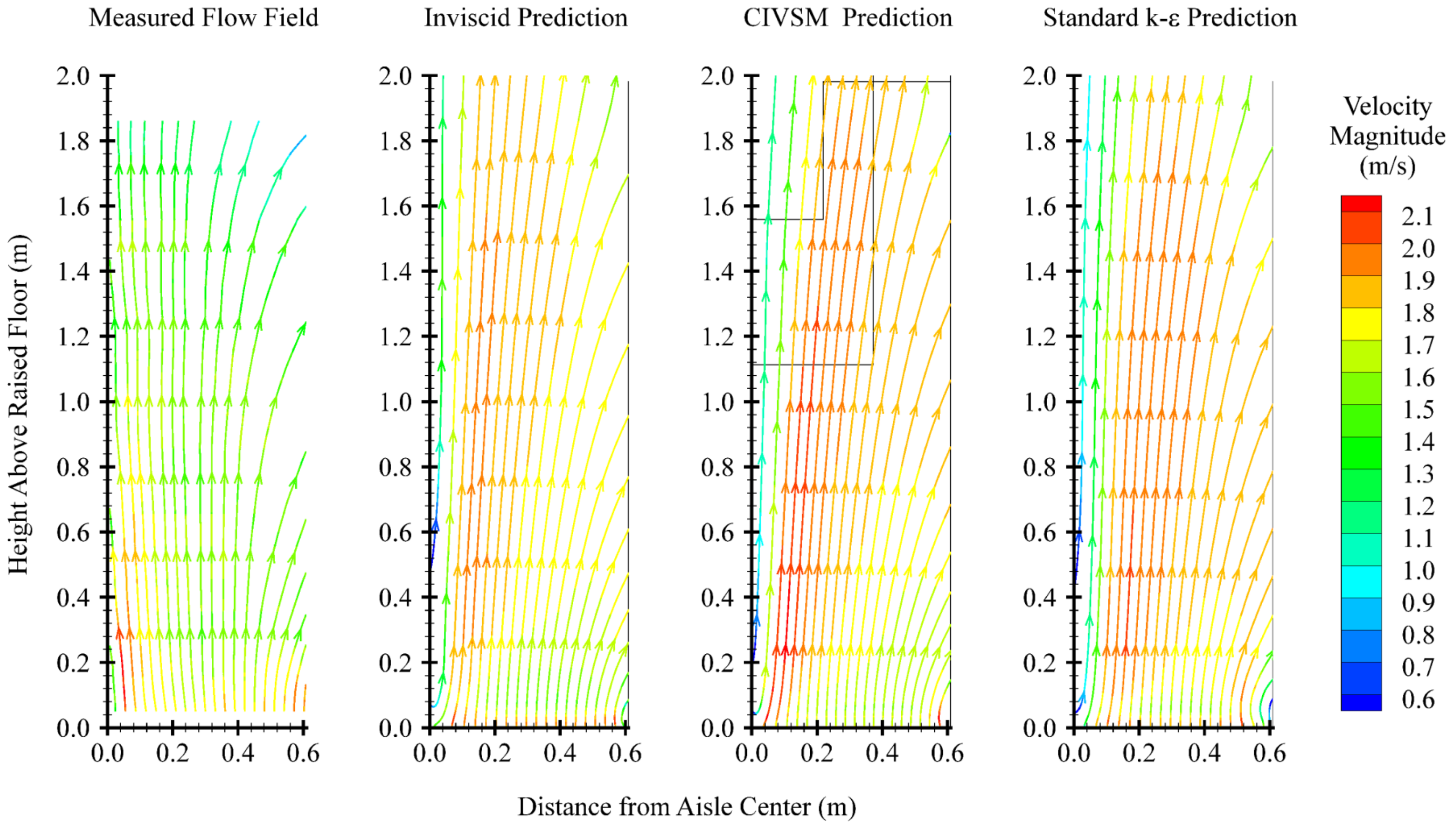


Figure 23. Measured and Predicted Streamtraces for the Full Tile (Case 1) Using the Modified Body Force Model

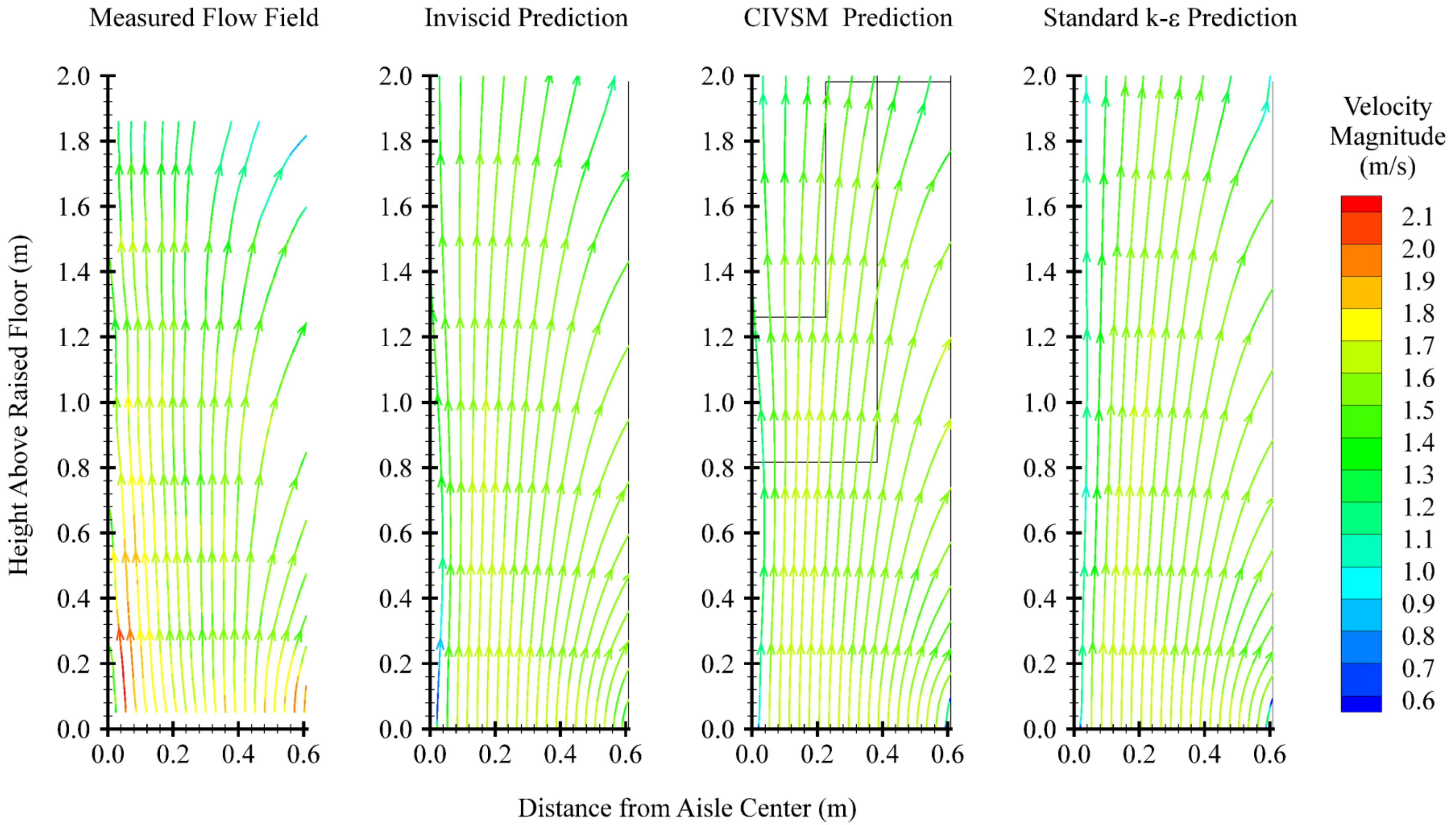


Figure 24. Measured and Predicted Streamtraces for the Full Tile (Case 1) Using the Porous Jump Model

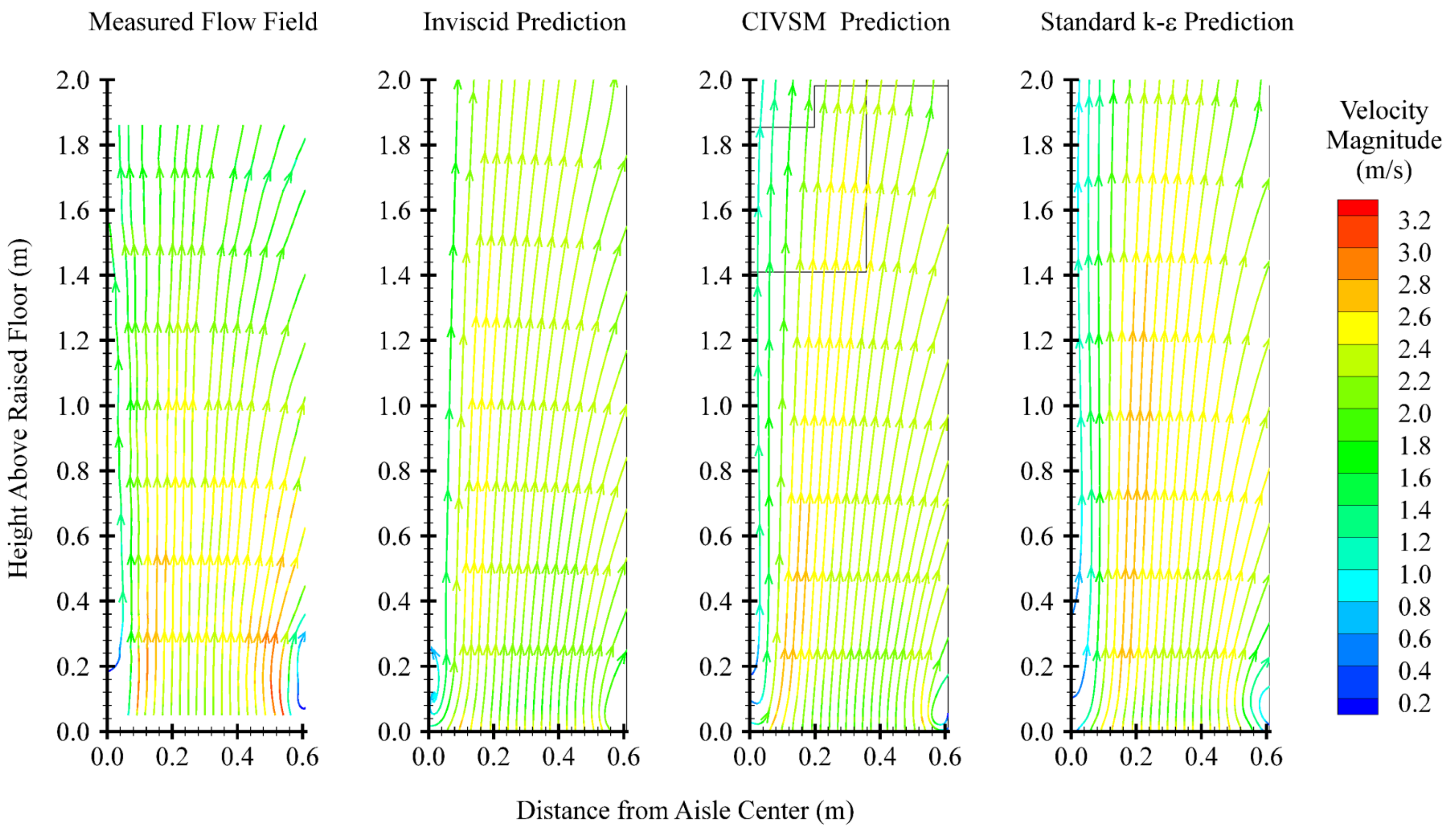


Figure 25. Measured and Predicted Streamtraces for the Framed Tile (Case 2) Using the Modified Body Force Model

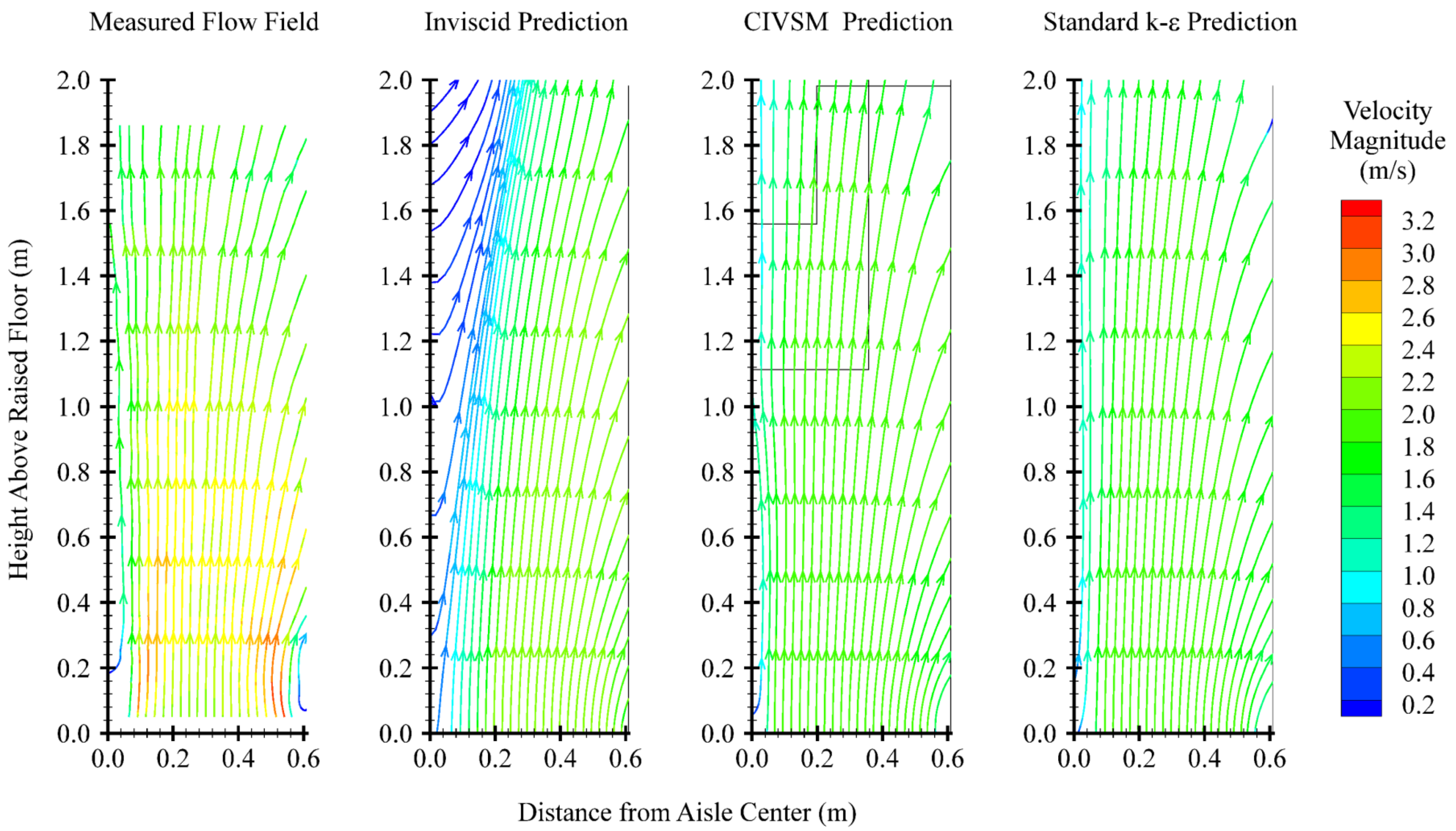


Figure 26. Measured and Predicted Streamtraces for the Framed Tile (Case 2) Using the Porous Jump Model

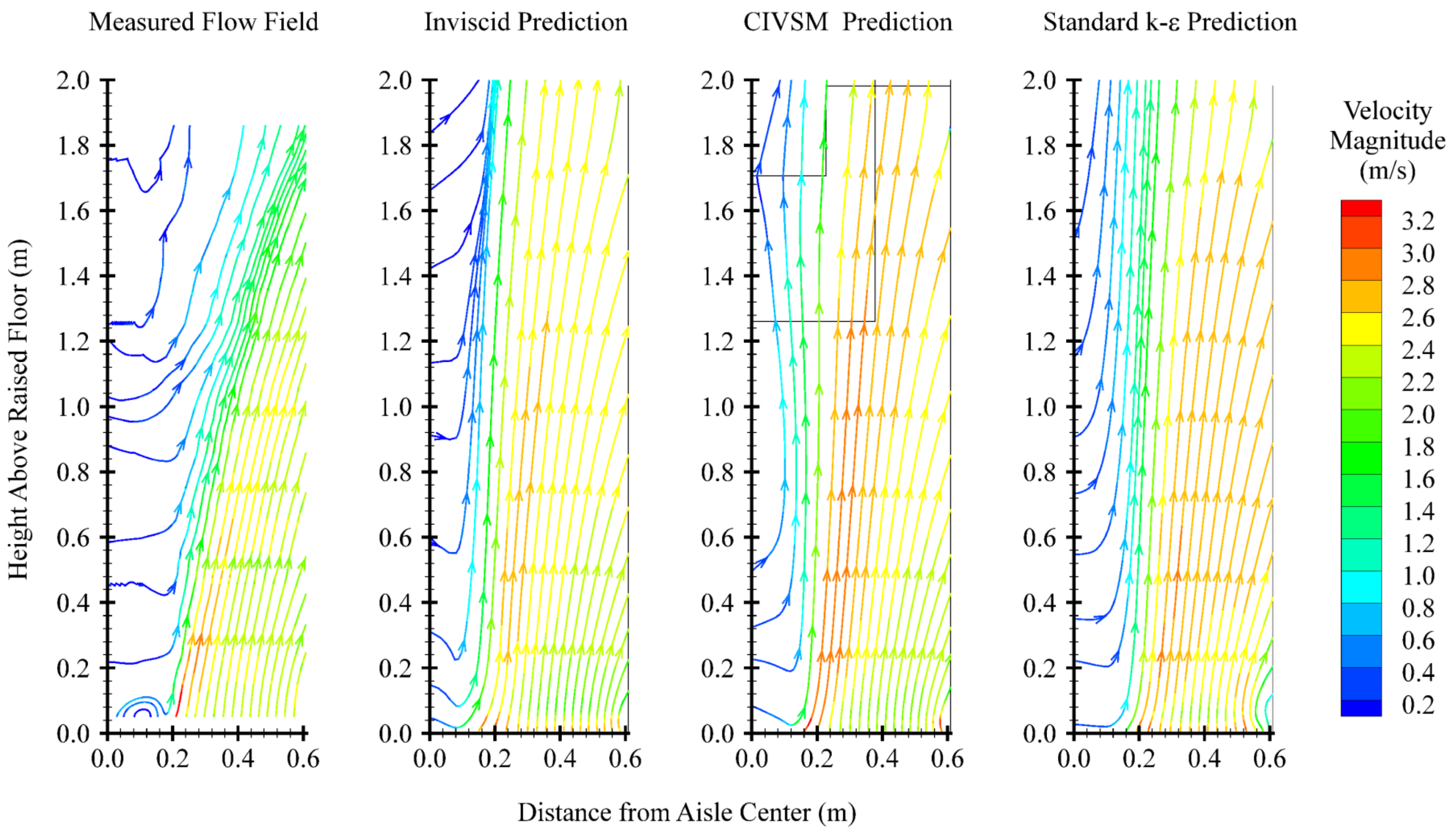


Figure 27. Measured and Predicted Streamtraces for the Blocked Tile (Case 3) Using the Modified Body Force Model

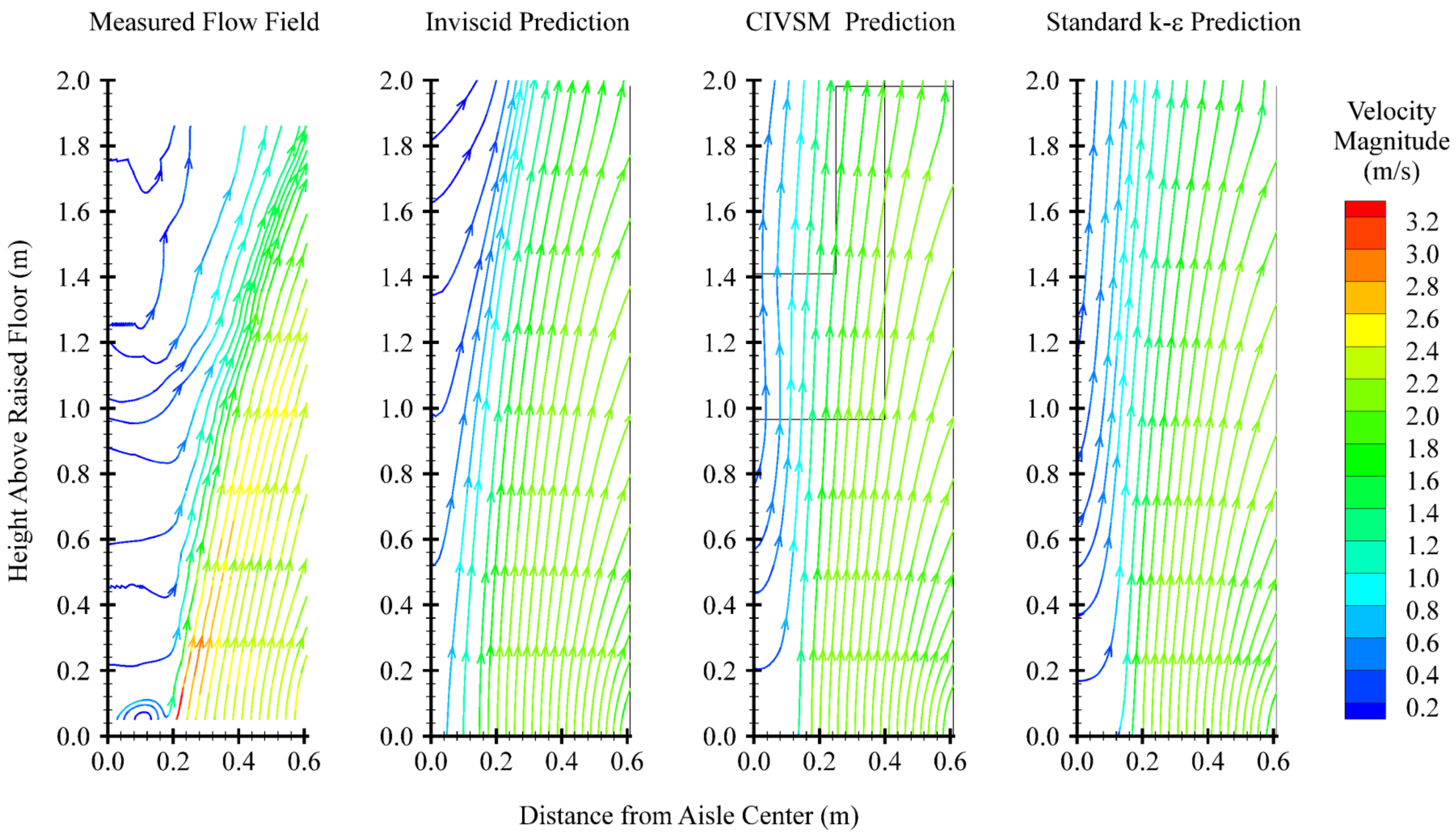


Figure 28. Measured and Predicted Streamtraces for the Blocked Tile (Case 3) Using the Porous Jump Model

also lead to artificially high momentum at the central portion of the tile. The high momentum levels could also be caused by the decreased momentum source height. This height is set by the perforated-tile pore size, which is half the size of the smallest pores from the previous studies [55-57].

