3-24-2016

Liquid Fuel Film Cooling: a CFD Analysis With Hydrocarbon Fuel

Jacob D. Bills

Liquid Fuel Film Cooling:
A CFD Analysis with Hydrocarbon Fuel

THESIS

Jacob D. Bills, Captain, USAF

AFIT-ENY-MS-16-M-199

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENY-MS-16-M-199

LIQUID FUEL FILM COOLING:
A CFD ANALYSIS WITH HYDROCARBON FUEL

THESIS

Presented to the Faculty

The Department of Aeronautics and Astronautics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Aeronautical Engineering

Jacob D. Bills, B.S.A.E.

Captain, USAF

March, 2016

AFIT-ENY-MS-16-M-199

LIQUID FUEL FILM COOLING:

A CFD ANALYSIS WITH HYDROCARBON FUEL

THESIS

Jacob D. Bills, B.S.A.E.
Captain, USAF

Committee Membership:

Maj Darrell S. Crowe, PhD
Chair

Maj James L. Rutledge, PhD
Member

Dr. Edward B. Coy
Member

Abstract

Cooling of liquid rocket engine combustion chambers and nozzles is a critical component to liquid rocket engine design. The temperatures within the engine can exceed the melting temperature of the wall materials by such a large margin that generally two methods of cooling are required. A common one is regenerative cooling—circulating fuel through the walls prior to injection into the combustion chamber. A second cooling method, often used in conjunction with regenerative cooling, is liquid fuel film cooling. Liquid fuel is injected along the surface of the wall to act as a barrier against the core combustion gases. A numerical model is developed for simulating liquid fuel film cooling in a rocket engine using a hydrocarbon fuel. The model incorporates turbulent multiphase flow with species transport using the commercial ANSYS® Fluent CFD software. Conjugate heat transfer is simulated through walls containing embedded cooling channels. A novel User Defined Function is written to incorporate heterogeneous chemical reactions between the liquid fuel and the freestream gases. Comparisons are made between simulations with and without the heterogeneous reactions as well as with simplifications to the cooling channel geometry. It is found that simplifications to the cooling geometry can artificially reduce cooling performance while neglecting the heterogeneous reactions can artificially increase cooling performance.

Acknowledgements

I would like to thank my adviser and co-adviser, Maj Darrell Crowe and Maj James Rutledge, for their help and guidance through this research. I would also like to thank the sponsor of this work, and my committee member, Dr. Edward Coy. Finally, I want to thank my wife for her never-ending support through these last 18 months. Her unending understanding of my working long days and weekends, even with the arrival of our son, was pivotal to my ability in accomplishing this project.

Jacob D. Bills

Table of Contents

Page

Page

List of Acronyms

SST Shear Stress Transport 24

TC Test Case 73

UDF User Defined Function 43

VoF Volume of Fluid 28

List of Symbols

List of Figures

List of Tables

LIQUID FUEL FILM COOLING:

A CFD ANALYSIS WITH HYDROCARBON FUEL

I. Introduction

From the very first rocket to reach space, a German V-2, to the launch of the first satellite, Sputnik, through today's modern space launch systems, Liquid Rocket Engines (LREs) have been the dominant mechanism used to reach orbit around the Earth. The reason for their use is due to the LRE's ability to provide the necessary thrust-to-weight ratio and performance required to lift payloads to orbit when compared to alternatives such as solid rocket motors. With the ability to provide the necessary performance for space launch comes great complexity in the design and operation of LREs. One of the main drivers of LRE design is cooling of the combustion chamber and nozzle walls. The adiabatic flame temperatures of common propellants can range from 2700 K to 4300 K [23]. This far exceeds the melting temperatures of common materials used to construct LRE combustion chambers and nozzles. These excessive temperatures necessitate the need for one or more active cooling systems.

Multiple methods for cooling LREs exist. These include, but are not limited to, regenerative, transpiration, ablative, and film cooling. The most common method used in LREs is regenerative cooling. Regenerative cooling circulates one of the propellants (generally the fuel) through channels within the nozzle and chamber walls. The circulating fuel cools the engine while also raising the fuel's temperature prior to injection into the combustion chamber, increasing the overall performance of the engine. Transpiration cooling involves the use of porous walls through which fuel seeps out, cooling the wall. Ablative cooling uses sacrificial materials on the walls

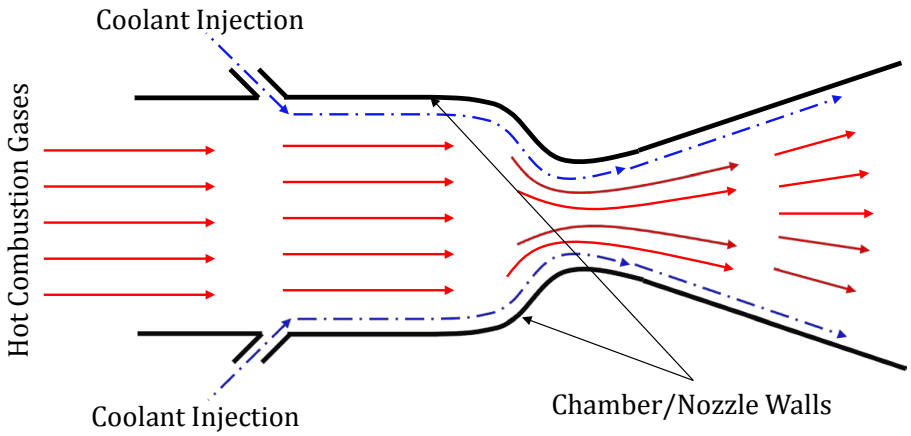

Figure 1. Overview of film cooling in a LRE.

designed to burn and flake off, absorbing and carrying away heat in the process. Film cooling, the subject of this thesis, is a method commonly used in conjunction with regenerative cooling [10]. A thin film of coolant (commonly propellant/fuel) is injected along the wall of the chamber and/or nozzle, shown in Figure 1. The solid red arrows represent the hot combustion gases flowing through the chamber and nozzle. The dashed blue lines show the injection of coolant along the wall, forming a thin film barrier between the walls and core gases. The film can be injected either through the side walls, as in Figure 1, or through injectors on the outer edges of the injector plate. This thin film can be in either a liquid or gaseous state and acts as a barrier between the hot gases in the core flow and the wall material. Liquid films have a benefit over gaseous due to the amount of heat that is absorbed as the liquid vaporizes and mixes with the core flow. In conjunction with the ability of liquid films to greatly reduce heat transfer to the walls, hydrocarbon fuels have another benefit when used for film cooling—their tendency to deposit carbon on the walls [10]. This layer of carbon has the potential to act as an insulator, further reducing heat transfer into the walls.

1.1 Motivation

Modern space launch systems tend to use Liquid Oxygen (LOX) and either Liquid Hydrogen (LH2) or a Liquid Hydrocarbon (LHC) fuel (e.g., kerosene or Rocket Propellant 1 (RP-1)) as their propellants. A LH2/LOX combination is generally considered to be an ideal propellant mixture for vehicle performance. However, due to the low density of LH2, it requires very large storage tanks and must be kept at cryogenic temperatures. While LHC fuels result in lower performance than LH2, they have benefits when it comes to storage and tank size. The less complex storage and smaller tank size required for LH2 fuels makes them of interest to the Air Force for use in future LRE designs. The Air Force Research Laboratory's (AFRL) Hydrocarbon Boost Technology Demonstrator (HBTD) program is investigating technologies to develop an affordable, high performing LRE for use with future launch systems [6]. Such a design would likely use Liquid Fuel Film Cooling (LFFC) as one of the cooling mechanisms.

Modeling and simulation (M&S) can be a critical part of the design process. With accurate M&S, designers can gain insight into problems that a physical experiment may be unable to provide due to either technological or cost limitations. Thus, reliable numerical models of LFFC behavior would help ensure a LRE design achieves the maximum performance possible. This is important since there is an inherent reduction in performance that results from the use of LFFC.

The fuel pumps on-board a launch vehicle will have some maximum mass flow rate ($\dot{m}$) limit. Some amount of $\dot{m}$ is required for LFFC. This reduces the overall amount of $\dot{m}$ available for the generation of thrust in the combustion chamber. Equation (1) shows that thrust (F), measured in N, and $\dot{m}$, measured in kg/s, are directly related through specific impulse (I_{sp}), measured in seconds, and the acceleration due to gravity (g_0), measured in m/s^2 [23]. Thus, a reduction in $\dot{m}$ of the propellants will

result in a reduction of available thrust. This reduction in thrust must be incorporated into a LRE design, or more fuel must be carried on-board and/or the size of the fuel pumps increased to compensate—resulting in an larger initial launch mass. In addition to the reduction in F, LFFC also results in a reduction of average exhaust temperature (T_1) and an increase in the average molecular weight ($\mathscr{M}$) of the exhaust. The combination of a reduction in T_1, measured in K, and an increase in $\mathscr{M}$, measured in kg/mol, reduces the averaged exhaust velocity (v_2), measured in m/s, shown in Equation (2), where γ is the ratio of specific heats, R_u is the universal gas constant [J/(mol·K)], and P_1 and P_2 are the pressures (P) in the combustion chamber and nozzle exit respectively, measured in Pa. A reduction in v_2 corresponds to a decrease in I_{sp} through Equation (3). The I_{sp} is a common parameter used to represent the efficiency of a rocket engine or motor. A higher I_{sp} corresponds to a higher efficiency thus less fuel is required to achieve the same thrust as an engine with a lower I_{sp} [23].

$$F = \dot{m} I_{sp} g_0 \tag{1}$$

$$v_2 = \left[\frac{2\gamma}{\gamma - 1} \frac{R_u T_1}{\mathscr{M}} \left(1 - \left[\frac{P_2}{P_1} \right]^{\frac{\gamma-1}{\gamma}} \right) \right]^{\frac{1}{2}} \tag{2}$$

$$I_{sp} = \frac{v_2}{g_0} \tag{3}$$

Because of this reduction in performance, LFFC needs to be specifically tuned to ensure there is enough $\dot{m}$ to adequately cool the walls—but not too much as to unnecessarily reduce overall engine performance due to the excess fuel being stripped away by the core flow. The tuning of the LFFC $\dot{m}$ is where accurate numerical models can give insight to an optimum LFFC design. For the study presented in this thesis, the numerical model is produced through Computational Fluid Dynamics (CFD)

simulations.

1.2 Research Objective & Investigative Question

The Air Force Research Laboratory has conducted two-dimensional (2D) CFD simulations of LFFC on a chamber wall in the past. When the results were compared with experiment, the CFD simulations showed an over-prediction of heat flux reduction to the wall at the fuel injection slot, while under-predicting the heat flux reduction downstream of the slot [8]. The primary objective of the CFD study presented in this thesis is to build upon what has been done by AFRL and create a new numerical model implementing the following physics:

1. Multiphase Flow
2. Heterogeneous Reactions
3. Conjugate heat transfer through stainless steel plate

The domain is based upon an experimental LFFC test setup being built at AFRL. An overview of the setup is shown in Figure 2. The hot gases (shown in red) flow through a 1-inch square channel of constant cross section, with a copper plate forming the upper surface. The plate is cooled by a series of water-cooled channels (shown in blue) along the backside of the plate. The bottom image is looking at the plate surface exposed to the hot gasses, with the dashed lines showing the locations of the water channels on the back side. The top image is a side-on view. The green arrows indicate where the fuel is injected into the free stream through a narrow slot.

The overarching investigative question associated with this work (not just the research presented in this document) is:
What are the driving parameters for heat transfer in the presence of LFFC?

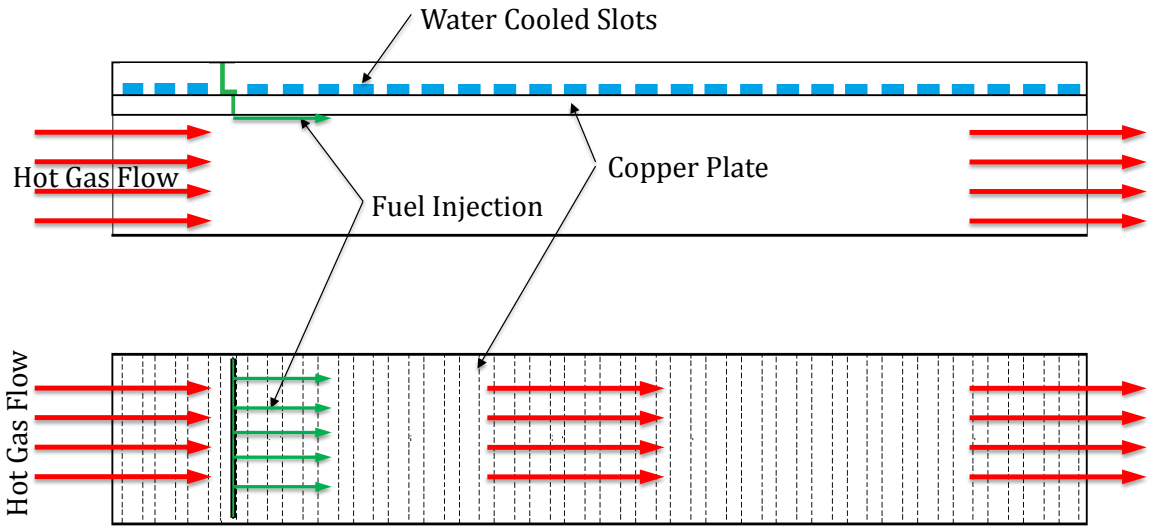

Figure 2. Overview of the AFRL LFFC experimental test setup

Answering this question will allow for more efficient LFFC designs using hydrocarbon fuels, directly supporting the design of more efficient LRE designs for future launch systems.

1.3 Summary

Chapter I introduced the topic of the present study including the underlying motivation, primary objective, and investigative questions. Chapter II presents the background and theory of the study. The background includes previous work done on LFFC in LREs with hydrocarbon fuels. Chapter II also discusses the fundamental theory of CFD simulations with the governing equations and various physical phenomena to be included in the numerical model such as conjugate heat transfer and multiphase flow. Chapter III presents the computational methodology used to develop the numerical model used for the study. The computational methodology covers grid generation, grid convergence, solution initialization, solution selection, solution convergence, and solver settings. Chapter IV presents the results of the grid convergence study and final test cases. Chapter V presents a summary of the conclusions

from the results in Chapter IV along with suggestions for future work to better answer the investigative questions.

II. Background & Theory

2.1 Previous Work

In the realm of film cooling, the majority of research efforts are directed towards gas turbine engines. There is extensive literature on the use of film cooling for turbine blades. These blades are located downstream of the combustion chamber and must be kept sufficiently cool to operate. The film coolant is generally air siphoned off of the compressor. This air is capable of cooling the blades, as the temperatures and pressures are not as extreme as those found in a LRE. The volume of research for film cooling in LREs is more limited. Of this research, studies on liquid film cooling using hydrocarbon fuels at supercritical pressures are few [8].

While general research on film cooling in LREs is not as vast as film cooling in other areas such as gas turbine engines, there has been interest in liquid film cooling dating back to the 1950s and 1960s [7, 13]. However, many of the earlier studies neglected certain phenomena that are significant in a realistic LRE combustion chamber such as radiation, turbulence, or combustion of the liquid film [20].

A summary of general film cooling CFD studies is found in reference [15], with studies up through 2004. The authors note that there are some concerns with much of the previous work. For example, many of the studies relied on dissipative numerical schemes, which were not second-order accurate. The studies also relied on wall functions for turbulence modeling and grid convergence studies were not always accomplished, due to cost/computing limitations. In addition to these concerns, CFD studies for film cooling in LREs are further hampered by the lack of reliable experimental data. As recently as 2009, Arnold et al. noted a lack of experimental studies of film cooling in LREs operating at realistic conditions [2]. While some experimental studies have occurred since 2009, more are still needed.

Some of the studies conducted recently investigated liquid film cooling in rocket combustion chambers. Zhang et al. conducted a numerical study using a gas/water system, where they studied the effects on the liquid film length [32]. They demonstrated good agreement between experiment and numerical simulation. However, their comparisons were limited, due to the noted lack of available experimental data.

An experimental study was conducted by Kirchberger et al. [12] on film cooling in a hydrocarbon/GOX (gaseous oxygen) combustion chamber. The experimental setup was capable of replicating realistic operating conditions for a LRE. The goal of the research was to investigate what variables affect film cooling efficiency and understanding the mechanisms of heat transfer within the combustion chamber. Numerical analysis was performed for comparison. This analysis achieved some success with modeling the film cooling sufficiently far downstream but not near the film coolant injection location.

Yang and Sun have conducted two Three-Dimensional (3D), steady, numerical simulations of liquid film cooling in a LOX/RP-1 engine using time-averaged Navier-Stokes equations and the standard κ-ϵ turbulence model. The first study [30] included effects of RP-1 chemical combustion through the use of a 17-step reaction table. They found that in addition to the $\dot{m}$ of the coolant, the geometry of the film coolant hole affects the cooling effectiveness. Their model did not include conjugate heat transfer through the wall or radiation from the core gases. No comparison with experimental results was made—potentially due to the lack of experimental data. The second study [31] investigated the coupled heat transfer between film cooling and regenerative cooling. Comparison with experimental results, which used nitrogen tetroxide and hydrazine/monomethyl-hydrazine, were performed with a $< 7\%$ discrepancy. Their study investigated the use of two film coolant slots for injection. It was found that the use of two slots produced more effective cooling, even when the total amount of

$\dot{m}$ for the coolant remained the same.

Himansu et al. performed research using CFD simulations of LFFC using hydrocarbon fuels in 2D [8]. The results demonstrated large differences between the amount of heat flux through the wall when compared with experimental results at similar operating conditions, which are also detailed in Reference [8]. These simulations did not include some physics, which could partially explain the differences with experimental results. For example, the formation of soot on the wall was not simulated. Instead, the layer of coke on the wall was modeled using a constant thickness. The exact thickness was determined using an iterative process of matching results with experiment. Also, liquid fuel was instead simulated as a dense gas, and the geometry of the cooling channels was not included.

2.2 Key Parameters

The results and analysis presented later in this paper use multiple key parameters. These key parameters are comprised of, but are not limited to, the thermal conductivity (k), density (ρ), and specific heat capacity at constant pressure (c_p), velocity (v), dynamic viscosity (μ), the mass flow rate ($\dot{m}$), the kinematic viscosity (ν), and chemical composition. The following is a list of the key parameters used in this study, along with their definitions.

Reynolds number (Re): The ratio of inertial forces to viscous forces is known as the Reynolds number. This non-dimensional value is used to compare similar flow phenomena across different physical circumstances.

$$Re \equiv \frac{\rho u l}{\mu} \tag{4}$$

Prandtl number (Pr): The ratio of viscous diffusion to thermal diffusion rates in

a fluid. A small Pr indicates a thermal gradient will diffuse through the fluid faster than a velocity gradient.

$$Pr \equiv \frac{\nu}{\alpha} = \frac{c_p \mu}{k} \tag{5}$$

Nusselt number (Nu): The ratio of convective to conductive heat transfer across a fluid/solid boundary (in the normal direction) is the Nusselt number. A large Nu implies convection dominates, while conversely a small Nu implies conduction is the dominant factor of heat transfer within the fluid.

$$Nu \equiv \frac{hl}{k_f} \tag{6}$$

Blowing ratio (M): The ratio of the film coolant mass flux to the freestream mass flux.

$$M \equiv \frac{\rho_f u_f}{\rho_\infty u_\infty} \tag{7}$$

Density ratio (DR): The density ratio is simply the ratio of the fuel and freestream densities.

$$DR \equiv \frac{\rho_f}{\rho_\infty} \tag{8}$$

Momentum flux ratio (I): The ratio of the film coolant and freestream kinetic energies is the momentum ratio. Note, the momentum ratio, density ratio, and blowing ratio are all interrelated through velocity and density.

$$I \equiv \frac{\rho_f u_f^2}{\rho_\infty u_\infty^2} \tag{9}$$

Overall Effectiveness (Φ): The measure of how effective the film coolant is at reducing the wall temperature is known as the overall effectiveness. The values of Φ range from zero to one. A value of one represents an ideal situation where the wall temperature is reduced to the temperature of the coolant. A Φ of zero corresponds

to no reduction in wall temperature from the freestream value.

$$\Phi \equiv \frac{T_\infty - T_s}{T_\infty - T_c} \tag{10}$$

2.3 The Governing Equations

The basis of CFD is to numerically simulate and track changes in the physical properties of a fluid flow. To achieve this goal of accurately simulating a flow's behavior, CFD codes rely on the continuum assumption and a set of governing equations. The continuum assumption assumes a fluid is a continuous medium and is valid if the Knudsen number (Kn) is $\ll$ 1. The Knudsen number is found using the mean free path (λ_m) of a molecule in the fluid and a length scale (l).

$$Kn \equiv \frac{\lambda_m}{l} \tag{11}$$

The governing equations are also known as the conservation laws. In the context of a fluid flow, the conservation laws are collectively referred to as the Navier-Stokes equations. These equations are generally represented in one of two ways: differential or integral form. The integral form of the governing equations is commonly used in many CFD codes, via the finite volume method, due to their generality. The basic form can be written for the change of some scalar quantity (η) with respect to time (t) as

$$\frac{\partial}{\partial t}\int_V \rho\eta \,\mathrm{d}V + \oint_S [\rho\eta(\vec{v}\cdot\vec{n}) - \alpha\rho(\nabla\eta\cdot\vec{n})]\,\mathrm{d}S = \int_V Q_V \,\mathrm{d}V + \oint_S (\vec{Q}_S\cdot\vec{n})\,\mathrm{d}S \tag{12}$$

Where η is the intensive version of an extensive property N. An extensive property is dependent upon mass while an intensive property is independent of mass. The control volume is represented by V, $\vec{n}$ is the unit normal vector, α is the thermal

diffusivity, ρ is the density, and Q_V and Q_S are source terms. The change of η in control volume $\mathbb{V}$ over time is

$$\frac{\partial}{\partial t}\int_{\mathbb{V}} \rho\eta \,\mathrm{d}\mathbb{V} \tag{13}$$

The amount of η entering or leaving $\mathbb{V}$ through the surface with some velocity $\vec{v}$, called the convective flux, is

$$\oint_S \rho\eta(\vec{v}\cdot\vec{n})\,\mathrm{d}S \tag{14}$$

The diffusive flux is

$$\oint_S \alpha\rho(\nabla\eta\cdot\vec{n})\,\mathrm{d}S \tag{15}$$

And finally, any sources of η within the volume or on the surface are represented as

$$\int_{\mathbb{V}} Q_V \,\mathrm{d}\mathbb{V} + \oint_S (\vec{Q}_S\cdot\vec{n})\,\mathrm{d}S \tag{16}$$

Equation (12) can also be rewritten for any vector quantity $\vec{\eta}$. As a result, the convective and diffusive fluxes and the surface source term become tensors.

$$\frac{\partial}{\partial t}\int_{\mathbb{V}} \rho\vec{\eta}\,\mathrm{d}\mathbb{V} + \oint_S [\rho\vec{\eta}(\vec{v}\cdot\vec{n}) - \overline{\overline{F}}_D\cdot\vec{n}]\,\mathrm{d}S = \int_{\mathbb{V}} \vec{Q}_V\, d\mathbb{V} + \oint_S (\overline{\overline{Q}}_S\cdot\vec{n})\,\mathrm{d}S \tag{17}$$

The selection of the property η, or a corresponding N, can transform Equation (17) into any one of the three governing equations: mass (continuity), momentum, and energy. Note: the derivations above and through Section 2.3.6 may be found in Reference [4].

2.3.1 Continuity Equation

The selection of mass (m) for N yields the first governing equation, the continuity equation. In this equation, the intensive property η reduces to 1. A fluid at rest would would have no velocity, thus no mass would be exchanged. This results in Equation

(15) equaling zero. As mass cannot be produced or destroyed, the two source terms in Equation (16) go to zero. This yields the final form of the continuity equation as

$$\frac{\partial}{\partial t}\int_V \rho \, \mathrm{d}V + \oint_S \rho(\vec{v}\cdot\hat{n}) \, \mathrm{d}S = 0 \tag{18}$$

2.3.2 Momentum Equation

The second governing equation is the conservation of momentum with $N = m\vec{v}$ and $\eta = \vec{v}$. The diffusive flux term remains zero, while the source terms in Equation (16) take special forms. From Newton's second law, the time rate of change of momentum is equal to the sum of the forces. This results in the source terms being represented by different forces acting on the fluid. The $\vec{Q}_V$ term becomes the sum of the body forces ($\vec{f_b}$) denoted as $\vec{F}_V$,

$$\vec{F}_V = \int_V \rho \vec{f_b} \, \mathrm{d}V \tag{19}$$

The surface forces ($\overline{\overline{Q}}_S$) become the combination of pressure and viscous forces acting on the boundaries of the control volume,

$$\overline{\overline{Q}}_S = -P\overline{\overline{I}} + \overline{\overline{\tau}} \tag{20}$$

where P is pressure, $\overline{\overline{I}}$ is the unit tensor, and $\overline{\overline{\tau}}$ is the viscous stress tensor. Plugging in $m\vec{v}$ for N and $\vec{v}$ for η into the left hand side of Equation (17), with the above expressions for the surface and body forces on the right hand side yields the momentum equation,

$$\frac{\partial}{\partial t}\int_V \rho\vec{v} \, \mathrm{d}V + \oint_S \rho\vec{v}(\vec{v}\cdot\hat{n}) \, \mathrm{d}S = \int_V \rho\vec{f_b} \, \mathrm{d}V - \oint_S P\hat{n} \, \mathrm{d}S + \oint_S (\overline{\overline{\tau}}\cdot\hat{n}) \, \mathrm{d}S \tag{21}$$

2.3.3 Energy Equation

The first law of thermodynamics states that the change in total energy (E) of a system is equal to the heat (Q) added to the system and the work (W) done on the system.

$$dE = \delta Q + \delta W \tag{22}$$

In the context of Equation (17), the intensive and extensive terms become $N = me + \frac{1}{2}m|v|^2$ and $\eta = e + \frac{1}{2}|v|^2$ with e representing the specific energy. Using these terms in Equation (17) and the right hand side of Equation (22) gives,

$$\frac{\partial}{\partial t}\int_V \rho\left(\ e + \frac{1}{2}|v|^2\ \right)\mathrm{d}V + \oint_S \rho\left(\ e + \frac{1}{2}|v|^2\ \right)(\vec{v}\cdot\hat{n})\,\mathrm{d}S = \frac{dQ}{dt} + \frac{dW}{dt} \tag{23}$$

The first term on the right hand side represents the rate of heat change done on the body through phenomena such as combustion, conduction, and radiation. This can be expressed as a combination of Fourier's law and a heat generation term ($\dot{q}_b$)

$$\frac{dQ}{dt} = \oint_S k(\nabla T\cdot\hat{n})\,\mathrm{d}S + \int_V \rho\dot{q}_b\,\mathrm{d}V \tag{24}$$

The $\dfrac{dW}{dt}$ term can be expressed by the work done by pressure, viscous, and body forces

$$\frac{dW}{dt} = \oint_S (-P(\vec{v}\cdot\hat{n}) + (\overline{\overline{\tau}}\cdot\vec{v})\cdot\vec{n})\ \mathrm{d}S + \int_V \rho\vec{f_b}\ \mathrm{d}V \tag{25}$$

Combining Equations (23), (24), and (25) yields the final form of the energy equation:

$$\begin{aligned}\frac{\partial}{\partial t}\int_V \rho\left(\ e + \frac{1}{2}|v|^2\ \right)\mathrm{d}V + \oint_S \rho\left(\ e + \frac{1}{2}|v|^2\ \right)(\vec{v}\cdot\hat{n})\,\mathrm{d}S = \oint_S k(\nabla T\cdot\hat{n})\,\mathrm{d}S + \\ \int_V \rho\dot{q}_b\,\mathrm{d}V - \oint_S P(\vec{v}\cdot\hat{n})\,\mathrm{d}S + \int_V (\overline{\overline{\tau}}\cdot\vec{v})\cdot\vec{n}\ \mathrm{d}S + \int_V \rho\vec{f_b}\ \mathrm{d}V\end{aligned} \tag{26}$$

2.3.4 Viscous Stress Tensor

The viscous stress tensor ($\overline{\overline{\tau}}$) has remained undefined so far. It is defined as

$$\overline{\overline{\tau}} = \begin{bmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \tau_{zz} \end{bmatrix} \tag{27}$$

The individual stress components are represented using a τ_{ij} notation indicating the component of stress is in the j direction on the i face of a fluid element. Assuming a Newtonian fluid, the viscous stress tensor is symmetric, with the diagonal entries being normal stresses. For Newtonian fluids and using index notation (Reference [27]), τ_{ij} is defined as

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) + \delta_{ij} \lambda \partial_i u_i \tag{28}$$

In the above equation λ is known as the second viscosity coefficient. The second viscosity coefficient is generally eliminated through use of the Stokes hypothesis, which relates λ and μ in the following manner:

$$\lambda + \frac{2}{3}\mu = 0 \tag{29}$$

Substituting this relation for λ and μ into Equation (28) yields the following

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3}\mu \delta_{ij} \partial_i u_i \tag{30}$$

Using the final form of the viscous stress tensor for a Newtonian fluid, along with the governing equations shown previously, it is possible to express the complete set of the Navier-Stokes Equations in a form commonly used in CFD.

2.3.5 Complete Navier-Stokes Equations

Collecting like terms in Equations (18), (21), and (26), results in

$$\frac{\partial}{\partial t}\int_V \vec{W}\,\mathrm{d}V + \oint_S (\vec{F}_c - \vec{F}_v)\,\mathrm{d}S = \int_V \vec{Q}\,\mathrm{d}V \tag{31}$$

This system of equations is collectively called the vector form of the Navier-Stokes Equations [4]. Each term consists of five components, one from the continuity equation, three from the momentum equation, and one from the energy equation. The first term ($\vec{W}$) is the vector of conserved variables

$$\vec{W} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{bmatrix} \tag{32}$$

The second term is the vector of convective fluxes

$$\vec{F}_c = \begin{bmatrix} \rho\tilde{U} \\ \rho u\tilde{U} + n_x p \\ \rho v\tilde{U} + n_y p \\ \rho w\tilde{U} + n_z p \\ \rho H\tilde{U} \end{bmatrix} \tag{33}$$

with $\tilde{U}$ being the contravariant velocity (velocity normal to a surface element) defined

as $(\vec{v} \cdot \vec{n})$, and H being the total enthalpy. The third term is the viscous flux vector

$$\vec{F}_v = \begin{bmatrix} 0 \\ n_x\tau_{xx} + n_y\tau_{xy} + n_z\tau_{xz} \\ n_x\tau_{yx} + n_y\tau_{yy} + n_z\tau_{yz} \\ n_x\tau_{zx} + n_y\tau_{zy} + n_z\tau_{zz} \\ n_x\Theta_x + n_y\Theta_y + n_z\Theta_z \end{bmatrix} \tag{34}$$

where the Θ terms are

$$\begin{aligned} \Theta_x &= u\tau_{xx} + v\tau_{xy} + w\tau_{xz} + k\frac{\partial T}{\partial x} \\ \Theta_y &= u\tau_{yx} + v\tau_{yy} + w\tau_{yz} + k\frac{\partial T}{\partial y} \\ \Theta_z &= u\tau_{zx} + v\tau_{zy} + w\tau_{zz} + k\frac{\partial T}{\partial z} \end{aligned} \tag{35}$$

The final term, $\vec{Q}$, represents the sources

$$\vec{Q} = \begin{bmatrix} 0 \\ \rho f_{e,x} \\ \rho f_{e,y} \\ \rho f_{e,z} \\ \rho \vec{f_b} \cdot \vec{v} + \dot{q}_b \end{bmatrix} \tag{36}$$

The form of the Navier-Stokes equations presented here still requires special treatment if turbulence or species transport is to be considered. Species transport requires extra transport equations and is covered in the following section. The integration of turbulence into the equations may be done by two methods: Reynolds-averaging and Favre-averaging which are covered in the section following species transport (Section 2.3.7).

2.3.6 Species Transport

For situations that involve physics such as real gas effects or chemical reactions, additional terms are required to account for species transport. Species transport introduces an additional n-1 transport equations for n species [4]. This results in the the vector of conserved variables becoming

$$\vec{W} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \\ \rho Y_1 \\ \vdots \\ \rho Y_{n-1} \end{bmatrix} \tag{37}$$

The vector of convective fluxes becomes

$$\vec{F}_c = \begin{bmatrix} \rho\tilde{U} \\ \rho u\tilde{U} + n_x p \\ \rho v\tilde{U} + n_y p \\ \rho w\tilde{U} + n_z p \\ \rho H\tilde{U} \\ \rho Y_1\tilde{U} \\ \vdots \\ \rho Y_{n-1}\tilde{U} \end{bmatrix} \tag{38}$$

The vector of convective fluxes becomes

$$\vec{F}_v = \begin{bmatrix} 0 \\ n_x\tau_{xx} + n_y\tau_{xy} + n_z\tau_{xz} \\ n_x\tau_{yx} + n_y\tau_{yy} + n_z\tau_{yz} \\ n_x\tau_{zx} + n_y\tau_{zy} + n_z\tau_{zz} \\ n_x\Theta_x + n_y\Theta_y + n_z\Theta_z \\ n_x\Upsilon_{x,1} + n_y\Upsilon_{y,1} + n_z\Upsilon_{z,1} \\ \vdots \\ n_x\Upsilon_{x,n-1} + n_y\Upsilon_{y,n-1} + n_z\Upsilon_{z,n-1} \end{bmatrix} \tag{39}$$

where the Θ and Υ terms are

$$\begin{aligned} \Theta_x &= u\tau_{xx} + v\tau_{xy} + w\tau_{xz} + k\frac{\partial T}{\partial x} + \rho\sum_{m=1}^{n} h_m D_m \frac{\partial Y_m}{\partial x} \\ \Theta_y &= u\tau_{yx} + v\tau_{yy} + w\tau_{yz} + k\frac{\partial T}{\partial y} + \rho\sum_{m=1}^{n} h_m D_m \frac{\partial Y_m}{\partial y} \\ \Theta_z &= u\tau_{zx} + v\tau_{zy} + w\tau_{zz} + k\frac{\partial T}{\partial z} + \rho\sum_{m=1}^{n} h_m D_m \frac{\partial Y_m}{\partial z} \end{aligned} \tag{40}$$

$$\begin{aligned} \Upsilon_{x,m} &= \rho D_m \frac{\partial Y_m}{\partial x} \\ \Upsilon_{y,m} &= \rho D_m \frac{\partial Y_m}{\partial y} \\ \Upsilon_{z,m} &= \rho D_m \frac{\partial Y_m}{\partial z} \end{aligned}$$

The vector of source terms becomes

$$\vec{Q} = \begin{bmatrix} 0 \\ \rho f_{e,x} \\ \rho f_{e,y} \\ \rho f_{e,z} \\ \rho \vec{f_b} \cdot \vec{v} + \dot{q}_b \\ \dot{s}_1 \\ \vdots \\ \dot{s}_{n-1} \end{bmatrix} \tag{41}$$

The mass fraction is Y_m, h_m is the enthalpy, D_m is the effective binary diffusivity, and $\dot{s}_m$ is the rate of change of each species m due to chemical reactions [4].