The second case depicts the air-flow measurements and predictions from the framed tile, shown in figures 25 and 26 for the modified body force and porous jump models, respectively. In this case the perforated tile was framed such that the perforations 0.038 m (1.5 in.) around the periphery of the tile where blocked. Since the total air-flow rate was maintained from the previous case the decrease in open area meant the exit velocities were increased compared with the full tile case.

The modified body force models seem to do a much better job of predicting the velocity field than the porous jump models. Again the CIVSM seems to bridge the difference between the two CFD models. The CIVSM and standard k-ε models both show the same trends relative to the measured data by over-predicting the velocities at about 0.2 m from the center aisle, and under-predicting the velocities near the 0.5 m mark. The inviscid model using the porous jump tile model appears to severely under-predict the flow velocities toward the center of the cold aisle.

Figures 27 and 28 present the third case, where only 0.46 m (1.5 ft) of the tile was open and 0.15 m (0.5 ft) was blocked. This simulates the use of a smaller aisle width and further increases the exit velocity as compared with the framed case. It is interesting to note that a greater fraction of the cold air is entering the IT equipment simulator for this case than in the previous two cases.

None of the numerical models do a good job of predicting the air-flow patterns in this case. However, the CIVSM does do a good job of splitting the difference between the two CFD predictions. The CIVSM prediction using the modified body force model does appear to have a peculiar streamtrace near the center of the cold aisle. The streamtrace takes a turn at the boundary of the interface and inviscid regions. This is due in part to the low velocity in that region and also due to the solution being patched together from the viscous and inviscid domains at that location.

In order to better evaluate the differences between the predictions and the measured velocity data, the vertical and horizontal velocities at three heights were plotted against the distance from the center of the aisle. Figure 29 shows the measured and predicted vertical and horizontal velocities for the full tile cases using the modified body force tile flow model at a height of 0.15 m (0.5 ft) above the perforated tile. The three models do a relatively good job of predicting the vertical velocities, but over-predict the horizontal velocities. Figure 30 shows the same case and height as figure 29 (case 1 @ 0.15 m), but with the porous jump tile flow predictions. The porous jump model does not do as good of a job capturing the vertical velocities as the modified body force model, but also does not over-predict the horizontal velocities by as much either.

Figures 31 and 32 show the velocity plots for case 1 at the highest of the three heights, namely 1.68 m (5.5 ft), for the modified body force and porous jump models, respectively. Here it is clear to see that the modified body force perforated-tile flow model severely over-predicts the vertical velocities, while the porous jump model does a relatively good job. Both models do a very good job at predicting the horizontal velocities at this height, although the porous jump model predictions are almost all within the error bars of the measurements.

The velocity plots for the other two cases at all three heights, as well as the middle height for case 1 are displayed in Appendix C.

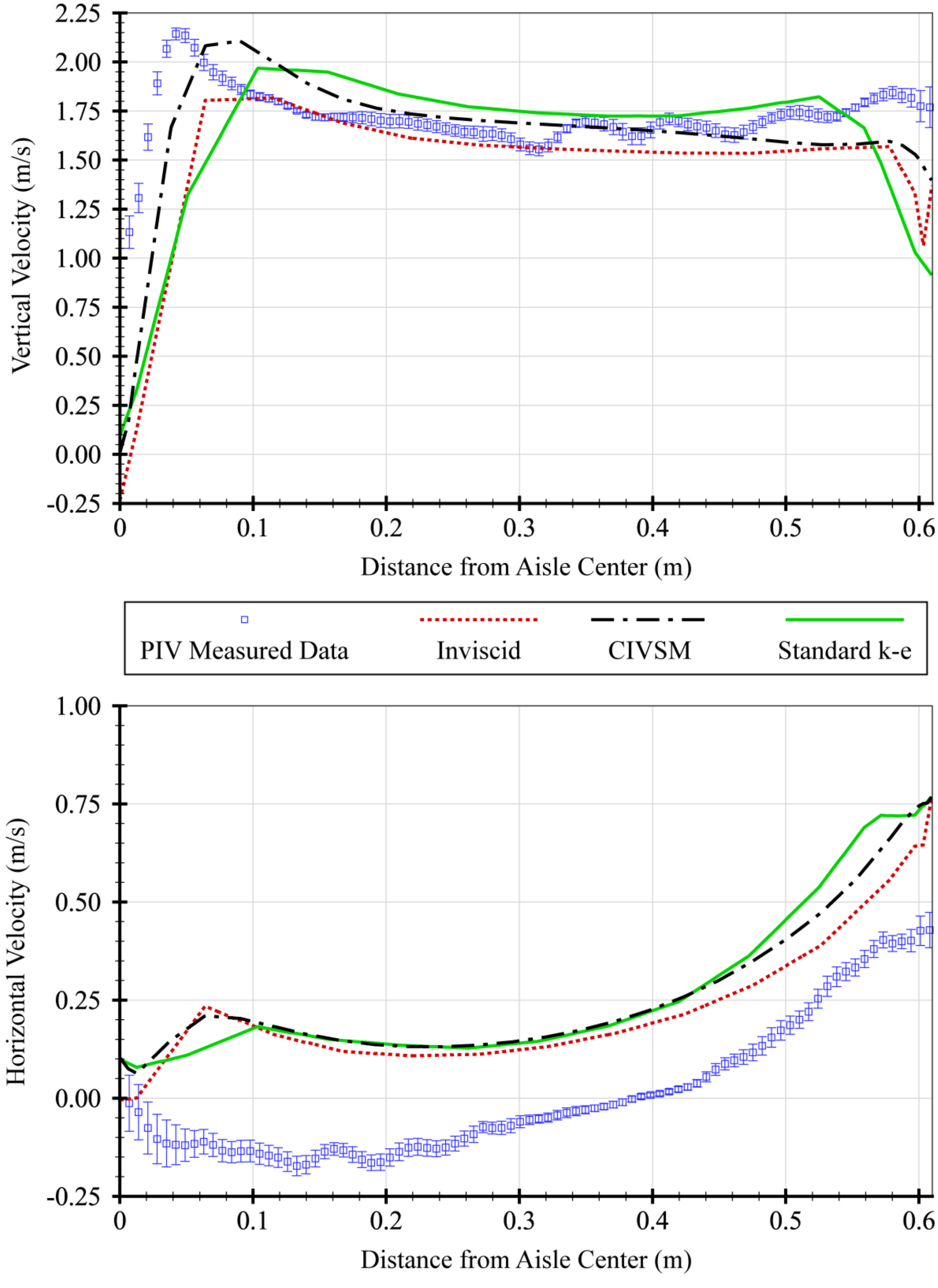


Figure 29. Velocity Plots for Case 1, Modified Body Force Model @ 0.15 m (0.5 ft)

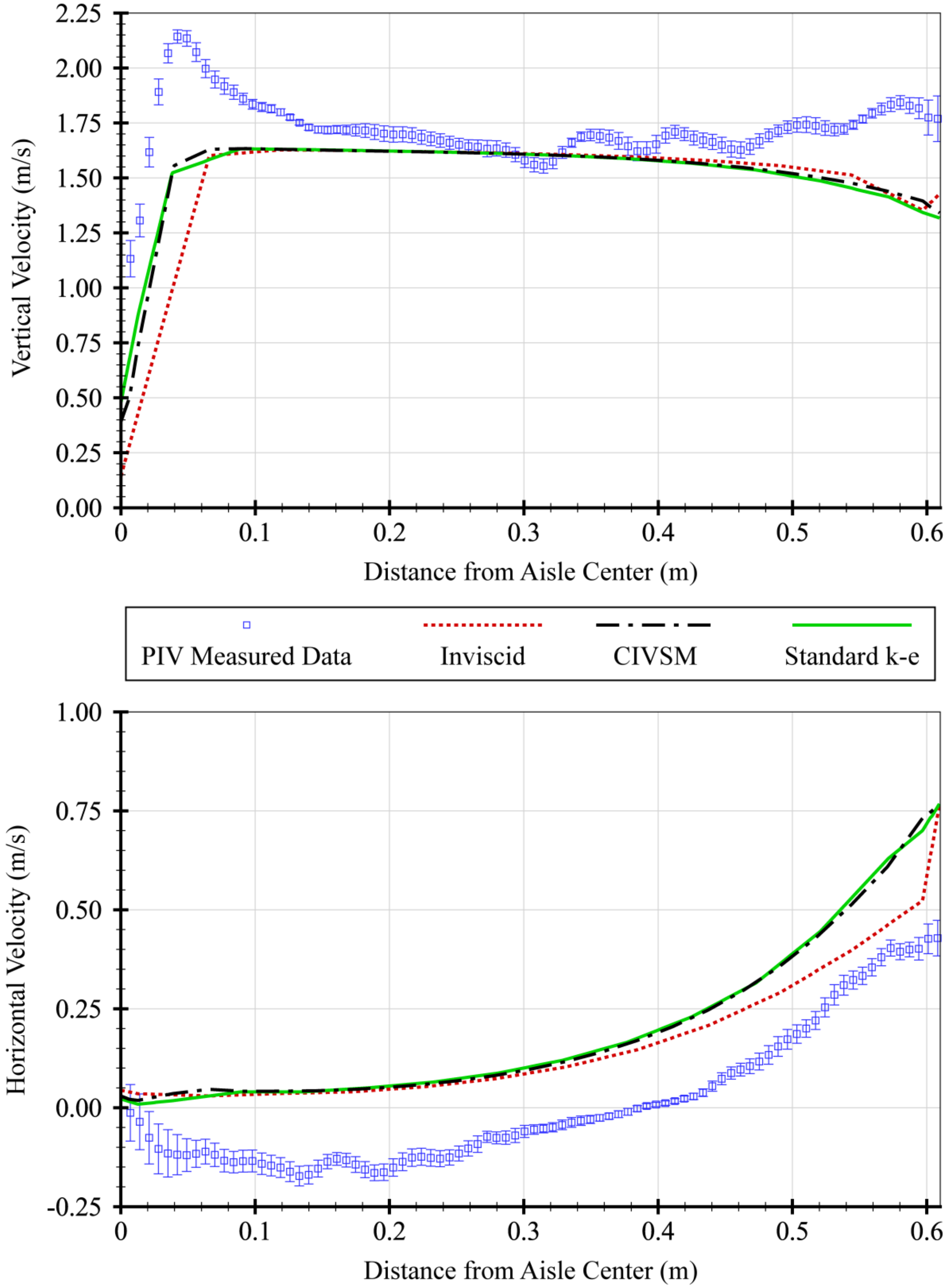


Figure 30. Velocity Plots for Case 1, Porous Jump Model @ 0.15 m (0.5 ft)

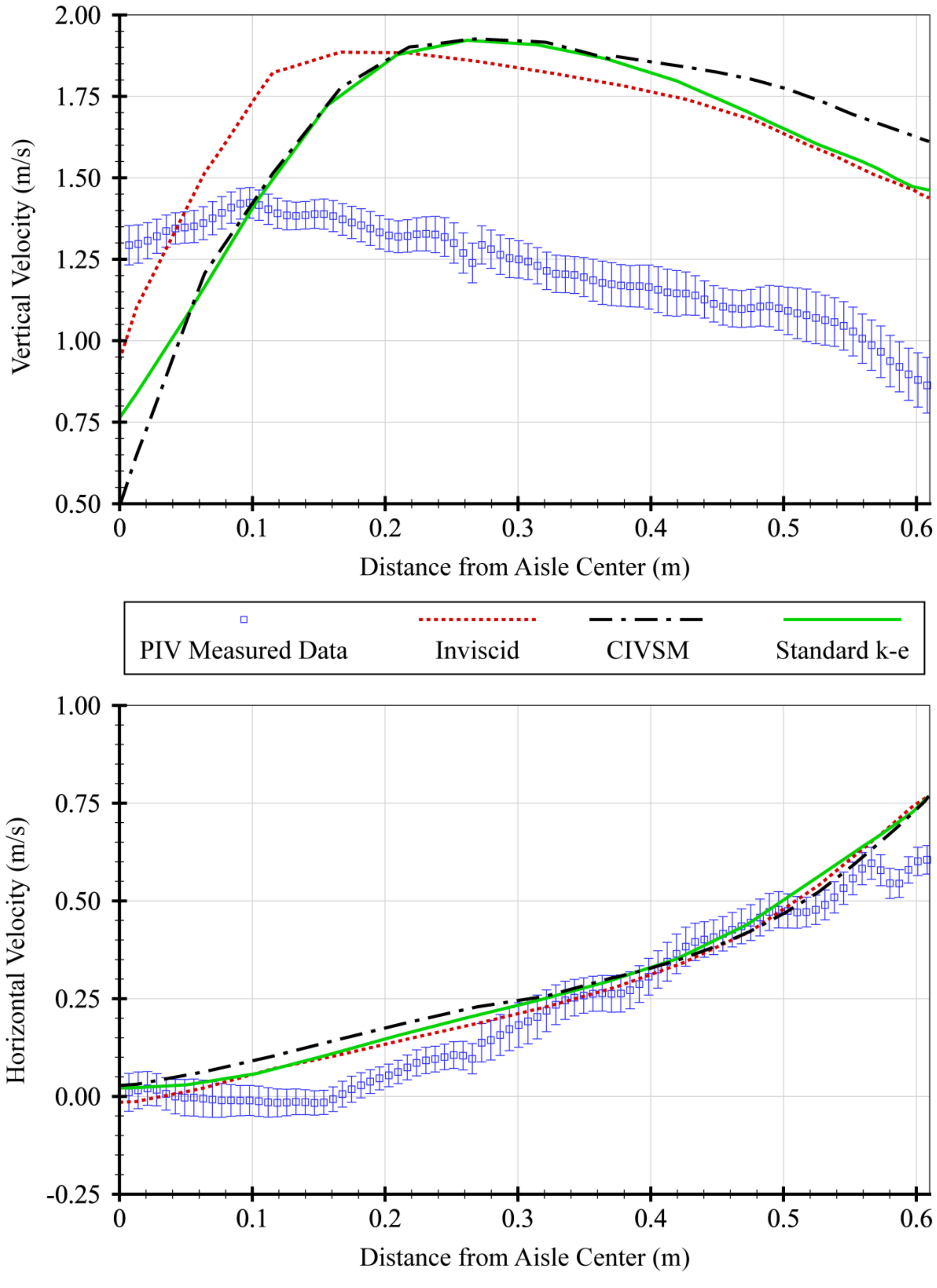


Figure 31. Velocity Plots for Case 1, Modified Body Force Model @ 1.68 m (5.5 ft)

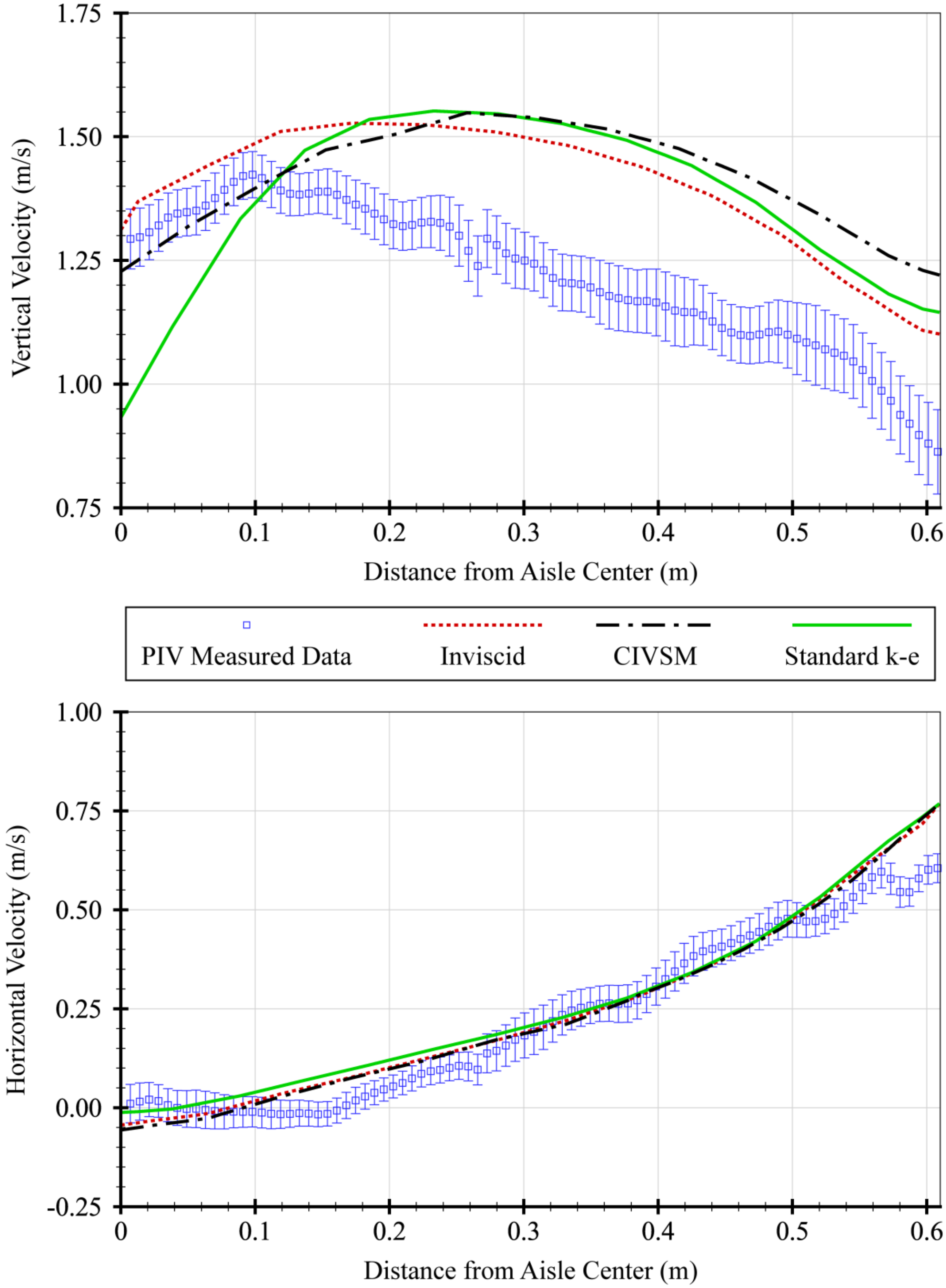


Figure 32. Velocity Plots for Case 1, Porous Jump Model @ 1.68 m (5.5 ft)

The lack of agreement between all of the numerical models and the measured data implies an error in the modeling techniques used. Modeling errors are most likely due to the boundary condition treatments. The differences between the two tile modeling techniques can be easily seen in figures 23-32. The modified body force model appears to do a much better job of predicting the vertical velocities near the tile than the porous jump model, but then it severely over-predicts the vertical velocities at higher elevations. The uniform rack inlet velocity boundary condition may also play a part in the discrepancies between measured and modeled results.