2.3.7 Favre and Reynolds-Averaged Navier-Stokes Equations

The addition of turbulence to the Navier-Stokes equations is commonly achieved through a method known as Reynolds-averaging. This method breaks up the variables into a time-averaged mean and fluctuating component shown in Equation (42). The mean components are denoted with a bar while the fluctuating components are denoted with an apostrophe.

$$u = \bar{u} + u' \tag{42}$$

This time averaging decomposition is imposed on all the variables in the Navier-Stokes equations. When the results are substituted back into the equations simplifications can be made. For example, the average of a fluctuating component is zero ($\overline{u'} = 0$). The resulting equations are known as the Reynolds-Averaged Navier-Stokes (RANS) equations [28]. For compressible flows the Navier-Stokes equations can be simplified through a combination of Reynolds-averaging and another technique known as Favre-

averaging. Favre-averaging uses a mass-average in place of a time-average. The mean mass-averaged components are denoted with a curly bar while the fluctuating components are denoted with a double apostrophe, both shown in Equation (43). Decomposing the variables in this manner and substituting back into the Navier-Stokes equations allows for similar simplifications as in the Reynolds-averaging case where some terms will go to zero while others will simplify (e.g. $\overline{\rho u_i} = \bar{\rho}\tilde{u}_i$) [28].

$$u = \tilde{u} + u'' \tag{43}$$

The combination of Reynolds-averaging for density and Favre-averaging for velocity and temperature produces the Favre and Reynolds-Averaged Navier-Stokes equations shown below in differential form with index notation [28].

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial}{\partial x_i}(\bar{\rho}\tilde{u}_i) = 0 \tag{44}$$

$$\frac{\partial}{\partial t}(\bar{\rho}\tilde{u}_i) + \frac{\partial}{\partial x_j}(\bar{\rho}\tilde{u}_j\tilde{u}_i) = -\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j}(\bar{\tau}_{ji} - \bar{\rho}\overline{u_i''u_j''}) \tag{45}$$

$$\frac{\partial}{\partial t}(\bar{\rho}\tilde{E}) + \frac{\partial}{\partial x_j}(\bar{\rho}\tilde{u}_j\tilde{H}) = \frac{\partial}{\partial x_j}(k\tfrac{\partial T}{\partial x_j} - \overline{\rho u_j''h''} + \overline{\tau_{ji}u_i''} - \overline{\rho u_j''\tfrac{1}{2}u_i''u_i''}) + \frac{\partial}{\partial x_j}[\tilde{u}_i(\bar{\tau}_{ij} - \bar{\rho}\overline{u_i''u_j''})] \tag{46}$$

In addition to the above three equations, an equation is required to solve for the turbulent kinetic energy (κ), Equation (47). The turbulent kinetic energy comes out of the Favre-averaged Reynolds-stress tensor ($-\bar{\rho}\overline{u_i''u_j''}$) which introduces six additional terms that require closure [4].

$$\frac{\partial}{\partial t}(\bar{\rho}\kappa) + \frac{\partial}{\partial x_j}(\bar{\rho}\tilde{u}_j\kappa) =$$
$$-\bar{\rho}\overline{u_i''u_j''}\frac{\partial \tilde{u}_i}{\partial x_j} - \overline{\tau_{ji}\tfrac{1}{2}(\tfrac{\partial u_i''}{\partial x_j} + \tfrac{\partial u_j}{\partial x_i})} + \frac{\partial}{\partial x_j}(\overline{\tau_{ji}u_i''} - \overline{\rho u_j''\tfrac{1}{2}u_i''u_i''} - \overline{p'u_j''}) - \overline{u_i''}\frac{\partial P}{\partial x_i} + \overline{P'\frac{\partial u_i''}{\partial x_i}} \tag{47}$$

The pressure term (P) is solved using

$$P = \bar{\rho}\frac{R_u}{\mathscr{M}}\tilde{T} \tag{48}$$

While the total energy and enthalpy, $\tilde{E}$ and $\tilde{H}$ respectively, are solved using

$$\begin{aligned} \tilde{E} &= \tilde{e} + \tfrac{1}{2}\tilde{u}_i\tilde{u}_i + \kappa \\ \tilde{H} &= \tilde{h} + \tfrac{1}{2}\tilde{u}_i\tilde{u}_i + \kappa \end{aligned} \tag{49}$$

Equations (44) through (47) require additional equations for closure. Multiple methods exist for closure and are generally separated into zero-, one-, and two-equation models (higher order models also exist). To achieve closure it is common to ignore the molecular diffusion and turbulent transport of κ ($\frac{\partial}{\partial x_j}[\overline{\tau_{ij}u_i''}]$ and $\frac{\partial}{\partial x_j}[\bar{\rho}\overline{\kappa u_i''}]$) while modeling the Reynolds-stress tensor and the turbulent heat-flux vector. The Reynolds-stress tensor is modeled using the Boussinesq approximation based on a parameter labeled the eddy viscosity (μ_T) and is shown in Equation (50). The turbulent heat-flux vector uses another parameter, the turbulent Prandtl number (Pr_T), in addition to the eddy viscosity and is shown in Equation (51) [28]. The turbulent Prandtl number is commonly taken to be a constant (e.g. 0.9 for air) [4].

$$\begin{aligned} \tau_{ij}^F - \overline{\rho u_i'' u_j''} &= 2\mu_T\left(\tilde{S}_{ij} - \frac{1}{3}\frac{\partial \tilde{u}_k}{\partial x_k}\delta_{ij} \right) - \frac{2}{3}\bar{\rho}\kappa\delta_{ij} \\ \tilde{S}_{ij} &= \frac{1}{2}\left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} \right) \end{aligned} \tag{50}$$

$$\overline{\rho u_j'' h''} = -\frac{\mu_T}{Pr_T}\frac{\partial \tilde{h}}{\partial x_j} \tag{51}$$

There are two common two-equation methods used to model the eddy viscosity. The first is the κ-ϵ model. The κ-ϵ model uses equations for the turbulent kinetic

energy (κ) and the turbulent dissipation rate (ϵ). A drawback of the κ-ϵ model is the requirement of a damping function near walls to retain accuracy in the viscous sublayer [4]. The κ-ϵ model also suffers in regions where an adverse pressure gradient exists [4]. The second two-equation model is the κ-ω model based on the turbulent kinetic energy (as in the κ-ϵ model) and specific dissipation rate (ω). The κ-ω model requires no damping function but is very sensitive to the value of ω in the freestream [4]. There exists a third two-equation model that blends the κ-ϵ and κ-ω models into one, known as the κ-ω Shear Stress Transport (SST) model [4].

2.3.8 The κ-ω SST Turbulence Model

The κ-ω SST model attempts to take the best features of the κ-ϵ and κ-ω models and combine them into one model. The κ-ω model is applied near the wall as it requires no damping function. The κ-ω model is also superior to the κ-ϵ model in this region for compressible flow and adverse pressure gradients. Sufficiently far from the wall, in the wake region, and in free shear layers the κ-ϵ model is used. The formulation of the κ-ω SST model presented in this section may be found in References [4] and [14].

The turbulent kinetic energy and specific dissipation transport equations are formulated as

$$\begin{aligned}
&\frac{\partial \rho\kappa}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_j \kappa) = \frac{\partial}{\partial x_j}\left[\ (\mu_L + \sigma_\kappa \mu_T)\frac{\partial \kappa}{\partial x_j}\ \right] + \tau_{ij}^F S_{ij} - \beta^* \rho\omega\kappa \\
&\frac{\partial \rho\omega}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_j \omega) = \\
&\frac{\partial}{\partial x_j}\left[(\mu_L + \sigma_\omega \mu_T)\frac{\partial \omega}{\partial x_j}\right] + \frac{C_\omega \rho}{\mu_T}\tau_{ij}^F S_{ij} - \beta\rho\omega^2 + 2(1-f_1)\frac{\rho\sigma_{\omega 2}}{\omega}\frac{\partial \kappa}{\partial x_j}\frac{\partial \omega}{\partial x_j}
\end{aligned} \tag{52}$$

The turbulent eddy viscosity is modeled as

$$\mu_T = \frac{a_1 \rho \kappa}{max(a_1 \omega, f_2 \|curl(\vec{v})\|_2)} \tag{53}$$

The parameter f_1 in Equation (52) is a blending function for the coefficients of turbulence model (κ-ω) in the boundary layer and turbulence model (κ-ϵ) in the free shear layer and freestream. The value of f_1 is calculated using the following, with d_w being the distance to the wall and arg_1 and $CD_{\kappa\omega}$ being intermediate equations:

$$\begin{gathered} f_1 = tanh(arg_1^4) \\ arg_1 = min\left[max\left(\frac{\sqrt{\kappa}}{0.09\omega d_w}, \frac{500\mu_L}{\rho\omega {d_w}^2} \right), \frac{4\rho\sigma_{\omega_2}\kappa}{CD_{\kappa\omega} d^2} \right] \\ CD_{\kappa\omega} = max\left(2\frac{\rho\sigma_{\omega 2}}{\omega}\frac{\partial\kappa}{\partial x_j}\frac{\partial\omega}{\partial x_j}, 10^{-20} \right) \end{gathered} \tag{54}$$

The value for f_2 appearing in the eddy viscosity, Equation (53), is found through (arg_2 is another intermediate equation):

$$\begin{gathered} f_2 = tanh(arg_2^2) \\ arg_2 = max\left(\frac{2\sqrt{\kappa}}{0.09\omega d_w}, \frac{500\mu_L}{\rho\omega {d_w}^2} \right) \end{gathered} \tag{55}$$

The constants a_1, β^*, and κ_2 are:

$$a_1 = 0.31, \quad \beta^* = 0.09, \quad \kappa_2 = 0.41 \tag{56}$$

The remaining coefficients in the SST model (β, C_ω, σ_κ, σ_ω) are found by blending the κ-ω and κ-ϵ model coefficients, ϕ_1 and ϕ_2 respectively, using the following correlation for each coefficient.

$$\phi = f_1\phi_1 + (1 - f_1)\phi_2 \tag{57}$$

The coefficients used in place of ϕ_1 (κ-ω) are

$$\begin{gathered}\sigma_{\kappa 1} = 0.85, \quad \sigma_{\omega 1} = 0.5, \quad \beta_1 = 0.075, \\ C_{\omega 1} = \beta_1/\beta^* - \sigma_{\omega 1}\kappa_2^2/\sqrt{\beta^*} = 0.533\end{gathered} \tag{58}$$

And finally the coefficients used in place of ϕ_2 (κ-ϵ) are

$$\begin{gathered}\sigma_{\kappa 2} = 1.0, \quad \sigma_{\omega 2} = 0.856, \quad \beta_2 = 0.0828, \\ C_{\omega 2} = \beta_2/\beta^* - \sigma_{\omega 2}\kappa_2^2/\sqrt{\beta^*} = 0.440\end{gathered} \tag{59}$$

2.4 Liquid Film Cooling Heat Transfer Analysis

A heat transfer analysis of Liquid Film Cooling (LFC) can be presented using a control volume and the conservation of energy. Figure 3 presents a simplified LFC scenario within a control volume. The film enters the control volume at some mass flow rate and temperature. Heat flows into the control volume due to radiation and convection. The film absorbs energy through heating. The heat which is not absorbed by the film is then transferred to the wall to be removed by conduction. In LREs the fuel is typically used as the film coolant to avoid oxidation issues. Assuming the film, or fuel, enters as a liquid and fully vaporizes within the control volume this analysis can be represented mathematically with Equation 60 [9].

$$(q_{rad} + q_{conv})_{in} = q_{cond} + \dot{m}_f \left(\int_{T_{vap}}^{T_{f,out}} c_{p,vap}\, \mathrm{d}T + h_{fg} + c_{p,f}[T_{vap} - T_{f,in}] \right) \tag{60}$$

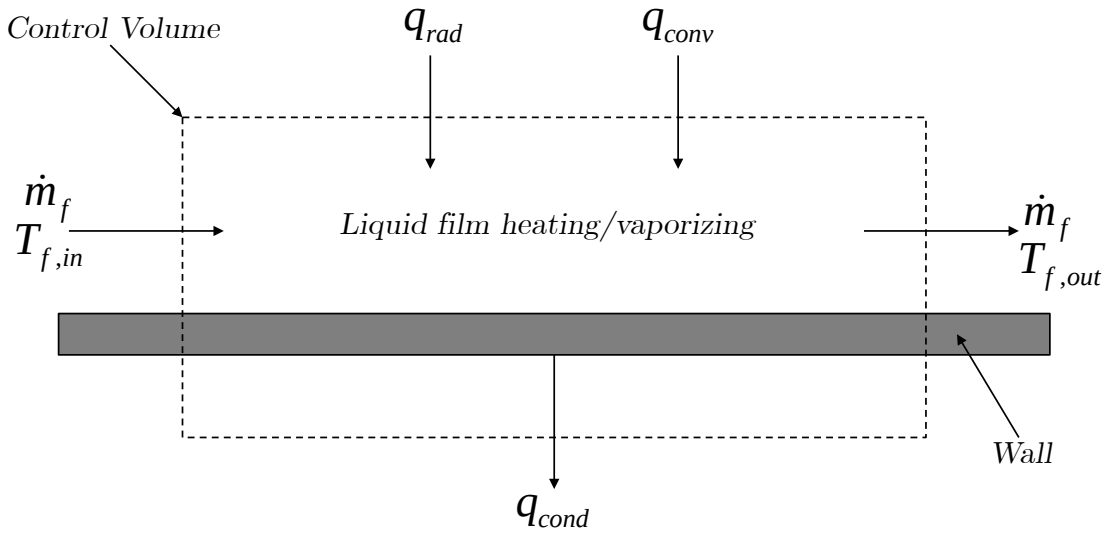

Figure 3. Heat transfer analysis of LFFC. Recreated from Figure 5.32 in Reference [9].

2.5 Conjugate Heat Transfer

A conjugate heat transfer problem occurs when there is heat transfer across a solid/fluid interface [16]. LFFC is inherently a conjugate heat transfer problem as the heat flux from the film (fluid) to the wall (solid) is of main interest. At this fluid/solid interface the heat flux conducted by the fluid must balance out with the heat flux conducted by the solid and any radiation (if not neglected) as shown in Equation (61).

$$\left(k\frac{\partial T}{\partial y} \right)_{fluid} + q''_{rad} = \left(k\frac{\partial T}{\partial y} \right)_{solid} \tag{61}$$

2.6 Multiphase Flow

Multiphase flow is the presence of two or more phases within a flow field. LFFC is inherently a multiphase flow as the core flow is comprised of gaseous combustion products and the cooling film is liquid fuel. This results in a gas-liquid interface.

Other multiphase flows can include gas-solid[1], liquid-solid, liquid-liquid, and three-phase flow. ANSYS® Fluent—the software selected to perform the CFD simulations presented in this thesis and discussed more in Section 3.3—splits multiphase flow into six different regimes shown in Figure 4 and described below [1]:

1. Slug Flow: Large bubbles or "slugs" within a continuous flow.

2. Particle-laden/bubbly Flow: Numerous droplets or particles within a continuous flow.

3. Slurry Flow: A semifluid mixture with many particles suspended in a fluid.

4. Stratified/Free-surface Flow: Two fluids with a clear boundary at their interface.

5. Sedimentation: A collection of particles initially well dispersed within a fluid which then settling to the bottom over time.

6. Fluidized Bed: Fluid distributed through a bed of suspended particles.

To simulate the different multiphase flows presented above Fluent offers three main models whose use is dependent upon the multiphase flow regime. The first, and most complicated, is the Eulerian model. The Eulerian model solves a set of momentum and continuity equations for each phase. The Eulerian model is appropriate for particle-laden/bubbly flows, slurry flows, fluidized beds, and sedimentation. The second is the mixture model. The mixture model solves a mixture momentum equation and calculates velocities for the dispersion of each phase. The mixture model can be used for particle-laden/bubbly flows and sedimentation. The final model is the Volume of Fluid (VoF) model. The VoF model tracks the interface of two or more fluids. A single

[1]Note the term "solid" used here is in reference to a solid particulate suspended within a fluid, not a solid interface such as a wall.

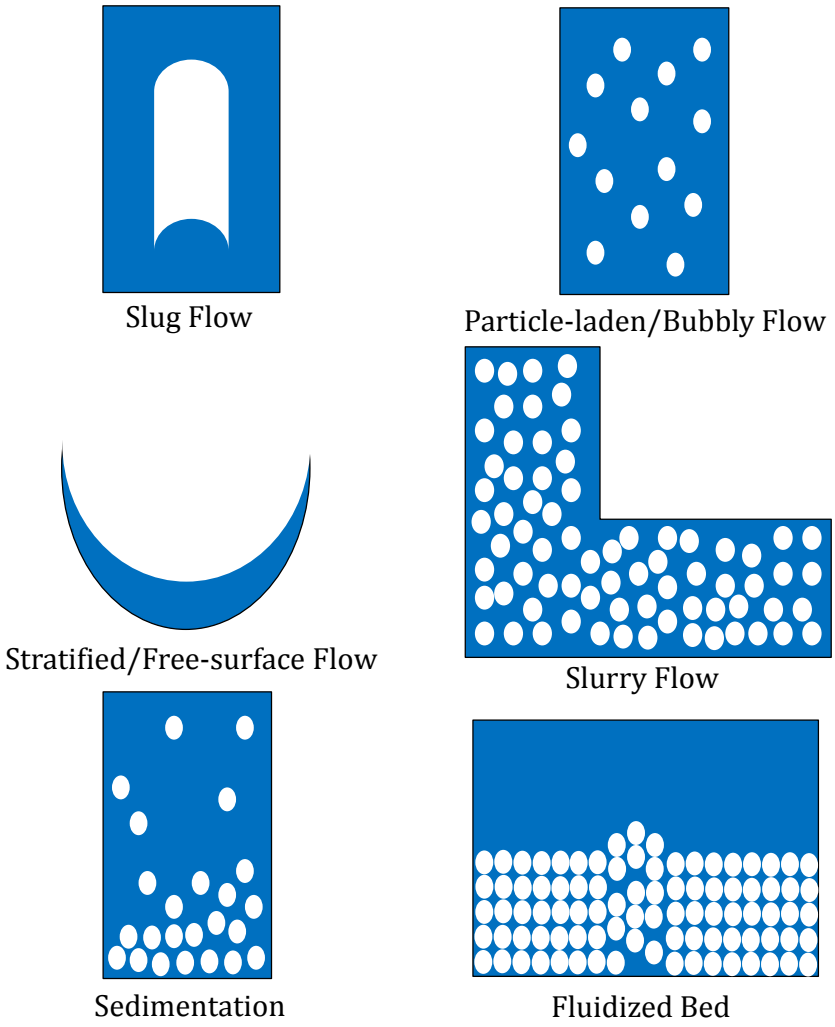

Figure 4. Six regimes of multiphase flow. Recreated from Figure 17.1 in Reference [1].

set of momentum equations is solved for the fluids and a volume fraction is tracked for each fluid in every cell. The VoF model can be used for stratified/free-surface flows and is useful when the interface of a fluid-gas flow is of interest [1].

III. Research Methodology

In general, the CFD solution process may be broken down into three main steps [24]:

1. Pre-process
2. Solve
3. Post-process

The first step is the pre-processing of the overall problem. This step can be broken down into four sub-steps. The first is the creation of the geometry. The geometry must be a physical representation of the particular flow situation to be solved. The geometry also bounds the overall problem and sets up the computational domain, or the domain where the flow properties are to be calculated. Geometry can include simplifications to reduce the complexity of the problem while still maintaining an acceptable level of accuracy in the calculations. An example is reducing the flow between parallel plates to a 2D geometry and ignoring the third dimension.

The second step of pre-processing is mesh generation. Once the geometry has been established a mesh, or grid, needs to be generated to fill the computational domain. This divides the domain into some number of discrete cells. It is at these individual cells that the specific fluid properties are calculated.

The third step is the selection of what fluid properties and physical phenomena are to be simulated. This can include whether the simulation will be steady or unsteady. Will species transport and combustion be modeled? Will a turbulence model be used? Is heat transport going to be accounted for? There are numerous other physical phenomena that can be included and the choices available for a particular CFD solution are generally only limited by what has been included with the solver of choice.

The fourth and final step of pre-processing is the selection of boundary conditions. Boundary condition selection includes the creation of inlets/outlets for fluid to flow, designations of walls, heat transfer properties, etc. The selection of boundary conditions must be representative of the physical flow conditions or the results of the CFD simulation will be nonphysical.

The second step in the CFD process is to solve the flow properties. This step can be broken down into two sub-steps. The first is solution initialization. Before a simulation can be run, all flow properties in every cell must have some initial value. The selection of this initial value can be crucial in ensuring a converged solution. Specific methods for calculating the flux of flow properties across cell faces are also selected in this step. This step is highly solver-specific as some solvers allow the user to specify a vast number of flow parameters/models while others are very limited in their options. The second sub-step is convergence monitoring. Convergence monitoring is a multi-step process where calculations of flow properties, residual monitoring, and solution monitoring all occur. As the calculations of the flow properties progress, values known as residuals track the imbalances in the flow properties. These residuals can be used to ensure the simulation is converging to some final solution. When all residuals reach some user defined threshold the solution can be said to have converged. The solution can also be monitored using other flow-specific variables. This can include lift or drag coefficients for an airfoil or the heat transfer coefficient for flow inside a heat exchanger. Solution convergence can also be decided by these flow-specific variables in addition to, or in place of, the residuals.

The final step in the CFD process, post-processing, comes after a converged solution is achieved. This step includes data analysis and flow visualization and is explained in the next chapter.

3.1 Computational Domain

As mentioned in Chapter I, the geometry for the CFD analysis presented in this thesis is based upon a physical experimental test rig developed by AFRL. The purpose of the test rig is to study LFFC. Figure 1 on page 2 shows an overview of the test setup. The physical setup uses a one inch square channel through which the hot combustion gases flow. One of the channel walls is an instrumented copper plate through which the liquid fuel is injected through a slot oriented 90 degrees to the freestream. Figure 5 shows the copper plate (first component on the left, shown in green) and the various components behind the plate that allow for the fuel injection, water channels, and temperature measurements.

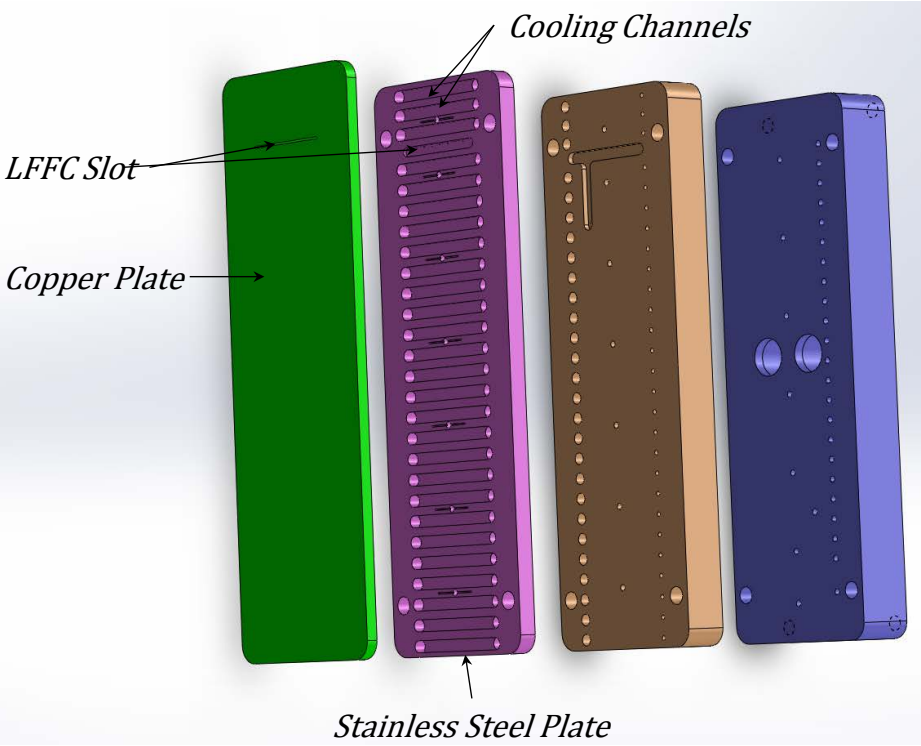

Figure 5. Copper and stainless steel plates along with instrumentation components.

The water channel plate is the component second from the left shown in pink, just behind the copper plate, and is manufactured with stainless steel. The backside of

the water channel plate is treated as adiabatic in the simulations. This assumption is made as it is assumed the water channels will be responsible for the majority of the heat removal and any conduction to the other two components behind the water channel plate (also manufactured with stainless steel) will be small due to the low thermal conductivity of stainless steel ($\sim$16 W/m/K) and large convective heat transfer coefficient of the water channels. As a result of this adiabatic assumption the two components located behind the water channel plate are not included in the computational domain. Figure 6 shows the 2D computational domain used in the final simulations. The axis are non-dimensionalized by the LFFC slot—also called the fuel injection slot—width (0.51 mm). The flow field is 390.24d long (x-axis) and 49.80d high (y-axis), with d representing the width of the LFFC slot. The LFFC slot is 16.20d in height. The solid plates are split into two regions, located on either side of the LFFC slot. The upstream section, which receives no LFFC, is 36.35d long. The downstream section is 259.47d long. The heights of the copper and stainless plates are 6.24d and 9.96d respectively. The cooling channels have dimensions of 6.24d $\times$ 1.49d. Both the flow field and LFFC slot have sections that are not adjacent to a solid domain. These are regions where the walls were modeled as inviscid to help with numerical stability in the simulations. The flow field inviscid region is 93.41d long and the fuel injection slot inviscid region is 9.96d long.

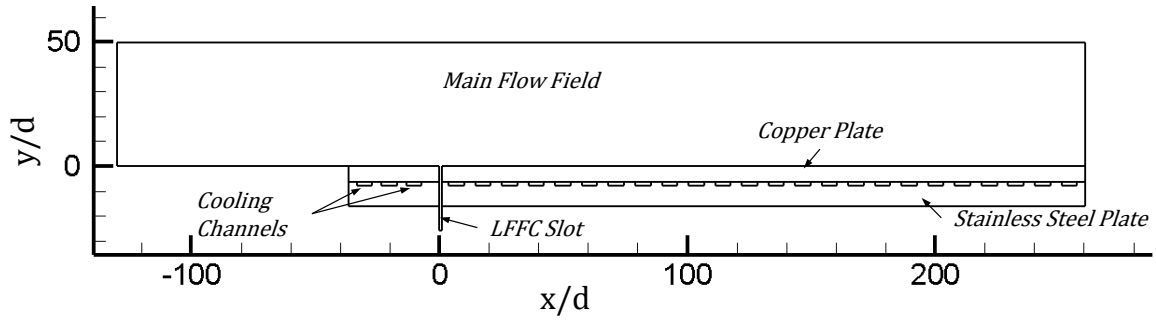

Figure 6. 2D computational domain.

3.2 Grid Generation

Grids are commonly separated into two categories that depend upon how the grid points are mapped. The first type is the structured grid and has been in use since the very early days of CFD. The second, and more recent, is the unstructured grid.

3.2.1 Structured Grids

Structured grids are defined as a grid whose nodes are sequentially numbered in each of the coordinate directions ($\hat{i}$, $\hat{j}$, $\hat{k}$). This means that the neighbors of each node are implicitly known from the node's coordinates. A benefit of this implicit numbering system is it negates the need to separately track and store the neighboring cells of each grid point which can result in increased memory or computing needs. However, to achieve this ordered connectivity of grid points structured grids require equal numbers of grid points on opposing grid boundaries. Having equal numbers of grid points on opposing boundaries can result in increased grid resolution in areas where it is not needed. As a result, the amount of computing time required can increase. Benefits of a structured grid are regularly shaped cells that avoid producing skewed areas in a mesh, and good performance resolving boundary layers, as high aspect ratio cells can capture the gradients normal to the surface. An example of a structured grid is shown in Figure 7 in the "Fluid" domain.

3.2.2 Unstructured Grids

Unstructured grids fundamentally differ from structured grids in that there is no defined manner in which the nodes are numbered. Instead of sequentially numbered coordinates, the neighbors of each cell face must be explicitly defined and stored. Unstructured grids may have difficulty capturing boundary layers as boundary layer refinement can result in highly skewed cells. Unstructured grids also have difficulty

with higher order schemes. Despite these drawbacks, unstructured grids can excel over structured grids due to their ability to easily mesh highly complex geometry. Unstructured grids can also reduce the number of cells required to fill a domain as they do not require equal numbers of grid points on opposing sides as structured grids require. An example of an unstructured grid is shown in Figure 7 the "Copper" and "Stainless Steel" domains.

3.2.3 Final Grids

The final grids used for this CFD analysis were generated using the Pointwise®mesh generation software. The grids used a combination of structured and unstructured topologies. The decision to use both structured and unstructured was to capitalize on the benefits offered by both types. The main flow channel and LFFC slot are meshed with a structured grid to resolve the boundary layer and interface between the liquid film and core flow. The square channel is also perfectly suited to a structured grid as there are no curves or complex geometry present to skew the grid.

The copper and stainless steel plates were meshed with an unstructured grid, point-matched to the structured grid in the main flow channel and the LFFC slot. This means that the grid in the channel, LFFC slot, and the plates all share the same nodes at their respective interfaces (walls) between the regions. This helps ensure accurate results when modeling conjugate heat transfer as no interpolation is required between mismatched boundaries. The use of an unstructured grid also greatly reduces the number of cells required in the solid plates. Had a structured grid been used in the plates the spacing used in the fuel injection slot would have propagated through the entire solid domain, greatly increasing the number of cells and computing time.

An overview of the 2D grid used is shown in Figure 7. The figure shows a close-up

view of the LFFC slot to show the structured grid used in the fluid domains and the unstructured used in the solid domains. Note the resolution shown in the figure is not the resolution used for the final grid. A larger resolution (fewer cells) is shown for visualization purposes to adequately convey the general grid structure.

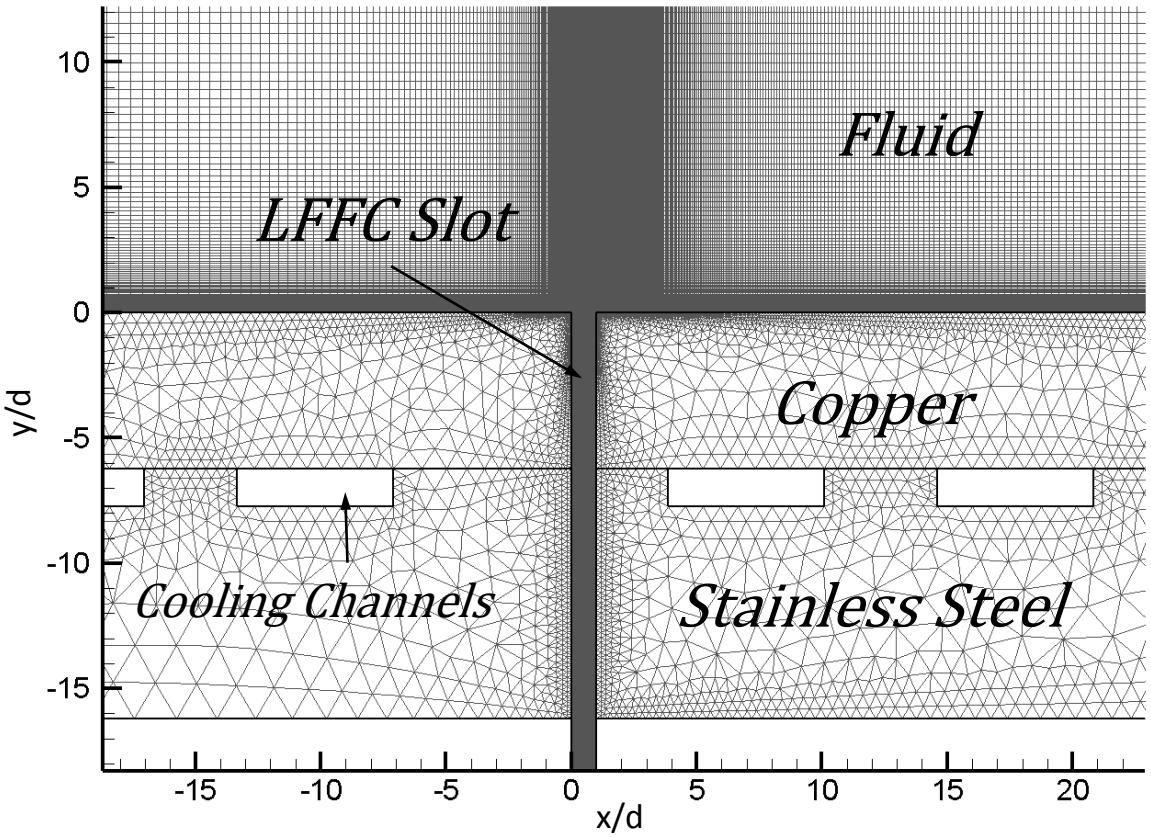

Figure 7. Zoomed view of 2D grid structure. Resolution is coarser than final resolution for visual purposes only.

Due to the selection of the turbulence model used (κ-ω SST) the dimensionless wall distance (y^+) value off the walls in the fluid domain must be ≤ 1. This required calculating a specific distance off the wall (Δs) at which to place the first grid point. The value of Δs is calculated using the wall shear stress (τ_w), density (ρ), friction velocity (u_τ), and kinematic viscosity (ν) through Equations (62) and (63). The value for τ_w is approximated using the skin friction coefficient (C_f) and Reynolds number with Equations (64) and (65). Equation (64) is based on turbulent flat-plate data

[27].

$$u_\tau \equiv \sqrt{\frac{\tau_w}{\rho}} \tag{62}$$

$$y^+ \equiv \frac{u_\tau \Delta s}{\nu} \tag{63}$$

$$C_f \approx \frac{0.027}{Re_x^{1/7}} \tag{64}$$

$$\tau_w = \frac{C_f \rho u_\infty^2}{2} \tag{65}$$

$$\Delta s = \frac{y^+ \nu}{u_\tau} \tag{66}$$

To calculate Δs using the above formulas, the values for u_∞, ρ, and μ were needed along with a length scale (l) and the desired y^+ value. The values used are shown in Table 1 along with the resulting Δs value used. The values used in the calculation were taken from an initial simulation using kerosene vapor as a freestream material designed to very roughly approximate the experimental flow conditions. The values were then taken from the entrance of the main fluid flow channel. The resulting actual y^+ values in the final simulations are not covered in Chapter IV but are instead presented in Appendix A. Note, the data in Table 1 did not need to be exact as the resulting Δs does not need to produce an exact y^+ value, it only needs to result in $y^+ \leq 1$.

Table 1. Values for y^+ calculation.

Variable	Value
u_∞	39.3 m/s
ρ	30 kg/m^3
μ	7×10^{-6} kg/m/s
l	0.15138 m
y^+	1
Δs	1.76×10^{-7} m

3.3 Solver

The selected solver for the presented numerical model was the ANSYS® Fluent v16.2 CFD software package. Fluent is a commercially available program that is widely used in multiple industries. An earlier version of Fluent was used in previous AFRL research on LFFC with hydrocarbon fuels [8]. The numerical model developed in Fluent for this research used 2D, steady, multiphase, turbulent flow, with species transport and heterogeneous reactions (reactions between fluids of differing phases). The specifics of the numerical model developed within Fluent are presented below while the exact settings used within Fluent are detailed in Appendix B.

3.3.1 Boundary Conditions

An overview of the boundary conditions used are shown in Figure 8 with exact boundary condition names and locations included in Appendix C. Both the main flow field and the LFFC slot used mass flow rate inlets. The main flow channel inlet was set to 25.64 kg/s with $T_\infty = 3680$ K. The LFFC slot had $T_\infty = 350$ K with a $\dot{m}$ that was varied across different simulations. Both inlets used turbulence intensity and hydraulic diameter (d_h) for the turbulence conditions. A default intensity of 5% was used for both inlets while the main flow field used a hydraulic diameter of 25.4 mm and the fuel injection slot used 0.999 mm. The d_h was calculated with Equation (67) using the area (A) and perimeter (L_p) of either the main flow channel or fuel injection slot. The outlet of the main flow field was set to a pressure outlet of 4.8 MPa. The walls of the main flow field and LFFC slot that extend beyond the plates were set to adiabatic inviscid walls for numerical stability. The walls switch to viscous/no-slip walls in the regions adjacent to the plates. The upper wall of the main flow field was set to adiabatic. The wall adjacent to the copper plate, and the interfaces between the copper and stainless steel plates were set to coupled heat transfer conditions which

are explained in Section 3.3.3. The copper and stainless steel plate walls which lay on the exterior of the domain (i.e. do not share a wall with the flow field or other solid plate) were set to adiabatic heat transfer conditions. The water channels were set to convective heat transfer boundaries with $T_\infty = 350$ K and $h = 25,500$ W/m^2/K. The values for the temperatures and flow rates were taken to roughly represent the AFRL experimental setup. The value for h was calculated with the Dittus-Boelter equation, Equation (68), using the properties of water flowing at approximately 144 m/s [29].

$$d_h = \frac{4\mathrm{A}}{L_p} \tag{67}$$

$$Nu_D = 0.023 Re_D^{4/5} Pr^n \tag{68}$$

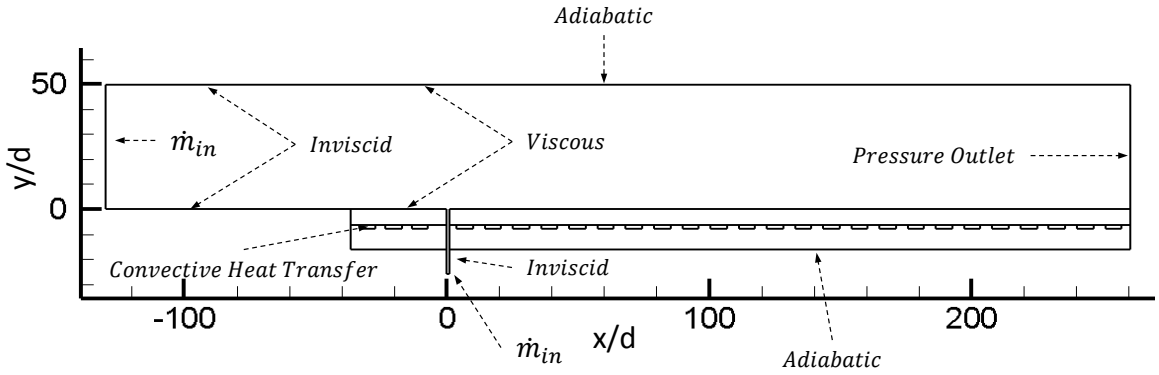

Figure 8. Overview of boundary conditions.

3.3.2 Multiphase Flow

Section 2.6 discusses the three different models available in Fluent for multiphase flows. The flow that is of interest for this research is a stratified flow with a fluid-gas interface (liquid kerosene and a gaseous freestream). Thus the VoF model was selected for use to model the multiphase flow.

The fuel was modeled as liquid kerosene from Fluent's built in materials database. The chemical formula Fluent uses to physically represent kerosene is $C_{12}H_{23}$. The freestream composition was modeled as a compressible ideal gas mixture of products resulting from the combustion of RP-1 and O_2 with a mixture ratio ($\dot{m}_o/\dot{m}_f$) of 2.8. The value for the mixture ratio was selected to be similar to previous work [8]. This combustion mixture was obtained using NASA's Chemical Equilibrium with Applications (CEA) program [22]. A copy of the output of the CEA results is included in Appendix D. It was assumed that the selection of RP-1 in CEA was an adequate substitute for kerosene in Fluent due to the similar hydrogen to carbon ratios (H to C). CEA uses a H to C ratio of 1.95 to represent RP-1 while Fluent uses a H to C ratio of 1.92 for kerosene. The mole fractions of each species used for the freestream mixture are listed in Table 2. Three species, H_2O_2, HCO, and HO_2, were omitted as they were not available in Fluent's database. It was determined that neglecting these species would not adversely affect the solution as they had mole fractions less than 5×10^{-4} when included. All other products from the RP-1/O_2 combustion had a mole fraction of less than 5×10^{-6} and were not included in the CEA results by default.

Table 2. Mole fractions of freestream species.

Species	Mole Fraction	Species	Mole Fraction
CO	0.29004	H_2O	0.31227
CO_2	0.15778	O_2	0.03739
H	0.03256	O	0.02113
H_2	0.06730	OH	0.08154

3.3.3 Conjugate Heat Transfer

The simulation of conjugate heat transfer requires the use of what Fluent refers to as "two-sided walls". Two-sided walls are a pair of boundary connectors that phys-

ically represent the same boundary, or wall. One connector will represent one side of the wall while the other connector, called a shadow wall, represents the opposite side. One of the options for the thermal conditions for a two-sided wall is a condition labeled "coupled". This coupled option notifies Fluent that the pair of boundary connectors are physically the same and uses information from adjacent cells to calculate the heat flux across the boundary.

As the grids were generated in Pointwise®, extra processing was required after importing the grid to Fluent as the two-sided walls were not automatically recognized. Instead the two-sided walls were identified as separate, unrelated boundaries. Each pair of connectors for every two-sided wall within the domain were fused together to form a single connector. This changed the "type" of the connector to the default option which is "interior", a type of boundary condition that allows fluid to freely flow through. Once the type was correctly changed back to "wall", Fluent recognized the two-sided wall and automatically created the corresponding shadow connector. The process of fusing the connectors and altering their type back to a wall type was automated using a journal script (Appendix E). A journal script is a method of automating tasks within Fluent.

3.3.4 Species Transport & Heterogeneous Reactions

Species transport was enabled to allow for the creation of the freestream mixture comprised of the combustion products listed in Table 2. The freestream mixture is only defined by what species comprise the mixture. The mixture does not include any information about the amounts of each species. To account for the proper ratio of each species, the mole fractions were specified in the species tab of the main flow field mass flow rate inlet boundary condition (Figure 100 in Appendix B).

Heterogeneous reactions (chemical reactions between two or more phases) were

included in the numerical model through reaction mechanisms for kerosene/RP-1 given in Reference [26] and shown below in Table 3. The two reactions are the paraffin and naphthene reactions between the liquid fuel ($C_{12}H_{24}$) and free O_2 in the gaseous freestream mixture. As Fluent uses $C_{12}H_{23}$ for a molecular representation of kerosene the coefficient for the H_2 product was entered as 11.5 to account for the single H difference.

Table 3. Heterogeneous reaction mechanisms.

Reaction	A $((gmol/cm^3)^{\varepsilon-1}/s)$	B	E/R (K)	Form[1]
Paraffin Global Step				
$C_{12}H_{24} + 6O_2 \rightarrow 12CO + 12H_2$	3.888×10^4	1	1.220×10^4	$P^{0.3}[C_{12}H_{24}]^{0.5}[O_2]$
Naphthene Global Step				
$C_{12}H_{24} + 6O_2 \rightarrow 12CO + 12H_2$	2.312×10^7	1	1.965×10^4	$P^{0.3}[C_{12}H_{24}]^{0.5}[O_2]$

The constants A, B, and E/R as well as the "Form"[1] are for use in calculating the forward reaction rate constant (k_f) of the reactions using an Arrhenius rate function shown in Equation (69) [25]. The Arrhenius rate form is an empirical method for calculating chemical reaction rate constants. The reaction order (ε) is the sum of the exponents of the concentrations of the two reactants in the "Form" column (see x and y in Equation (70)). This ensures that the resulting reaction rate (R')—shown generically in Equation (70)—has units consistent with moles per unit volume. The bracket notation (e.g. $[O_2]$) used in the Form column in Table 3 and in Equation (70) denote species concentrations in moles per unit volume.

$$k_f = AT^B e^{-E/RT} \tag{69}$$

$$R' = k_f T[X_F]^x[X_{ox}]^y \tag{70}$$

The implementation of the paraffin and napthene reactions were included in the

[1]Note that the pressure used here is in unis of atm, not Pa.

numerical model through the Graphical User Interface (GUI) under the phase interactions dialog box (e.g. Figure 60 in Appendix B). However, while the reaction—$C_{12}H_{24} + 6O_2 \rightarrow 12CO + 12H_2$—may be included in this manner, the reaction rate method may not be entered through the GUI. The method for calculating the reaction rate must be included in the numerical model through the use of a User Defined Function (UDF). A UDF is a custom program developed by the user—written in the C programming language—that allows for functions/calculations to be performed within the baseline code that are otherwise not available. Once compiled within Fluent—Figure 130, Appendix B—the UDF becomes available for use in the designated area of the solver. A UDF was written to calculate the appropriate reaction rate using the data from Table 3 along with pressure and temperature information in each cell. This enabled the selection of either the paraffin or naphthene global step reaction rate in the phase interaction dialog box where the heterogeneous reactions were entered (bottom left corner of Figure 60, Appendix B). The UDF is included in Appendix F.

3.3.5 Solution Initialization

Before a CFD simulation may be performed all variables in the domain must be set to some initial value or set of values. This process is called initializing the solution. A combination of different methods were used to develop an initial solution that could be used for the final test cases presented in Chapter IV. The overall method consisted of beginning with simplified flow conditions and then introducing more complex physics incrementally. A coarse grid was used in this initialization process to reduce the amount of computation time required. The piece-wise introduction of more complex physics was done to help with solution stability. It was found that attempting to initialize solutions with complex flows (e.g. multiphase or heterogeneous reactions) would quickly cause the solution to diverge.

Fluent offers several different built in methods for solution initialization. These methods are generally used when no other initial solution currently exists. As such, a built in method—Hybrid initialization—was used for early solutions using simplified flow conditions. The hybrid initialization method solves the Laplace equation to produce a velocity and pressure field in the domain while patching—setting a constant value through the domain—the remaining variables[1]. These simplified flow solutions did not include any heterogeneous reactions or LFFC. The solutions were first run using first-order upwind (FOU) solution methods for the density, momentum, turbulent kinetic energy, dissipation rate, and temperature discretization schemes. The FOU method produces a large amount of numerical dissipation [17]; areas of high gradients become smeared out. While the FOU method reduces the accuracy of the solution, the dissipation introduced generally results in a more stable solution. This can be extremely useful when performing early solution initializations where solution instability may be unavoidable when using higher order methods, such as second-order upwind (SOU). In addition to reducing the order of accuracy to first order the under-relaxation factors were also adjusted as needed to help with solution stability. Under-relaxation factors control how much of a change each flow field variable sees between iterations. For example, a factor of 0.8 for temperature would indicate that whatever ΔT is calculated at the current iteration n, the temperature will only change by $0.8\Delta T$ for iteration $n+1$. Under-relaxation factors help prevent large deltas in flow variables which may lead to instabilities. An under-relaxation factor of 1 would produce no limit in the change from iteration n to $n+1$. Under-relaxation factors below 1 also result in an increase in computation time as the solution requires more iterations to achieve an equivalent change in flow variables. The lower the under-relaxation factor, the larger the increase in computation time. The simulations were allowed to run until the flow field and temperature profiles had been well established

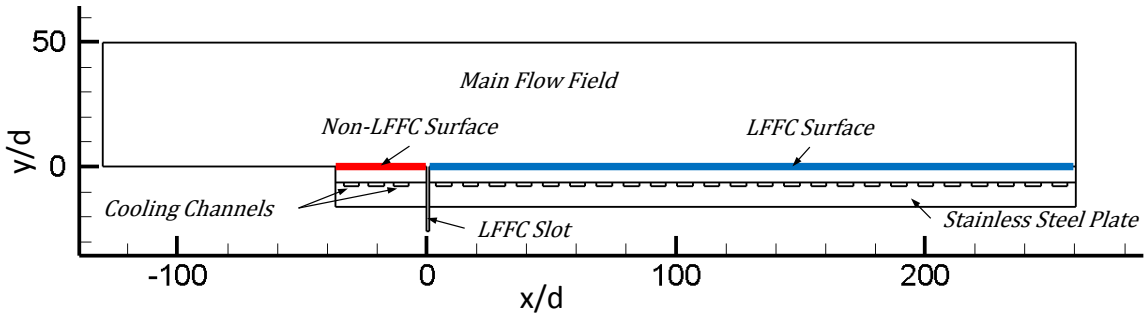

Figure 9. Location of the surfaces used to monitor the solution.

through the domain.

To track the solution—and determine when it was well established—four monitors were used:

1. The area-averaged surface temperature ($\overline{T}_s$) of the non-LFFC portion of the copper plate (upstream of the fuel injection slot).

2. The $\overline{T}_s$ of the LFFC portion of the copper plate (downstream of the fuel injection slot).

3. The area-averaged surface heat flux ($\overline{q''}_s$) through the non-LFFC portion of the copper plate.

4. The $\overline{q''}_s$ through the LFFC portion of the copper plate.

The non-LFFC and LFFC surfaces of the copper plate are shown in red and blue respectively in Figure 9. The solution was determined to be well established once the residuals had become periodic and the monitors had visually approached some average value. This process typically took 10,000 to 20,000 iterations. Once a solution had been established using the simplified flow conditions and FOU methods, the density, momentum, turbulent kinetic energy, dissipation rate, and temperature discretization schemes were set to SOU and again run until the monitors and residuals had visually stabilized. The discretization schemes were then switched back to FOU and the

mass flow rate of the fuel for the fuel inlet was set at the appropriate value and the volume fraction of fuel in the entire fuel injection slot was patched to 1. Patching the volume fraction sped up the solution initialization as fuel would already be fully occupying the fuel injection slot at the first iteration. The solution was then run again until the monitors and residuals had visually stabilized. The process was then repeated with SOU schemes. Finally, the heterogeneous reactions were included in the numerical model. Immediately implementing the reactions at the full reaction rates was found to cause too much instability and cause the solution to diverge. So in addition to using the FOU/SOU switching described above, the values of A in Table 3 were incrementally increased to slowly ramp up to the full reaction rates. Once a solution was stable with the full reaction rates and SOU methods it was saved as an interpolation file for use in initializing the corresponding simulation on the final grid.

The use of interpolation files allowed for a solution developed on one grid (e.g. grid-A) to be used on another grid (e.g. grid-B). Both grid-A and grid-B must physically represent the same domain but can be different resolutions (e.g. grid-A may have 100k cells while grid-B may have 5M cells). The ability to use a coarser grid to develop the solution allowed for an initial solution to be developed much more efficiently and with fewer computational resources.

3.4 Solution Convergence

Once a solution has been initialized it must be run some number of iterations until solution convergence is achieved. Solution convergence is when the the solution has approached some constant or average value. A common method for checking for solution convergence in CFD simulations is checking to see if all residuals have decreased to below some value after having been normalized by the residual of the first iteration. Two common thresholds are 1×10^{-3} and 1×10^{-6}. However, residuals

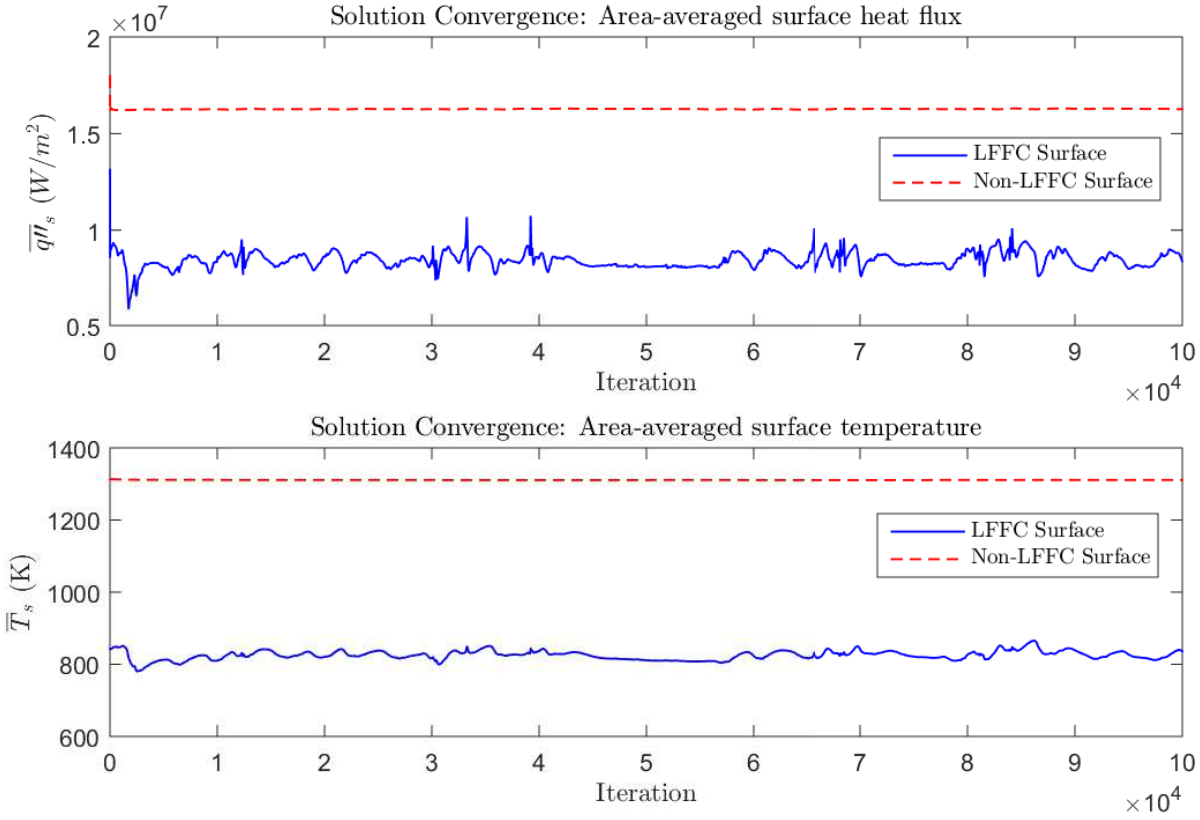

Figure 10. Example of solution evolution over 100k iterations.

are not the only indicator of solution convergence and may never decrease to below the desired threshold in certain situations. It was observed in early simulations that the monitors listed in Section 3.3.5 would still be evolving even after the residuals had already dropped below 1×10^{-6} or had leveled off at some higher value. Thus the residuals were not used to determine solution convergence. Convergence was instead determined using the surface temperature and heat flux monitors for the LFFC portion of the copper plate. The monitors for the non-LFFC portion of the copper plate were not used. The reason for only using the monitors on the LFFC portion of the plate is shown in Figure 10. This figure shows an example of the evolution of the four monitors of one of the well initialized simulations over 100,000 iterations. The monitors for the non-LFFC surface show almost no change over the

iteration range indicating that they are converged. The LFFC surface monitors show large fluctuations over the entire span. Because of these fluctuations the LFFC surface monitors were selected to check for solution convergence. Note that the convention used for plotting heat flux in Figure 10, and through the rest of this document, is a positive heat flux represents heat flow into the copper plate.

The final simulations were run to 100,000 iterations after being well initialized as described in Section 3.3.5. The last 30,000 iterations were selected to check for convergence. The 30,000 iteration span was split into three equal and overlapping samples, shown in Figure 11 as the orange lines. The mean—Figure 11 dashed black lines—and standard deviation of each sample was calculated. An average of the sample means was calculated along with the standard deviation of the sample means. If the standard deviation of the sample means is less than the mean of the sample standard deviations ($\sigma_{\text{means}} < \overline{\sigma}$), then the solution is converged. The MATLAB script used for this procedure is included in Appendix G.

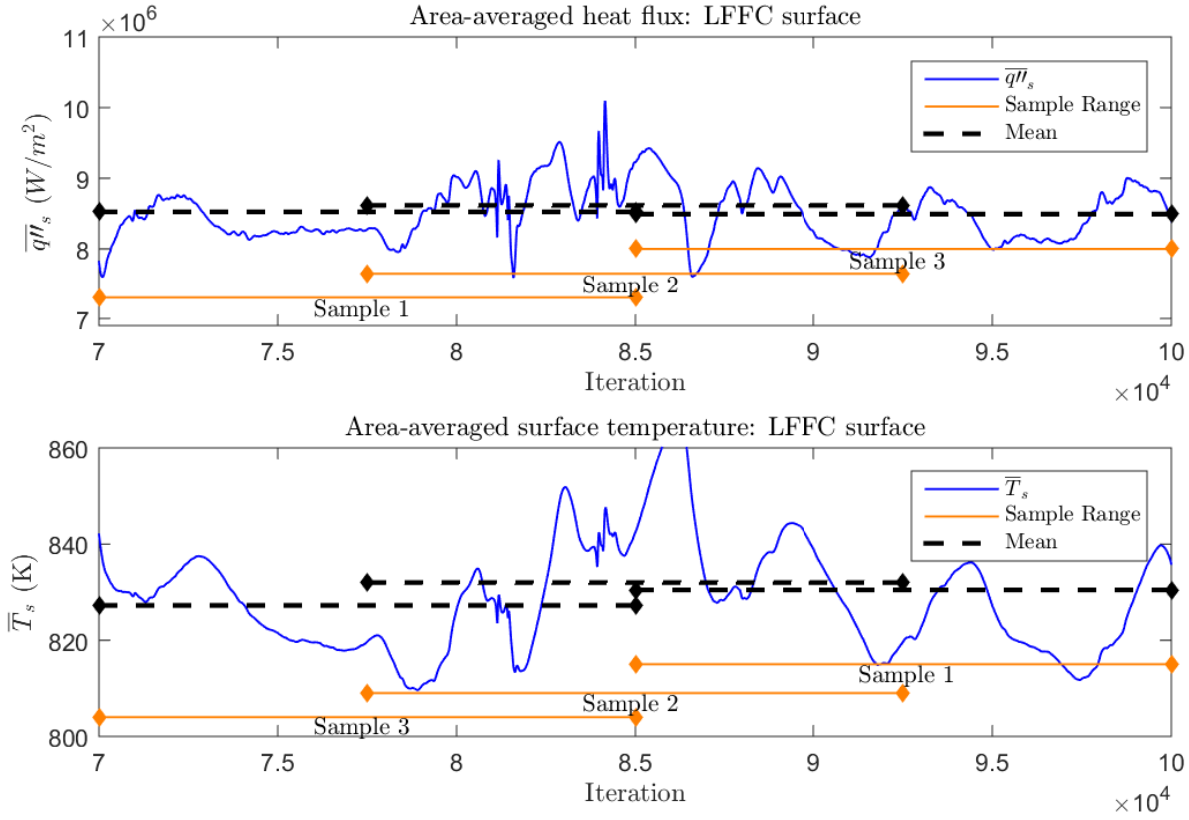

Figure 11. Example of solution convergence check.

3.5 Grid Convergence Study

A necessary step in any CFD research is the execution of a grid convergence study. A grid convergence study is used to ensure that the final grid, or mesh, used in the simulations will produce a solution that is independent of the grid's resolution—also called a grid independent solution. Having a solution that is grid independent indicates that the discretization error introduced into the solution due to the use of a mesh with some finite number of cells is sufficiently small to be considered negligible. Grid convergence studies are generally performed by running simulations using identical numerical models on successively finer grid resolutions. One or more figures of merit (f) are then compared across the different solutions. Differences in

f are indicative of the discretization error. Note that the increase in grid resolution corresponds to an increase in cell count which results in an increase in computation time. Thus a balance must be met between a grid of sufficient resolution that is fine enough to reduce the discretization error to an acceptable value but not overly fine that the computational time required becomes unfeasible.

The method used for performing the grid convergence study—and quantifying the discretization error—for the simulations presented in Chapter IV is the grid convergence index (GCI) method [18]. The GCI method has become a commonly accepted method in literature for calculating discretization error [5]. The method is based on Richardson extrapolation (RE) and can be broken down into four steps presented below [5, 18, 19].

The first step involves the calculation of the grid resolution (χ) and grid refinement factor (r). This starts with the creation of multiple grids with successively finer resolutions. A minimum of two grids are needed for the GCI method while three or more are recommended [18]. The results for the grid convergence study presented in Chapter IV use a minimum of three grid resolutions. The grid resolution (χ) is then calculated for each grid. The value for χ is a representation of the grid size and is shown in Equation (71) for a 2D grid (a 3D grid would use an exponent of 1/3 in place of 1/2). The number of cells is represented by N_{cells} while A_i is the cell area.

$$\chi = \left[\frac{1}{N_{\text{cells}}} \sum_{i=1}^{N_{\text{cells}}} (\text{A}_i) \right]^{1/2} \tag{71}$$

Once the values for χ are known for each grid, r is calculated for each pair of grids. The value for r represents the ratio of grid sizes for successive grids and is shown in Equation (72) with $\chi_2 > \chi_1$ (grid 1 is finer than grid 2). It is recommended that r

be larger than 1.3 [5].

$$r_{21} = \frac{\chi_2}{\chi_1} \tag{72}$$

The second step is selecting the figures of merit and calculating their apparent order of accuracy (p). The equations presented below—and used in the results of the grid convergence study shown in Chapter IV—use three grids with $\chi_3 > \chi_2 > \chi_1$. The values for p (one per each figure of merit) were symbolically solved using a MATLAB script with Equations (73) through (77). The MATLAB scrip is included in Appendix H for the fluid domain.

$$p = \frac{1}{\ln(r_{21})}\left|\ln\left|\frac{\epsilon_{32}}{\epsilon_{21}}\right| + q(p)\right| \tag{73}$$

$$q(p) = \ln\left(\frac{r_{21}^{p} - s}{r_{32}^{p} - s}\right) \tag{74}$$

$$s = 1 \cdot \mathrm{sgn}(\epsilon_{32}/\epsilon_{21}) \tag{75}$$

$$\epsilon_{32} = f_3 - f_2 \tag{76}$$

$$\epsilon_{21} = f_2 - f_1 \tag{77}$$

After the values for p are known the third step is to calculate the extrapolated values for the figures of merit, f_{ext}^{21}. The extrapolated values represent the solution on an "infinite" grid and can be considered an asymptotic limit representing the exact numerical solution[2].

$$f_{\mathrm{ext}}^{21} = \frac{r_{21}^{p} f_1 - f_2}{r_{21}^{p} - 1} \tag{78}$$

The fourth and final step is calculating the approximate relative error and extrapolated relative error ($e_a{}^{21}$ and $e_{\mathrm{ext}}{}^{21}$) along with the GCI for the fine grid ($\mathrm{GCI}_{\mathrm{fine}}^{21}$).

[2]The exact numerical solution may still differ from the true physical solution.

The GCI is reported as a percent.

$$e_a{}^{21} = \left| \frac{f_1 - f_2}{f_1} \right| \tag{79}$$

$$e_{\text{ext}}{}^{21} = \left| \frac{f_{\text{ext}}^{21} - f_1}{f_{\text{ext}}^{21}} \right| \tag{80}$$

$$\text{GCI}_{\text{fine}}^{21} = \frac{1.25 e_a{}^{21}}{r_{21}^p - 1} \tag{81}$$

The 1.25 in Equation (81) represents a factor of safety. A value of 3 is recommended for a factor of safety when only two grids are used for a grid convergence study. The use of three or more grids with a 1.25 factor of safety results in a GCI that approximately represents a 95% confidence interval [19]. Note that e_a and e_{ext} do not represent errors from experimental data but rather error due to discretization.

Another check for grid convergence is to ensure that each grid refinement produces a solution within the asymptotic range of convergence from the computed p [21]. With three grid resolutions this can be verified with Equation (82). For the results presented in Chapter IV this validation is reported using the asymptotic convergence check ($\mathscr{A}$) variable, shown in Equation (83), and verifying that is close to 1.

$$\text{GCI}^{32} = r_{21}^p \cdot \text{GCI}^{21} \tag{82}$$

$$\mathscr{A} = \frac{\text{GCI}^{32}}{r_{21}^p \cdot \text{GCI}^{21}} = 1 \tag{83}$$

As the final grids used for the simulations used two different types of meshes—structured in the fluid domain and unstructured in the solid domain—two different grid convergence studies were conducted with the GCI method. The first study looked at grid convergence in the structured fluid domain. This was done by either doubling or halving—depending on if a finer or coarser mesh was desired—the points on every

grid connector that touched the fluid domain. Once a grid independent mesh was found a second grid convergence study was conducted on the solid domain. This second convergence study used the grid independent mesh for the fluid domain as the starting grid. The the grid points on connectors that only touched the solid domain—connectors shared by both the solid and fluid domains were not changed—were again either doubled or halved. The decay rate for the unstructured grids was also increased to ensure a $r > 1.3$. The decay rate is a factor used to determine how many cells are generated within an unstructured domain. A larger decay rate will produce more cells.

Ideally, a grid convergence study would be performed for every individual test case. The results presented in Chapter IV detail six different test cases (Section 4.2.1). Performing a grid convergence study for all these test cases would be unfeasible for both time requirements and computing resources. Generally grid convergence studies are only performed for one test case and an assumption is made that the study will be valid for the remaining test cases provided they do not drastically differ. Thus the grid convergence studies performed for both the solid and fluid domains used 2.9% LFFC with non-reacting flow. See Section 4.2.1 for specifics on the test cases.

3.6 Solution Selection

The methods for solution convergence and grid convergence presented in Sections 3.4 and 3.5 are based on average values from the domain over the iteration history. At each iteration there exists an independent solution for the entire computational domain. Final analysis of the domain requires the solution at either one specific iteration or a solution averaged over many iterations. Due to the large oscillations shown in the monitor values (examples shown in Figures 10 and 11) the best approach would be to use an averaged solution over many iterations. Unfortunately, the nu-

merical solver used did not have the capability to average the solution for a steady simulation and attempting to export the solution at every iteration and average it manually would have required an unfeasible amount of storage. A suitable method for data analysis was created by examining the solutions of a simulation selected specifically due to large oscillations present in the monitor values and setting the solver to automatically save the solution every 500 iterations—a rate that would not require excessive amounts of storage. Several iterations both near and away from the mean of the heat flux for the LFFC surface were selected for analysis. The monitor for the area-averaged total surface heat flux on the LFFC portion of the plate was selected as it shows the largest amplitudes in the oscillations. These iterations and their monitor values are shown on Figures 12 and 13 (iterations selected are shown as orange dots). The corresponding profiles for the surface temperature and surface heat flux are shown in Figure 14. It is shown in Figure 14 that the solution taken near the mean (i.e. iterations 75k and 90k in Figure 12) produces a smother surface heat flux and temperature profile. Thus the solutions selected for analysis in Chapter IV were selected by locating a solution within the iteration range analyzed for solution convergence that had an area-averaged total surface heat flux monitor near the mean and corresponded to one of the available saved solutions—one every 500 iterations. The area-averaged heat flux values for iterations 75k and 90k were found to both differ from the mean by $< 4\%$. This difference value was used to inform a threshold value for the solutions used in the final simulations. To ensure the solutions analyzed in Chapter IV represented an average value, a conservative difference from the mean of 2.5% was used.

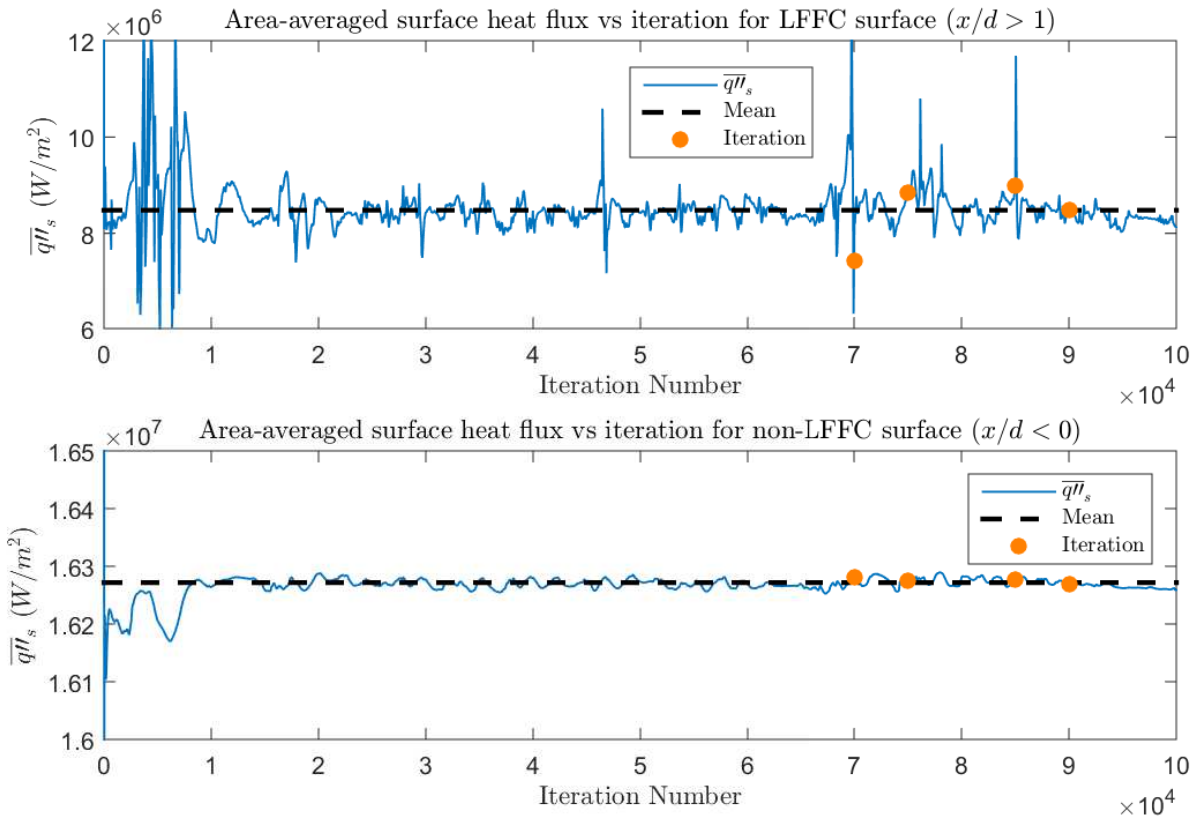

Figure 12. Total surface heat flux through both sections of the copper plate. Orange dots show iterations corresponding to data shown in Figure 14.