For all of the cases and perforated-tile flow models, the CIVSM produced similar results to traditional CFD using the standard k-ε turbulence model. In many of the areas where the CIVSM deviated from the standard k-ε turbulence model, it produced similar results to the inviscid model. The good overall agreement between the numerical models implies that if the modeling errors are corrected, then the CIVSM should be able to reproduce the results of the standard k-ε turbulence model, but at much lower computation expense.

5.3.5 Solution Time

Figure 33 shows the solution time for each of the flow cases and perforated-tile flow models. The inviscid models all ran the fastest, followed by the CIVSM, and then the standard k-ε turbulence model. On average the CIVSM models ran an order of magnitude faster than the standard k-ε turbulence models, and the inviscid ran 1.7 to 7.6 times as fast as the CIVSM models. Even when running the standard k-ε turbulence model for the second solve step, the CIVSM still provided significant reductions in solution time.

5.4 Summary

The CIVSM produced similar flow predictions to the standard k-ε turbulence model while reducing solution time by about an order of magnitude. However, convergence issues necessitated the use of the standard k-ε turbulence model for the second solve step, which increased the CIVSM solution times. It also brings into question using the zero-equation turbulence model for the second solve step on future implementations that use the standard k-ε turbulence model for the viscous domain solver.

Although the CIVSM has the potential to significantly reduce computational effort, it does have a few weaknesses, and potential problems that need to be overcome to be competitive with traditional CFD/HT.

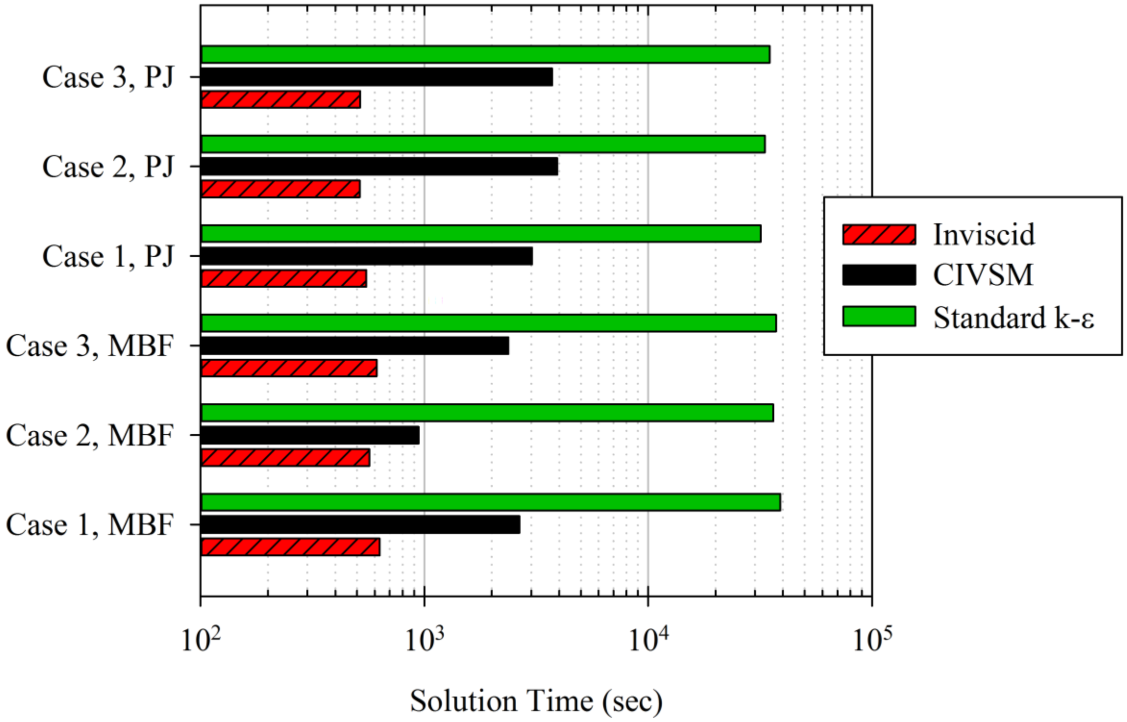


Figure 33. Solution Time for the Flow Verification Test Cases

CHAPTER 6

FUTURE WORK, RECOMMENDATIONS, AND CONCLUSIONS

A novel CIVSM was developed, and has proven to provide significant time savings compared with traditional CFD/HT methods while not sacrificing accuracy. However, there are still a number of known issues and potential sources of problems that can arise when using the CIVSM. There are also a number of parameters and techniques that have yet to be fully explored when implementing the CIVSM technique. These may lead to further decreases in computational time or increased accuracy of the models.

6.1 CIVSM Partitioning Algorithm

The CIVSM's partitioning algorithm is key to the success of the method. If the defined viscous regions are too small, then the CIVSM will not accurately model the domain. Conversely, if the defined viscous regions are too large, then the CIVSM solution time will increase to the point where it is no longer advantageous as compared with a traditional, fully viscous CFD/HT model.

Even though it is easily quantified and understood, the overall size of the viscous regions is not the only aspect which determines the partitioning algorithm's effectiveness. Other factors, both within the partitioning algorithm and external to it, have been identified as having a significant influence over the definition of the viscous regions, and subsequently the solution time and accuracy of the CIVSM. These factors are described in detail in the following sections.

6.1.1 Non-Uniform Grid

One of the most difficult things to do when building and running CFD/HT models is to appropriately grid the domain. Too fine of a grid leads to increased compute time, but too coarse of a grid can lead to inaccurate and misleading results. One tool that has been used extensively to reduce the grid size is to employ a non-uniform grid. Certain "rules-of-thumb" have been developed over time for determining what is acceptable and what will produce poor results for traditional CFD/HT. These generalized rules have not been fully developed for the CIVSM modeling technique, although a few issues have been noted.

When growing the viscous regions, the CIVSM's partitioning algorithm sweeps across the faces of every cell in the domain and examines the χ-value of the adjacent cells. When evaluating the χ-value of an adjacent cell, the partitioning algorithm does not take into consideration the face area, or any other parameter of the neighboring cells. The adjacent cell may be much larger, or smaller than the cell in which the evaluations are being conducted, and this may be an important factor that is being overlooked by the current implementation of the partitioning algorithm. The partitioning algorithm has shown a tendency to place viscous region boundaries at the interfaces of non-uniform grid cells.

The way in which the interface regions are defined can also lead to problems due to non-uniform grid. When bounding a fine grid, simply defining a fixed number-of-cells to include in the interface region may not allow for an effective interface region. This is especially true when there is a limited amount of grid in the flow direction. The partitioning algorithm also allows for a distance variable to be set for the definition of the interface region, but all implementations of the CIVSM described in this dissertation have set the distance parameter to zero. This was done in order to avoid irregular viscous boundaries in the form of stair-casing. Stair-casing is when the boundaries of a domain take the shape of a stair case, which leads to non-physical grid effects.

6.1.2 Viscous Region Smoothing

One problem that was discovered early-on in the development of the CIVSM was the need for relatively smooth viscous region boundaries. Irregularly shaped viscous regions can lead to boundary condition applications that produce non-physical grid effects. This can lead to extremely slow convergence, or even divergence of the overall solution.

These non-smooth boundaries are naturally created by the partitioning algorithm during the formation of the viscous and interface regions. The viscous regions grow outwardly from the seed cells in a “spherical” manner, resulting in stair-cased viscous regions. Similarly, when the interface regions are defined by a certain distance from the viscous regions, stair-cased boundaries are defined at the corners and along the edges of the viscous regions. The stair-casing can be mitigated by only using the distance in number-of-cells from a boundary, since this parameter only forms interface regions normal to the viscous boundaries, and does not create irregularly shaped boundaries on its own.

A smoothing function is currently incorporated in the algorithm, and is an integral part of the growth of the viscous region (see table 1). However this function always grows the viscous region outwardly from the seed cells, and never trims the extraneous cells that sometimes end up on the face of an otherwise smooth boundary. An additional smoothing function has been formulated to eliminate these irregularities, but it tends to create extremely large viscous regions, which can cause the partitioning algorithm to diverge. A trimming function would help to ensure convergence of the partitioning algorithm when the smoothing functions fail to contain the viscous region growth.

Advanced smoothing techniques would lead to better viscous region definition, and eliminate a source of oscillations within the CIVSM. Smaller viscous regions are desired for reduced computational effort, but too small of a region could produce poor predictions. A fine balance must be made to create appropriately sized viscous regions that are not irregularly shaped, which can cause convergence issues.

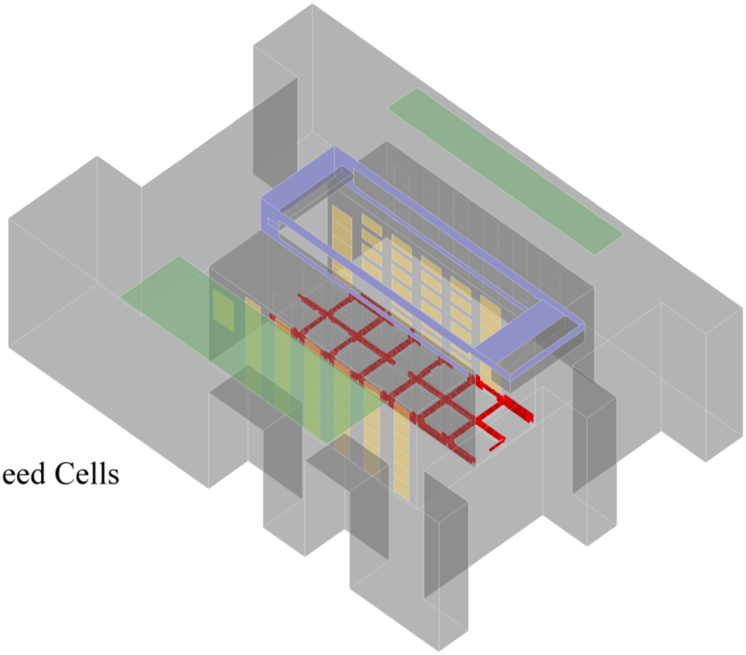
6.1.3 Partitioning Algorithm Parameters

Vorticity magnitude was chosen as the default parameter used in the partitioning algorithm to determine the viscous regions. Other parameters could also be used, but have not been extensively researched. Three new parameters were explored for the one CRAC case, using the modified body force perforated-tile flow modeling technique from the DCL at Georgia Tech. The three parameters analyzed were turbulent kinetic energy, turbulent dissipation, and turbulent viscosity.

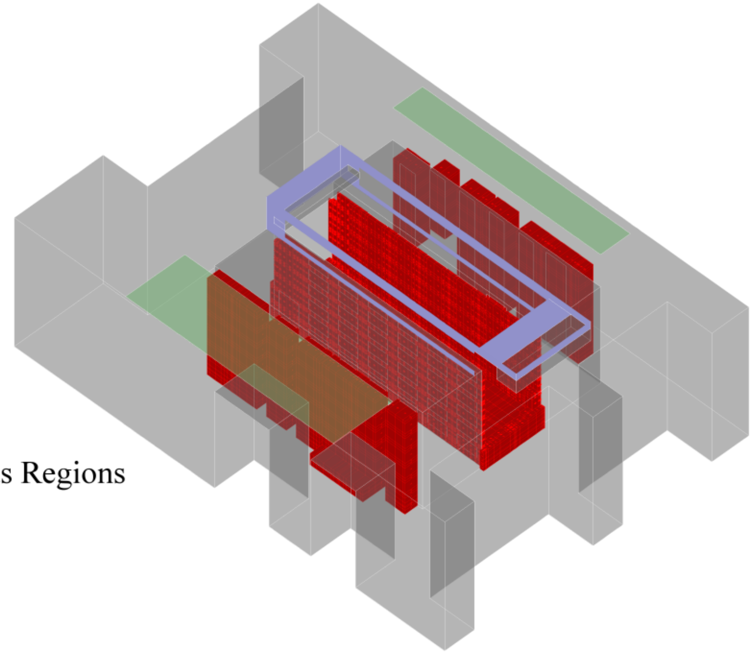
The first two parameters require the use of a k-ε turbulence model (e.g. standard, RNG) in order to calculate the parameters for the domain. This can be a drawback since it requires the use of a more complicated model than the zero-equation turbulence model, which ultimately results in increased solution times for the second solve step of the CIVSM. Although, as seen in the flow verification cases, using the same turbulence model for the second solve step and the iterative portion of the CIVSM may be required for overall convergence.

Figures 34-37 show the seed cells and final viscous regions defined by the partitioning algorithm for vorticity magnitude, turbulent kinetic energy, turbulent dissipation, and turbulent viscosity, respectively. The first three seed cell locations and viscous regions look quite similar. However, the turbulent viscosity defined regions are very different from the others.

Even though the seed cells for the first three figures are nearly the same, the other factors in the partitioning algorithm managed to create slightly different final viscous regions. What aren’t shown in these figures are the initial χ-values that were used to create the viscous regions. The location of the cells with χ-values greater than zero affect the growth of the viscous regions.

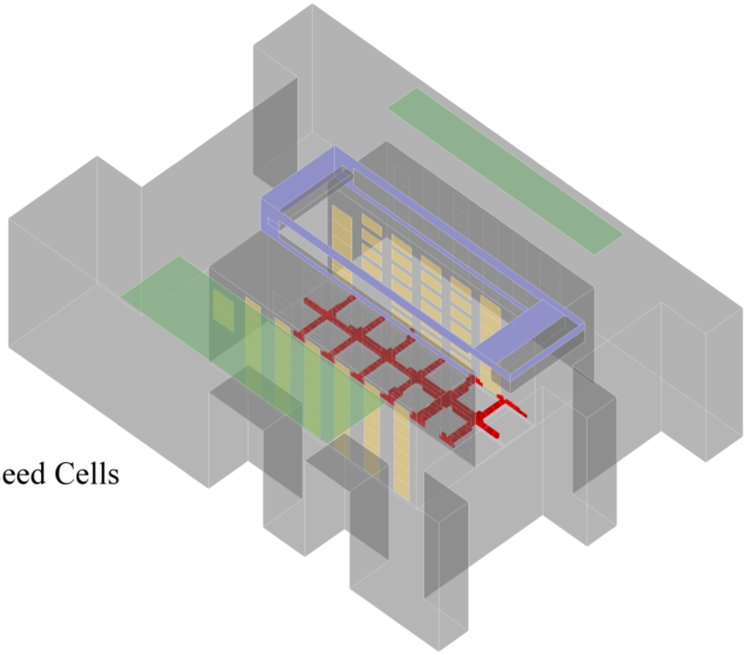


A. Seed Cells



B. Viscous Regions

Figure 34. Seed Cell and Viscous Region Locations for 1 CRAC, Modified Body Force Model when Using Vorticity Magnitude


A. Seed Cells

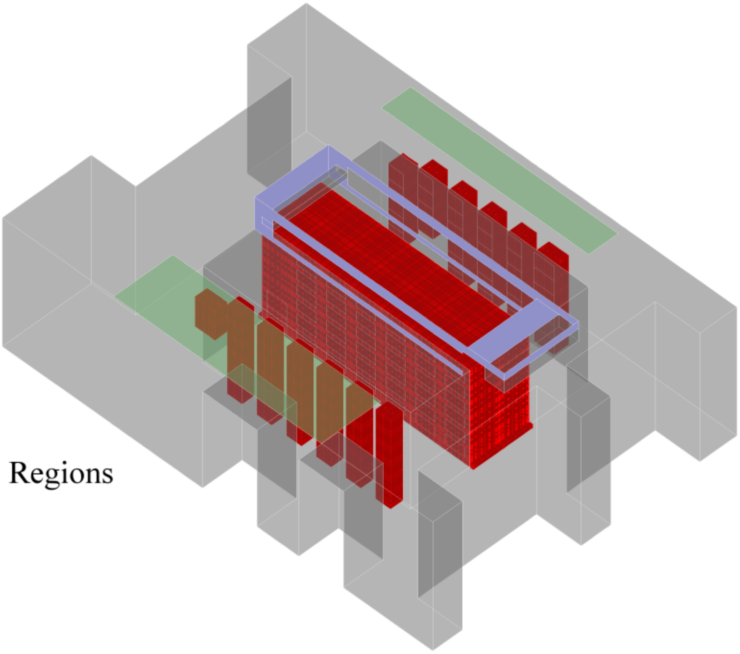
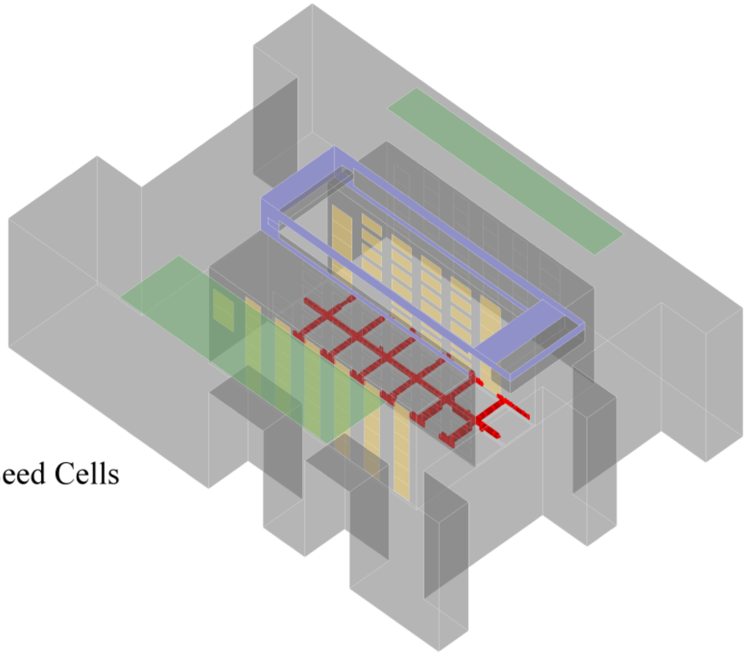
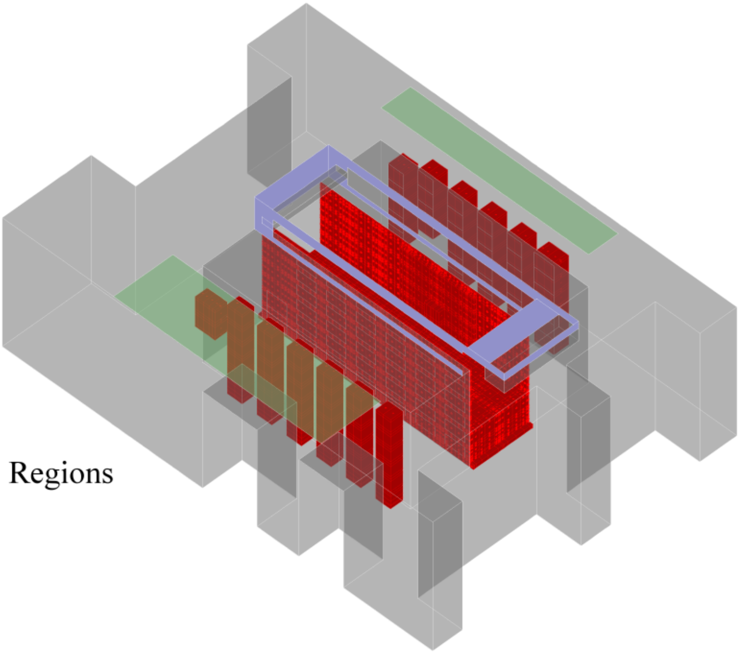

B. Viscous Regions

Figure 35. Seed Cell and Viscous Region Locations for 1 CRAC, Modified Body Force Model when Using Turbulent Kinetic Energy

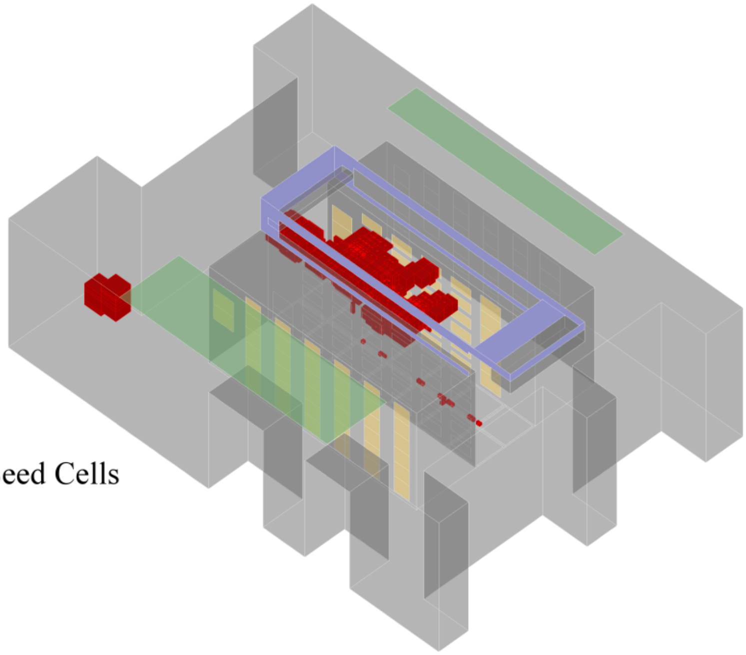


A. Seed Cells

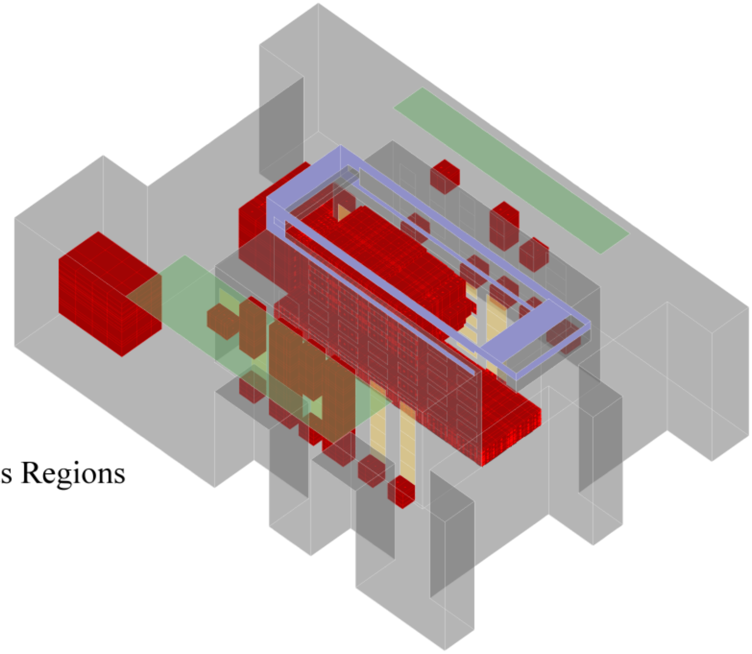


B. Viscous Regions

Figure 36. Seed Cell and Viscous Region Locations for 1 CRAC, Modified Body Force Model when Using Turbulent Dissipation



A. Seed Cells



B. Viscous Regions

Figure 37. Seed Cell and Viscous Region Locations for 1 CRAC, Modified Body Force Model when Using Turbulent Viscosity

The turbulent kinetic energy defined viscous region (figure 35) was the largest of the first three final viscous regions shown. It included the entire cold-aisle within the viscous region. This is not surprising since high turbulent kinetic energy indicates the location of high inertial regions within the domain rather than highly viscous regions.

Using the turbulent dissipation as a criterion, the partitioning algorithm defined a viscous region nearly identical to that of the turbulent kinetic energy viscous region (figure 36), except that the upper two-thirds of the center of the cold aisle was not included. The vorticity magnitude defined region (figure 34) was even smaller in the center of the cold-aisle than the turbulent dissipation defined region, but contained more grid on the exhaust of the IT equipment.

Both the turbulent kinetic energy and turbulent dissipation defined partitions used the standard k-ε turbulence model for the second solve step, while the vorticity magnitude used the zero-equation model. The viscous regions defined by the partitioning algorithm using vorticity magnitude were minimally different when using the standard k-ε turbulence model instead of the zero-equation model.

The viscous regions defined by the partitioning algorithm using turbulent viscosity as the criterion (figure 37) were completely different than those defined by the other parameters. The partitioning algorithm using turbulent viscosity defined one viscous region in the middle of one corner of the data center, and another at the end of the cold aisle near the solitary CRAC unit. Although the flow may be mixing in these regions, modeling these zones as a viscous regions would likely not alter the predictions of the inlet air temperatures of the IT equipment and would significantly increase the solution time. Also there is a large gap between a viscous region at the perforated floor tiles and a viscous region in the center of the cold aisle. Although this gap is large enough to prevent interface region overlap, it would likely increase the solution time compared to a viscous region that included the entire cold aisle due to the additional boundary conditions and non-solving operations that would be required. Like the turbulent kinetic energy and turbulent dissipation defined partitions, the turbulent viscosity defined partitions used the standard k-ε turbulence model for the second solve step.

Overall, the viscous regions defined by the partitioning algorithm using turbulent viscosity as the criterion (figure 37) are not located where one would expect viscous effects to be important. This suggests that unlike vorticity magnitude and turbulent dissipation, the turbulent viscosity is a poor criterion for determining the viscous regions. The turbulent viscosity is defined as:

$$\mu_t = \rho C_\mu \frac{k^2}{\varepsilon} \tag{21}$$

where μ_t is the turbulent viscosity, C_μ is the empirical constant defined by the turbulence model (0.9 for the standard k-ε model), *k* is the turbulent kinetic energy, and ε is the turbulent dissipation. This confirms the notion that the use of turbulent viscosity as the criterion for the partitioning algorithm is a poor choice, since viscous effects will be highest where the turbulent dissipation is high.

The CIVSM was run using the same parameters as in section 4.4.1 except that the standard k-ε turbulence model was used for the second solve step and the partitioning algorithm used turbulent kinetic energy, or turbulent dissipation for the viscous region definitions. The results from these test cases are shown in figure 38 along with the previous results for the one CRAC case from chapter 4.

Of the CIVSM models, the original model using vorticity magnitude produced the best results in terms of both RMS difference, and solution time. The two new partitions took longer to converge, but also included more of the domain in the viscous region. Both produced similar RMS difference results to the traditional CFD/HT standard k-ε model, while solving in less than a quarter of the time.

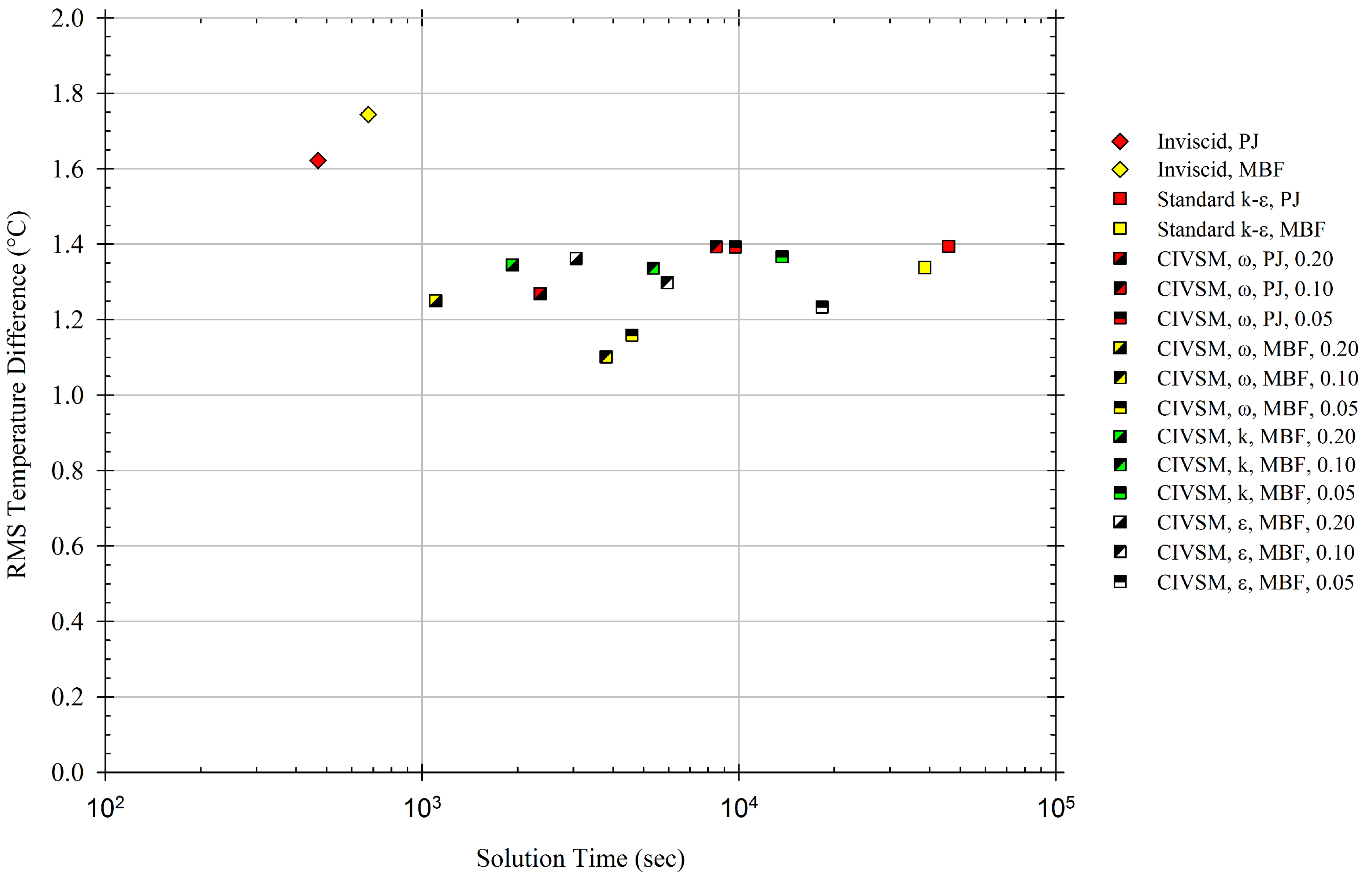


Figure 38. RMS Temperature Difference vs. Solution Time for the 1 CRAC Case

Based on these limited results, either turbulent kinetic energy, or turbulent dissipation may offer a good alternative to vorticity magnitude as the partitioning algorithm criterion. However, vorticity magnitude can be easily calculated from any viscous flow solution, and is not reliant on a specific turbulence model. Further research into the optimization of the different parameters, and verification of the effectiveness of the criteria is needed for a thorough comparison.

6.1.4 Relatively Small Viscous Regions

The partitioning algorithm can form relatively small viscous regions in zones where there are only a few cells with high vorticity in the vicinity. In some cases, these may not be important to the overall solution, and could probably be neglected without affecting the areas of interest in the domain. Currently, there is no mechanism to find and eliminate these small viscous regions.

A good example of this is in the single-rack data center test cell, where the air flow exiting the bypass tiles is included in a small viscous region. This region is located on the other side of the data center from the area of interest and is much smaller than the other two viscous regions. It may not add any benefit to the overall solution, but requires a significant amount of computational effort.

Without user interaction, defining what constitutes a negligible viscous region is an extremely difficult task. One possibility that has been considered for implementation is to define the areas of interest (i.e. in the case of data centers, IT equipment inlets) prior to running the first solve step of the CIVSM. Then all viscous regions that fall outside of those areas would be eliminated if their contiguous regions were less than a predefined percentage of the overall viscous domain. This would help eliminate any small extraneous viscous regions that may not add to the accuracy of modeling the areas of interest.

6.2 Mass Balance Algorithm

The mass balance algorithm does an excellent job of minimizing the changes to the boundary conditions while ensuring continuity. However, it does have two major inherent problems. The first, which has already been described, is the interaction with the SIMPLE algorithm's pressure-velocity coupling. This could pose a significant challenge for large models that approach the limits of the computer's memory since the coupled pressure-velocity method requires roughly twice memory as compared with the SIMPLE method.

The second issue with the mass balance algorithm is the potential for interactions with other flow boundary conditions. The mass balance algorithm can induce undesired fluctuations in the solver when coupled with a boundary condition that does not specify the mass flux. For example, a pressure-outlet boundary will satisfy the continuity equation by changing the mass flux through the domain. If a pressure-outlet boundary is coupled with the mass balance algorithm, then both boundary conditions simultaneously, and independently attempt to satisfy continuity. This could lead to divergence of the solution.

This potential problem has been mitigated by stopping the mass balance algorithm when it encounters a net mass flux increase between successive iterations. This typically occurs only for the first few outer iterations of the CIVSM, and has not led to any noticeable degradation in overall performance.

However, it is worth noting that the mass balance algorithm was developed only for cases in which the mass fluxes were specified at all of the boundaries. Therefore this potential problem is easily addressed by simply turning off the mass balance algorithm when there is at least one boundary condition where the mass fluxes are not fixed. In these cases, the non-fixed boundary condition will automatically balance the mass flow for the domain.

6.3 Under-Relaxation Parameters

Under-relaxation of the boundary conditions has helped speed convergence, but has only been statically implemented. Large fluctuations in boundary conditions for the first few iterations necessitate small under-relaxation values. However, later iterations may benefit from higher under-relaxation values and could decrease the number of outer iterations. An optimum set of under-relaxation values that slowly increase with the number of outer iterations could significantly reduce convergence times. This could be implemented in a similar manner to the way the convergence criteria are updated after the first few iterations.

6.4 Non-Solving Operations

In a few of the cases it was found that all of the non-solving operations of the CIVSM took just as much time as the solving operations. There can be a lot of overhead with the CIVSM, especially for problems that solve relatively quickly and have many viscous regions. The two of the biggest time consumers are loading and saving the individual viscous and inviscid models. If the models are not so large as to create memory problems, then there is the possibility of keeping both models continuously open to save the time of closing and loading the individual models. This would also allow for solving one domain while simultaneously saving the other's solution, thereby parallelizing a few of the operations and further reducing the overall solution time.

6.5 Conclusions

Inviscid solution methods have proven to greatly reduce solution time relative to traditional CFD/HT for bounded domains. However, inviscid methods are not capable of capturing the effects of mixing hot and cold air streams which is vital to accurately predicting the inlet air temperatures for IT equipment in data centers. While certainly more accurate than inviscid solution methods, traditional CFD/HT models are computationally expensive and can severely limit the ability to optimize a data center layout. There is a need for a modeling technique that provides more accurate results than the inviscid methods and faster solutions than traditional CFD/HT.

A novel approach to predicting the flow and temperature fields of bounded domains, such as data centers, was presented. The technique couples an inviscid and a viscous solver to create a solution that uses less computational effort than a traditional CFD/HT solver, and produces more accurate results than an inviscid solver. The CIVSM allows for an increased number of cases to be run for better optimization of data center designs and layouts without loss of accuracy as compared with traditional CFD/HT.

APPENDIX A

TURBULENCE MODELING

All of the turbulence models that were used in this dissertation are of the Reynolds-Averaged Navier-Stokes (RANS) type (see section 3.2.1). The additional terms that are introduced from the Reynolds decomposition require relations in order to close the set of equations. Although numerous turbulence models have been implemented for many different applications, only the zero-equation turbulence model [39,40], the one-equation Spalart-Allmaras model [41,42], and variations of the two-equation k-ε models [43-47] have been used here. All turbulence modeling constants were kept at their respective default values.

The simplest of the models is the zero-equation turbulence model, which was first proposed by Prandtl in 1925 [39]. The implementation used here was developed by Smagorinsky [40]. Although crude, this model does a good job of predicting flow in thin shear layers and requires minimal computational expense. This modeling technique has found widespread use in external aerodynamics applications, however it does not describe flows with separation [48, 54]. For data center applications, the zero-equation turbulence model can provide a good compromise of fast computation and similar temperature field predictions of much more complicated turbulence models [13-15].

The Spalart-Allmaras turbulence model [41,42] adds one transport equation to the computation by solving for the kinematic eddy viscosity. Originally created for aeronautics applications, the Spalart-Allmaras turbulence model also performs quite well for turbomachinery applications, but not in free shear flows [52]. The specific implementation used here was developed by Dacles-Mariani et al. [42]. Although not as fast as the zero-equation turbulence model, the Spalart-Allmaras turbulence model does a relatively good job of predicting the temperature field in a data center [13-15].

The most common forms of RANS turbulence modeling are the two-equation models, with the k-ε turbulence model variations being the most widely used technique. Of the two-equation models, the standard k-ε turbulence model, which was originally developed by Jones and Launder [43], is the most commonly used turbulence model for data center modeling [17]. Many other variations have been proposed, but the standard k-ε turbulence model forms the basis for most applications [48].

Three k-ε turbulence models were used here, namely the standard, RNG, and realizable. The two additional transport equations in the various k-ε turbulence models solve for the turbulent kinetic energy (k) and its dissipation (ε). The different variations all have similar forms, but go about calculating the turbulent viscosity, turbulent Prandtl number, and the generation and destruction of ε differently [51]. Enhanced wall treatments were also run with the three k-ε turbulence models, but in general showed little difference with the models run with their standard wall functions [13,14]. Additional information on enhanced wall treatments can be found in [48,54,51].

The standard k-ε turbulence model is quite robust, but does have some limitations and weaknesses. The main weaknesses for application to data centers are that the standard k-ε turbulence model is developed only for fully turbulent flows and that it performs poorly in rapidly strained flows. Also due to the additional transport equations, all of the k-ε turbulence models are more computationally expensive per iteration than the zero-equation and Spalart-Allmaras models.

The RNG k-ε turbulence model [45,46] was derived with renormalization group theory, a statistical technique, in order to account for smaller scale effects in the flow than what the standard k-ε turbulence model can capture. Theoretically the RNG k-ε turbulence model should provide the best results in a data center environment since it improves the accuracy of the modeling of low-Reynolds number effects and for rapidly strained flows. In practice, the RNG k-ε turbulence model showed nearly the same accuracy as the standard k-ε turbulence model, but required increased solution time [13,14].

The realizable k-ε turbulence model [47] alters the formulation for the turbulent viscosity and the transport equation for the turbulent dissipation. Unlike in the standard and RNG k-ε turbulence models, the turbulent dissipation transport equation is derived from an exact equation. The realizable k-ε turbulence model also satisfies certain constraints of turbulent flows which mathematically make it “realizable”. These changes are meant to improve upon some of the weaknesses of the standard k-ε turbulence model in certain situations. Although it required the least computational effort, the realizable k-ε turbulence model was also the least accurate performing two-equation model [13,14].

For general data center CFD modeling, the RNG k-ε turbulence model should return the most accurate results for a RANS type turbulence model. However, this was not shown to be the case in [13,14]. Both the zero-equation and Spalart-Allmaras models produced lower RMS differences in those cases. Of the two simpler models, the zero-equation required nearly an order of magnitude less solution time. Based on the results from [13,14], the zero-equation turbulence model should be used as an initial pass turbulence model due to its speed and accuracy. This is what was initially used for the coupled inviscid-viscous solution method (CIVSM) presented here.

APPENDIX B

NUMERICAL ERROR ANALYSIS

There are three main sources of errors in numerical analyses: modeling errors, discretization errors, and iteration errors. Modeling errors are defined as the difference between the actual flow and the exact solution to the analytical model [60]. Discretization error is defined as the difference between the exact solution of the analytical model and the exact solution of the numerical model, which is the system of equations which result from the discretization of the analytical model [60]. Iteration error defined as the difference between the exact and iterative solutions of the numerical model [60]. Only the discretization and iterative errors will be estimated here.

The discretization errors are estimated using the grid convergence index (GCI) procedure defined by Roache [61] and Celik et al. [35]. Table 5 shows the values for the calculations for the rack inlet air temperature at 1.067 m (3.5 ft) above the raised floor for the different flow models used in the single-rack data center test cell. The minimum, maximum, and average values are reported for the seven monitor points at the inlet to the rack, which correspond to the points measured using the three-dimensional mapping tool. The maximum fine GCI reported for the single-rack data center test cell using traditional CFD is for the standard k-ε turbulence model at 0.15%, which corresponds with a discretization error of 0.42°C.

Similarly, table 6 shows the same calculations for the coupled inviscid-viscous solution method (CIVSM) for the single-rack data center test cell. However, the results presented for the CIVSM models were from the intermediate grid rather than the fine grid, which effectively increases the GCI and discretization error by a factor of four. The maximum intermediate GCI reported for the single-rack data center test cell using

CIVSM is for the zero-equation turbulence model at 0.43%, which corresponds with a discretization error of 1.25°C.