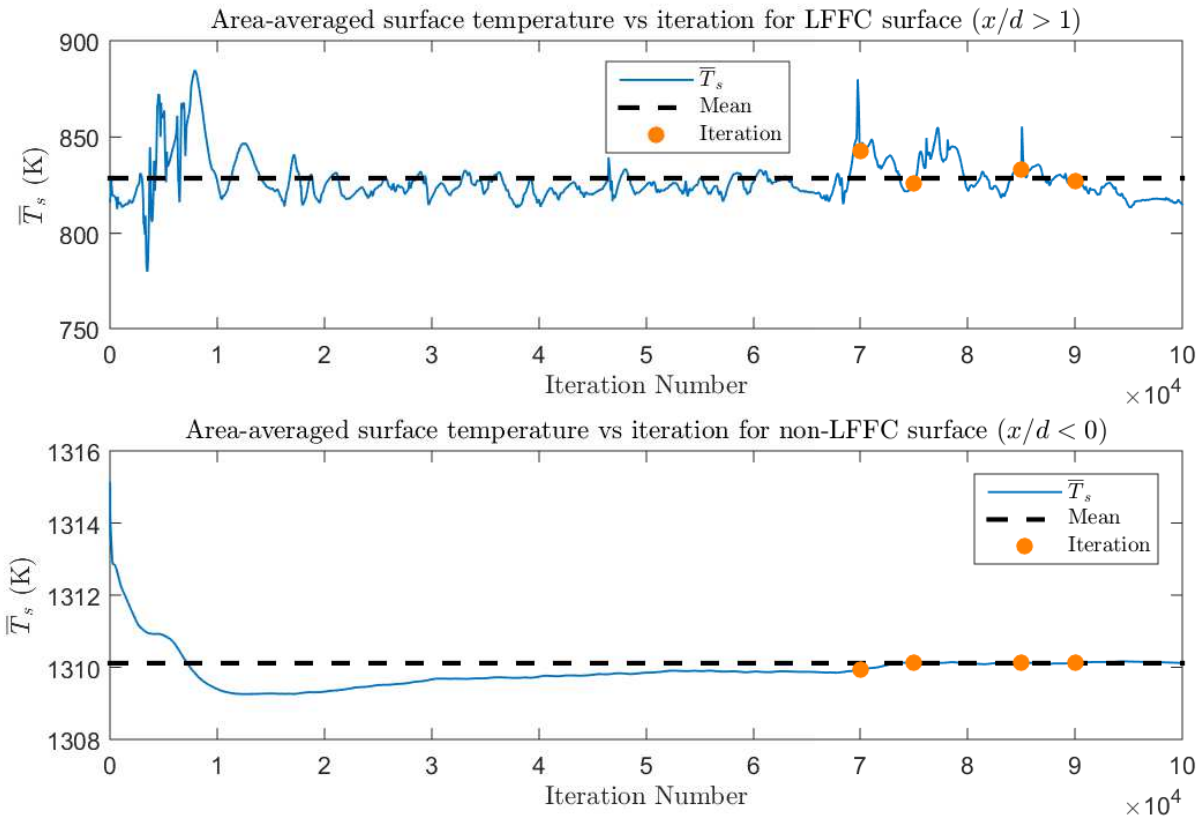

Figure 13. Area averaged surface temperature on both sections of the copper plate. Orange dots show iterations corresponding to data shown in Figure 14.

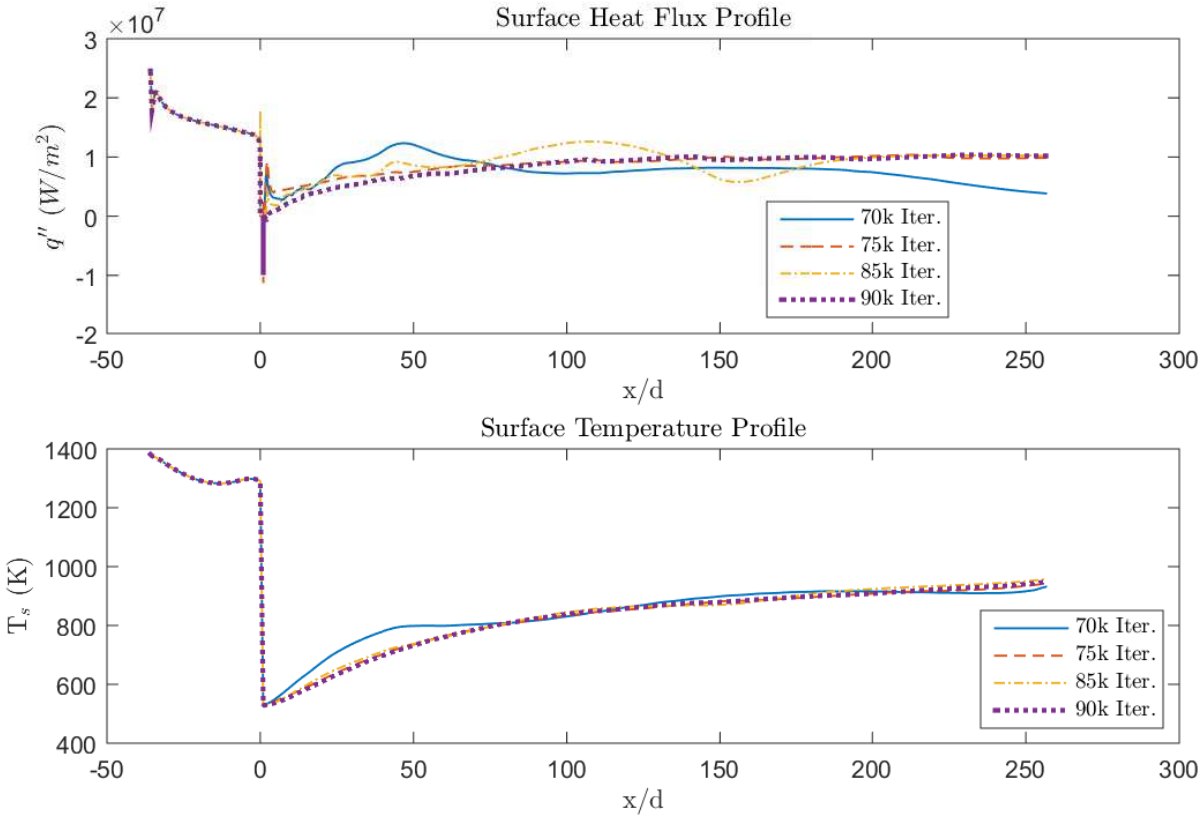

Figure 14. Copper plate surface profiles of temperature and heat flux at the 4 different iterations shown in Figures 12 and 13.

IV. Results

The results of the grid convergence studies and the final simulations are detailed in the following sections. In Section 3.3.5 the process of incrementally adding more complex physics to the simulation was explained. Early results produced during the process of incrementally adding more complex physics are available in Reference [3]. The early results compared two variations of a simplified freestream gas. One comprised of kerosene vapor and the other a freestream gas mixture comprised of combustion products similar to the final results presented in this chapter. The early results also used a simplified heterogeneous reaction mechanism that did not use a UDF or Arrhenius rate functions.

4.1 Grid Convergence Study

Section 3.5 detailed the two grid convergence studies performed—one for the fluid domain and one for the solid domain. Both studies were performed using 2.9% LFFC with non-reacting flows (see Section 4.2.1). The results of the two studies are presented in the following two sections.

4.1.1 Fluid Domain

Table 4 shows the solution convergence data for the fluid domain grid convergence study. For all grid resolutions the standard deviation of the means (σ_{means}) is less than the mean of the standard deviations ($\overline{\sigma}$) for both heat flux and surface temperature–designated by the associated units—indicating solution convergence. The plots corresponding to the data in Table 4 are available in Appendix I. The span used to determine solution convergence and grid independence was 80k to 90k iterations for the fluid domain grid convergence study. This differs from the solid domain study

Table 4. Solution convergence results for fluid domain grid convergence study; $\sigma_{\text{means}} < \overline{\sigma}$ indicates solution convergence.

Grid	σ_{means} (W/m^2)	$\overline{\sigma}$ (W/m^2)	σ_{means} (K)	$\overline{\sigma}$ (K)
Coarse	2.77×10^4	5.26×10^5	1.14	9.99
Medium	3.64×10^4	2.47×10^5	1.52	6.36
Fine	1.27×10^5	3.18×10^5	2.05	11.67
Extra Fine	7.33×10^4	2.47×10^5	2.75	12.16

Table 5. Grid data for fluid domain grid convergence study (Figure 15).

Grid	N_{cells}(k)	χ(m)	r
Coarse (3)	139	2.12×10^{-4}	n/a
Medium (2)	520	1.10×10^{-4}	1.93
Fine (1)	2033	5.54×10^{-5}	1.98
Extra Fine (0)	4548	3.70×10^{-5}	1.50

and the final test cases which used 70k to 100k. The reason the span is smaller for the fluid domain was the differences in the grid resolutions were sufficient enough to produce large changes in the solution. The large changes allowed for a smaller iteration span to be used for analysis. Tables 5, 6, and 7 and Figure 15 show the results of the grid convergence study for the fluid domain. Table 5 lists the specifics of each grid used, the grid designation (0, 1, 2 or 3), the number of cells (N_{cells}) in thousands, the grid resolution (χ) in meters, and the grid refinement factor (r) between the the current grid and previous (coarser) grid (e.g. r for the medium grid is actually r_{32}). The coarse grid does not have an associated r as it is the coarsest grid. The number of cells range from ∼140 thousand in the coarse grid to over 4.5 million in the finest grid. The grid resolution (χ), Equation (71), is a measure of the grid spacing used—the smaller the number the finer the grid—and ranges from 2.12×10^{-4} to 3.70×10^{-5} meters. The grid refinement factor (r), Equation (72), ranges from 1.98 to 1.50. The two r's near 2 indicate a near doubling in all grid connectors which results in roughly a four times increase in the number of cells.

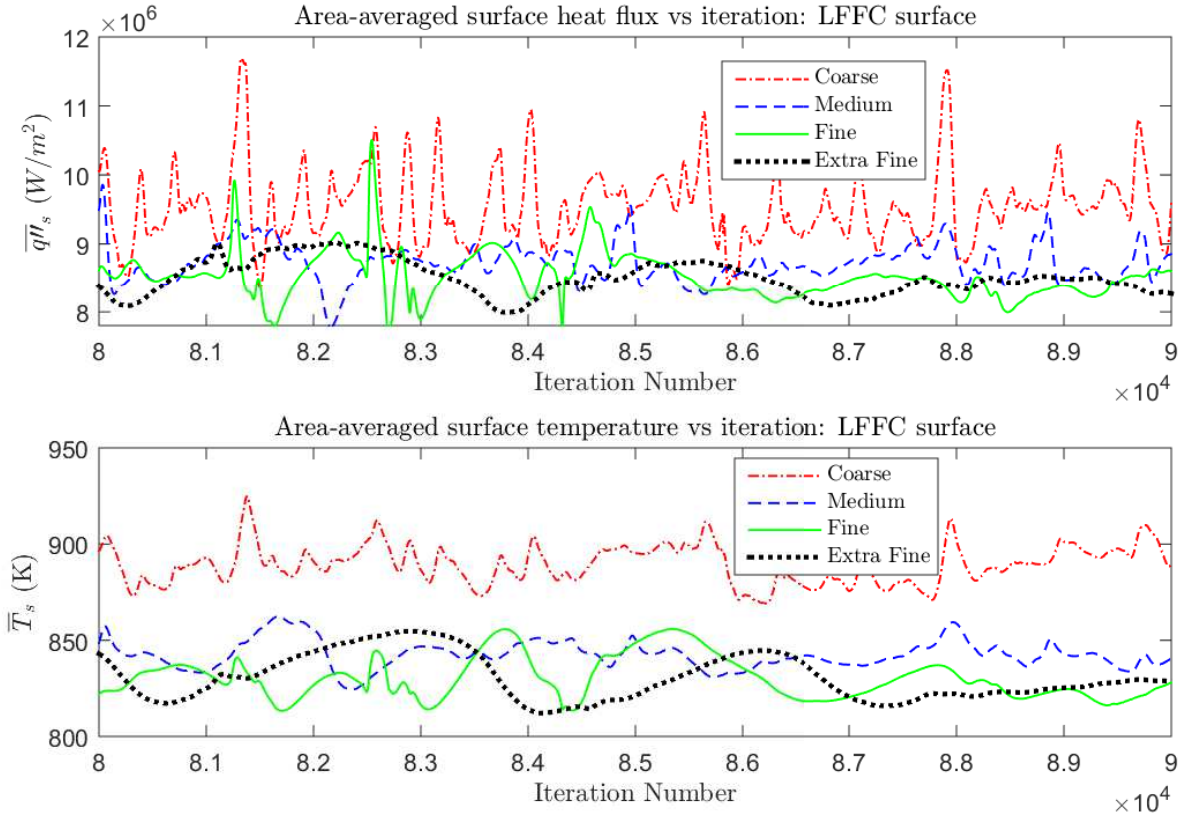

Figure 15. Monitor history for grid convergence study on fluid domain.

Figure 15 shows the iteration span used to find the averaged values for the figures of merit shown in Table 6. Table 6 also shows the extrapolated figure of merit calculated with Equation (78). The values for f_{ext}^{10} are nearly identical to the values on the extra fine grid with the differences occurring in the third and second decimal place for $\overline{q''}_s$ and $\overline{T}_s$ respectively. This suggests that the extra fine grid is approaching the asymptotic limit for the exact numerical solution.

Table 7 shows the approximate relative error (e_a), extrapolated relative error

Table 6. Figure of merit values averaged over span shown in Figure 15.

f	Coarse	Medium	Fine	Extra Fine	f_{ext}^{10}
$\overline{q''}_s$ (W/m^2)	9.54×10^6	8.72×10^6	8.52×10^6	8.50×10^6	8.50×10^6
$\overline{T}_s$ (K)	889.4	842.5	830.4	830.5	830.5

Table 7. Grid convergence results for fluid domain grid convergence study (Figure 15).

f	${e_a}^{32}$	${e_a}^{21}$	${e_a}^{10}$	${e_{\text{ext}}}^{10}$	$\text{GCI}^{10}_{\text{ext. fine}}$	$\text{GCI}^{21}_{\text{fine}}$	$\mathscr{A}$	p
$\overline{q''}_s$ (W/m^2)	9.5%	2.3%	0.02%	0.062%	0.08%	0.31%	0.998	3.4
$\overline{T}_s$ (K)	5.6%	1.5%	0.02%	0.002%	0.003%	0.030%	1.000	6.3

(e_{ext}), grid convergence index (GCI) and the apparent order of accuracy (p) for both figures of merit. The approximate relative errors get smaller with finer grids, as is expected, with the final approximate relative error (${e_a}^{10}$) being 0.02% for both f's. The extrapolated relative errors are both $\ll 1\%$ demonstrating the very small difference between the f values on the extra fine grid and the extrapolated f values. The $\text{GCI}^{10}_{\text{ext. fine}}$ is 0.08% for $\overline{q''}_s$ and 0.003% for $\overline{T}_s$ indicating a very small amount of uncertainty due to discretization error. The apparent order of accuracy for $\overline{q''}_s$ is 3.4 and 6.3 for $\overline{T}_s$. The values for p are for their respective figure of merit only and do not mean that the entire solution is greater than second order accurate. Values for p greater than 2 are also not unexpected and have been reported at values greater than 8 in literature [5]. The values for $\mathscr{A}$, calculated from Equation (83), were both near 1, indicating that the heat flux and temperature are in the calculated asymptotic range of convergence.

As the extra fine grid had approximate relative error and grid convergence index values much less than 1% it was selected as the grid converged mesh. The extra fine grid was used as the medium grid for the grid convergence study on the solid domain shown in the next section.

4.1.2 Solid Domain

The number of cells, grid resolution, and grid refinement factor for the solid domain grid convergence study are listed in Table 8. The values listed were calculated using only the cells in the solid domain which is why the N_{cells} values are far smaller

Table 8. Grid data for solid domain grid convergence study (Figure 16).

Grid	N_{cells}	χ(m)	r	% of Domain
Coarse (3)	22647	5.25×10^{-4}	n/a	0.5
Medium (2)	50865	3.50×10^{-4}	1.50	1.1
Fine (1)	90754	2.62×10^{-4}	1.34	2.0

Table 9. Grid convergence results for solid domain grid convergence study (Figure 16).

f	${e_a}^{32}$	${e_a}^{21}$	${e_{\text{ext}}}^{21}$	$\text{GCI}_{\text{fine}}^{21}$	p
$\overline{q''}_s$ (W/m^2)	0.47%	0.23%	0.61%	0.51%	1.10
$\overline{T}_s$ (K)	0.66%	0.037%	0.005%	0.0030%	7.30

than those listed in Table 5. Also listed in Table 8 is what percentage of all the cells in the entire grid reside in the solid domain. Tables 9 and 10 and Figure 16 all show that there is a much smaller amount of change between the different grids when compared with the changes seen in the fluid domain convergence study. For the solid domain the approximate relative errors and GCIs are all $< 1\%$ but do still show decreasing errors as the grid is refined. The very small approximate relative errors are likely due to the small number of cells in the solid domain—as compared to the fluid domain. As the solid domain only comprises between 0.5% and 2% of the total number of cells the total affect on the solution is small—producing very small values for e_a and GCI.

As the refinements in the solid domain all displayed small approximate relative errors all three grids may be considered converged. The solution convergence (Table 11) was thus taken into consideration for selecting the grid to be used in the final

Table 10. Figure of merit data for Figure 16.

f	Coarse	Medium	Fine	f_{ext}^{01}
$\overline{q''}_s$ (W/m^2)	8.45×10^6	8.49×10^6	8.51×10^6	8.56×10^6
$\overline{T}_s$ (K)	834.1	828.5	828.9	828.9

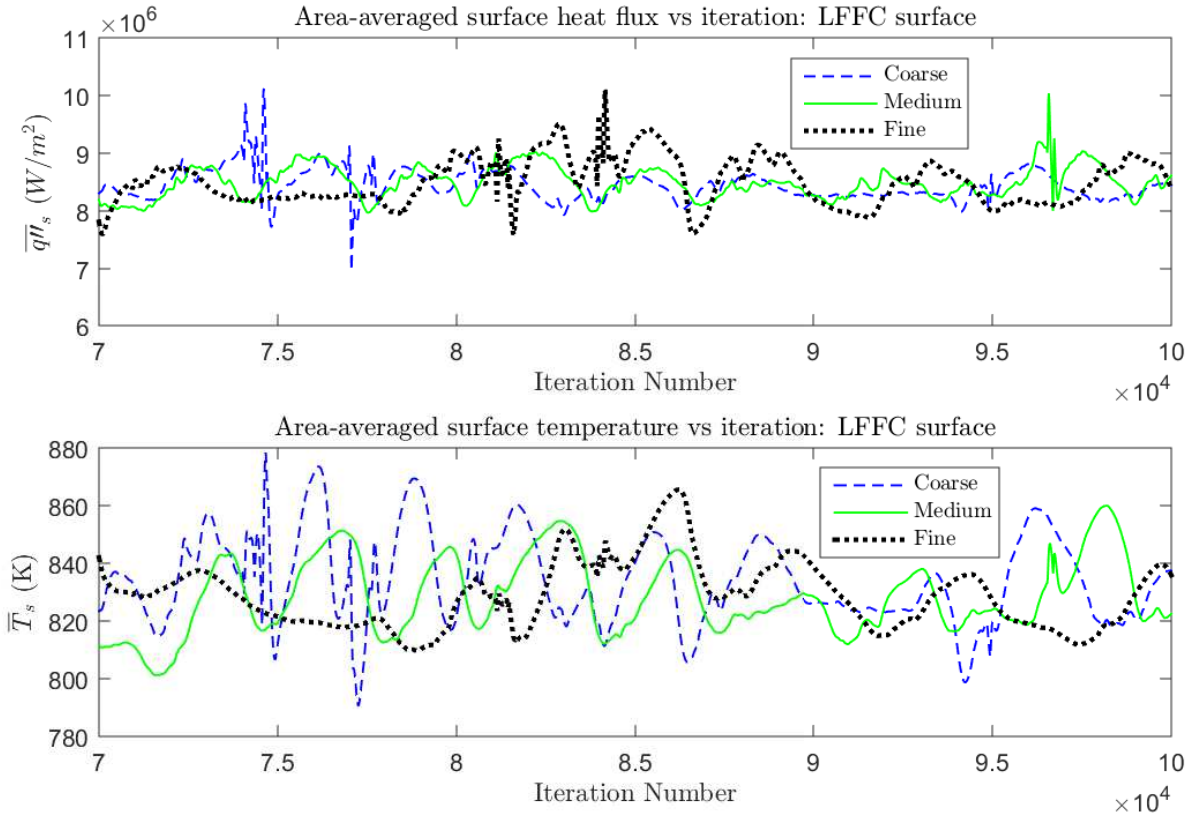

Figure 16. Monitor history for grid convergence study on solid domain.

Table 11. Solution convergence results for Figure 16; $\sigma_{\text{means}} < \overline{\sigma}$ indicates solution convergence.

Grid	σ_{means} (W/m^2)	$\overline{\sigma}$ (W/m^2)	σ_{means} (K)	$\overline{\sigma}$ (K)
Coarse	8.08×10^4	2.30×10^5	3.14	14.62
Medium	1.69×10^4	2.66×10^5	0.13	12.62
Fine	6.45×10^4	4.26×10^5	2.42	12.13

simulations. Note that the value for $\mathscr{A}$ was not used for the solid domain due to the solution convergence being used to determine the final grid choice. Both the coarse and fine grids show sporadic oscillations in their solution convergence plots—Figures 17 and 19. The means of the samples—bold dashed lines—are more dispersed resulting in a larger σ_{means} when compared to the medium grid. Figure 18 shows the solution convergence for the medium grid. The means of the samples are all nearly identical, resulting in the smallest σ_{means} value. The medium grid also shows a much more periodic solution with fewer spikes in the area-averaged surface temperature or heat flux. Due to this smoother trend in solution convergence the medium grid was selected for use in the final simulations. Note that this was also the grid selected in the fluid domain grid convergence study.

With the GCI method error bars can be overlaid on the data using the GCI value as shown in Reference [5] to visually show the amount of uncertainty due to discretization. However, as the GCI values for both monitors in both grid convergence studies were $\ll 1\%$ the error bars correspond to approximately the width of the line used to plot the results. As such, the error bars are neglected in this study for heat flux and temperature plots.

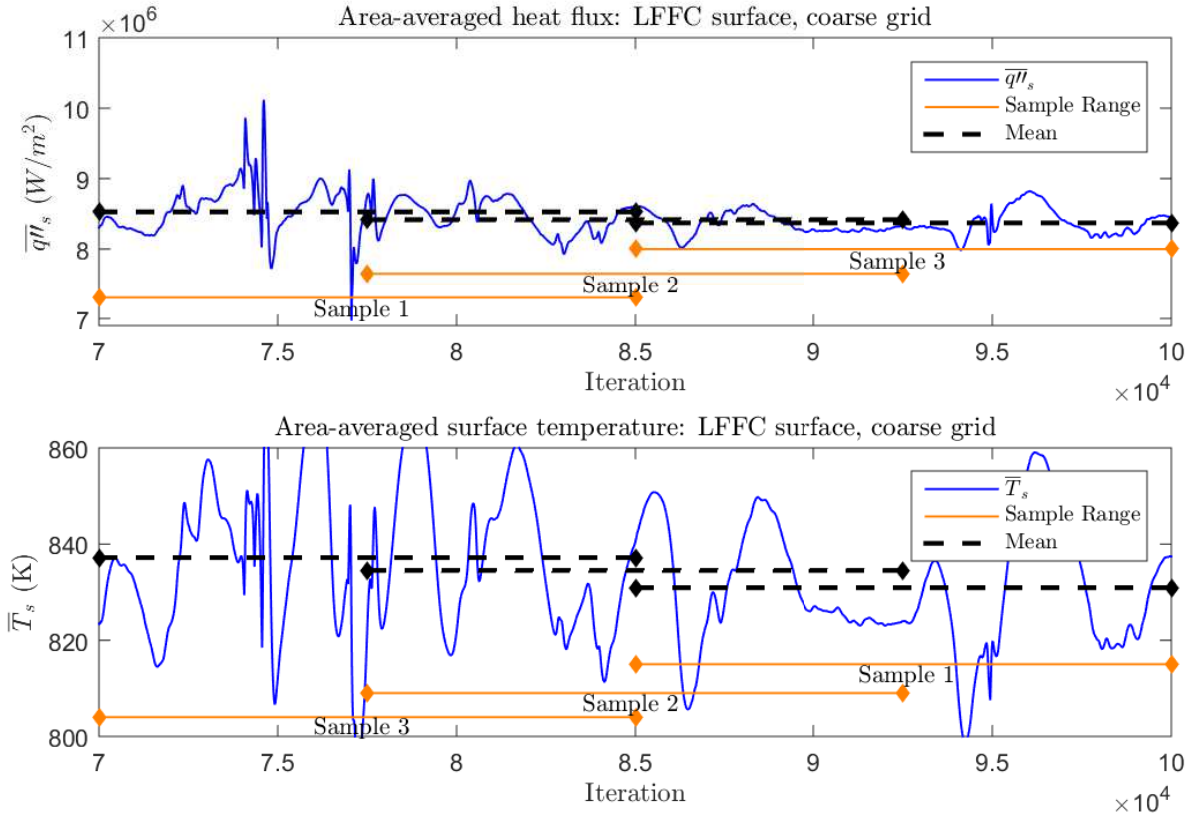

Figure 17. Solution convergence plot for coarse grid.

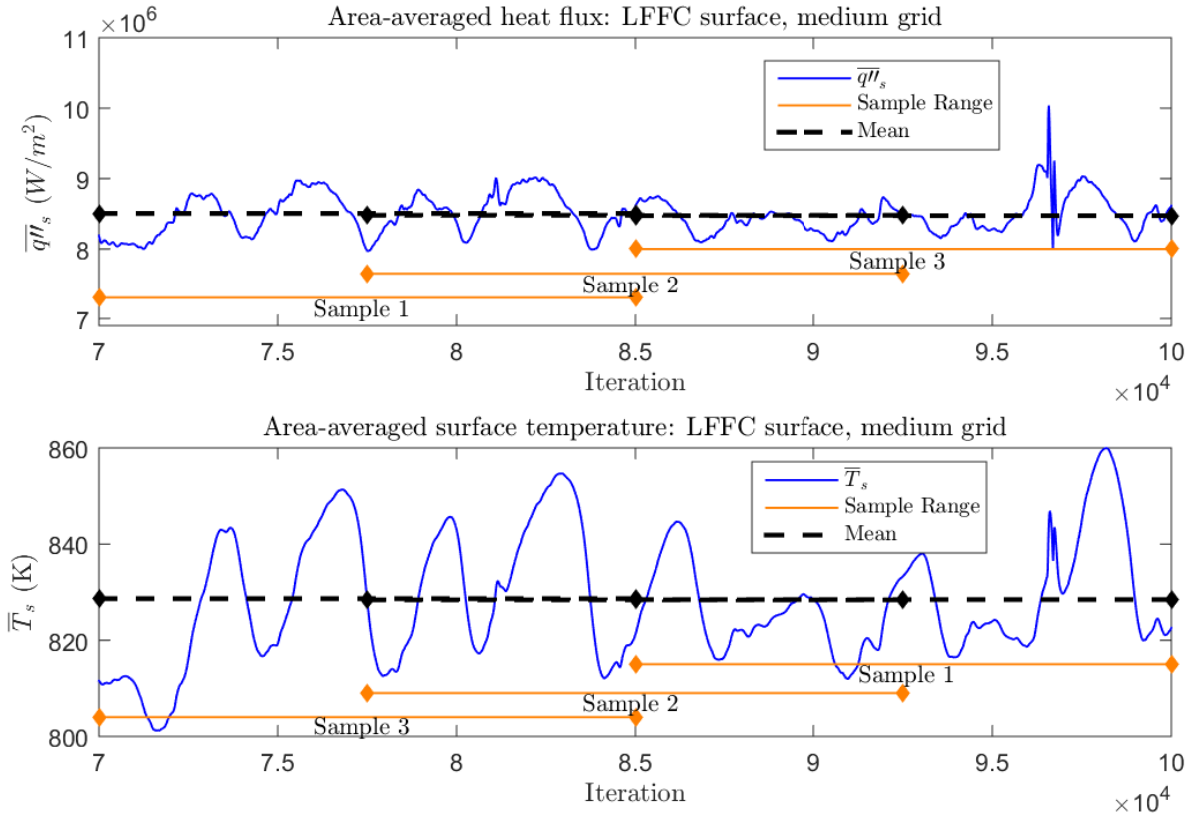

Figure 18. Solution convergence plot for medium grid.

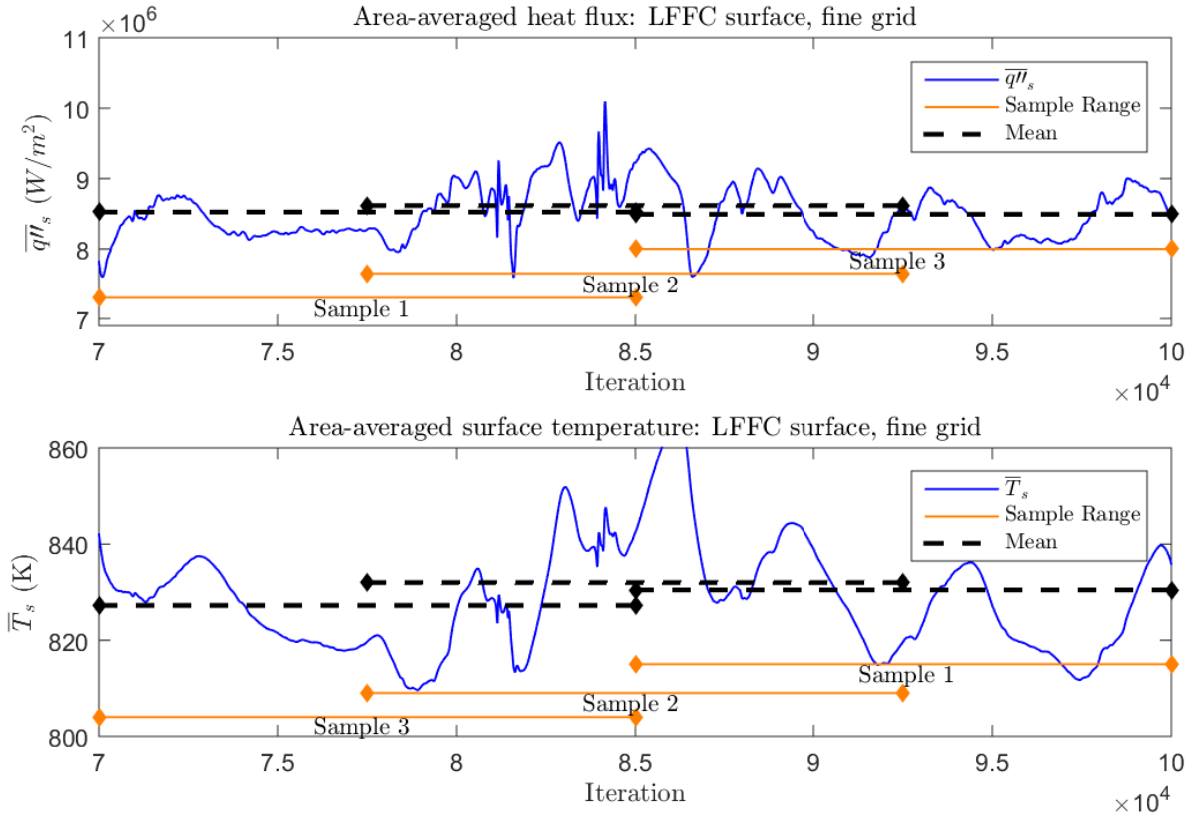

Figure 19. Solution convergence plot for fine grid.

4.2 Final Simulations

The results of the final simulations using the grid-converged mesh from Section 4.1.2 are presented in the following sections.

4.2.1 Test Cases

Table 12. Test cases used for the final simulations.

Test Case	Type	LFFC%	S.S. Plate
1	No RXN	0.0	Yes
2	RXN	1.6	Yes
3	No RXN	1.6	Yes
4	RXN	2.9	Yes
5	No RXN	2.9	Yes
6	No RXN	2.9	No

The six test cases used for the final simulations are detailed in Table 12. The test cases are split into two types called reacting (RXN) and non-reacting (No RXN) simulations. The reacting test cases included the heterogeneous reactions while the non-reacting did not. Test Case 1 is a non-reacting simulation automatically as there is no LFFC to react with the freestream. The LFFC% column indicates the amount of LFFC as a percentage of the main flow field $\dot{m}$ (25.64 kg/s). The last column—S.S. Plate—indicates if the effects of heat flux through the stainless steel plate with the water cooling channels were included in the simulation. Only one test case (TC6) did not include heat flux through the stainless steel plate. Aside from solution convergence and selection, Test Case 6 is only analyzed in Section 4.2.4.

Values for the Reynolds number using hydraulic diameter (Re_D), blowing ratio (M), Density Ratio (DR), and momentum flux ratio (I) are presented in Table 13. These values were calculated using the inlet conditions for the main flow field inlet and fuel injection slot inlet. All the values remain constant between reacting and

Table 13. Test case non-dimensional constants. Values calculated at the inlets of the main flow field and fuel injection slot.

Parameter	2.9% LFFC	1.6% LFFC	0.0% LFFC
Re_D (fuel inlet)	604	335	n/a
Re_D (main inlet)	1.49×10^6	1.49×10^6	1.49×10^6
M	1.44	0.80	n/a
DR	104	104	n/a
I	0.012	0.0061	n/a

non-reacting cases of the same LFFC%. In addition, the Re_D for the main inlet is constant across all LFFC%. The Re_D is constant as the $\dot{m}$ in the main flow field is held constant, resulting in constant velocity. The DR is constant for both non-zero LFFC% as the densities at both inlets are constant across all test cases. The DR is very high due to the vast difference in the densities of the freestream gas and the liquid kerosene,$\sim$8 kg/m^3 and $\sim$780 kg/m^3 respectively. The small values for I show that the kinetic energy present in the freestream gas is much larger than that present in the liquid kerosene.