Table 5. Single-Rack Data Center Test Cell Discretization Error Calculations for Traditional CFD Flow Models

		Inviscid	Laminar	Zero-Equation	Spalart-Allmaras	Standard k-ε
Grid Size	1	17061	248148	248148	248148	248148
	2	4165	17061	17061	17061	17061
	3	1361	3140	3140	3140	3140
Refinement Factor	r_{21}	1.60	2.44	2.44	2.44	2.44
	r_{32}	1.45	1.76	1.76	1.76	1.76
Temperature (K)	1	287.78	287.13	286.81	287.05	286.64
	2	287.62	287.78	287.21	287.11	286.86
	3	287.99	287.63	287.06	287.34	286.93
Apparent Order	Local	2.07	1.30	0.90	2.53	0.57
	Minimum	0.19	0.14	0.56	0.17	0.14
	Maximum	7.28	3.44	3.42	2.99	1.83
	Average	2.26	1.32	1.14	1.88	0.81
Extrapolated Temperature (K)	T_{ext}^{21}	287.88	286.83	286.48	287.04	286.30
Approximate Error	e_{a}^{21}	0.06%	0.23%	0.14%	0.02%	0.08%
Extrapolated Relative Error	e_{ext}^{21}	0.03%	0.10%	0.11%	0.00%	0.12%
Grid Convergence Index (GCI)	Fine	0.04%	0.13%	0.14%	0.00%	0.15%
Discretization Error (°C)		0.12	0.37	0.41	0.01	0.42
Oscillating Convergence	Number	4	2	5	1	0
	Percent	57.1%	28.6%	71.4%	14.3%	0.0%

Table 6. Single-Rack Data Center Test Cell Discretization Error Calculations for the CIVSM Models

		CIVSM Zero-Equation	CIVSM Standard k-ε
Grid Size	1	870144	870144
	2	108768	108768
	3	15584	15584
Refinement Factor	r_{21}	2.00	2.00
	r_{32}	1.91	1.91
Temperature (K)	1	287.88	289.06
	2	287.47	288.65
	3	288.51	288.60
Apparent Order	Local	1.41	2.68
	Minimum	1.41	0.15
	Maximum	8.32	4.73
	Average	2.80	1.90
Extrapolated Temperature (K)	T_{ext}^{21}	288.13	289.14
Approximate Error	e_a^{21}	0.14%	0.14%
Extrapolated Relative Error	e_{ext}^{21}	0.09%	0.03%
Grid Convergence Index	Fine	0.11%	0.03%
	Intermediate	0.43%	0.12%
Discretization Error (°C)		1.25	0.35
Oscillating Convergence	Number	6	3
	Percent	85.7%	42.9%

Figure 39 shows the rack inlet air temperature predictions from the three grids along with the extrapolated temperature from the CIVSM using the k-ε turbulence model. Figure 40 shows the temperature prediction from grid 2 along with discretization error bars as well as the measured temperatures and their error bars. Grid 2 was the final grid that was used for the comparison in chapter 3, and was the intermediate grid. The averaged apparent order of accuracy was used to create the discretization error bars shown.

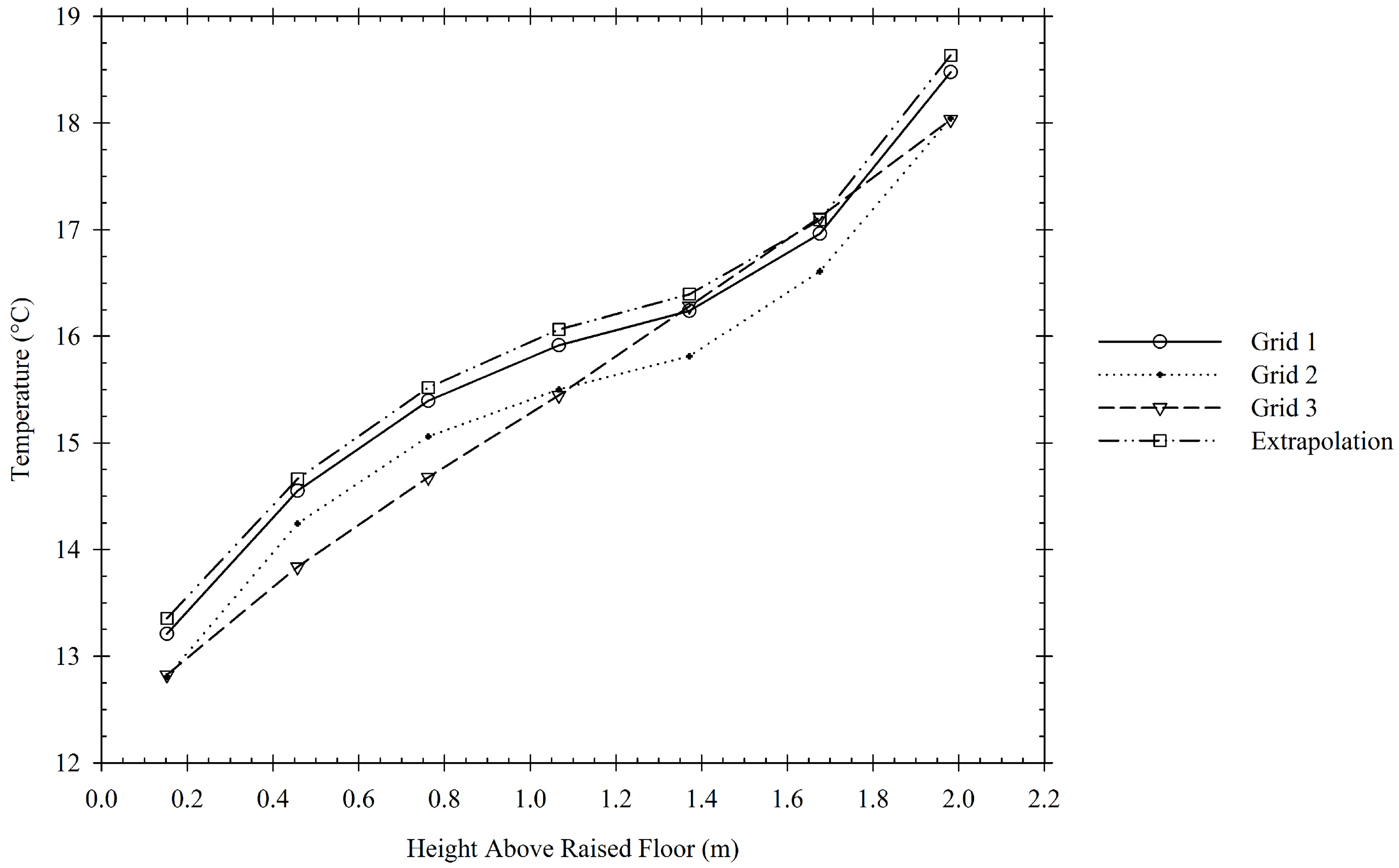


Figure 39. Rack Inlet Air Temperature Predictions for the CIVSM Using the Standard k-ε Turbulence Model

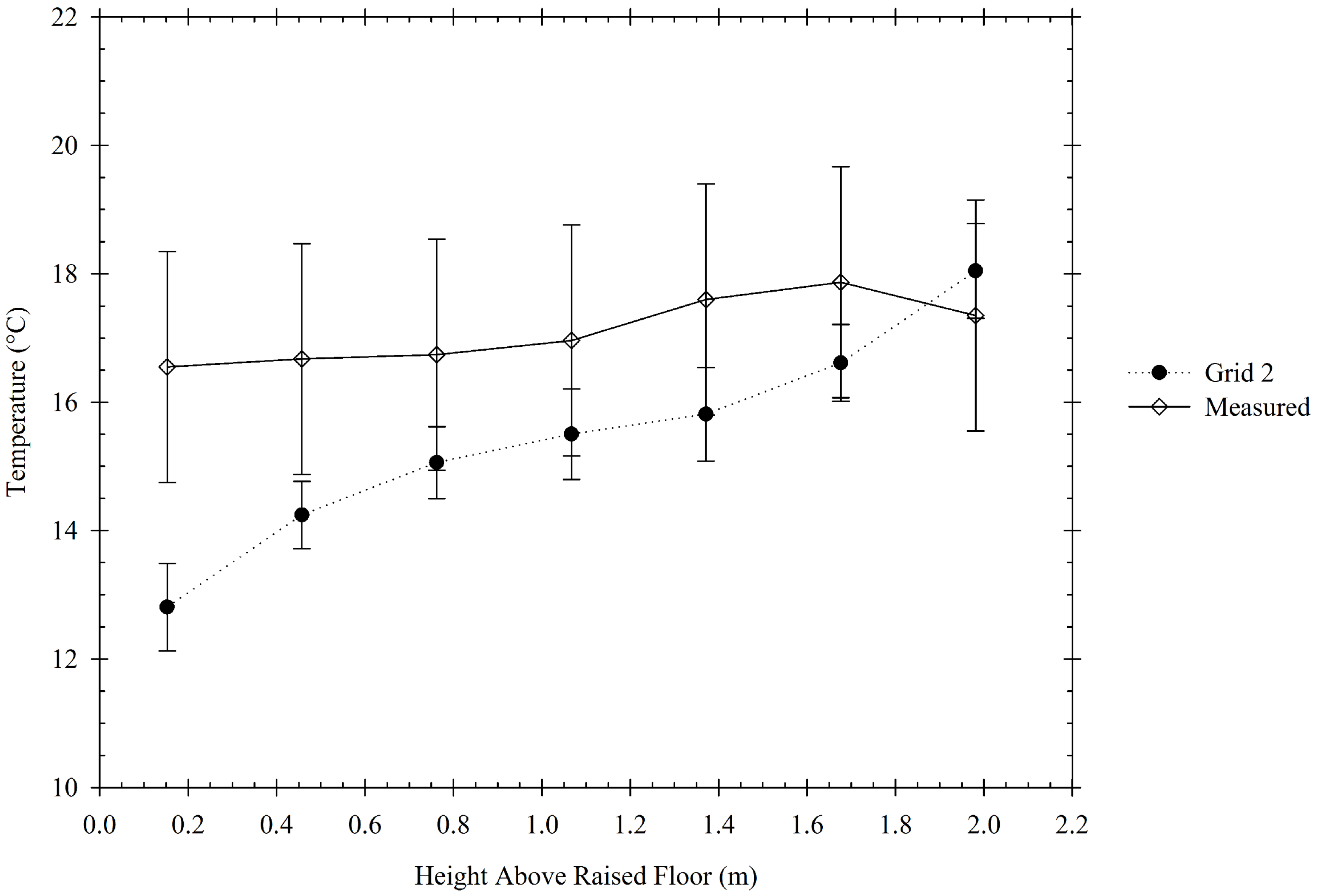


Figure 40. Rack Inlet Air Temperature Predictions for the CIVSM Using the Standard k-ε Turbulence Model and Measured Data

Table 7 shows the values for the calculations for the inlet air temperature of the highest server in the middle of the aisle of the 1 CRAC, multi-rack data center using the modified body force perforated-tile modeling. The minimum, maximum, and average values are reported for the inlet air temperatures of all 71 pieces of IT equipment in the data center. The GCI results from the inviscid model are based on only two grids, and as expected show the highest discretization error of the three model types. Both the CIVSM and standard k-ε models used the intermediate grids for their final grids and show quite similar discretization error. The intermediate GCI for the multi-rack data center using the standard k-ε turbulence model was 0.08%, which corresponds with a discretization error of 0.22°C while the CIVSM model was slightly higher at 0.09%, which corresponds with a discretization error of 0.27°C.

The other major source of numerical errors being estimated is iteration errors. Iteration errors should be at least an order of magnitude less than the discretization errors [35]. The highest iteration errors will be in those cases where the solution had not completely reached the convergence criteria. Two of these cases are examined here: the worst case convergence from the single-rack data center test cell, and the worst case convergence from the multi-rack data center.

The iteration errors are estimated using the procedure outlined by Celik et al. [35]. In the case of the single-rack data center test cell, the highest iteration errors occurred in the solution for the laminar flow models. The procedure resulted in an estimated iteration error of 0.0098°C for the same seven temperature monitor points used for the discretization error estimate. This fulfills the requirement of ensuring the iteration errors are at least an order of magnitude less than the discretization errors for the worst case condition.

For the multi-rack data center, only the inviscid model didn't achieve convergence. For this model, the iteration errors are estimated to be 0.073°C for the 71 inlet air temperatures. This estimate is right at an order of magnitude less than the discretization error for this case.

Table 7. Multi-Rack Data Center Discretization Error Calculations

		Inviscid	CIVSM	Standard k-ε
Grid Size	1	547584	2197688	4380672
	2	68448	274711	1198137
	3	-	68448	68448
Refinement Factor	r_{21}	2.00	2.00	1.54
	r_{32}	-	1.59	2.60
Temperature (K)	1	291.34	291.63	292.88
	2	291.16	291.45	292.97
	3	-	290.99	292.56
Apparent Order	Local	-	2.43	1.76
	Minimum	-	0.34	0.08
	Maximum	-	12.15	7.60
	Average	-	3.77	2.00
Extrapolated Temperature (K)	T_{ext}^{21}	291.41	291.67	292.80
Approximate Error	e_{a}^{21}	-	0.06%	0.03%
Extrapolated Relative Error	e_{ext}^{21}	-	0.01%	0.03%
Grid Convergence Index	Fine	0.26%	0.02%	0.04%
	Intermediate	-	0.09%	0.08%
Discretization Error (°C)		0.75	0.27	0.22
Oscillating Convergence	Number	-	32	23
	Percent	-	45.1%	32.4%

For both the single-rack data center test cell and the multi-rack data center, both the discretization and iteration errors for all of the worst case models were of acceptable levels.

APPENDIX C

VELOCITY PROFILE PLOTS

The following figures depict the velocity profiles of the measured PIV data, along with the predictions from models using an inviscid solver, the CIVSM, and traditional CFD/HT using the standard k-ε turbulence model (see chapter 5). Both the modified body force and porous jump tile modeling techniques (see section 4.2.2) were employed for three different tile cases: full tile, framed tile, and blocked tile. The velocity profiles are shown for the measured data and predictions along the plane (shown in figure 21) for three different heights: 0.15 m (0.5 ft), 0.91 m (3.0 ft), and 1.68 m (5.5 ft).

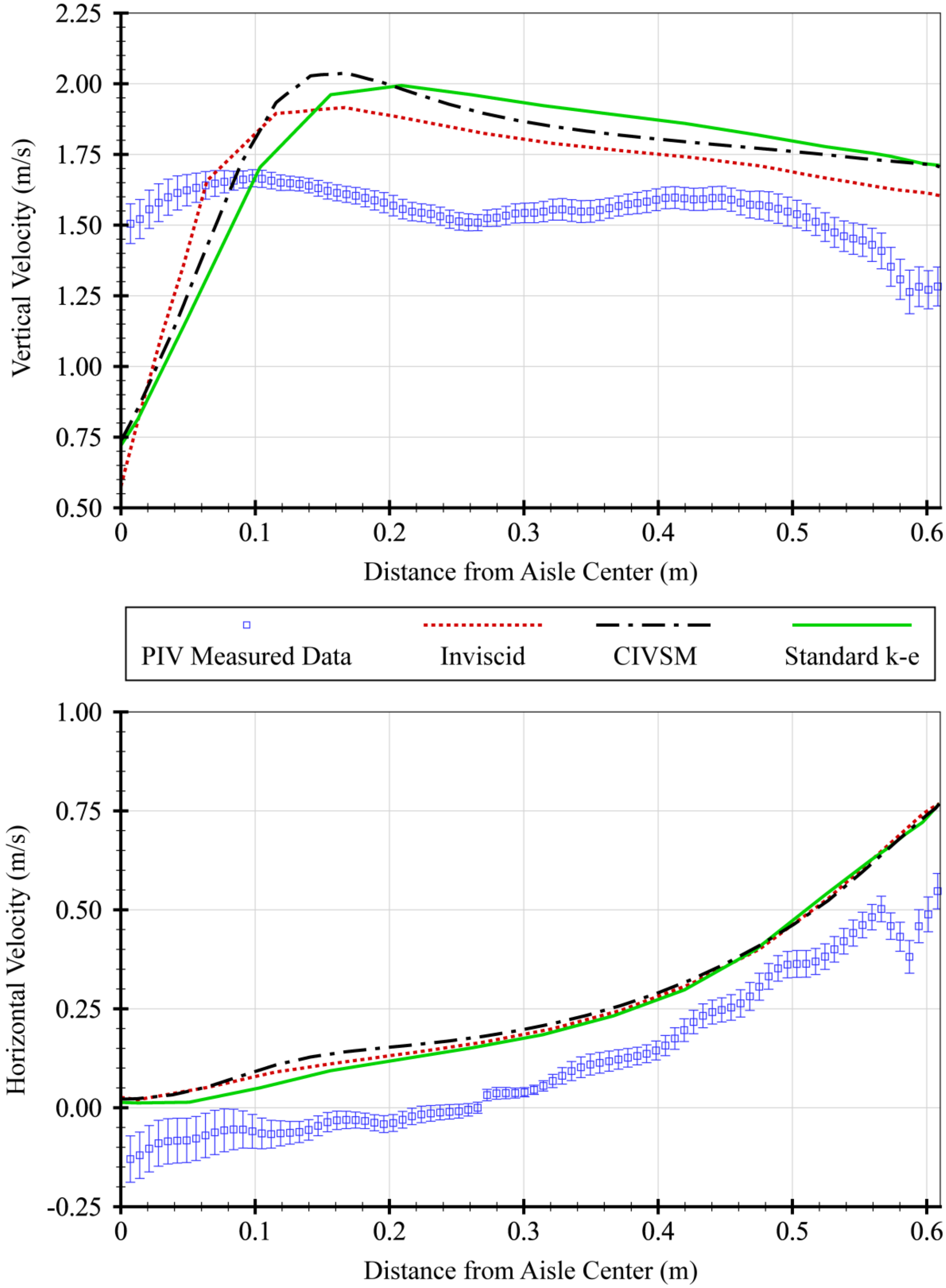


Figure 41. Velocity Plots for Case 1, Modified Body Force Model @ 0.91 m (3.0 ft)

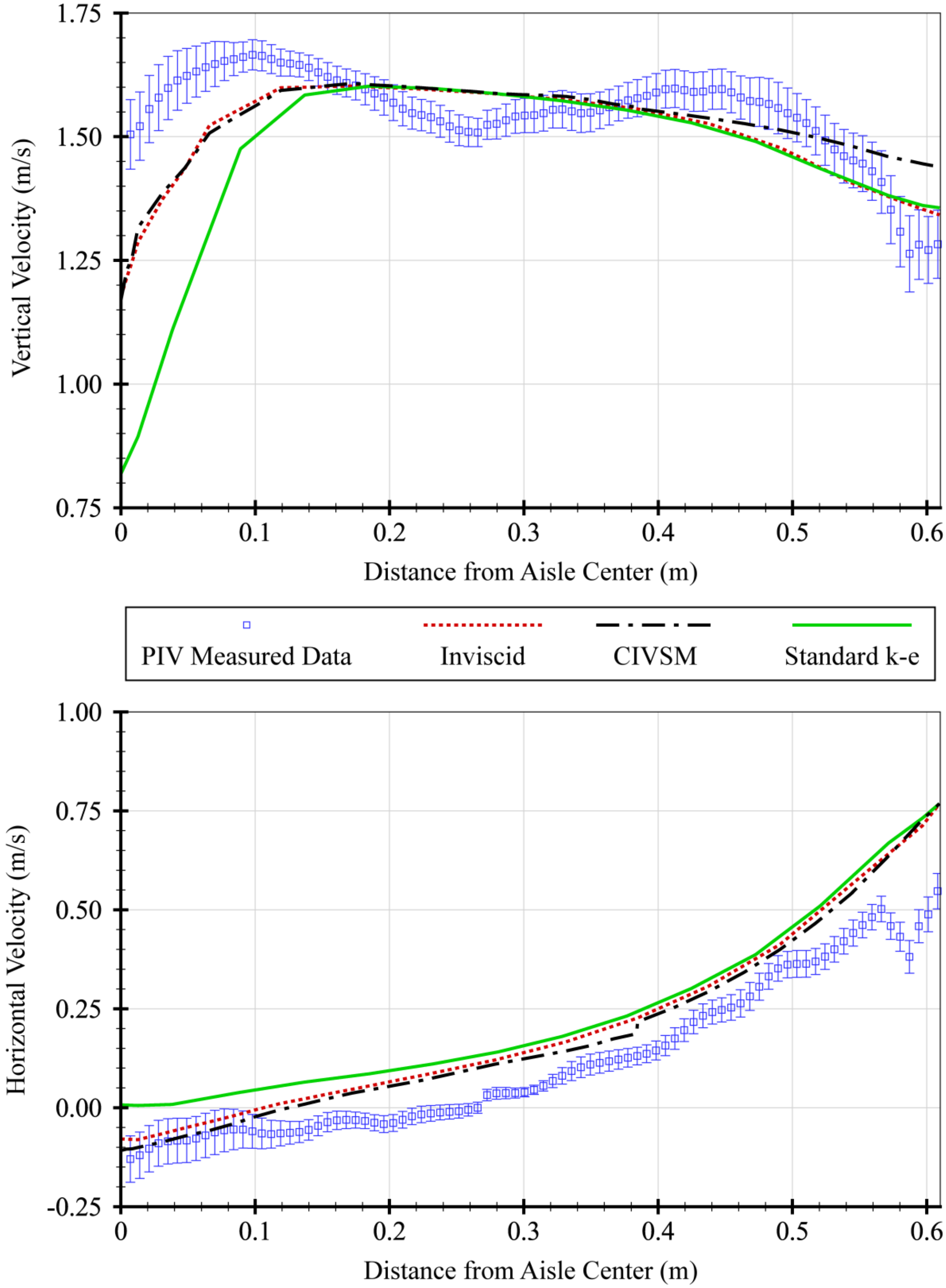


Figure 42. Velocity Plots for Case 1, Porous Jump Model @ 0.91 m (3.0 ft)

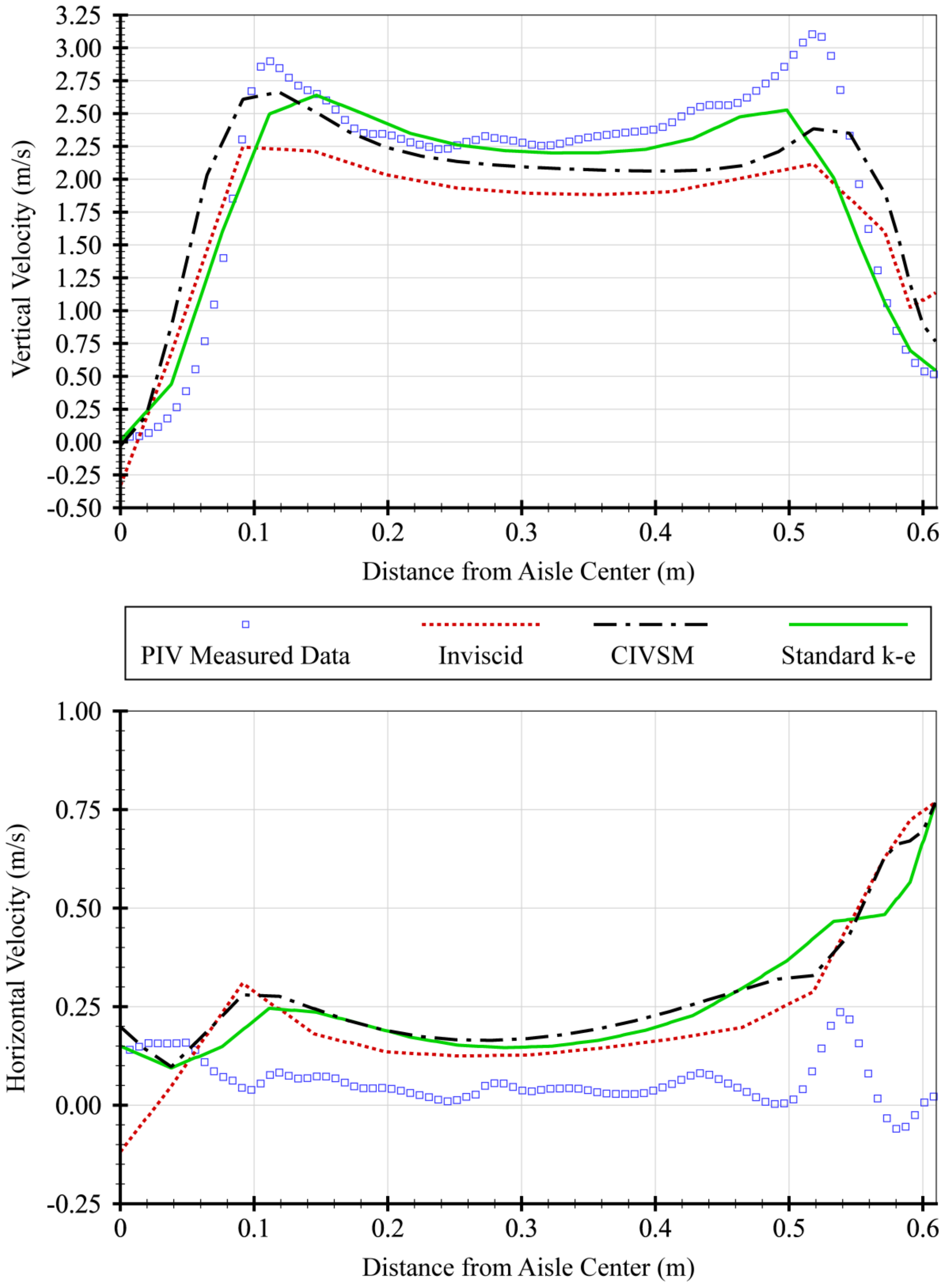


Figure 43. Velocity Plots for Case 2, Modified Body Force Model @ 0.15 m (0.5 ft)

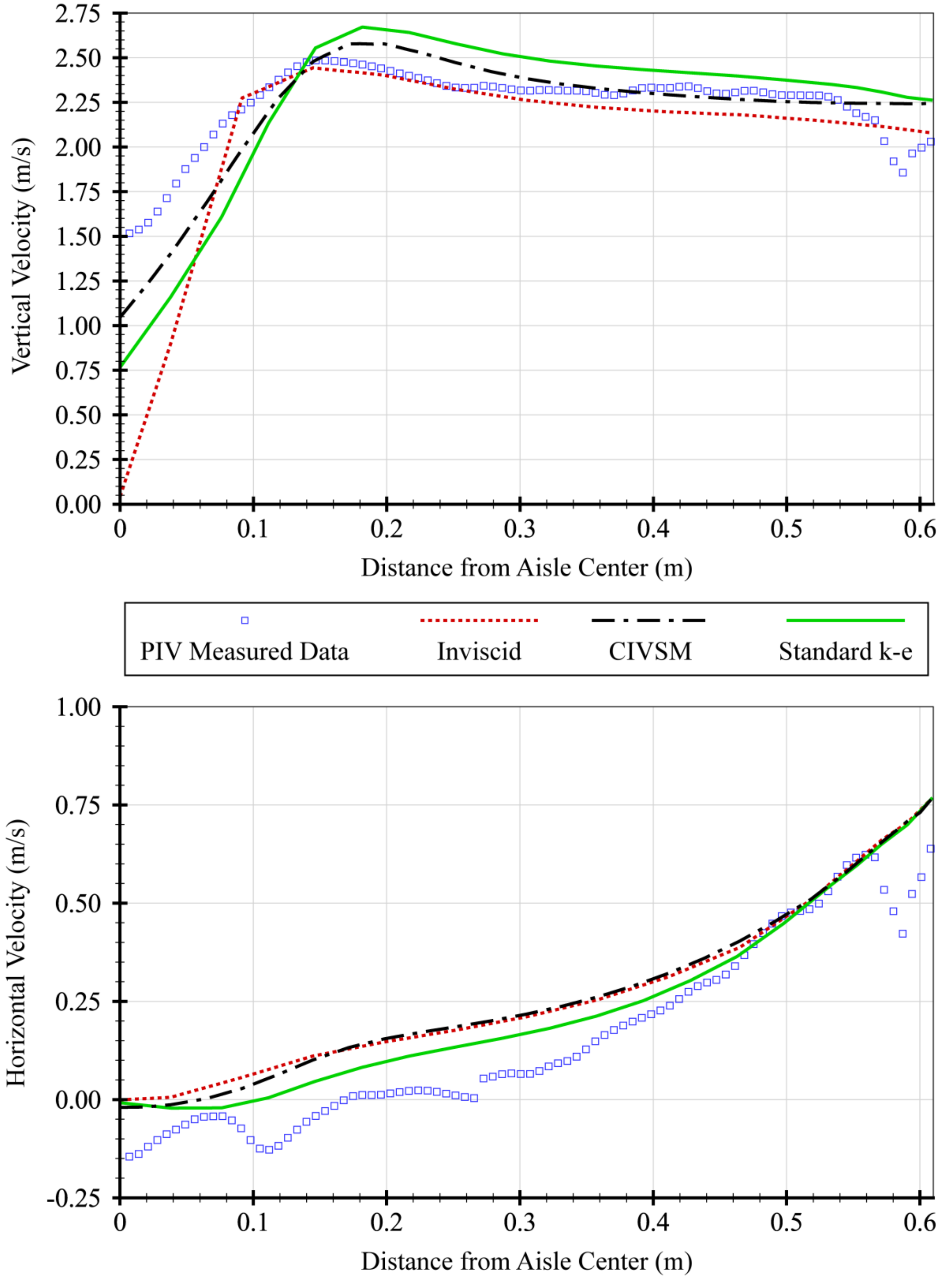


Figure 44. Velocity Plots for Case 2, Modified Body Force Model @ 0.91 m (3.0 ft)

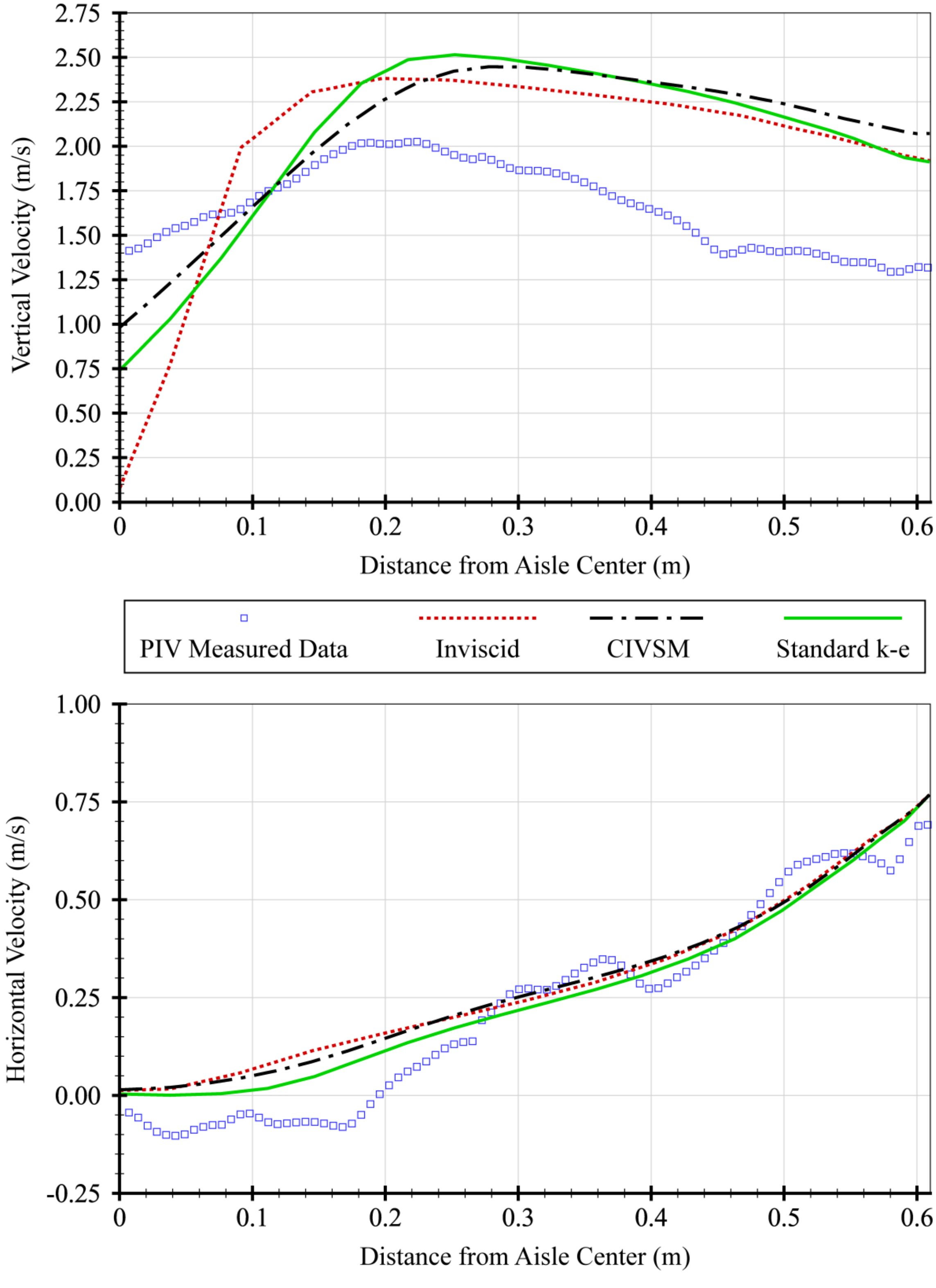


Figure 45. Velocity Plots for Case 2, Modified Body Force Model @ 1.68 m (5.5 ft)

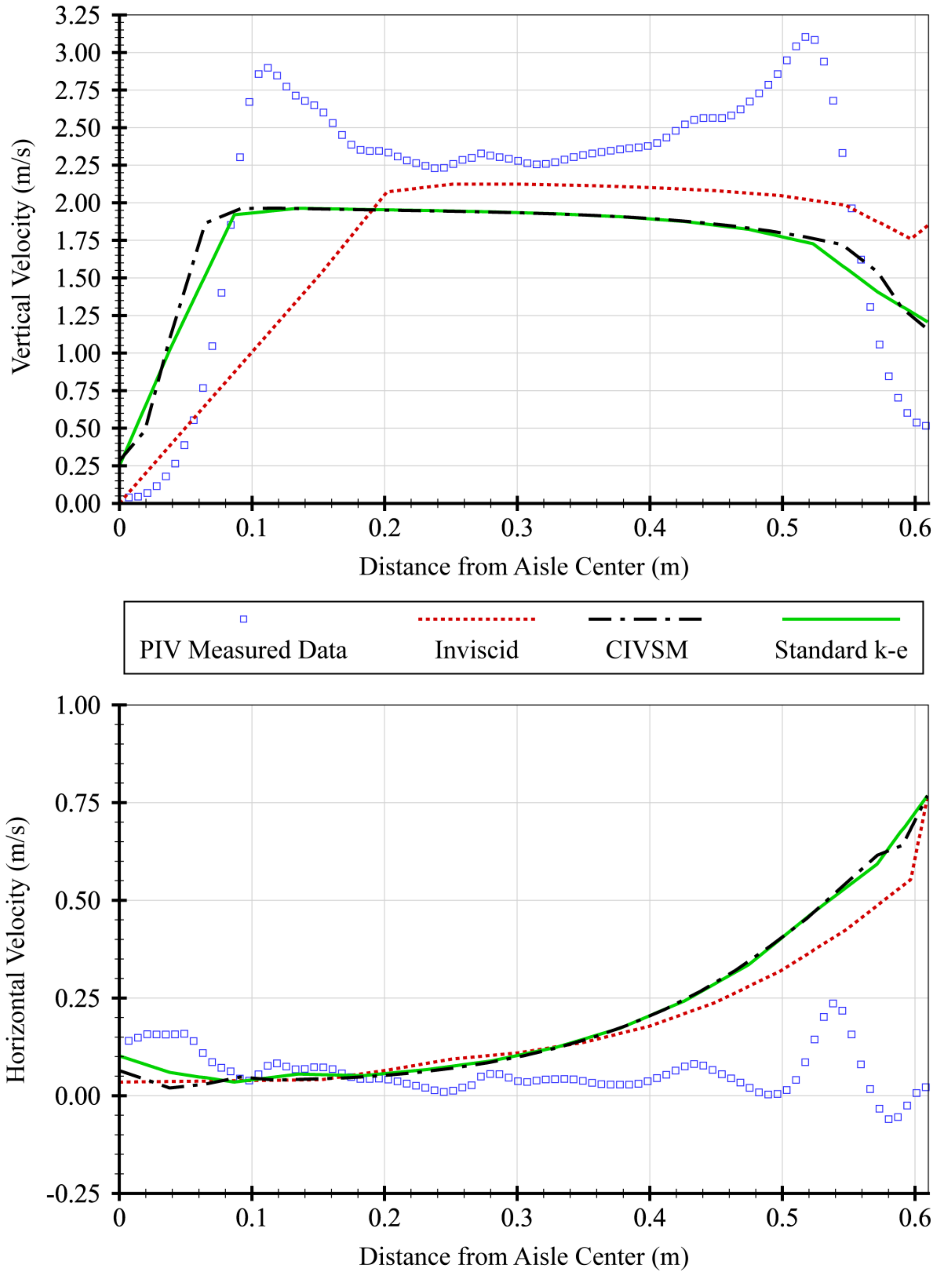


Figure 46. Velocity Plots for Case 2, Porous Jump Model @ 0.15 m (0.5 ft)

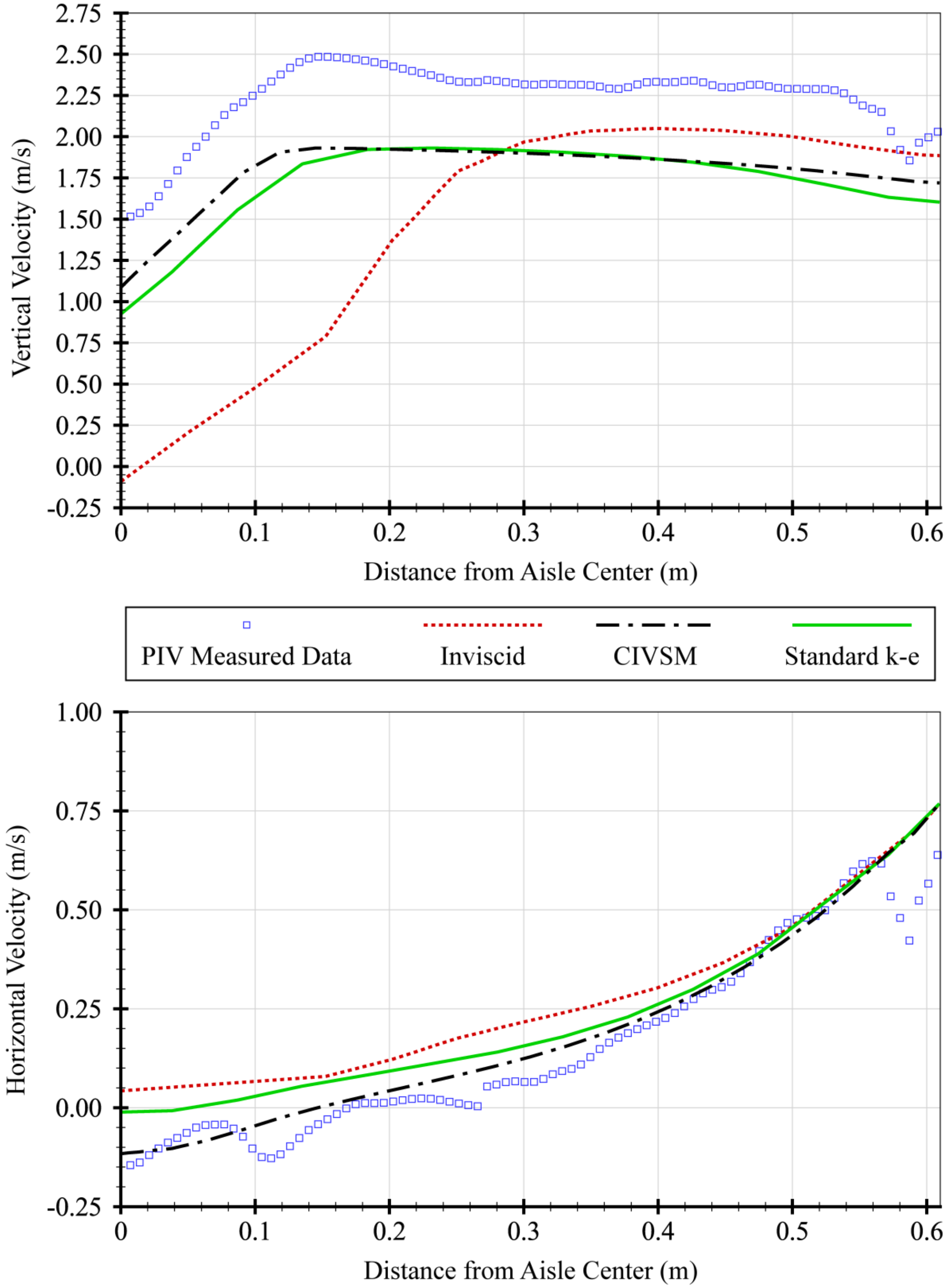


Figure 47. Velocity Plots for Case 2, Porous Jump Model @ 0.91 m (3.0 ft)