4.2.2 Solution History

The evolution of the figures of merit used to monitor the final simulations are shown in Figures 20 and 21 for Test Cases 1-5 (Test Case 6 is included in Appendix J). The entire solution history from iteration 0 through 100,000 is shown. The data is shown using a combination of line types, color, and data markers. Red triangles indicate no LFFC, green stars for 1.6% LFFC, and blue circles for 2.9% LFFC. The thin solid lines are non-reacting simulations while the bold dashed lines are reacting simulations. This format will be used through the rest of this document.

For the simulations with LFFC present the early iterations (prior to $\sim$50,000) display large oscillations on the LFFC surface (top plots on Figures 20 and 21) as the solution is attempting to converge at some steady state. The last $\sim$30,000 iterations

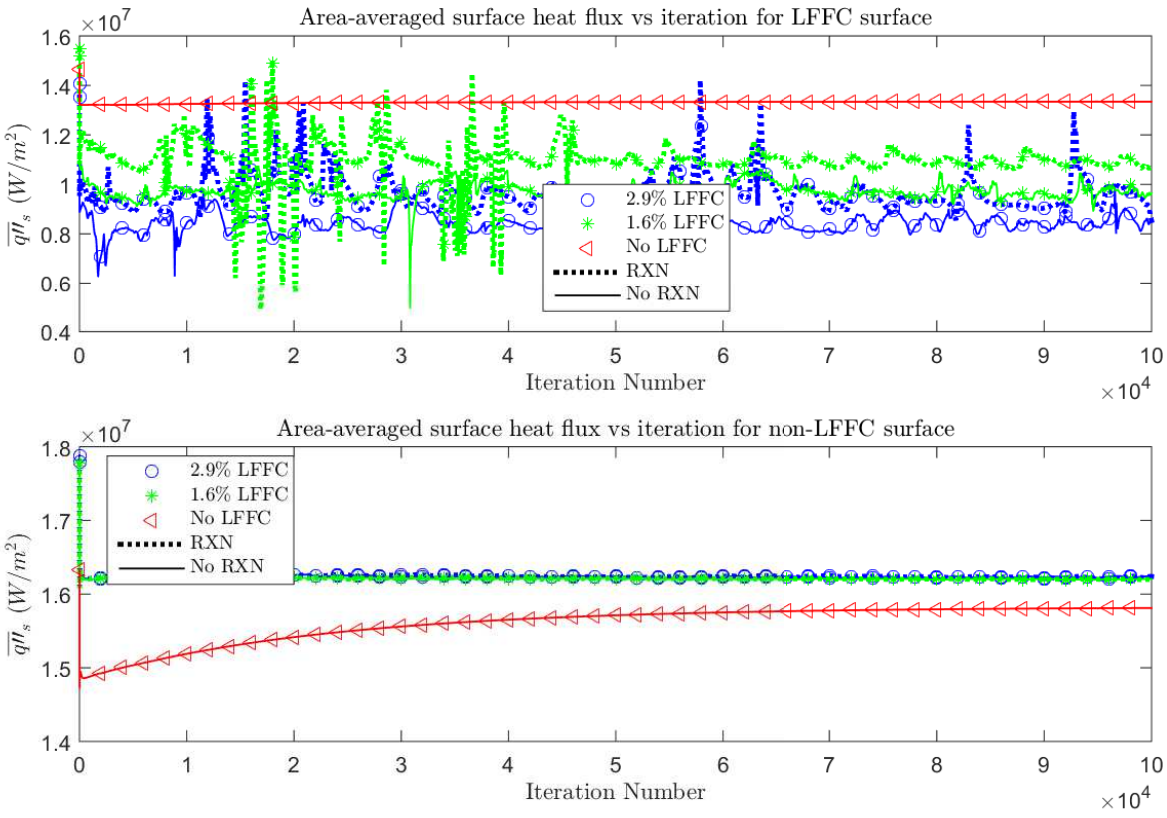

Figure 20. Solution history of area-averaged surface heat flux for final simulations.

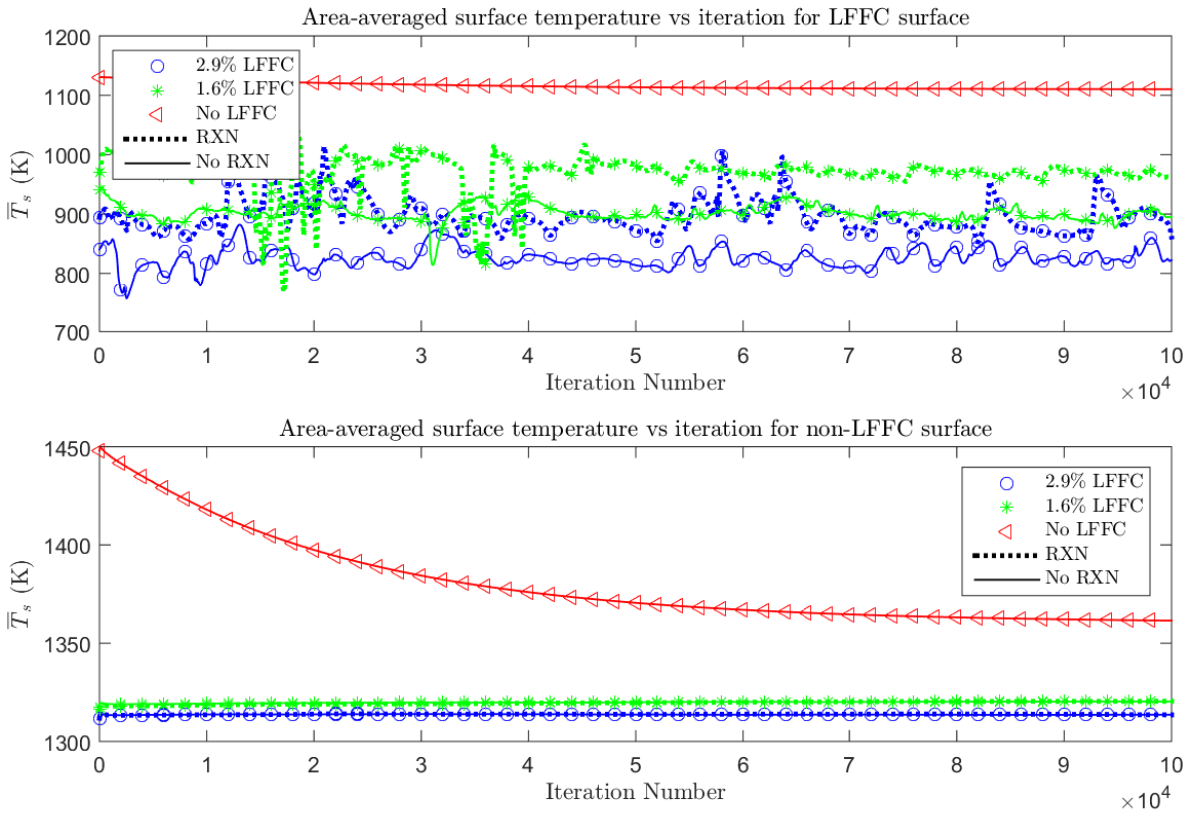

Figure 21. Solution history of area-averaged surface temperature for final simulations.

show the oscillations have reduced in frequency and amplitude and have become fairly regular as the solution has reached a quasi-steady state solution.

The plots for the non-LFFC surface (bottom plots on Figures 20 and 21) show a very smooth convergence with no large oscillations. The simulations with LFFC were very well initialized prior to their running on the grid independent mesh. Because of this initialization the monitors show almost no change on the non-LFFC surface in the simulations with LFFC. The simulation with no LFFC was not as well initialized which is why it does show some change in the iteration history before converging on a final value.

One key aspect of Figures 20 and 21 is the effect that the presence of LFFC has on the solution. With no LFFC (the no LFFC case and all plots on the non-LFFC surface) the monitors show very smooth convergence. When LFFC is introduced (2.9% and 1.6% on the LFFC surface plots) the monitors become erratic prior to convergence and then show periodic behavior once converged. The amount of LFFC also appears to affect the solution as the 2.9% LFFC case displays oscillations with larger amplitudes once it is converged when compared to the 1.6% LFFC case.

The overall trend of the plots for the LFFC surface (top plots on Figures 20 and 21) show the overall goal of the use of LFFC. The highest amount of LFFC (2.9%) has the lowest area-averaged surface temperature and heat flux (positive heat flux is heat flow into the copper plate). The 1.6% LFFC case shows a slightly higher area-averaged surface temperature and heat flux while the no LFFC case shows the highest area-averaged surface temperature and heat flux. The plots for the LFFC surface also show the effect of the heterogeneous reactions. The non-reacting cases all show a smaller value for area-averaged surface temperature and heat flux when compared to their respected reacting cases. This is expected as the heterogeneous reactions between the liquid fuel and the core gaseous freestream produce heat.

The plots for the non-LFFC surface (bottom plots on Figures 20 and 21) both show the 2.9% and 1.6% cases produce very similar results. This is expected as while there is LFFC on the LFFC surface (downstream of the fuel injection slot) the non-LFFC surface receives no LFFC as it is upstream of the slot. One could expect the no LFFC case to also lay near the LFFC cases for the non-LFFC surface, yet this is not the case. The reason is due to axial conduction. Since 2D conjugate heat transfer is included in the simulation the upstream portion of the copper plate experiences cooling through axial conduction due to the liquid fuel in the fuel injection slot. This reduces the average temperature resulting in the no LFFC case showing a lower $\overline{q''}_s$ and a higher $\overline{T}_s$.

4.2.3 Solution Convergence & Selection

Solution convergence results for the six test cases in Table 12 are shown in Table 14. All solutions show iterative convergence as each test case meets the criteria set in Section 3.4 ($\sigma_{\text{means}} < \overline{\sigma}$). Since Test Case 1 (TC1) had no LFFC it did not display the oscillations in the solution convergence that are characteristic of all the other TC's which did have LFFC. Thus solution convergence for TC1 was determined by finding the max iteration difference (Δ) for each figure of merit for the final 10,000 iterations using Equation 84 with n representing the current iteration. Table 15 shows the maximum percentage difference for both figures of merit for TC1. Both differences were $\ll 1\%$, thus the solution was considered converged. The figures showing the iteration history and sample means used to determine solution convergence are included in Appendix K.

$$\Delta = \left| \frac{f^n - f^{n-1}}{f^n} \right| \tag{84}$$

In addition to verifying solution convergence using the $\sigma_{\text{means}} < \overline{\sigma}$ method, the

Table 14. Solution convergence results for final test cases in Table 12; $\sigma_{\text{means}} < \overline{\sigma}$ indicates solution convergence.

Test Case	σ_{means} (W/m^2)	$\overline{\sigma}$ (W/m^2)	σ_{means} (K)	$\overline{\sigma}$ (K)
1	n/a	n/a	n/a	n/a
2	5.50×10^3	1.96×10^5	0.44	5.80
3	1.74×10^4	2.21×10^5	0.92	6.97
4	7.87×10^4	5.19×10^5	4.47	20.68
5	1.69×10^4	2.66×10^5	0.13	12.62
6	8.14×10^4	3.60×10^5	6.05	16.34

Table 15. Solution convergence results for Test Case 1 (No LFFC) in Table 12 for last 10,000 iterations using Equation (84).

f	Max Δ (%)
$\overline{q''}_s$	3.17×10^{-4}
$\overline{T}_s$	3.65×10^{-5}

moving average of the solution monitors was also used. Figure 22 shows the moving average of $\overline{q''}_s$ and $\overline{T}_s$ for all six test cases. Each data point on the plot represents the average of all prior data points. All test cases show the average of the monitors becoming independent of the iteration prior to reaching 70,000 iterations. The average becoming independent of iteration number indicates solution convergence. Note that Figure 22 includes Test Case 6—the adiabatic stainless steel plate—shown with orange squares.

For the results presented later in this chapter a solution at one specific iteration was selected for each TC. Figures 23 through 27 show the iteration selected for Test Cases 2-6 as well as the 2.5% margin referenced in Section 3.6. As TC1 did not display oscillatory convergence and the maximum difference in the f's for the final 10,000 iterations was $\ll 1\%$ the final iteration (100,000) was selected for TC1. The iterations for each TC along with the percentage differences from the mean $\overline{q''}_s$ or $\overline{T}_s$ are shown in Table 16.

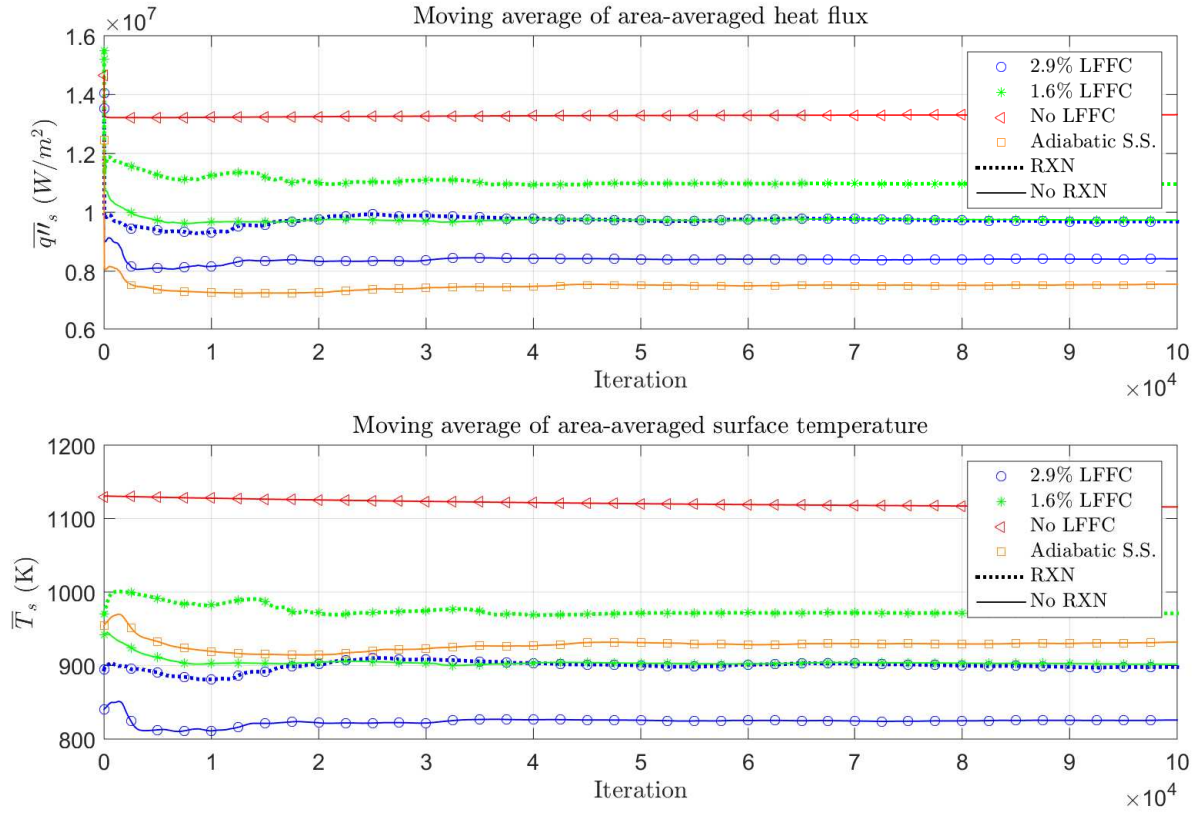

Figure 22. Moving average of solution monitors for all test cases.

Table 16. Selected iteration and difference from the mean.

Test Case	Iteration	$\Delta\overline{q''}_s$ (%)	$\Delta\overline{T}_s$ (%)
1	100,000	0.02	0.05
2	78,500	0.27	0.61
3	87,500	0.02	0.26
4	72,000	0.02	2.4
5	89,000	0.03	0.40
6	95,000	0.07	0.72

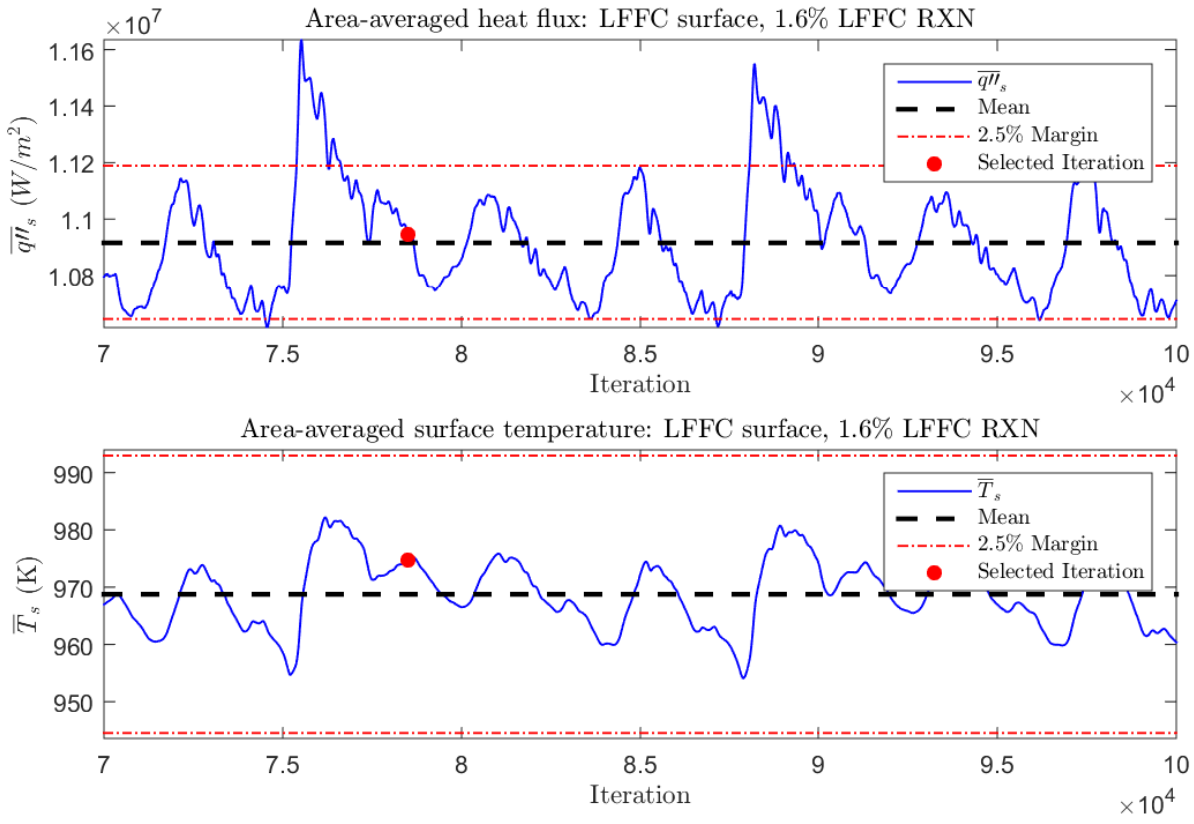

Figure 23. Solution selection for Test Case 2.

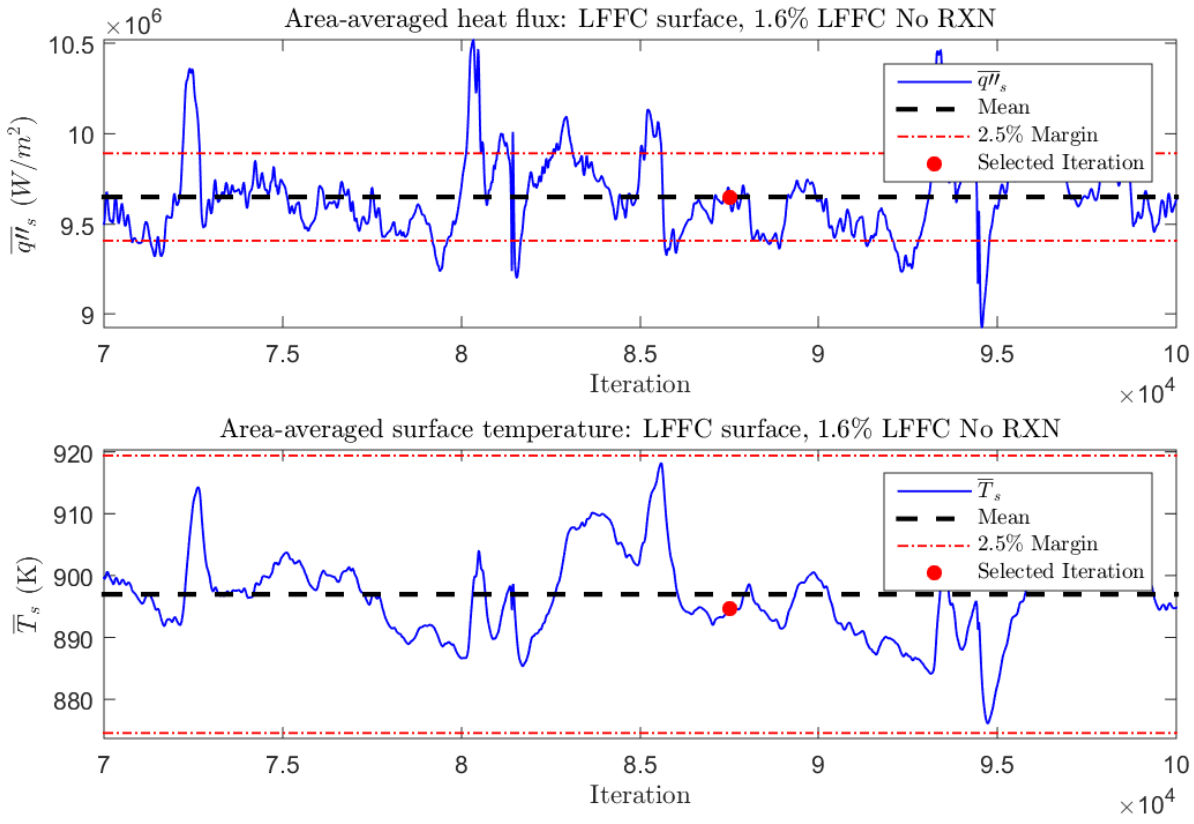

Figure 24. Solution selection for Test Case 3.

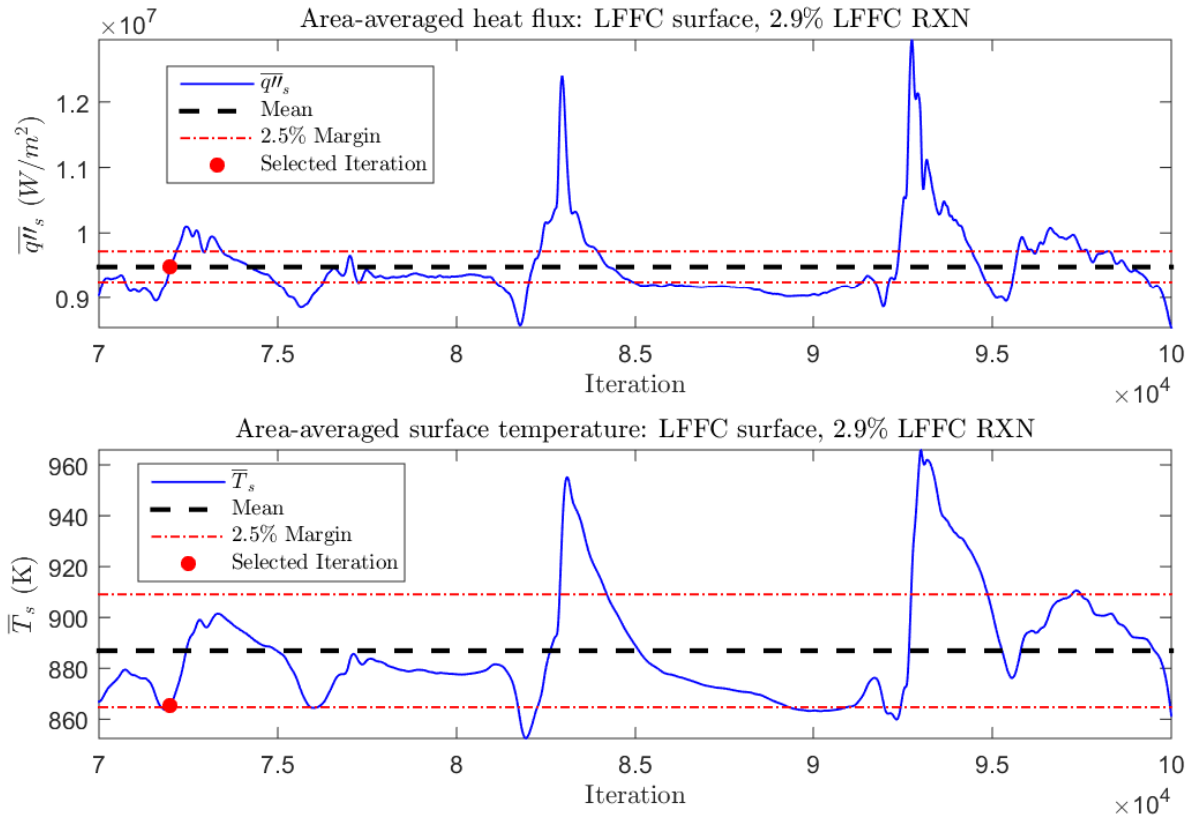

Figure 25. Solution selection for Test Case 4.

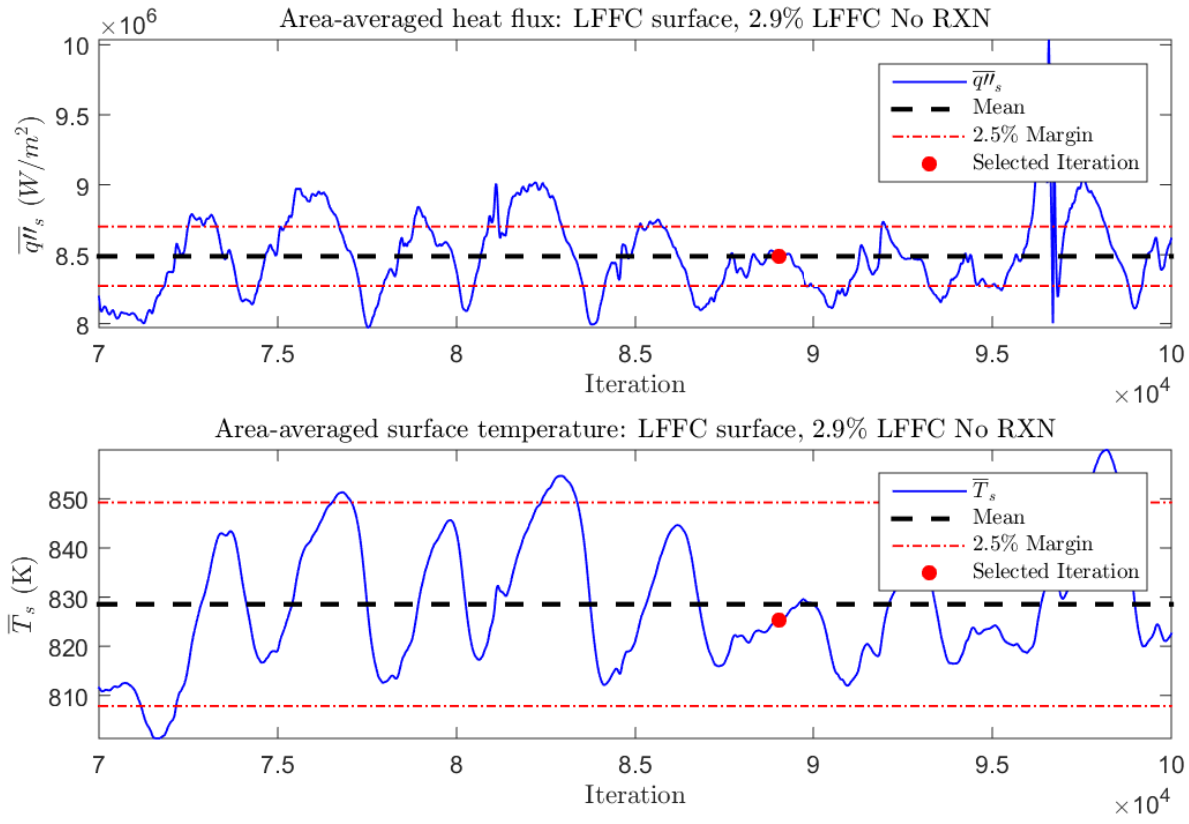

Figure 26. Solution selection for Test Case 5.

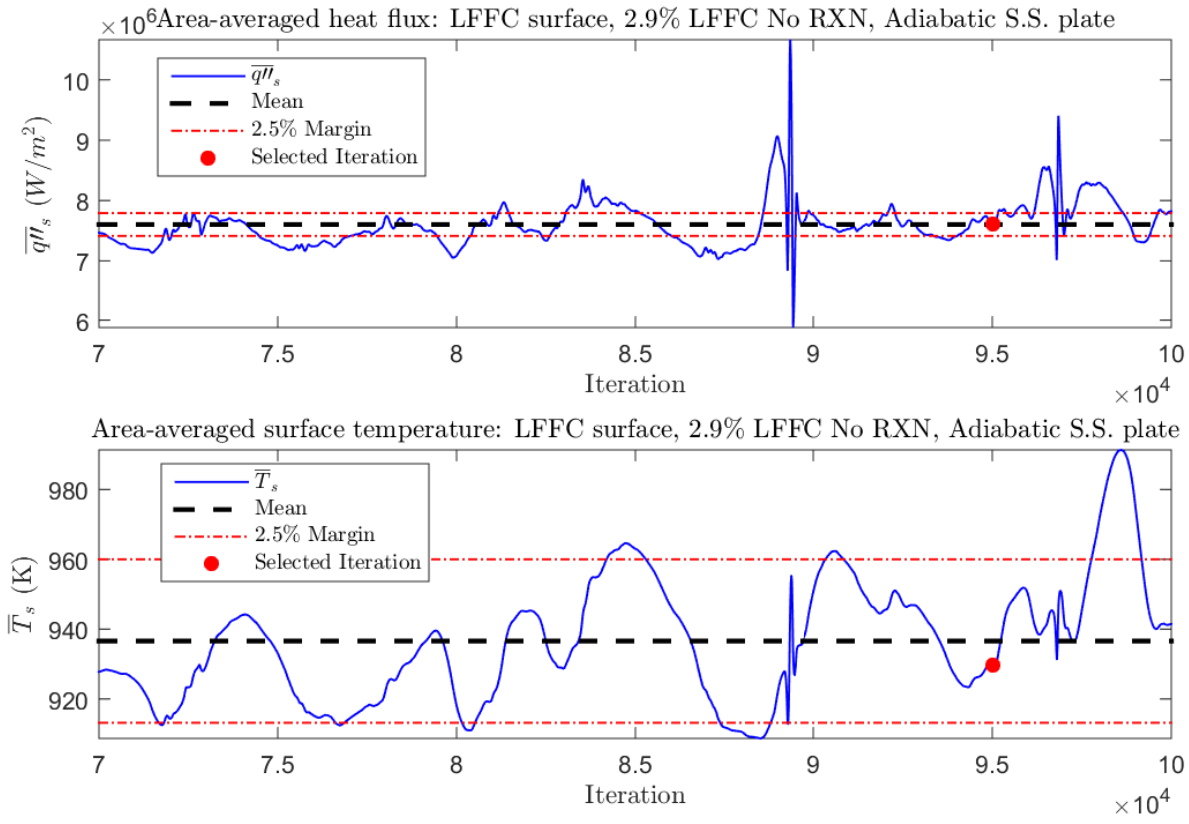

Figure 27. Solution selection for Test Case 6.

4.2.4 Stainless Steel Plate Geometry

In previous work the stainless steel plate has been assumed to be adiabatic [8]. Test Case 6 was used to test the effects of this adiabatic assumption. To make the stainless steel plate adiabatic in the domain all interfaces between the stainless steel and copper plates were set to zero-heat-flux boundary conditions. Figure 28 shows a close-up view of the interfaces between the two plates near the fuel injection slot (all other interfaces not shown were also set to zero-heat-flux). The two walls of the stainless steel plate that touched the fuel injection slot were also set to have zero heat flux.

Figures 29 through 31 show the surface profiles of temperature, heat flux, and overall effectiveness (Φ) between the two test cases. The x-axis is non-dimensionalized using the slot width (d). This is the same scale used in Figure 7. The upstream wall of the LFFC slot is located at $x/d = 0$ and and the downstream wall at $x/d = 1$. In Figure 29 the adiabatic assumption (TC6) shows a significantly higher surface temperature through the domain and a correspondingly lower Φ in Figure 30. The values for Φ is calculated using Equation 10 and are similar to the temperature profiles (but inverted vertically). A Φ of 1 would indicated perfect cooling while a Φ of 0 would indicate no cooling. The temperature difference ranges from $\sim$200 K upstream of the fuel injection slot to $\sim$50 K immediately down stream of the slot and over 100 K far downstream ($x/d \simeq$200).

Figure 31 shows the surface heat flux profiles for the two test cases. The adiabatic case (TC6) shows a slightly lower heat flux across nearly the entire surface. This figure is an example where heat flux data could be misleading to a designer. A lower heat flux could be interpreted as a lower temperature. However, Figure 29 shows that is not the case. The reason for the lower heat flux in Figure 31 is a smaller temperature gradient between the surface of the copper plate and the freestream.

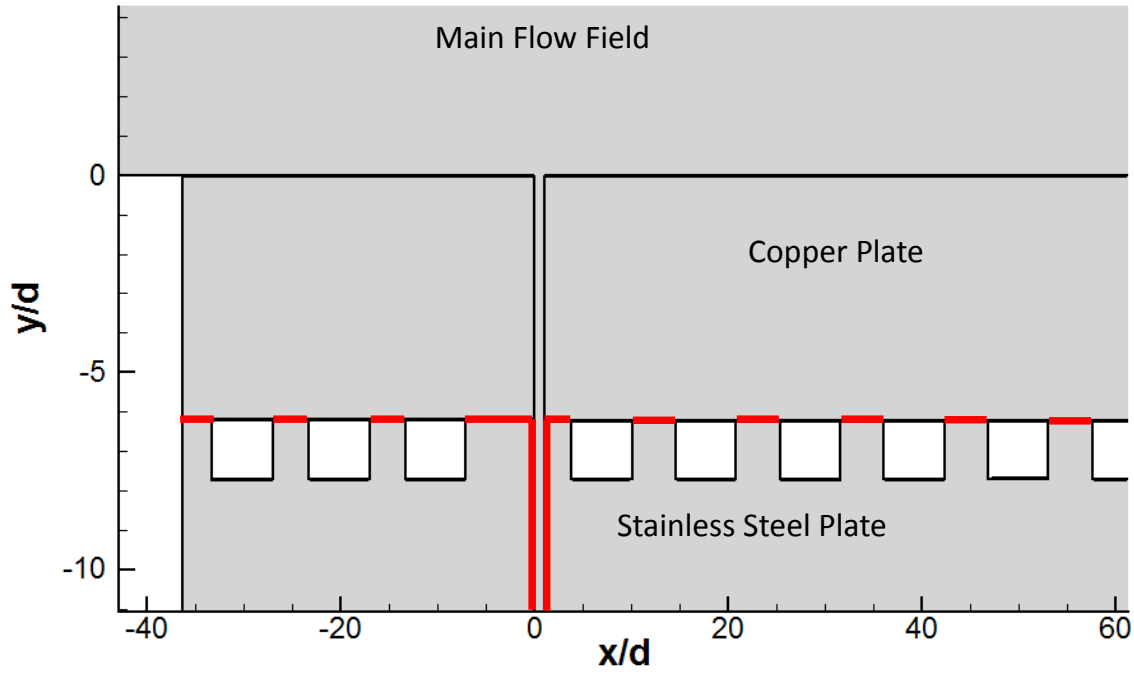

Figure 28. Locations of new zero heat flux boundaries (shown in red) to make the stainless steel plate adiabatic in Test Case 6.