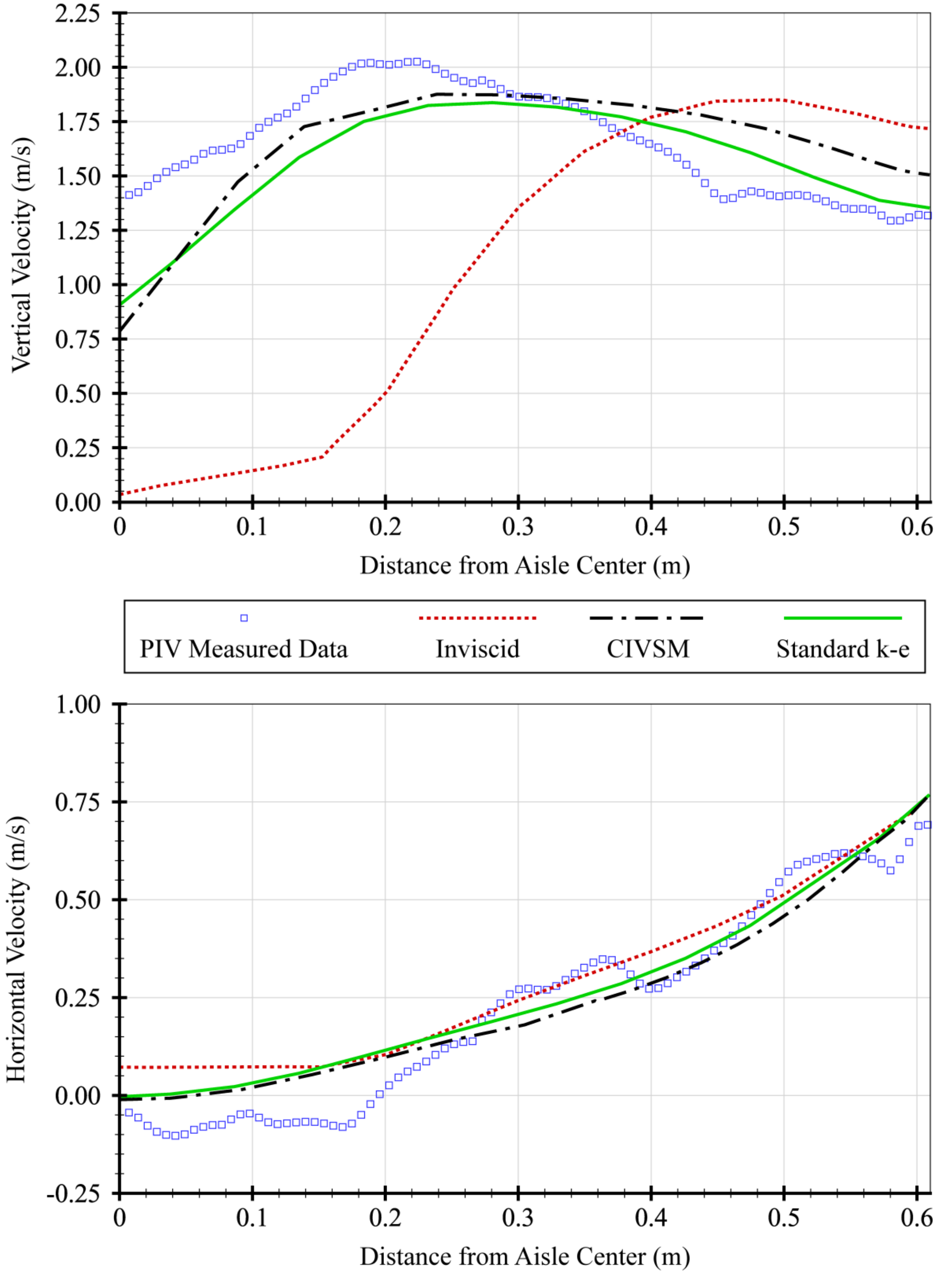


Figure 48. Velocity Plots for Case 2, Porous Jump Model @ 1.68 m (5.5 ft)

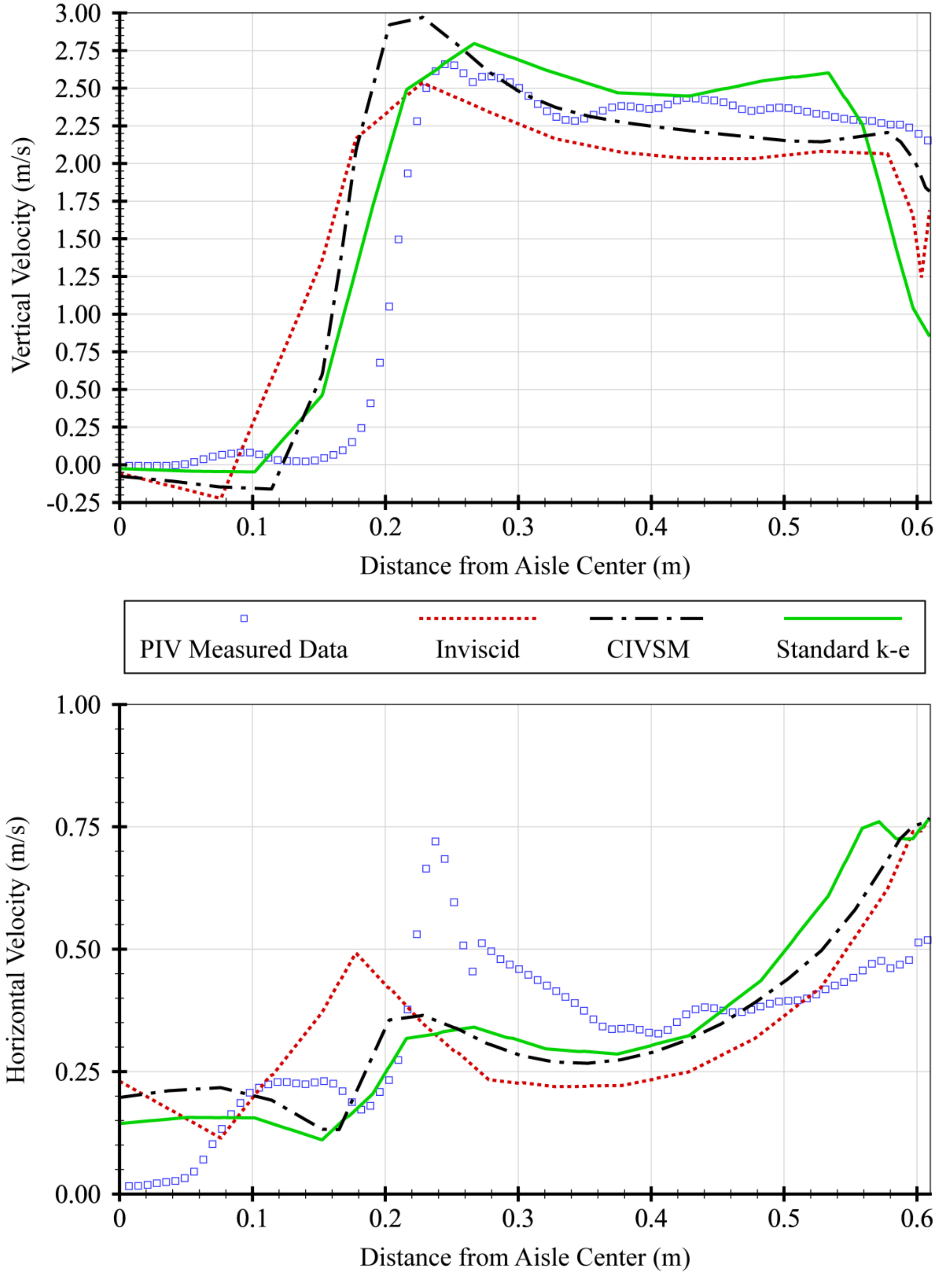


Figure 49. Velocity Plots for Case 3, Modified Body Force Model @ 0.15 m (0.5 ft)

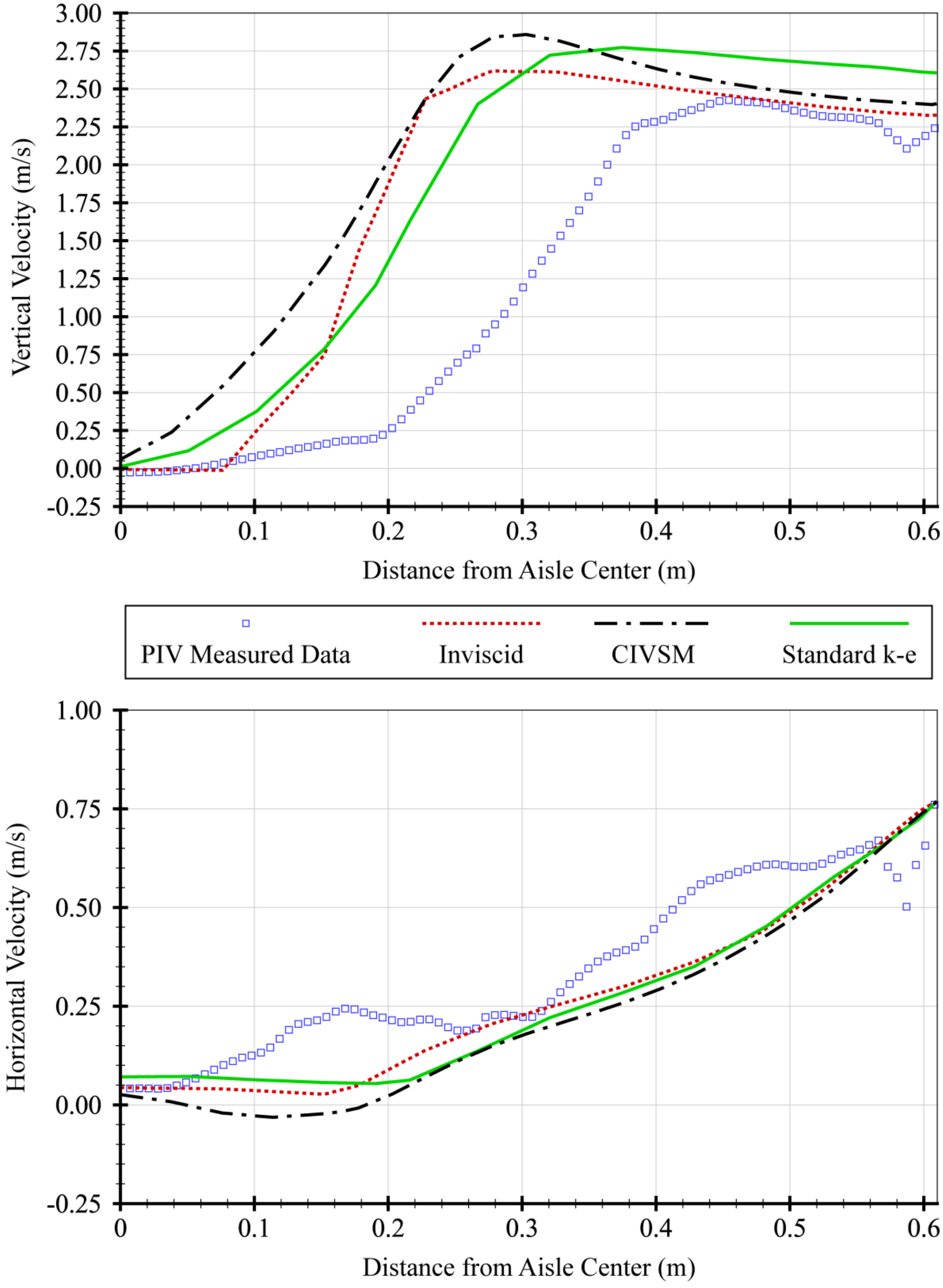


Figure 50. Velocity Plots for Case 3, Modified Body Force Model @ 0.91 m (3.0 ft)

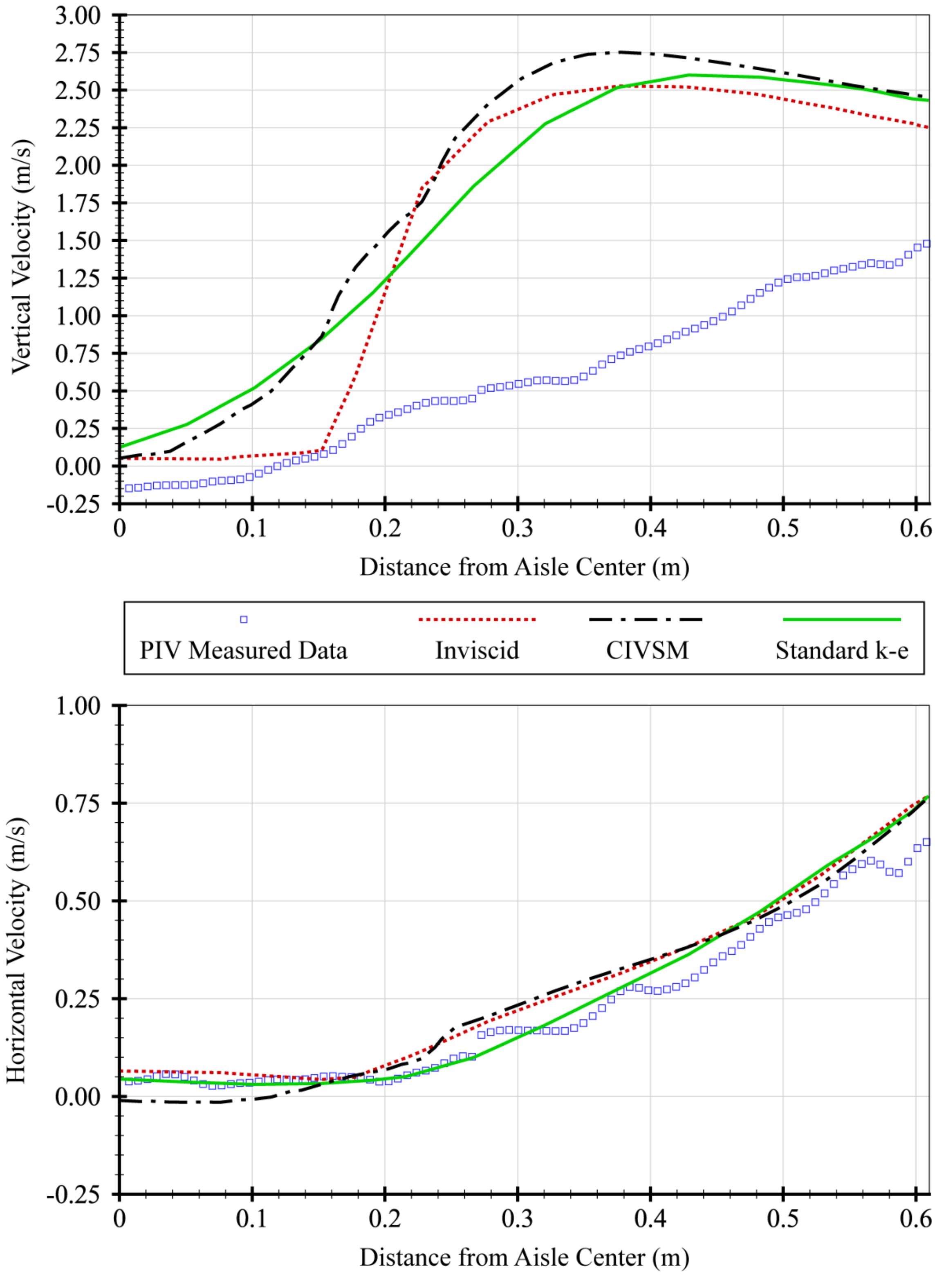


Figure 51. Velocity Plots for Case 3, Modified Body Force Model @ 1.68 m (5.5 ft)

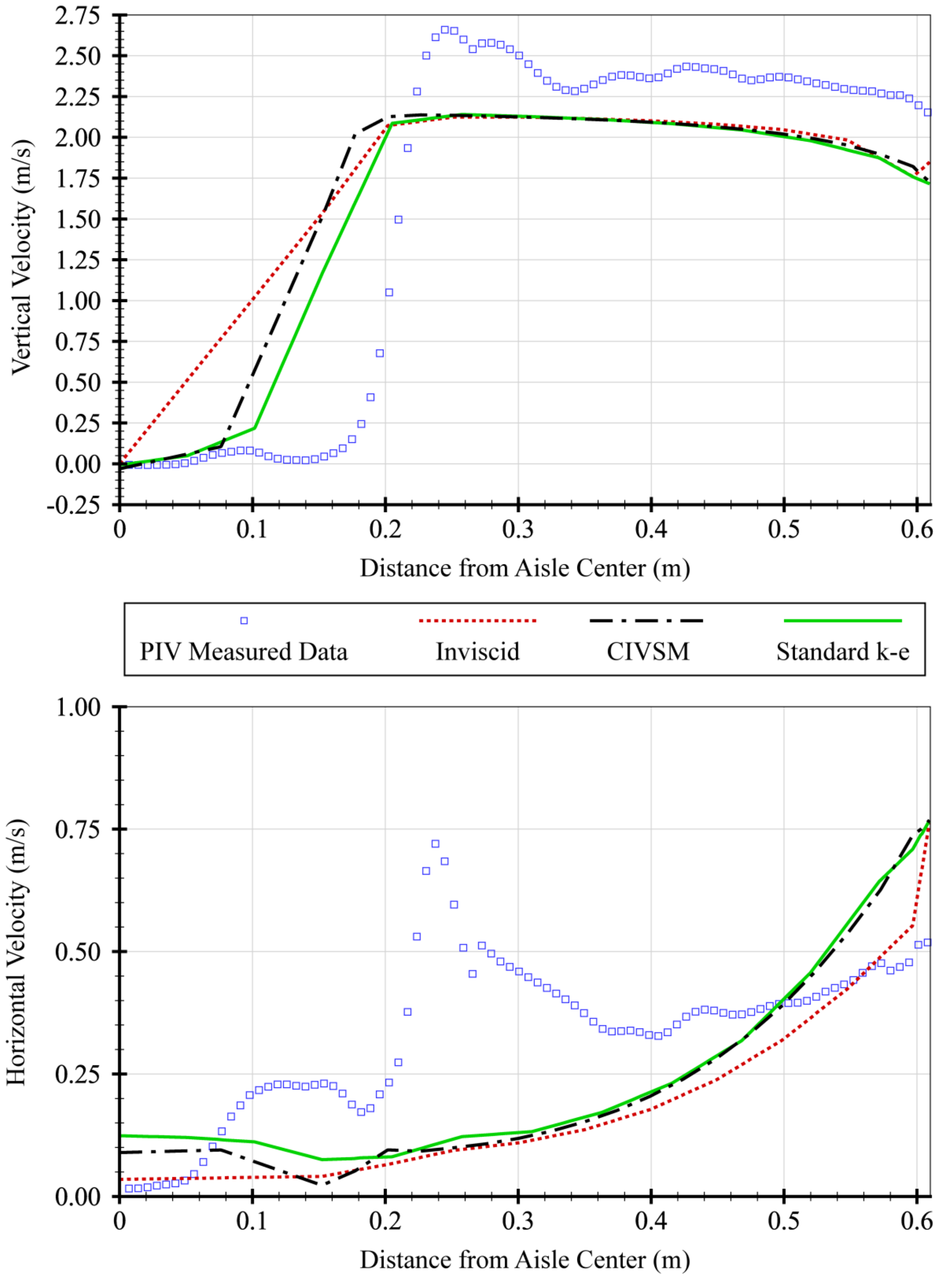


Figure 52. Velocity Plots for Case 3, Porous Jump Model @ 0.15 m (0.5 ft)