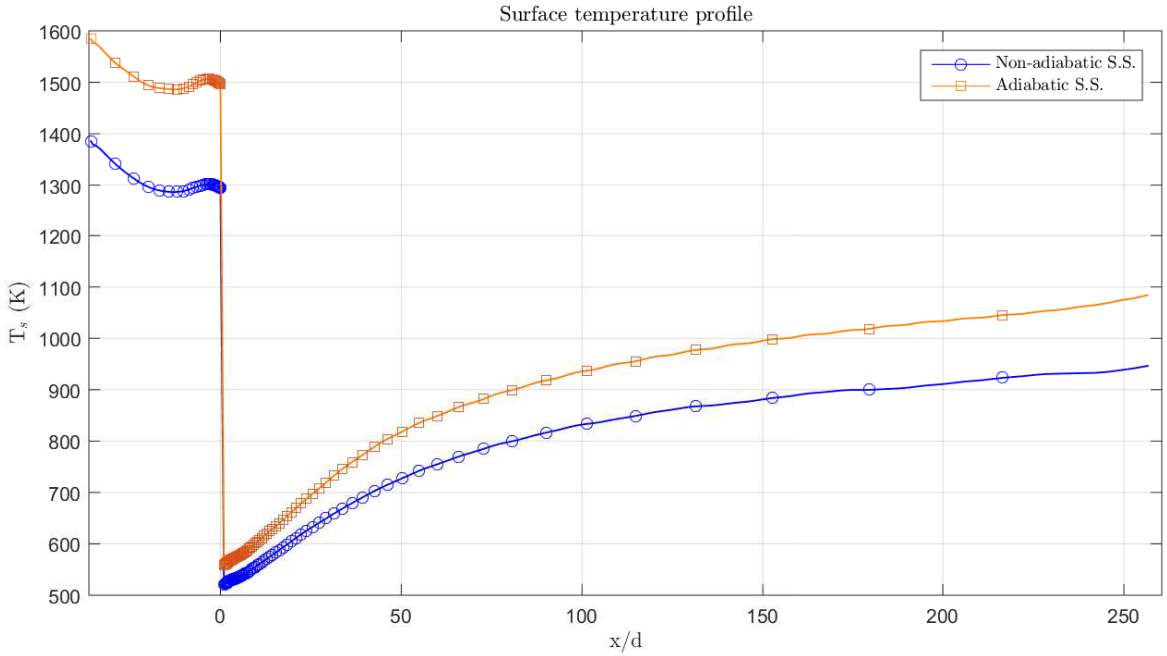

Figure 29. Surface temperature (T_s) profile comparison of the non-adiabatic stainless steel plate (Non-adiabatic S.S.) and the adiabatic stainless steel plate (Adiabatic S.S.). Test Cases 5 and 6.

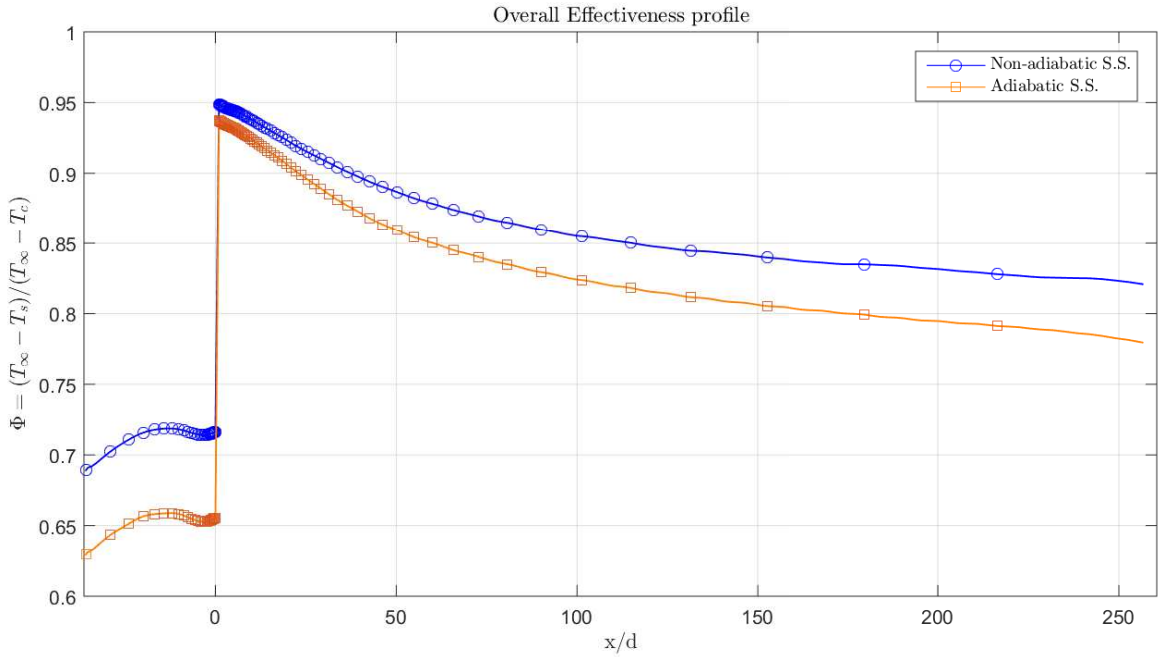

Figure 30. Overall Effectiveness (Φ) profile comparison of the non-adiabatic stainless steel plate (Non-adiabatic S.S.) and the adiabatic stainless steel plate (Adiabatic S.S.). Test Cases 5 and 6.

An increase in surface temperature drives down the difference with the freestream temperature, reducing heat flux.

From the surface temperature plot in Figure 29 it is shown that the stainless steel plate cannot be assumed to be adiabatic. This indicates that there are significant temperature gradients in the stainless steel plate, resulting in heat flow to the backside of the water cooling channels. Figure 32 shows the temperature contours between the two test cases with and without the adiabatic stainless steel plate. The top contour (TC6) shows the adiabatic plate (grey indicating zero heat flux across the boundaries). The bottom contour (TC5) shows the results of allowing heat flux across the interfaces and through the stainless steel plate. The thin dashed lines on the contours represent isotherms (lines of constant temperature). Heat flow is always perpendicular to isotherms [11]. Thus the bottom contour in Figure 32 shows heat flow passing across the copper/stainless steel plate interface and flowing into the sides

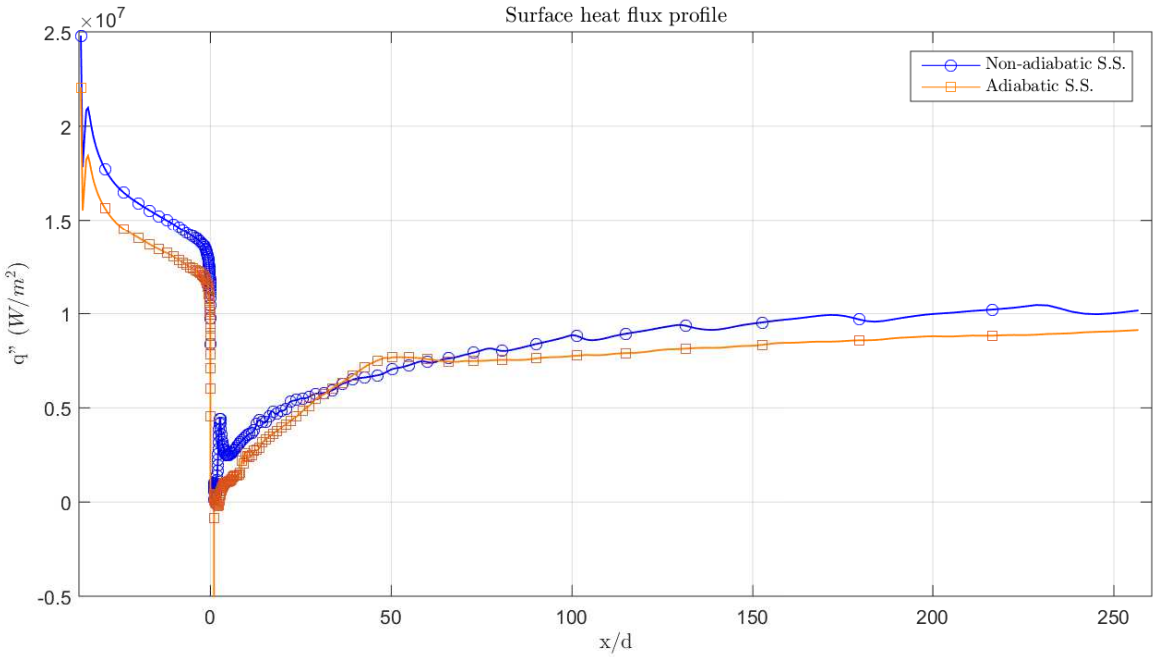

Figure 31. Surface heat flux (q'') profile comparison of the non-adiabatic stainless steel plate (Non-adiabatic S.S.) and the adiabatic stainless steel plate (Adiabatic S.S.). Test Cases 5 and 6.

of the cooling water channels as well as to the back side of the channels. An example of the heat flow vectors are shown in Figure 33. Temperature contour plots of all test cases are available in Appendix L.

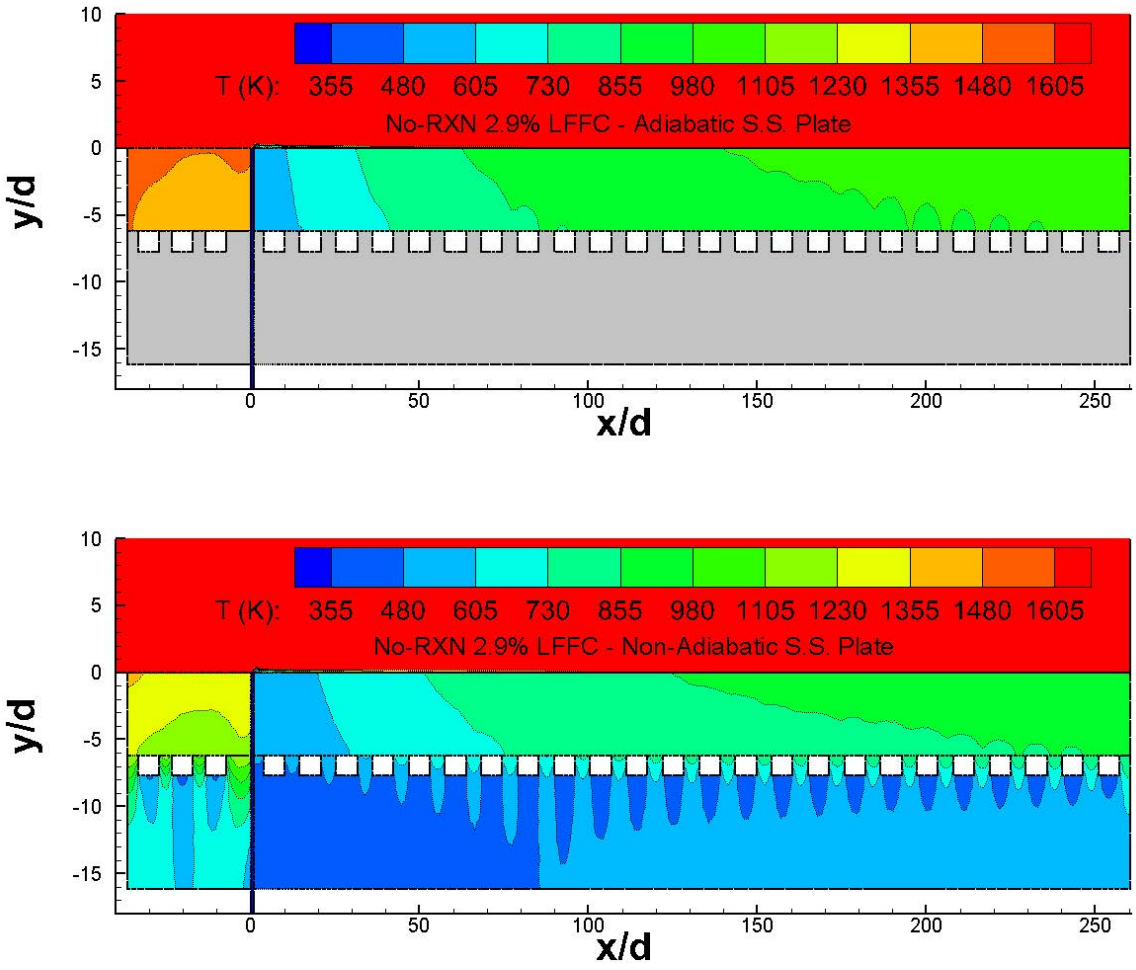

Figure 32. Temperature contour comparison of Test Cases 5 and 6.

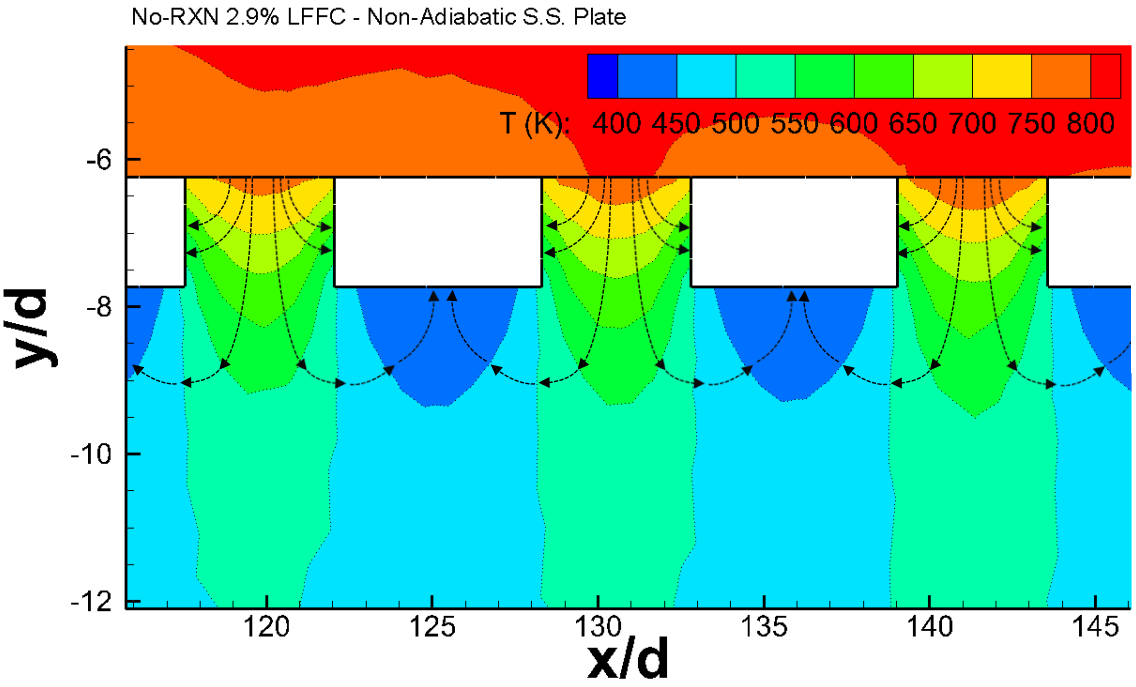

Figure 33. Example of heat flux vectors (dashed arrows) for Test Case 5. The heat flows perpendicularly to the isotherms (thin dotted contour lines).

4.2.5 Surface Plots

Figures 34 and 35 show the heat flux and temperature profiles on the surface of the copper plate (both LFFC and non-LFFC surfaces). For Figure 35, a positive heat flux represents heat flow into the copper plate. Thus an ideal case would show a heat flux (q'') profile either at or near 0 on the positive side of the axis.

In Figure 34, the temperature profiles show similar trends to the area-averaged surface temperature monitor in Figure 21. The non-LFFC surface ($x/d < 0$) shows the four cases with LFFC present (blue circles and green stars) have lower temperatures when compared to the case with no LFFC (red triangles). While all 5 cases have no LFFC upstream of the slot, the temperatures differ due to axial conduction through the upstream portion of the copper plate into the LFFC slot due to the cold fuel. This drives down the temperature on the non-LFFC surface for the cases with LFFC. The surface with LFFC ($x/d > 1$) shows the expected effects of film cooling. The

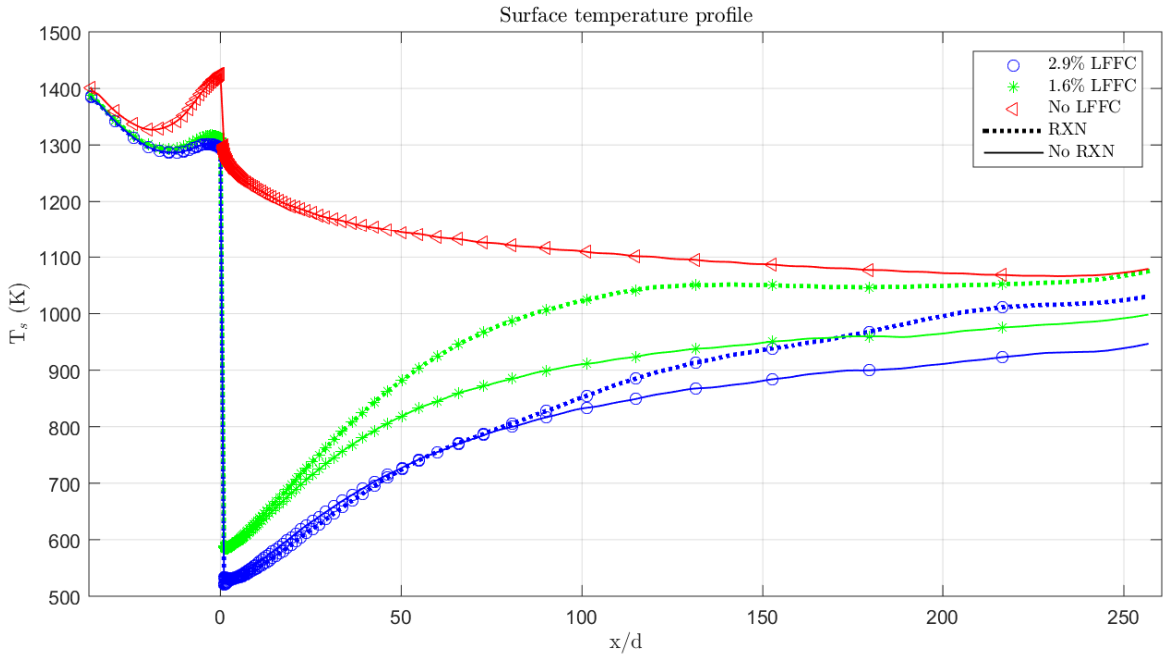

Figure 34. Surface temperature profile on copper plate, TC 1–5.

2.9% LFFC cases show the lowest temperatures while the case with no LFFC shows the highest temperatures. The cases with reactions (bold dashed lines) initially show similar temperatures to the non-reacting cases just downstream of the slot. However, as the flow moves further down the plate the fuel has time to react with the freestream, releasing heat and raising the temperature. By ∼200 slot widths downstream the effect of film cooling for the reacting 1.6% LFFC case is almost completely gone. At the exit of the domain the reacting cases are ∼75 K warmer than the non-reacting cases.

Figure 35 shows the q'' profile for the copper plate. Upstream of the slot on the surface of the copper plate that receives no film cooling ($x/d < 0$) the profiles of all cases are initially all roughly identical. As the profiles progress towards the slot the cases with film cooling move away from the one case with no film cooling. This is another effect of axial conduction due to the presence of the cold liquid fuel in the slot producing a larger temperature gradient, which drives heat flux. The case with

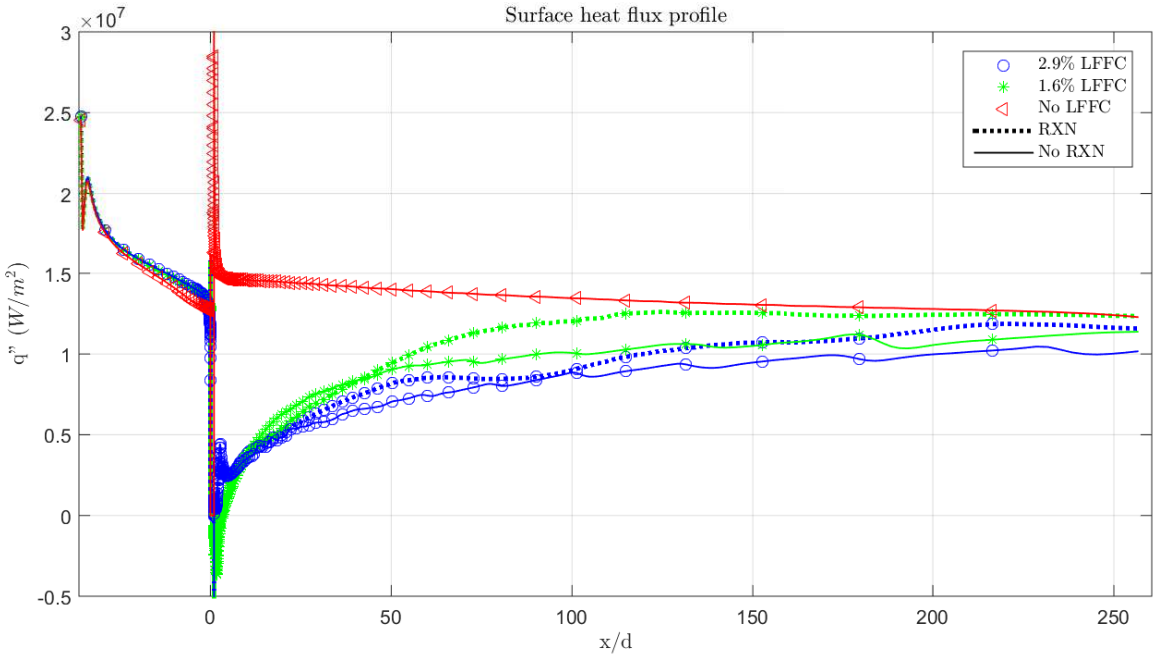

Figure 35. Heat flux profile on copper plate, TC 1–5.

no film cooling has a sharp positive spike in heat flux in the region near the slot. This is due to the slot being open to the flow with no liquid fuel to fill it. A recirculation region forms in the slot in the case with no film cooling resulting in an increase in temperature and heat flux. The cases with film cooling also show a spike near the slot but in the negative direction. The heat flux actually drops below zero briefly for all cases with film cooling. This indicates that heat is actually flowing out of the copper plate and into the surrounding fluid. This negative heat flux is corroborated by the temperature contours near the LFFC slot. Figure 36 shows the temperature contours for the reacting 1.6% LFFC case near the LFFC slot. The contours do show the copper plate has become slightly warmer than the surrounding fluid. The increase in the temperature of the copper plate in this region is likely due to a combination of 2D conjugate heat transfer allowing for the warming of the liquid fuel as it flows up the slot, viscous heating on the walls, and the recirculation region on the surface of the plate.

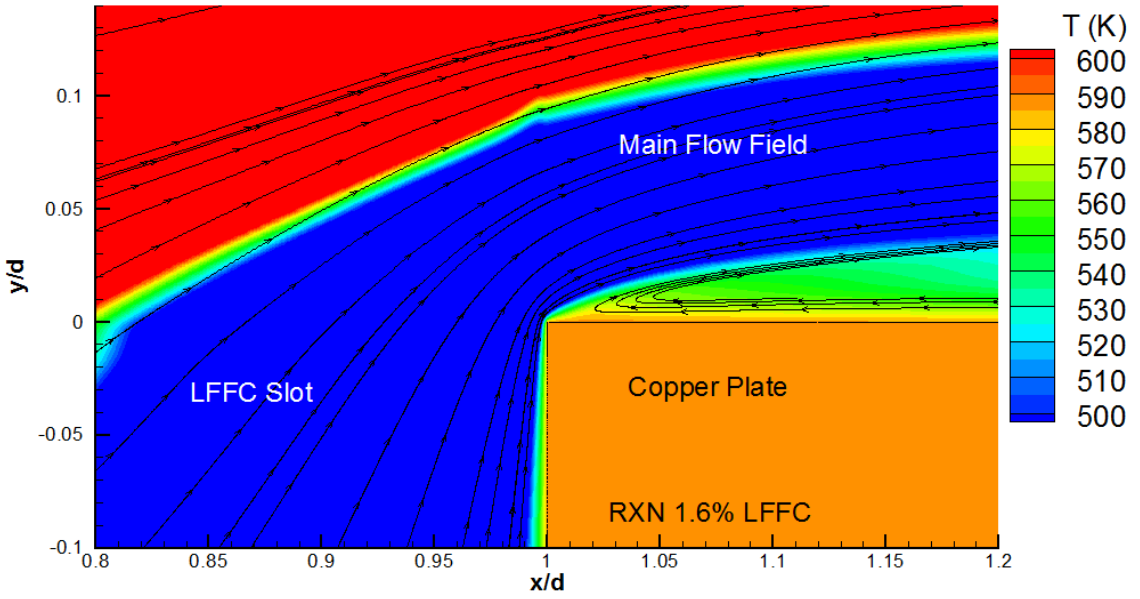

Figure 36. Temperature contour with streamtraces for Test Case 2 showing the section of copper plate that is warmer than the surrounding fluid.

The heat flux profiles in Figure 35 also display some undulations downstream of the injection slot for the cases with LFFC. This behavior is most apparent in the non-reacting 2.9% LFFC case downstream of $x/d \simeq 100$. One potential cause of this minor fluctuation in heat flux is the thickness of the liquid fuel film on the plate. Figure 37 shows the volume of fluid profile for the non-reacting 2.9% LFFC case. A VoF of 1 indicates pure freestream gas while a VoF of 0 indicates pure liquid kerosene. The profile in Figure 37 shows a relatively smooth contour in the region where the undulations occur in Figure 35 ($x/d >\sim 100$). The contour shows a gradual thickening of the fuel layer—likely due to the increasing boundary layer thickness—but does not show oscillations in the fuel layer thickness. The lack of oscillations in the fuel layer thickness in Figure 37 suggests that the undulations in Figure 35 are likely a product of the simulation being run to a steady state solution and are similar to the discussion for Figure 14. The solver is unable to reach an exact steady state solution. Thus it oscillates about some average solution—hence the need to select an

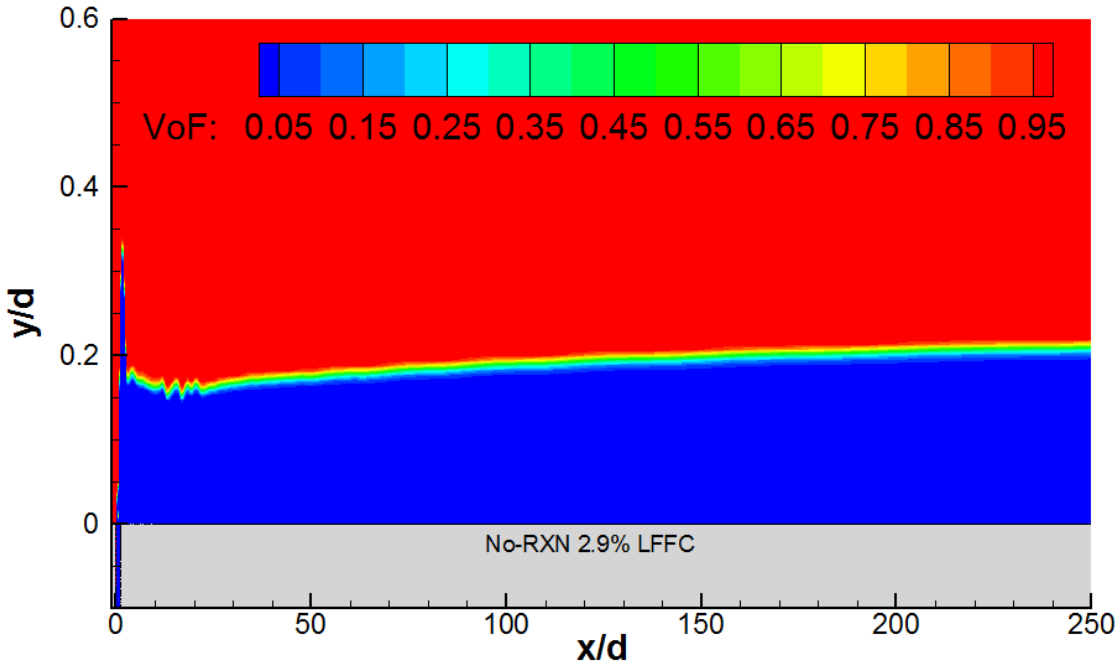

Figure 37. Volume of Fluid contour for non-reacting 2.9% LFFC (Test Case 5).

iteration that is near the average—which result in minor fluctuations in the surface profiles of temperature and heat flux.

Figure 38 shows the overall effectiveness (Φ) profiles for each test case. The cases with the most LFFC show the highest values of Φ (peaking at approximately 0.95) while the case with no LFFC shows the lowest (peaking at approximately 0.78). Note that the case with no LFFC does not have a Φ of 0. The no LFFC case still receives cooling from the water channels embedded within the stainless steel plate. Downstream of the LFFC slot, the no LFFC case shows an average Φ of $\sim$0.77. This shows that the majority of the cooling for all cases is being done by the cooling channels and points to their importance to being included in the computational domain. The no LFFC case also shows an increasing trend in Φ as it progresses downstream of the fuel injection slot while the cases with LFFC show a decreasing Φ over the same range. The reason for the increasing Φ in the no LFFC case is the surface temperature is gradually decreasing downstream of the fuel injection slot (see Figure 34). A decrease

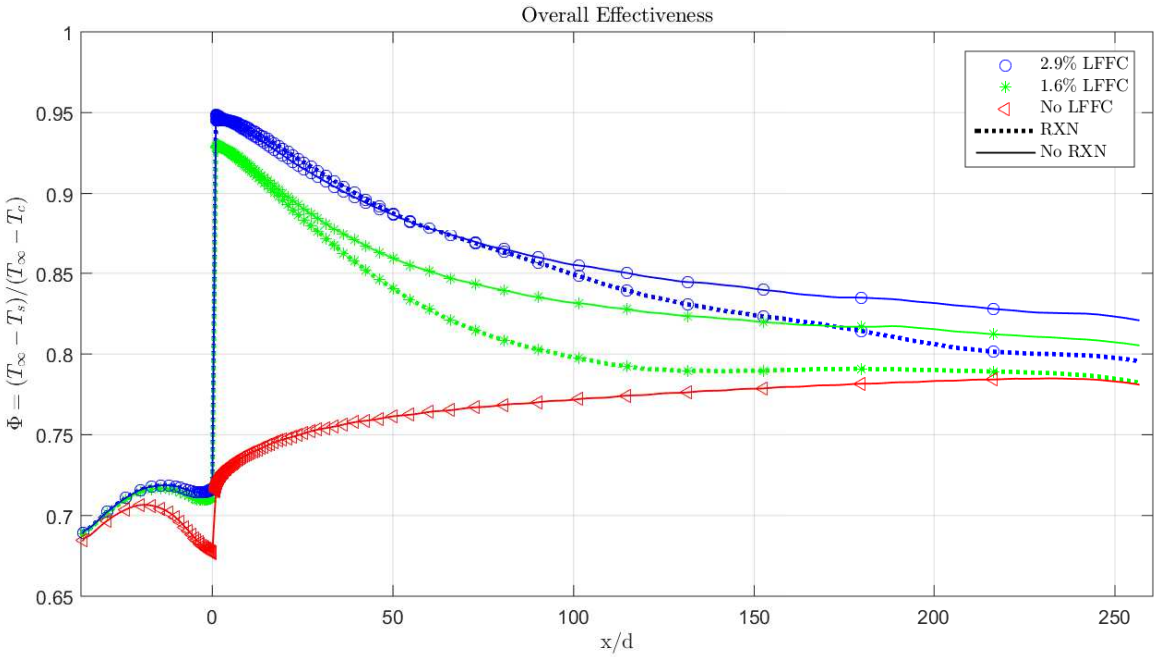

Figure 38. Overall effectiveness on copper plate. Test Cases 1–5.

in surface temperature will result in an increase in Φ. As the no LFFC case represents the baseline with no film cooling, the cooling produced exclusively by the LFFC can be determined by the difference between the no LFFC case (red triangles) and the LFFC cases (green stars and blue circles). Figure 39 shows the Φ due to only LFFC. The Φ profile of the no LFFC case has been subtracted from the LFFC cases. The information in Figure 39 can be used to determine when the introduction of a second fuel injection slot would be appropriate. For example, the reacting 1.6% LFFC case shows a Φ of nearly 0 at an $x/d = 100$. Thus a second fuel injection slot could be appropriate at $x/d = 100$ for the reacting 1.6% LFFC case.

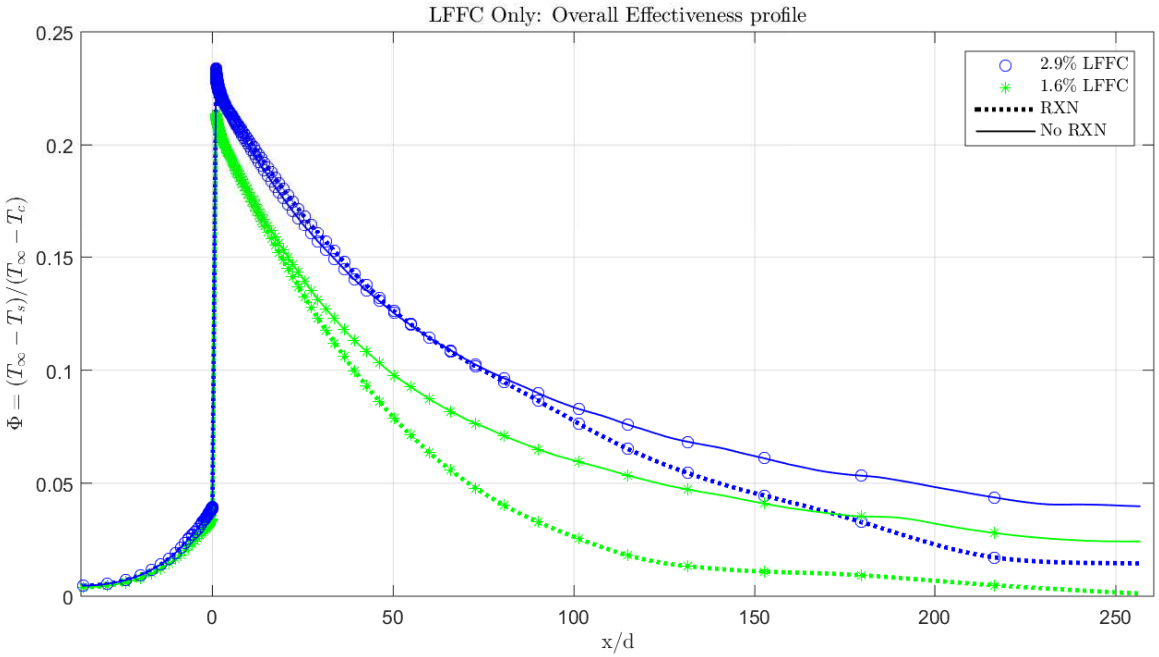

Figure 39. Overall Effectiveness on copper plate due to only the LFFC (Test Cases 2–5). The baseline no LFFC contribution to Φ has been subtracted out.