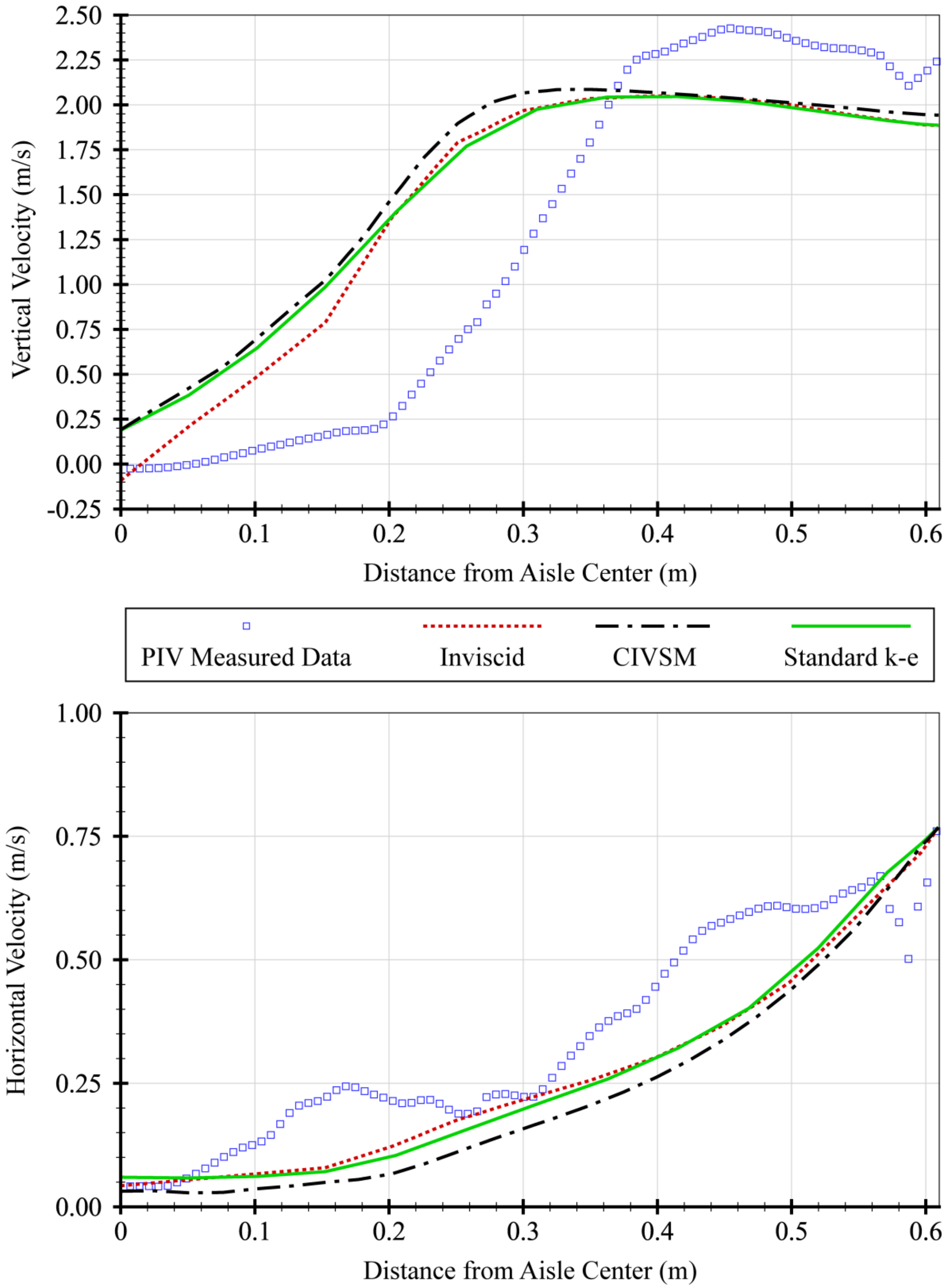


Figure 53. Velocity Plots for Case 3, Porous Jump Model @ 0.91 m (3.0 ft)

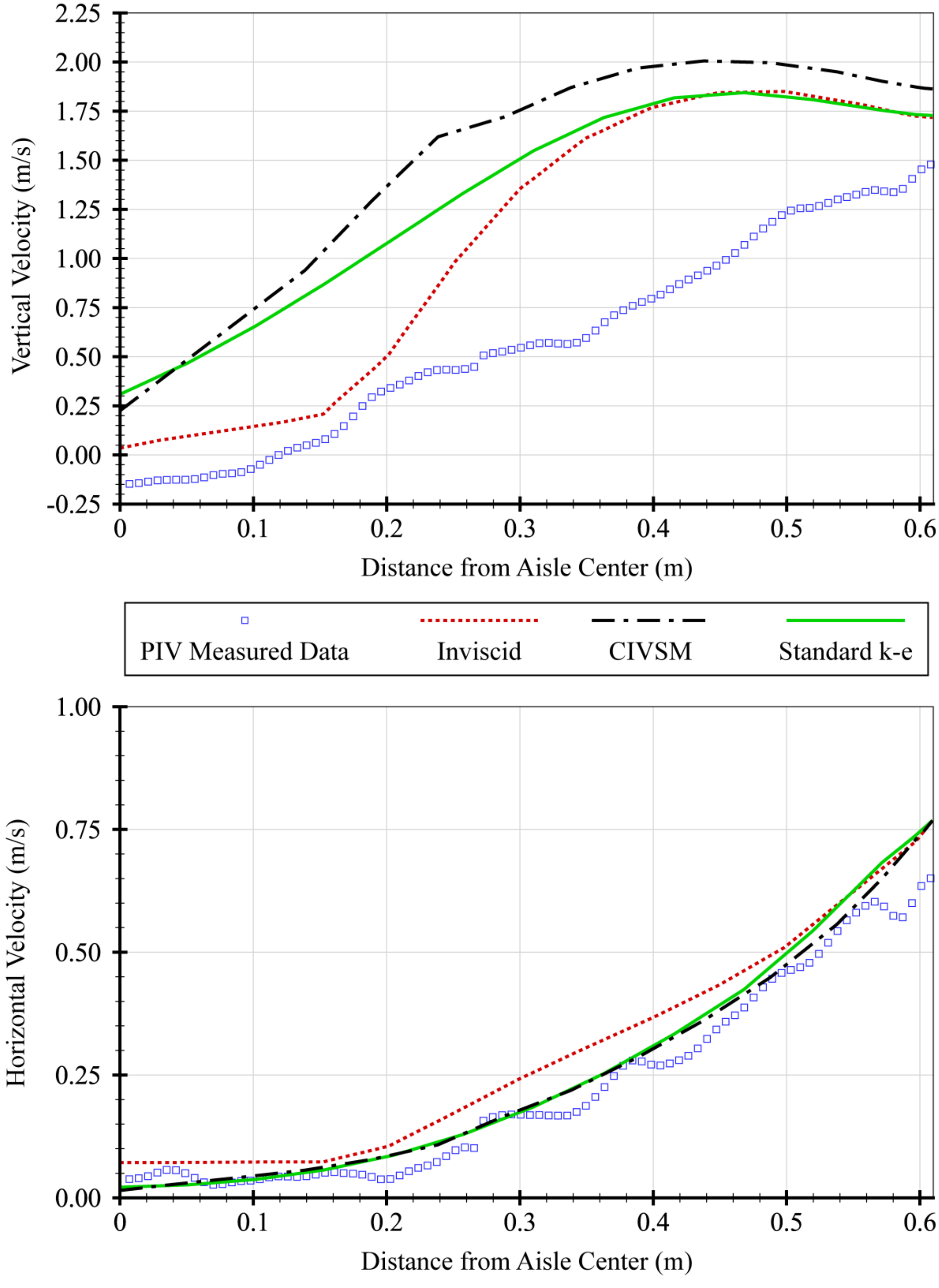


Figure 54. Velocity Plots for Case 3, Porous Jump Model @ 1.68 m (5.5 ft)

REFERENCES

[1] American Society of Heating, Refrigerating and Air Conditioning Engineers (ASHRAE), 2012, *Datacom Equipment Power Trends and Cooling Applications, 2nd Edition*, ASHRAE, Atlanta, Georgia, USA.

[2] Koomey, J., Growth in Data Center Electricity Use 2005 to 2010, July 2011, www.analyticspress.com/datacenters.html

[3] Schmidt, R., Cruz, E., and Iyengar, M., 2005, "Challenges of Data Center Thermal Management," IBM J. Res. & Dev., **49**(4/5), pp. 709-723.

[4] American Society of Heating, Refrigerating and Air Conditioning Engineers (ASHRAE), 2012, *Thermal Guidelines for Data Processing Environments, 3rd Edition*, ASHRAE, Atlanta, Georgia, USA.

[5] VanGilder, J., and Schmidt, R., 2005, "Airflow Uniformity Through Perforated Tiles in a Raised-Floor Data Center," Proceedings of IPACK'05, July 17-22, San Francisco, California, USA.

[6] Kumar, P., and Joshi, Y., 2010, "Experimental Investigations on the Effect of Perforated Tile Air Jet Velocity on Server Air Distribution in a High Density Data Center," Proceedings of the InterSociety Conference on Thermal Phenomena (ITherm), June 2-5, Las Vegas, Nevada, USA.

[7] Patel, C., Bash, C., and Belady, C., 2001, "Computational Fluid Dynamics Modeling of High Compute Density Data Centers to Assure System Inlet Air Specifications," Proceedings of IPACK'01, July 8-13, Kauai, Hawaii, USA.

[8] Shrivastava, S., Iyengar, M., Sammakia, B., Schmidt, R., and VanGilder, J., 2006, "Experimental-Numerical Comparison for a High-Density Data Center: Hot Spot heat Fluxes in Excess of 500 W/ft^2," Proceedings of the Inter Society Conference on Thermal Phenomena (ITherm), San Diego, California, USA.

[9] Schmidt, R., Iyengar, M., and Caricari, J., 2007, "Data Center Housing the World's 3rd Fastest Supercomputer - Above Floor Thermal Measurements Compared to CFD Analysis," Proceedings of IPACK'07, July 8-12, Vancouver, Canada.

[10] Amemiya, Y., Iyengar, M., Herman, H., O'Boyle, M., Schappert, M., Shen, J., and van Kessel, T., 2007, "Comparison of Experimental Temperature Results with Numerical Modeling Predictions of a Real-World Compact Data Center Facility," Proceedings of IPACK'07, July 8-12, Vancouver, Canada.

[11] Iyengar, M., Schmidt, R.R., Hamann, H., and VanGilder, J., 2007, "Comparison Between Numerical and Experimental Temperature Distributions in a Small Data Center Test Cell," Proceedings of IPACK'07, July 8-12, Vancouver, Canada.

[12] Zhang, X., VanGilder, J., Iyengar, M., and Schmidt, R., 2008, "Effect of Rack Modeling Detail on the Numerical Results of a Data Center Test Cell," Proceedings of the Inter Society Conference on Thermal Phenomena (ITherm), May 28-31, Orlando, Florida, USA.

[13] Cruz, E., Joshi, Y., Iyengar, M., and Schmidt, R., 2009, "Comparison of Numerical Modeling to Experimental Data in a Small Data Center Test Cell," Proceedings of IPACK'09, July 19-23, San Francisco, California, USA.

[14] Cruz, E., Joshi, Y., Iyengar, M., and Schmidt, R., 2009, "Comparison of Numerical Modeling to Experimental Data in a Small, Low Power Data Center Test Cell," Proceedings of the International Mechanical Engineering Congress and Exposition (IMECE), November 13-19, Lake Buena Vista, Florida, USA.

[15] Cruz, E., Joshi, Y., 2013, "Inviscid and Viscous Numerical Models Compared to Experimental Data in a Small Data Center Test Cell," J. Electron. Packag. **135**(3), pp. 030904.

[16] Lettieri, D., Toulouse, M., Bash, C., Shah, A., Carey, V., 2013, "Computational and Experimental Validation of a Vortex-Superposition-Based Buoyancy Approximation for the COMPACT Code in Data Centers," J. Electron. Packag. **135**(3), pp. 030903.

[17] Rambo, J., and Joshi, Y., 2007, "Modeling of Data Center Airflow and Heat Transfer: State of the Art and Future Trends," Distributed and Parallel Databases, **21**(2-3), pp. 193– 225.

[18] Hamann, H., Lacey, J., O'Boyle, M., Schmidt, R., and Iyengar, M., 2008, "Rapid Three Dimensional Thermal Characterization of Large Scale Computing

Facilities," IEEE Transactions on Components and Packaging Technologies, **31**(2), pp. 444-448.

[19] Moore, J., Chase, J., Farkas, K., and Ranganathan, P., 2005, "Data Center Workload Monitoring, Analysis, and Emulation," Proceedings of the Eighth Workshop on Computer Architecture Evaluation Using Commercial Workloads, February 12, San Francisco, California, USA.

[20] Moore, J., Chase, J., Ranganathan, P., and Sharma, R., 2005, "Making Scheduling Cool: Temperature-Aware Workload Placement in Data Centers," Proceedings of the Usenix Technical Conference, April 10-15, Anaheim, California, USA.

[21] Karlsson, M., Karamanolis, C., and Zhu, X., 2004, "Triage: Performance Isolation and Differentiation for Storage Systems," Proceedings of the Twelfth International Workshop on Quality of Service, June 7-9, Montreal, Canada.

[22] Sharma, R.K., Bash, C., Patel, C.D., Friedrich, R.J., and Chase, J.S., 2003, "Balance Of Power: Dynamic Thermal Management for Internet Data Centers," Whitepaper issued by Hewlett Packard Laboratories, Tech. Rept. HPL-2003-5.

[23] Samadiani, E., and Joshi, Y., 2010, "Reduced Order Thermal Modeling of Data Centers Via Proper Orthogonal Decomposition: a Review," Int. J. Numer. Methods Heat Fluid Flow, **20**(5), pp. 529-550.

[24] Toulouse, M., Doljac, G., Carey, V., and Bash, C., 2009, "Exploration of a Potential-Flow-Based Compact Model of Air-Flow Transport in Data Centers," Proceedings of the International Mechanical Engineering Congress and Exposition (IMECE), November 13-19, Lake Buena Vista, Florida, USA.

[25] Toulouse, M., Lettieri D., Carey, V., and Bash, C., 2010, "Experimental Validation of the COMPACT Code in Data Centers," Proceedings of the International Mechanical Engineering Congress and Exposition (IMECE), November 12-18, Vancouver, British Columbia, Canada.

[26] Samadiani, E., Joshi, Y., Hamann, H., Iyengar, M., Kamalsy, S., and Lacey, J., 2012, "Reduced Order Thermal Modeling of Data Centers via Distributed Sensor Data," J Heat Transfer, **134**(4), pp. 807-814.

[27] Lopez, V., and Hamann, H., 2011, "Heat Transfer Modeling in Data Centers," Int. J. Heat Mass Transfer, **54**(25-26), pp. 5306-5318.

[28] Hamann, H., Lopez, V., and Stepanchuk, A., 2010, “Thermal Zones for More Efficient Data Center Energy Management,” Proceedings of the Inter Society Conference on Thermal Phenomena (ITherm), June 2-5, Las Vegas, Nevada, USA.

[29] Das, R., Yarlanki, S., Hamann, H., Kephart, J., and Lopez, V., 2011, “A Unified Approach to Coordinated Energy-Management in Data Centers,” Proceedings of Network and Service Management (CNSM), October 24-28, Paris, France.

[30] Lopez, V., and Hamann, H., 2010, “Measurement-Based Modeling for Data Centers,” Proceedings of the Inter Society Conference on Thermal Phenomena (ITherm), June 2-5, Las Vegas, Nevada, USA.

[31] Sankar, L., Bharadvaj, B., and Tsung, F., 1993, “Three-Dimensional Navier-Stokes/Full-Potential Coupled Analysis for Viscous Transonic Flow,” AIAA J., **31**(10), pp. 1857-1862.

[32] Berezin, C., and Sankar, L., 1998, “Improvements to a Tightly Coupled Viscous-Inviscid Procedure for Three-Dimensional Unsteady Transonic Flow,” Comput. Fluids, **27**(5-6), pp.689-694.

[33] ANSYS Fluent, www.ansys.com, last accessed April 28, 2015.

[34] IBM IntelliStation, www.ibm.com, last accessed April 28, 2015.

[35] Celik, I. B., Ghia, U., Roache, P. J., Freitas, C. J., Coleman, H., and Raad, P. E., 2008, “Procedure for Estimation and Reporting of Uncertainty Due to Discretization in CFD Applications,” ASME J. Fluids Eng., **130**(7), pp. 078001.

[36] Cruz, E., US Patent No. 8744812, 2014.

[37] Cruz, E., US Patent No. 8756040, 2014.

[38] Schwarz, H. A., 1870. “Über einen Grenzübergang durch alternierendes Verfahren,” Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich **15**, pp. 272-286.

[39] Prandtl, L., 1925, “Bericht über Untersuchungen zur ausgebildeten Turbulenz,” Z. Angew. Math. Mech., **5**(2), pp. 136–139.

[40] Smagorinsky, J., 1963, “General Circulation Experiments with the Primitive Equations,” Mon. Weather Rev., **91**(3), pp. 99–164.

[41] Spalart, P., and Allmaras, S., 1992, “A One-Equation Turbulence Model for Aerodynamic Flows,” AIAA Pap. 92-0439.

[42] Dacles-Mariani, J., Zilliac, G. G., Chow, J. S. and Bradshaw, P., 1995, "Numerical/Experimental Study of a Wingtip Vortex in the Near Field," AIAA J., **33**(9), pp. 1561-1568.

[43] Jones, W. P., and Launder, B. E., 1972, "The Prediction of Laminarization with a Two-Equation Model of Turbulence," Int. J. Heat Mass Transfer **15**, pp. 301-314.

[44] Launder B., and Spalding, D., 1972, *Lectures in Mathematical Models of Turbulence,* Academic Press, London, England.

[45] Yakhot, V., and Orszag, S., 1986, "Renormalization Group Analysis of Turbulence I Basic Theory," J. Sci. Comput., **1**(1), pp. 1–51.

[46] Orszag, S., Yakhot, V., Flannery, W., Boysan, F., Choudhury, D., Maruzewski, J., and Patel, B., 1993, "Renormalization Group Modeling and Turbulence Simulations," Proceedings of the International Conference on Near-Wall Turbulent Flows, March 15-17, Tempe, Arizona, USA.

[47] Shih, T., Liou, W., Shabbir, A., Yang, Z., and Zhu, J., 1995, "A New k-epsilon Eddy-Viscosity Model for High Reynolds Number Turbulent Flows - Model Development and Validation," Comput. Fluids. **24**(3), pp. 227–238.

[48] Durbin, P., and Pettersson Reif, B., 2001, *Statistical Theory and Modeling for Turbulent Flows*, John Wiley & Sons Ltd., Chichester, England, Chap. 6.

[49] Kreyszig, E., 2011, *Advanced Engineering Mathematics, 10th Edition.*, John Wiley & Sons Ltd., Hoboken, N.J., USA, Chap. 19.

[50] Patankar, S. V. and Spalding, D.B., 1972, "A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows," Int. J. Heat Mass Transfer **15**(10), pp. 1787-1806.

[51] ANSYS FLUENT Theory Guide, 2012, ANSYS Inc., Canonsburg, PA, USA.

[52] ANSYS FLUENT User's Guide, 2012, ANSYS Inc., Canonsburg, PA, USA.

[53] Arghode, V., Sundaralingam, V., Joshi, Y., and Phelps, W., 2013, "Thermal Characteristics of Open and Contained Data Center Cold Aisle," J. Heat Trans., **135**(6), pp. 061901.

[54] Versteeg, H., and Malalasekera, W., 1995, *An Introduction to Computational Fluid Dynamics The Finite Volume Method*, Pearson Education Ltd., Harlow, England, Chap. 3.

[55] Arghode, V., Kumar, P., Joshi, Y., Weiss, T., and Meyer, G., 2013, “Rack Level Modeling of Air Flow Through Perforated Tile in a Data Center,” J. Electron. Packag. **135**(3), pp. 030902.

[56] Arghode, V., and Joshi, Y., 2014, “Room Level Modeling of Air Flow in a Contained Data Center Aisle,” J. Electron. Packag. **136**(1), pp. 011011.

[57] Arghode, V., and Joshi, Y., 2014, “Rapid Modeling of Air Flow through Perforated Tiles in a Raised Floor Data Center,” Proceedings of the Inter Society Conference on Thermal Phenomena (ITherm), May 27-30, Orlando, Florida, USA.

[58] Cruz, E., and Joshi, Y., 2015, “Coupled inviscid-viscous solution method for bounded domains: Application to data-center thermal management,” Int. J. Heat Mass Transfer **85**, pp. 181-194.

[59] Arghode, V., and Joshi, Y., 2015, “Experimental Investigation of Air Flow Through a Perforated Tile in a Raised Floor Data Center,” J. Electron. Packag. **137**(1), pp. 011011.

[60] Ferziger, J. H., and Perić, M., 2002, *Computational Methods for Fluid Dynamics*, *3rd Edition*, Springer, Berlin, Germany, Chaps. 2, 5, and 9.

[61] Roache, P. J., 1994, “Perspective: A Method for Uniform Reporting of Grid Refinement Studies,” ASME J. Fluids Eng., **116**(3), pp. 405-413.

VITA

Ethan E. Cruz was born in Houston, Texas to Mr. and Mrs. F. Edward Cruz. He attended public schools, graduating in 1996 from Bowie High School in Austin, Texas. He then enrolled in the College of Engineering at The University of Texas at Austin, graduating in 2000 with a Bachelor of Science in Mechanical Engineering. In 2000, he began his career for the IBM Corporation, where he is a hardware development engineer designing cooling solutions for server and storage products. In 2006, while working full time for the IBM Corporation, he graduated with a Master of Science in Mechanical Engineering from Southern Methodist University. In 2007, he enrolled in the doctoral program in the George W. Woodruff School of Mechanical Engineering at the Georgia Institute of Technology. He currently holds seven patents and has authored 13 technical papers. He resides with his wife, Ryan, and two children, Tino and Taryn, in Austin, Texas. S.D.G.