4.2.6 Temperature Contours

The x-y plots in Figures 34, 38, and 39 show the results of the reacting and non-reacting simulations in one dimension. Temperature contour plots expand the information to 2D and show the affects of the heterogeneous reactions propagating into the copper and stainless steel plate. Figures 40 and 41 show the temperature contours for the reacting and non-reacting 1.6% LFFC cases. In Figure 40 the heterogeneous reactions cause the surface temperature to reach 950 K at the upper surface (adjacent to freestream) around $x/d \sim 70$ and $x/d \sim 110$ on the bottom surface (adjacent to stainless steel plate). In the absence of heterogeneous reactions, Figure 41, the upper surface does not reach the 950 K mark until $x/d \sim 200$ while the bottom surface does not reach 950 K at any point downstream of the fuel injection slot. The same trend is visible in the 2.9% LFFC case. Temperature contours, including Φ contours, for Test Cases 1–5 are shown in Appendix L.

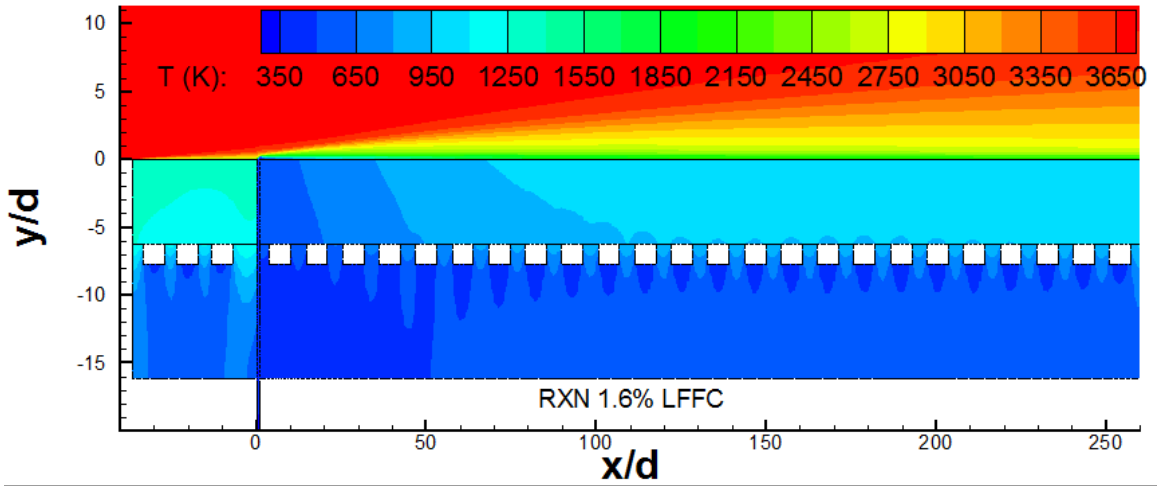

Figure 40. Temperature contour for the reacting 1.6% LFFC case (Test Case 2).

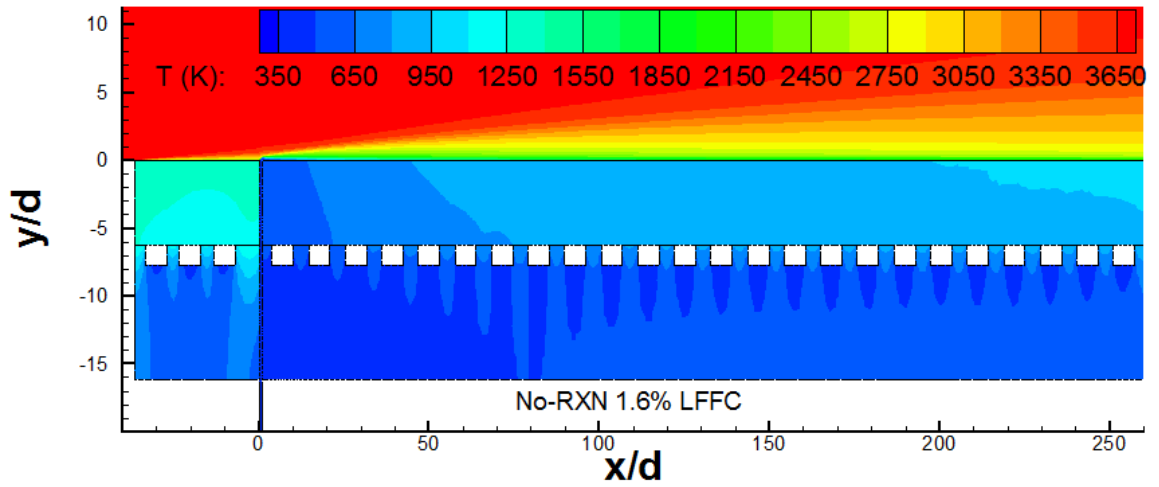

Figure 41. Temperature contour for the non-reacting 1.6% LFFC case (Test Case 3).

4.2.7 Heterogeneous Reactions

While the VoF contour for the non-reacting 2.9% LFFC case in Figure 37 shows a relatively constant fuel layer thickness with a well defined interface between the fuel and freestream, the respective reacting case reveals one of the results of the introduction of the heterogeneous reactions. Figure 42 shows the VoF contour for the

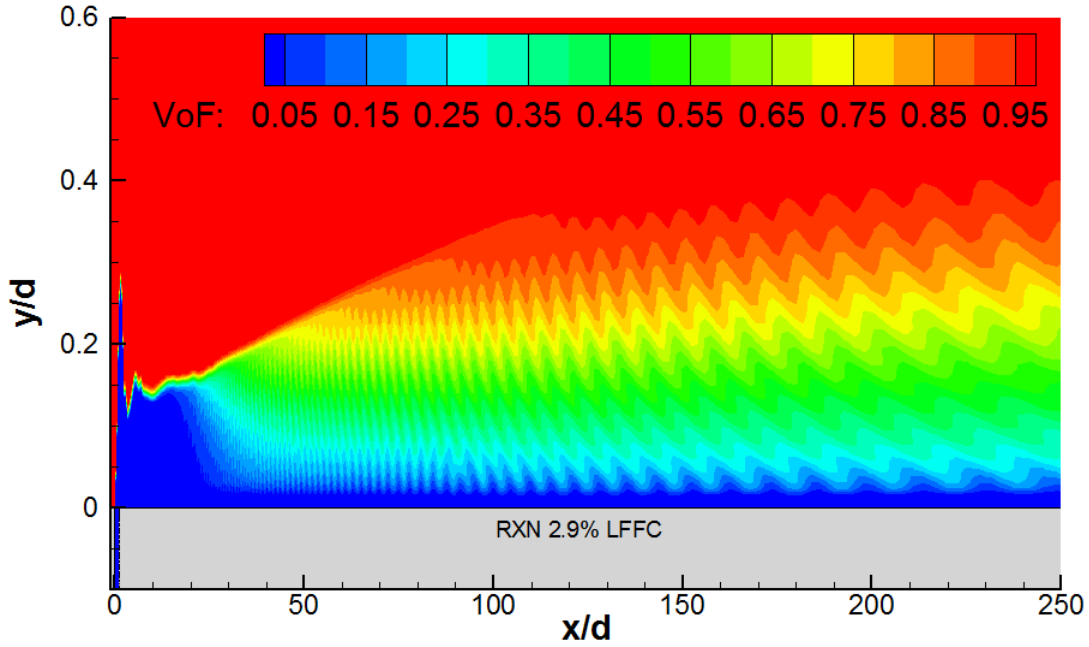

Figure 42. Volume of Fluid contour for reacting 2.9% LFFC (Test Case 4). A VoF of 1 is pure freestream gas, a VoF of 0 is pure liquid kerosene.

reacting 2.9% LFFC case. As with the the non-reacting case, the reacting case shows a fairly thick layer of fuel immediately down stream of the injection slot ($x/d <\sim 20$). However, once the fuel travels sufficiently far downstream it begins reacting with the freestream and the VoF contour shows the layer of pure kerosene (blue) becomes far thinner ($y/d \simeq 0.025$) than what is shown in Figure 37 ($y/d \simeq 0.2$). The interface between the fuel and freestream also becomes much thicker and less defined, demonstrating the change in phase that is occurring due to the heterogeneous reactions. The liquid kerosene is reacting with the free O_2 in the freestream and

becoming CO and H_2. The VoF contours also show periodic behavior which could be another artifact of a steady-state simulation (as opposed to a transient simulation). VoF contours of all test cases are available in Appendix M.

The effects of the heterogeneous reactions are also visible in contours of the density. Figure 43 shows the density contour for the non-reacting 2.9% LFFC case. The same sharp interface between the liquid kerosene—which has a high density compared to the freestream—is visible as with the VoF contour in Figure 37. When the heterogeneous reactions are enabled—shown in Figure 44—the sharp interface is replaced by a much larger interface with a smooth transition from the low density freestream ($\sim$10 kg/m^3) to the high density liquid kerosene ($\sim$780 kg/m^3). As with the VoF contours, density contours of Test Cases 1 through 5 are available in Appendix N.

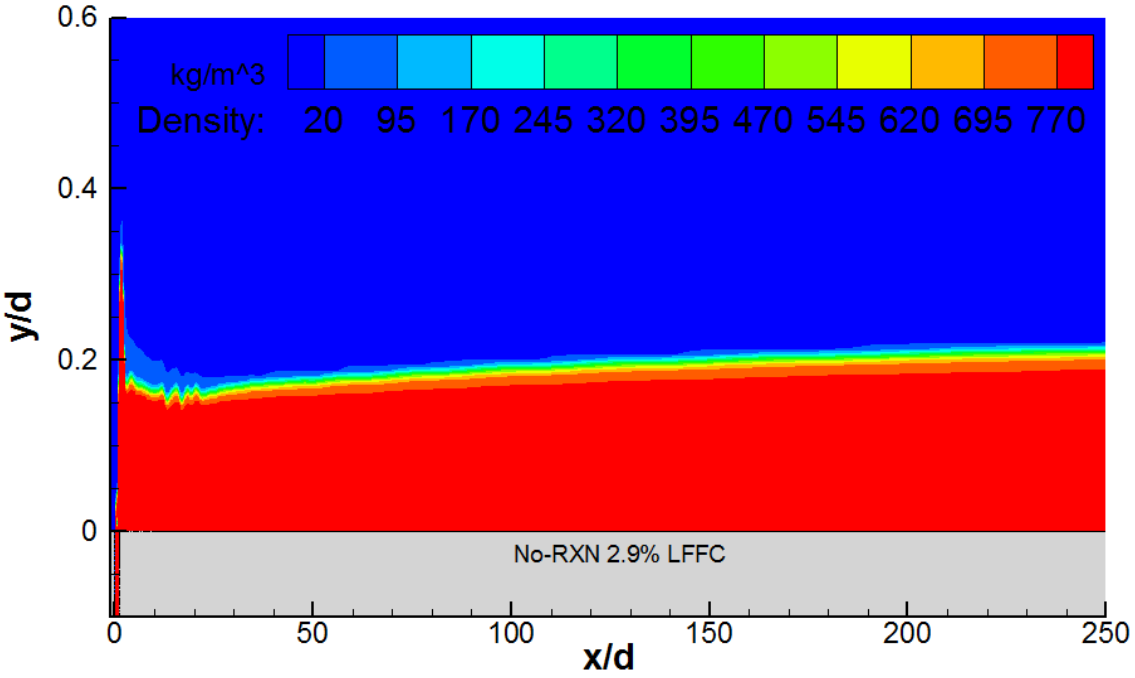

Figure 43. Density contour for non-reacting 2.9% LFFC (Test Case 5).

Contours of the heat released due to the heterogeneous reactions are shown in Figures 45 and 46 for the reacting 2.9% and reacting 1.6% cases. The heat released is focused at the interface region between the liquid kerosene and gaseous freestream.

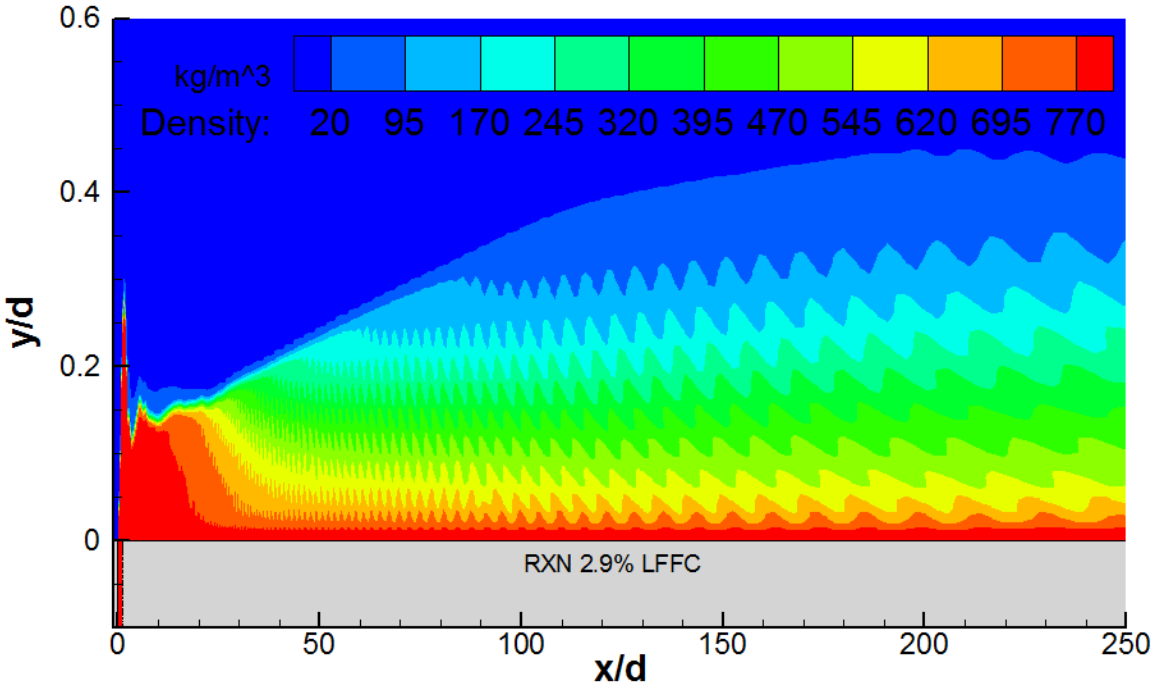

Figure 44. Density contour for reacting 2.9% LFFC (Test Case 4).

The reacting 2.9% case (Figure 45) shows the heat release at a larger y/d value (~ 0.5) when compared to the 1.6% case (Figure 46) due to the increase in the LFFC flow rate.

Figures 47 and 48 show the reaction rates, in kmol/m^3/s, for the two heterogeneous reactions for the reacting 2.9% case (TC4). Figure 47 shows the paraffin reaction has the highest rate not far downstream of the fuel injection slot ($x/d \simeq 25$). In comparison, the naphthene reaction rate has a small region of high reaction rate at roughly the same x/d as the paraffin reaction, but the naphthene reaction has the highest rates far downstream ($x/d > 150$). Note that the high reaction rates at the $x/d \simeq 25$ location correspond to where the interface between the liquid and freestream begins to thicken in Figure 42. The napthene reaction rate also shows much higher reaction rates, reaching a maximum of approximately 150 kmol/m^3/s. The paraffin reaction rate reaches a maximum rate of approximately 80 kmol/m^3/s, almost half that of the naphthene rate. The reacting 1.6% case shows the same trend with the

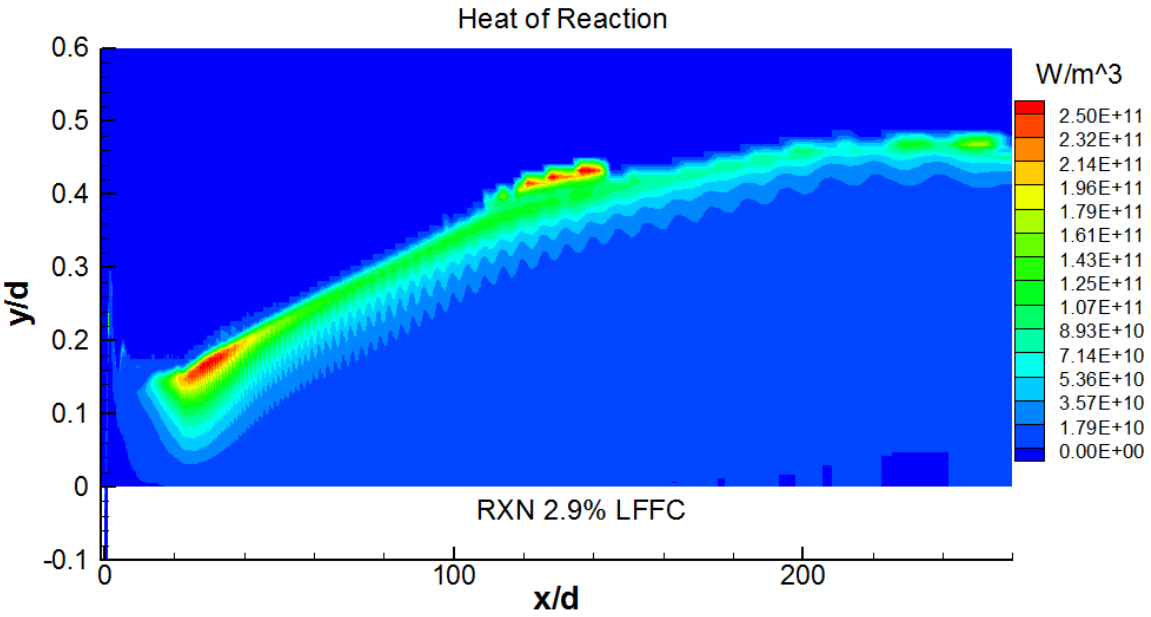

Figure 45. Heat of reaction (W) for reacting 2.9% LFFC (Test Case 4).

naphthene reaction occurring at a much faster rate than the paraffin, and farther downstream from the fuel injection slot. Contour plots of the reaction rates for both test cases are included in Appendix O.

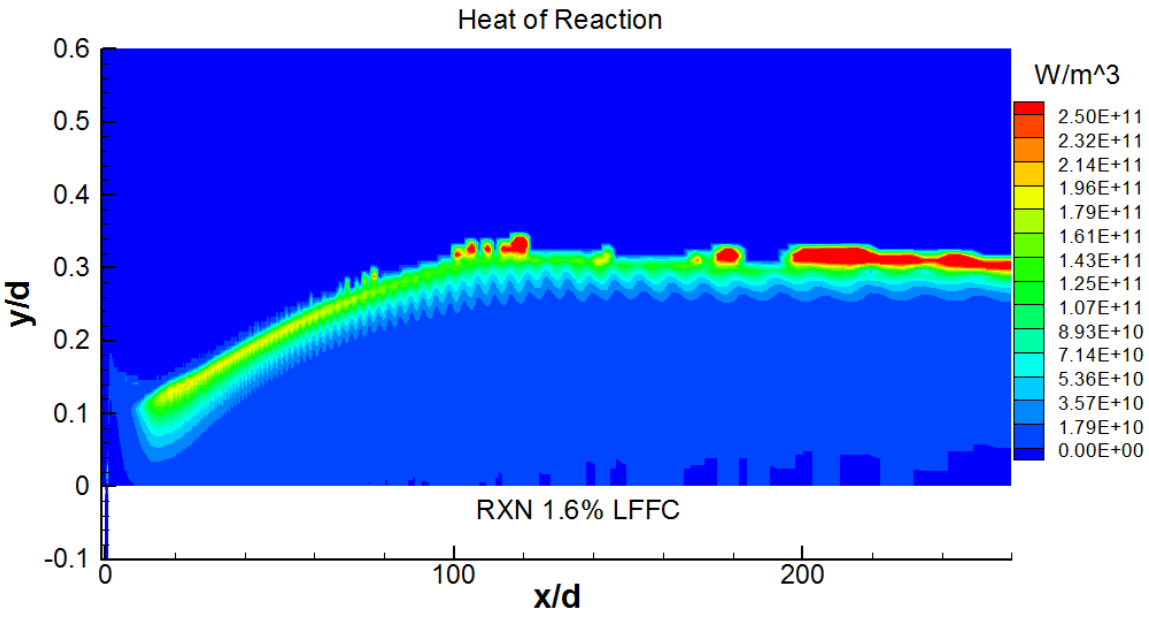

Figure 46. Heat of reaction (W) for reacting 1.6% LFFC (Test Case 2).

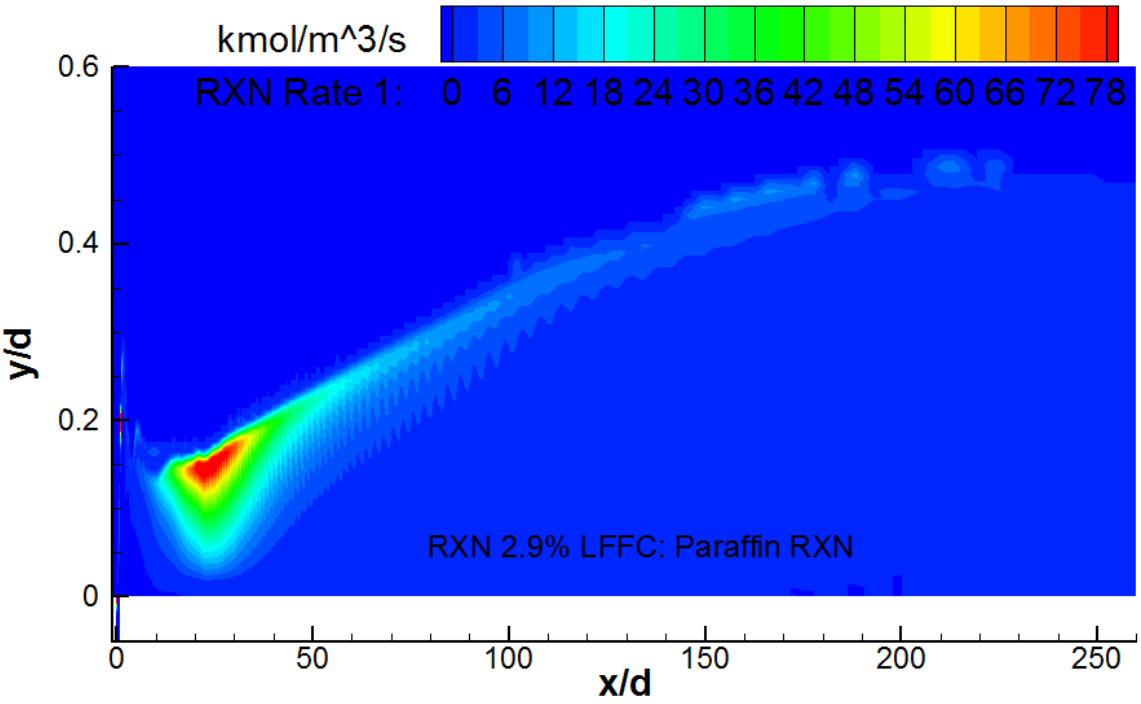

Figure 47. Reaction rate (kmol/m^3/s) of paraffin reaction for reacting 2.9% LFFC (Test Case 4).

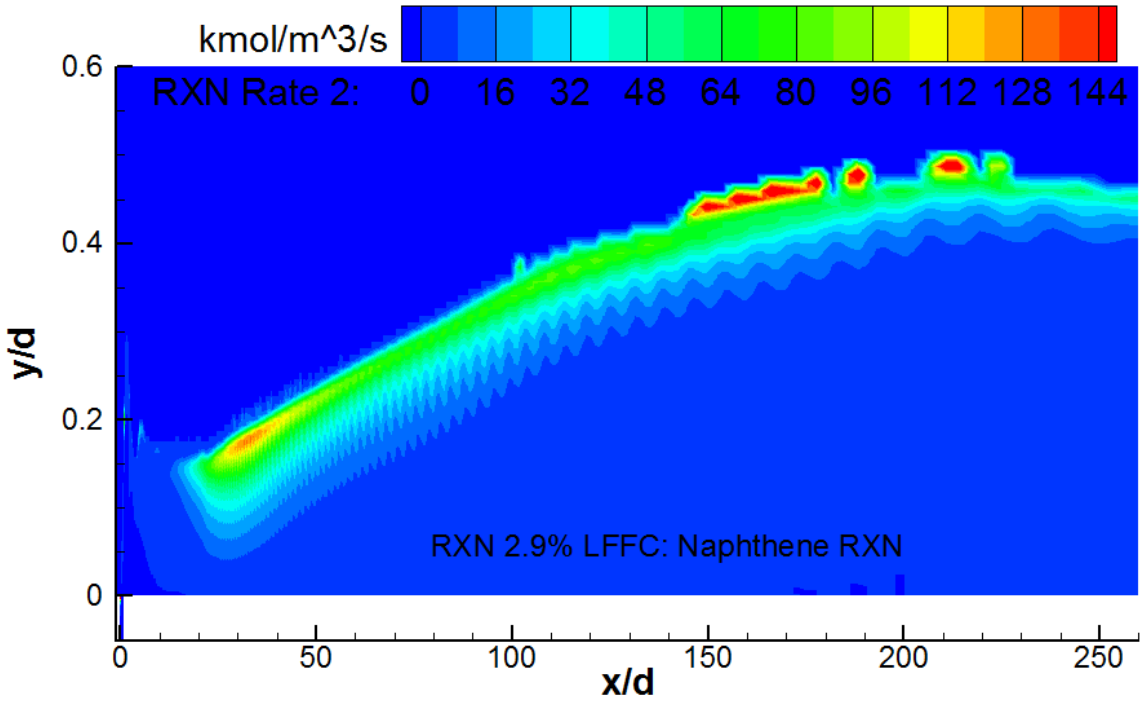

Figure 48. Reaction rate (kmol/m^3/s) of naphthene reaction for reacting 2.9% LFFC (Test Case 4).

V. Conclusions and Future Work

5.1 Summary

In Chapter I the concept of LFFC in LREs was introduced. The reasoning behind the use of LFFC as well as the motivation, background, and research objectives were also discussed. Chapter II delved more deeply into previous work done in the realm of LFFC in LREs with hydrocarbon fuel. Chapter II also explained key components of the research and the theory behind CFD simulations.

Chapter III presented the methodology used to develop the numerical model. The three steps of the CFD process were discussed along with the computational domain. The grid generation process was detailed as well as the grid convergence study. The specifics for implementing the various physics into the numerical model were also presented for multiphase flow, heterogeneous reactions, and conjugate heat transfer. Finally, the methods used for solution initialization and solution convergence were explained.

Chapter IV presented the results of the CFD simulations using the methodology presented in Chapter III. The results of the grid convergence studies for both the fluid and solid domains were detailed along with the solution convergence/selection results. The results of the six test cases in Table 12 were shown and analyzed for the three different LFFC% (0, 1.6, and 2.9) with and without the heterogeneous reactions. The results of the adiabatic stainless steel plate assumption were also presented.

5.2 Conclusions

From the grid convergence results it was found that attempting to model LFFC as a steady solution for the problem presented in this research produced large fluctuations in the simulation monitors. The solver is attempting to converge on a single

steady solution when none exists as the phenomena is inherently unsteady. This results in a solution that fluctuates around some average solution. Using a highly refined grid can help to dampen these fluctuations (Figure 15), but *care is needed when selecting an iteration from which to analyze a solution.*

The inclusion of the stainless steel plate geometry was found to be significant to the computational domain. The amount of heat flow through the stainless steel plate into the back side of the water cooling channels resulted in a ∼200 K difference upstream of the fuel injection slot and over 100 K far downstream ($x/d \simeq$200). In addition to the temperature difference, the overall effectiveness (Φ) for the case with no LFFC was ∼0.75 indicating that *the cooling channels are providing the vast majority of the cooling present in the computational domain.* This large value for Φ in the no LFFC case further demonstrates the need to include the heat flow through the stainless steel plate. The heat flux plots for the test cases used to compare the adiabatic stainless steel plate (TC5 and TC6) showed the need for analysis of both heat flux and temperature. Assuming the stainless steel plate was adiabatic resulted in a lower heat flux through the copper plate. *This lower heat flux could potentially mislead initial designs of a potential LRE as the lower heat flux actually corresponds to a higher temperature.* It is the surface temperature that is ultimately the deciding factor for the performance of a particular cooling scheme.

The results of the three different LFFC amounts shown in Figures 34 and 38 show that *as the amount of LFFC increases, the surface temperature decreases (*Φ *increases*). For the reacting cases, the 1.6% LFFC case provided no significant cooling beyond $x/d \simeq 150$ while the 2.9% LFFC provided cooling till $x/d \simeq 225$. *The locations where adequate cooling is no longer being achieved could be used to determine the locations of extra LFFC slots.* Both of the non-reacting cases with LFFC showed cooling through the entire domain due to the lack of heterogeneous reactions.

The inclusion of the heterogeneous reactions through the UDF resulted in an increase in surface temperature (Figure 34) due to the heat release at the interface of the liquid fuel and gaseous freestream (Figures 45 and 46 and Appendix O). *The result was a smoothing out of the interface between the two phases.* The cases without the reactions show a very sharp, thin, clear interface between the liquid and gas while the cases with reactions show a much thicker interface as the kerosene reacts with free O_2 molecules to form CO and H_2 (Figure 151 in Appendix M).

5.3 Research Objective & Investigative Question

In Chapter I the primary objective and investigative question were presented for this research. The primary objective of the CFD study presented in this thesis was to build upon what has been done by AFRL and create a new numerical model implementing the following physics:

1. Multiphase Flow
2. Heterogeneous Reactions
3. Conjugate heat transfer through stainless steel plate

The numerical model that was developed and presented did meet the objective of developing the foundation of a new numerical model and including the desired physics. Conjugate heat transfer, multiphase flow, and a simplified model of non-equilibrium chemistry (heterogeneous reactions) were successfully integrated. The overarching investigative question was:
What are the driving parameters for heat transfer in the presence of LFFC?
More work will likely be required to successfully answer the investigative question. It is very likely that radiation and a well-developed multiphase coking model will need

to be implemented along with expanding the domain to 3D. These additions to the numerical model are discussed in the Future Work section.

5.4 Future Work

At the time of this writing initial results of the physical experimental setup being replicated in the CFD simulations should become available in the near future by AFRL. One of the first things to accomplish for future work would be comparisons of experimental results with CFD. The inclusion of experimental data can better inform the numerical model.

The physical apparatus has a main flow channel that is approximately 25 mm square. Affects of the side walls and corners of the channel have the potential to affect the flow. The computational domain used in this research was 2D. Extending the domain to 3D could uncover the affects, if any, of the corners and sidewalls. Expanding the computational domain to 3D will also enable the simulation of the water flow through the cooling channels. As the cooling channel flow is perpendicular to the main flow field flow it was not able to be simulated in the 2D domain used in the final simulations presented in Chapter IV.

Simulating the conjugate heat transfer through the stainless steel plate with the cooling channel geometry demonstrated that it is important to the computational domain. The cooling channel plate has two more components (both stainless steel) located behind it in the physical experiment (Figure 5, page 32). Ensuring the heat flow through these two extra components is negligible may be done by expanding the computational domain in future simulations.

Oscillations of the solution monitors over the convergence history were presented in Chapter IV. The oscillations necessitated a specific methodology for selecting an iteration for analysis. Transitioning to a transient (i.e. time-accurate) simulation

could remove the oscillations in the solution and examine the unsteady phenomena.

In Himansu et al., radiation is called out as an important phenomena for the problem presented in this research [8]. Future work should include the affects of radiation. Initial simulations for the work presented in previous chapters did attempt to include the affects of radiation using the P1 model and information from Reference [8] but resulted in the simulations diverging.

The kerosene combustion mechanism used to model the heterogeneous reactions (Reference [26]) also includes a series of homogeneous elementary reactions. These reactions occur between the species in the freestream mixture. To further enhance the combustion within the numerical model these homogeneous reactions could be included. More research into their numerical stability would be required. An attempt was made at including them in the final numerical model using Arrhenius rate functions within Fluent. The forward reactions were successfully implemented but the reverse reactions resulted in instability and caused divergence. Using only the forward reactions was not realistic as they consumed all free O_2 in the immediate vicinity of the main flow field inlet. Thus there was no free O_2 remaining to react with the liquid kerosene, halting the heterogeneous reactions.

Finally, another item for future work that affects radiation and the kerosene combustion mechanism is the deposition of carbon (soot) on the walls. In Chapter I it was mentioned that these carbon deposits have the potential to further insulate the wall from the combustion gases and could be an important factor for LRE design. The carbon deposits would also be an important contribution for radiation. The formation and oxidation (destruction) of soot is also included as two separate heterogeneous reactions in Reference [26]. Inclusion of these reactions in Fluent would require the use of wall surface reactions for deposition onto the walls, and the formation of solid carbon particulate (C_s) from the breakdown of the kerosene hydrocarbon

chains. This would likely need to be done in conjunction with experimental work as some inputs, such as particulate diameter, would require knowledge from experimental measurements.

Appendices

Appendix A. Dimensionless Wall Distance (y^+) Plots: Final Test Cases

Figures 49 and 50 show the y^+ values on the surface of the copper plate and on the fuel injection slot walls. The plots for the surface of the copper plate show a y^+ value of approximately 0.4 upstream of the fuel injection slot. Downstream of the slot y^+ drops to roughly 0.01. This change in y^+ is due to the density change from the freestream gas adjacent to the wall upstream of the slot to the liquid kerosene downstream of the slot. Test Case 1 (no LFFC) stays nearly constant just over 0.4. The y^+ plots for the fuel injection slot walls are all approximately in the 0.01 range (except TC1, no LFFC). The small values of y^+ in both figures shows that the wall spacing could be increased in future simulations to reduce the number of grid cells and reduce the computational time required to perform the simulations.

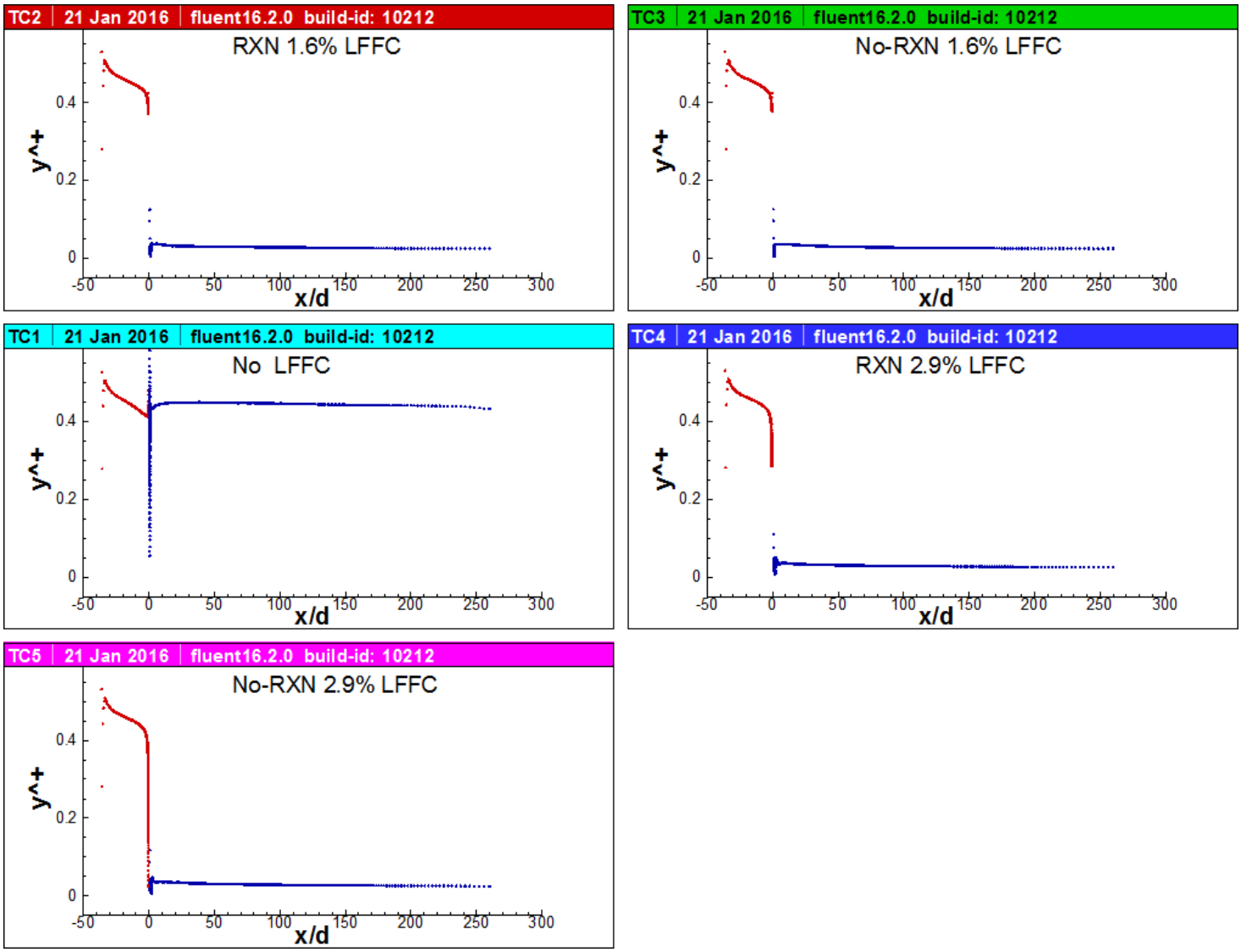

Figure 49. Dimensionless wall distance (y^+) values on the surface of the copper plate for Test Cases 1–5.

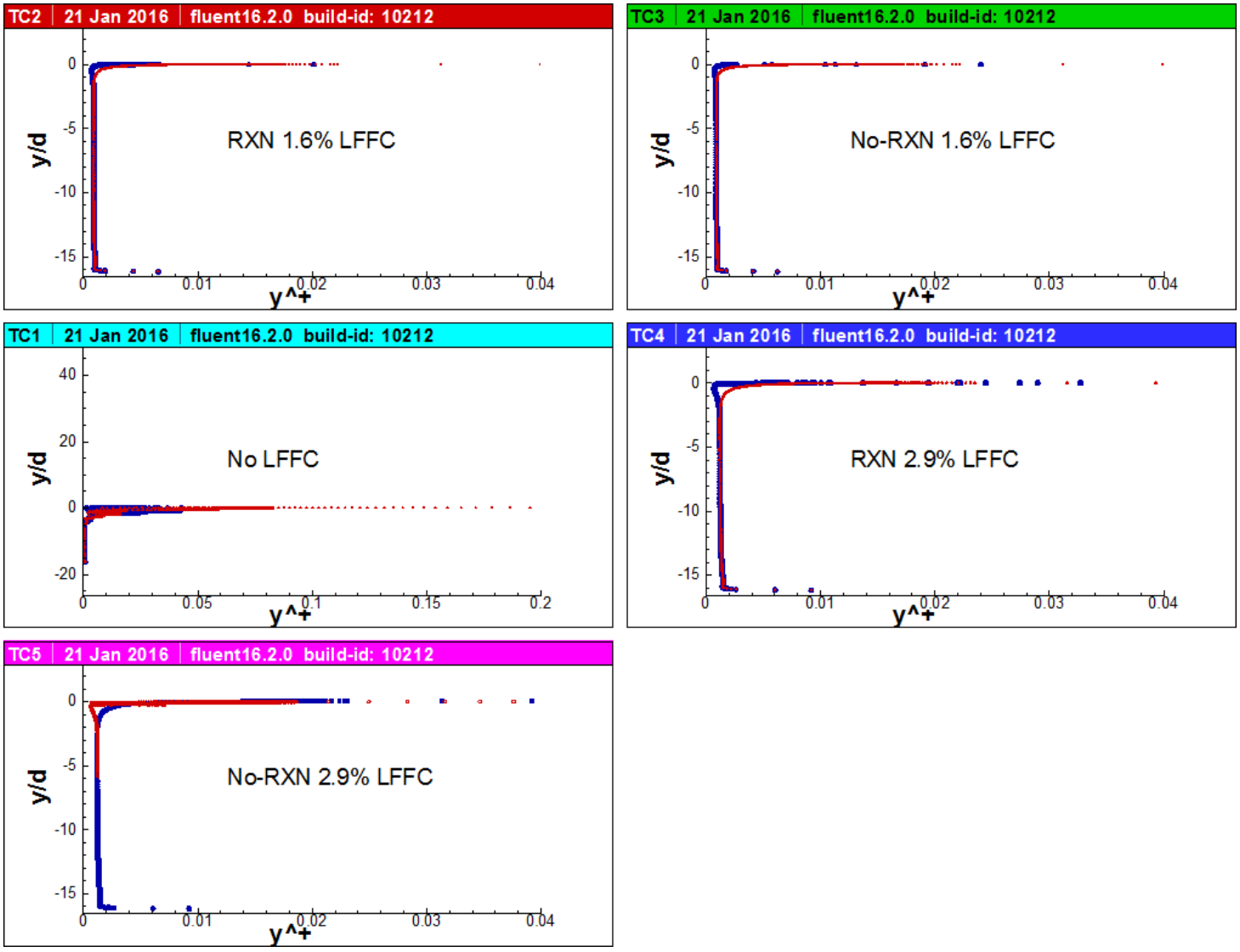

Figure 50. Dimensionless wall distance (y^+) values on the fuel injection slot walls for Test Cases 1-5.

Appendix B. Fluent Settings

The following images are the fluent settings used for the reacting 2.9% LFFC simulation (Test Case 4). For non-reacting cases, the heterogeneous reactions were not enabled in the numerical model by setting all stoichiometric coefficients to zero—see Figures 59 and 60. To change the LFFC%, the $\dot{m}$ of liquid kerosene injected into the fuel injection slot was altered—see Figure 88.

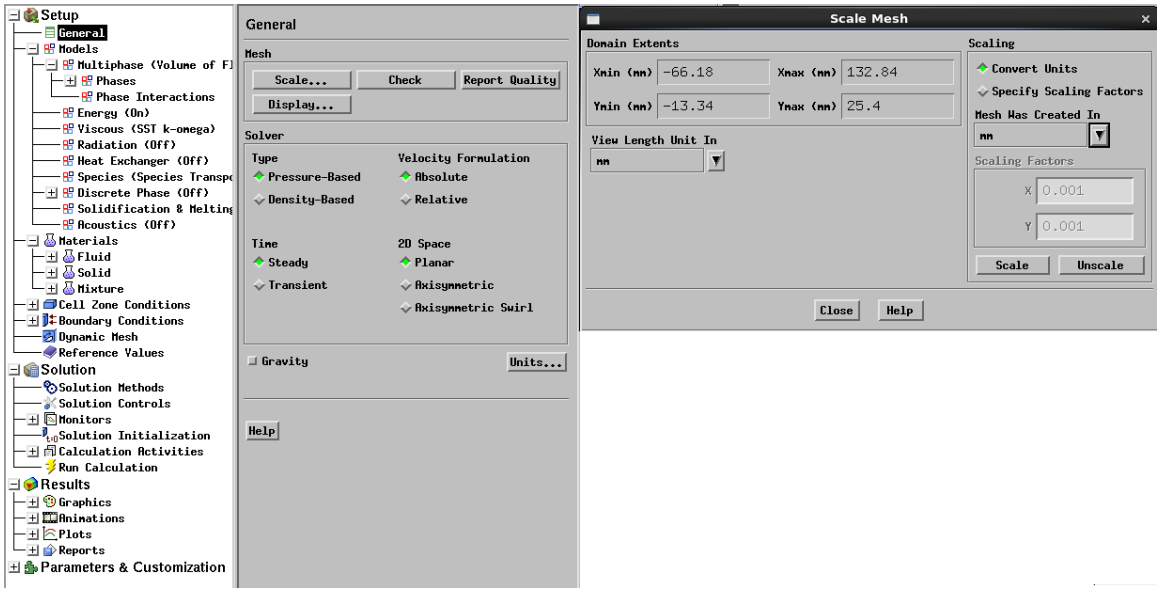

Figure 51. Fluent Settings—Scale Mesh Dialogue Box: Fluent reads a mesh in meters as the default setting. All meshes were created in mm and must be scaled as such upon first importing into Fluent.

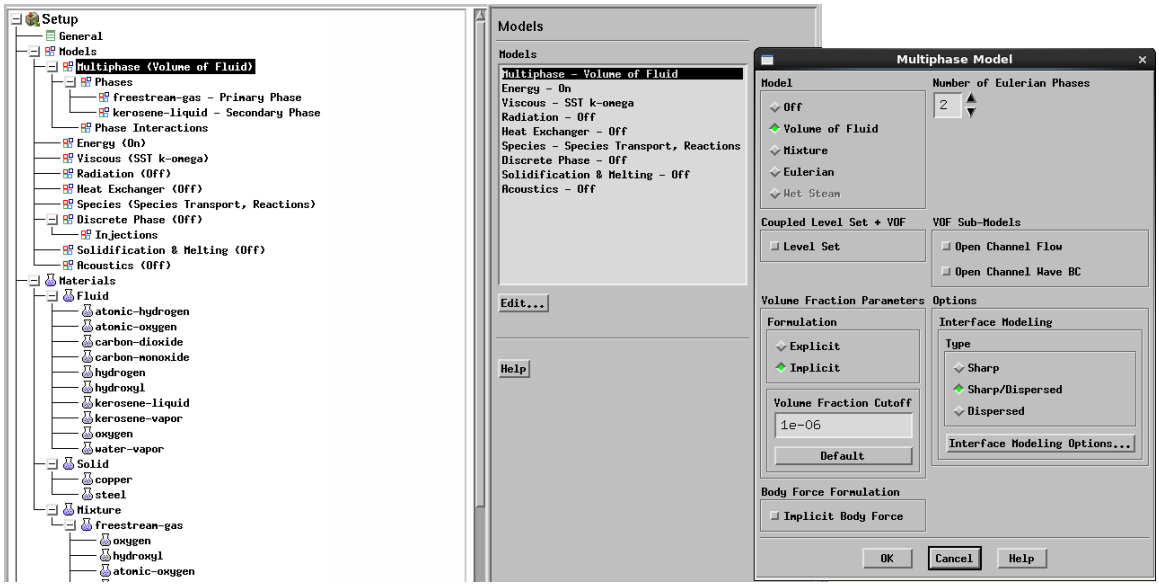

Figure 52. Fluent Settings: Multiphase Model Settings. Implicit formulation selected to allow for a steady simulation. Sharp/Dispersed type selected to allow for possible dispersion of fuel into the freestream.

```
/solve/set> vof-numerics
disable high order Rhie-Chow flux? [no]
Use hybrid treatment for high order Rhie-Chow flux? [no]
Use unstructured variant of PRESTO pressure scheme? [no]
Enable new framework for vof specific node based treatments? [no]
Enable enhanced compressible flow numerics? [yes] ]
```

Figure 53. Fluent Settings: VoF numeric option selections. All options left at default except enhanced compressible flow numerics for increased numerical stability. These options are accessed using the Text User Interface.

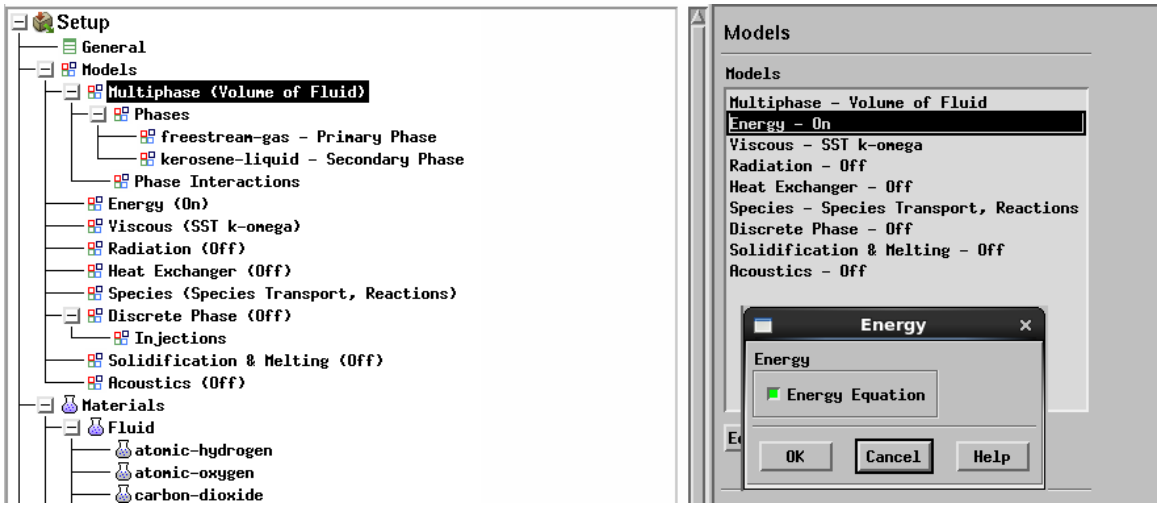

Figure 54. Fluent Settings: Energy Equation dialogue box

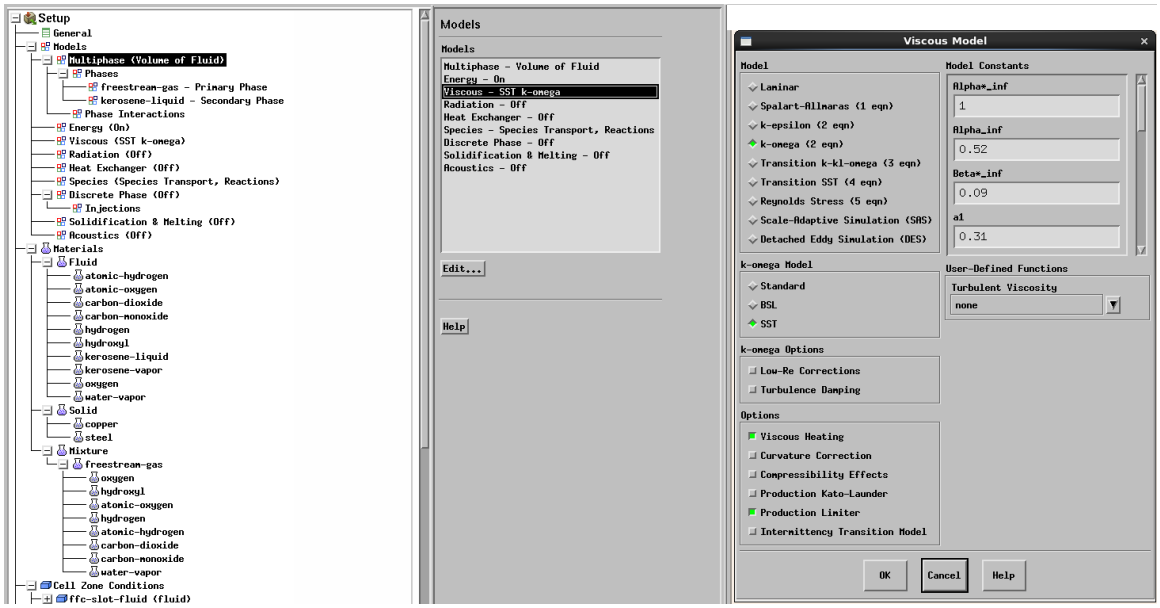

Figure 55. Fluent Settings: Viscous Model (Turbulence Model) Settings. Constants were left at default values.

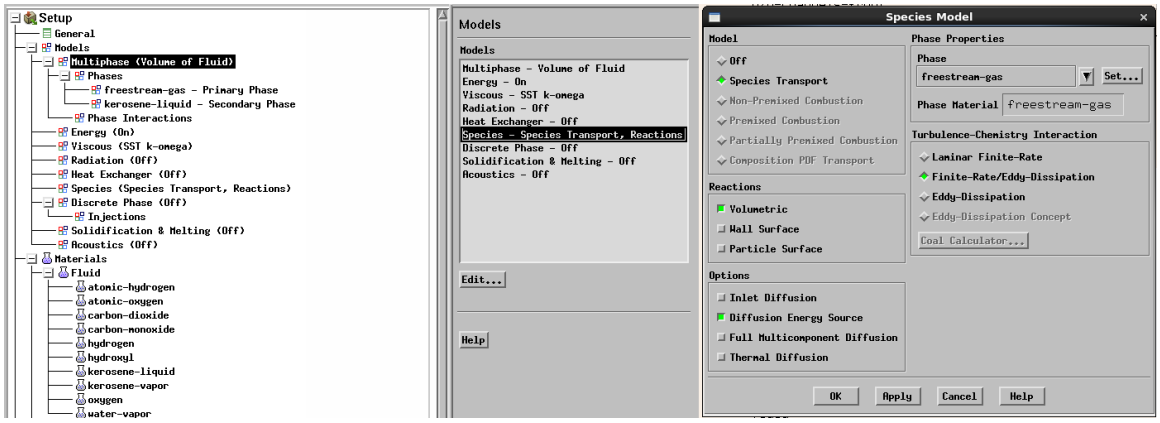

Figure 56. Fluent Settings: Species Transport Settings

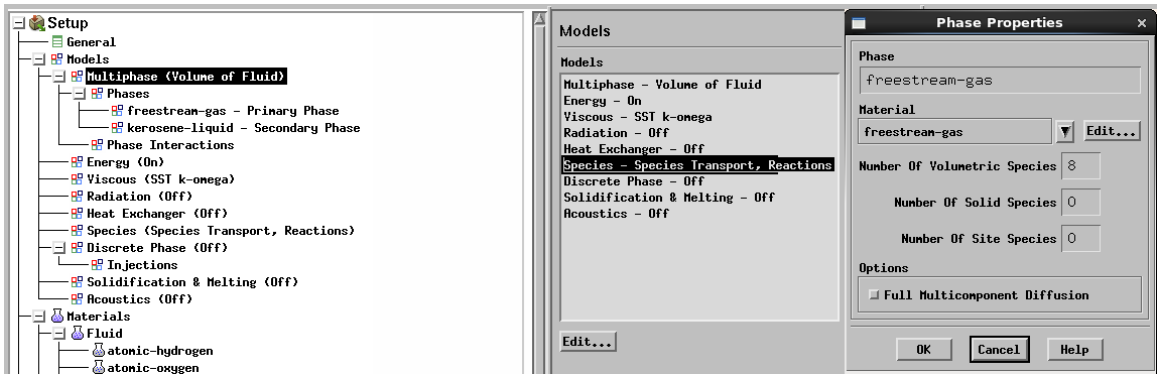

Figure 57. Fluent Settings: Phase Properties dialogue box

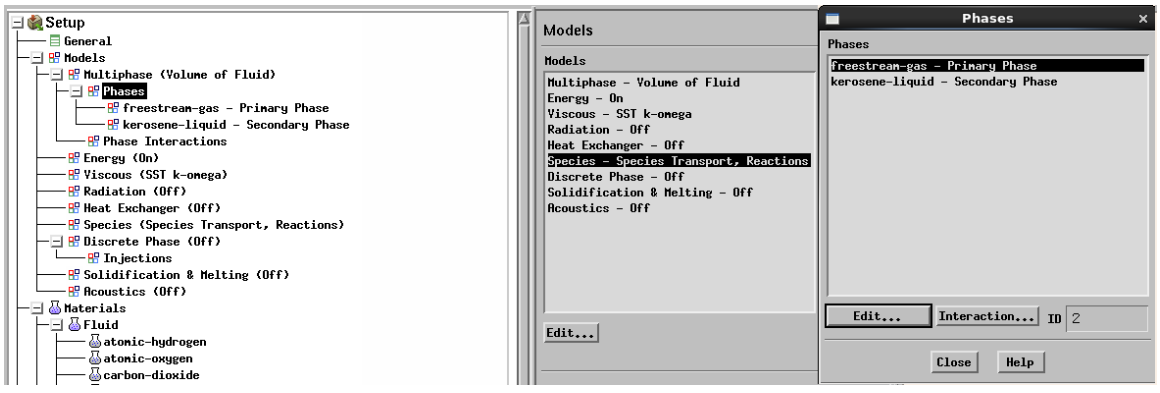

Figure 58. Fluent Settings: Primary and Secondary Phases

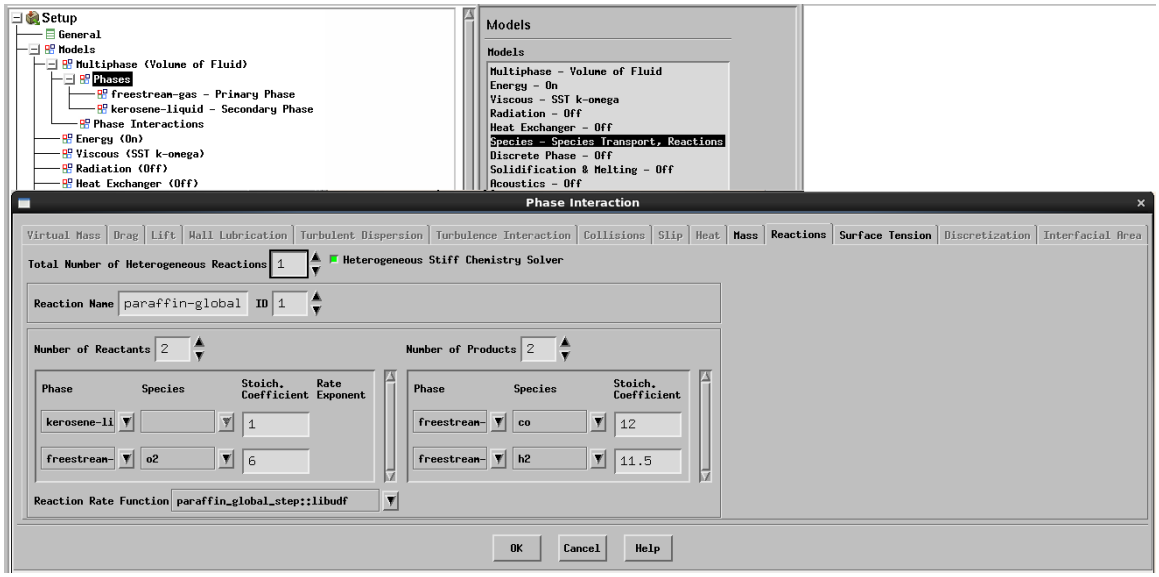

Figure 59. Fluent Settings: Phase Interaction dialogue box for paraffin global step. UDF must be loaded and compiled for selection of the Reaction Rate Function—see Figure 130.

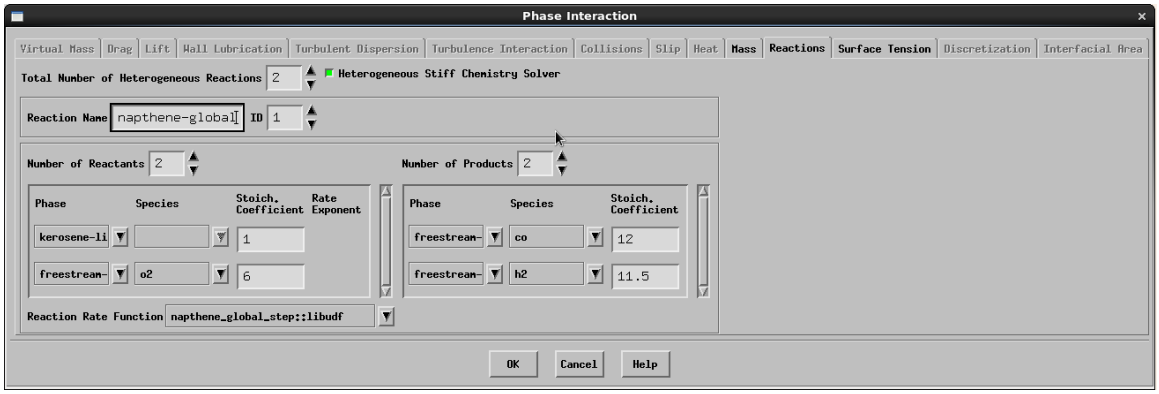

Figure 60. Fluent Settings: Phase Interaction dialogue box for napthene global step. UDF must be loaded and compiled for selection of the Reaction Rate Function—see Figure 130.

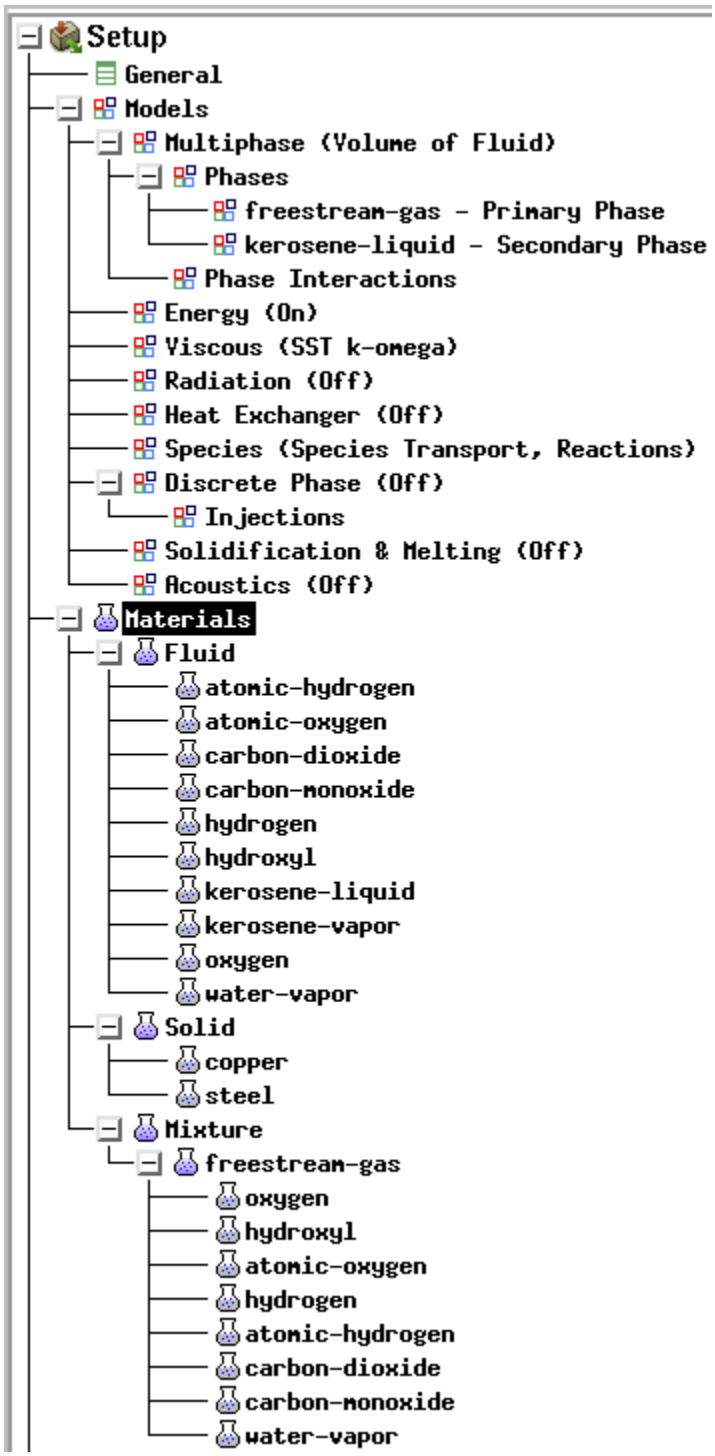

Figure 61. Fluent Settings: Materials list

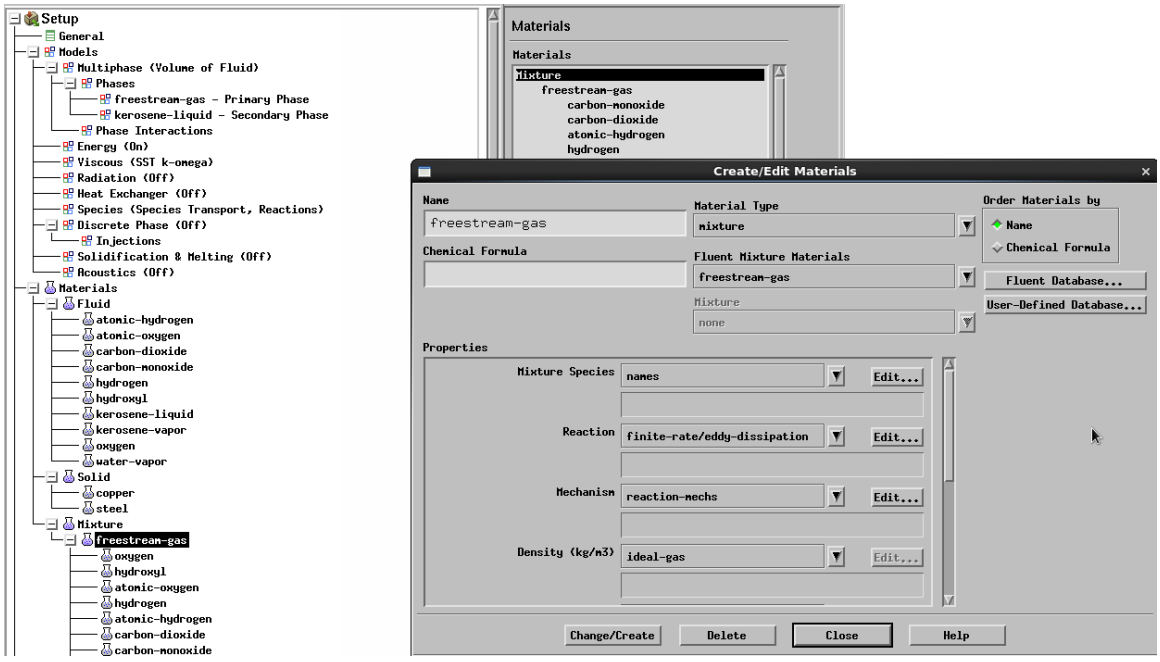

Figure 62. Fluent Settings: freestream-gas material dialogue box.

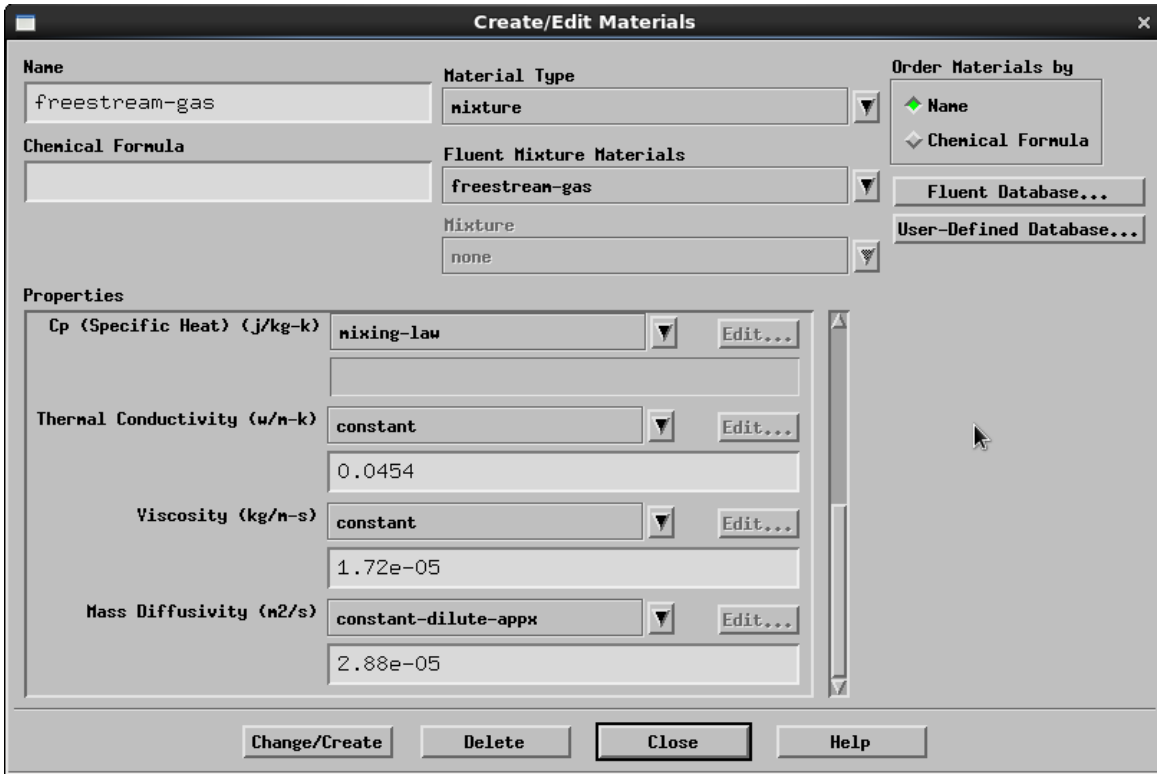

Figure 63. Fluent Settings: freestream-gas material dialogue box continued.

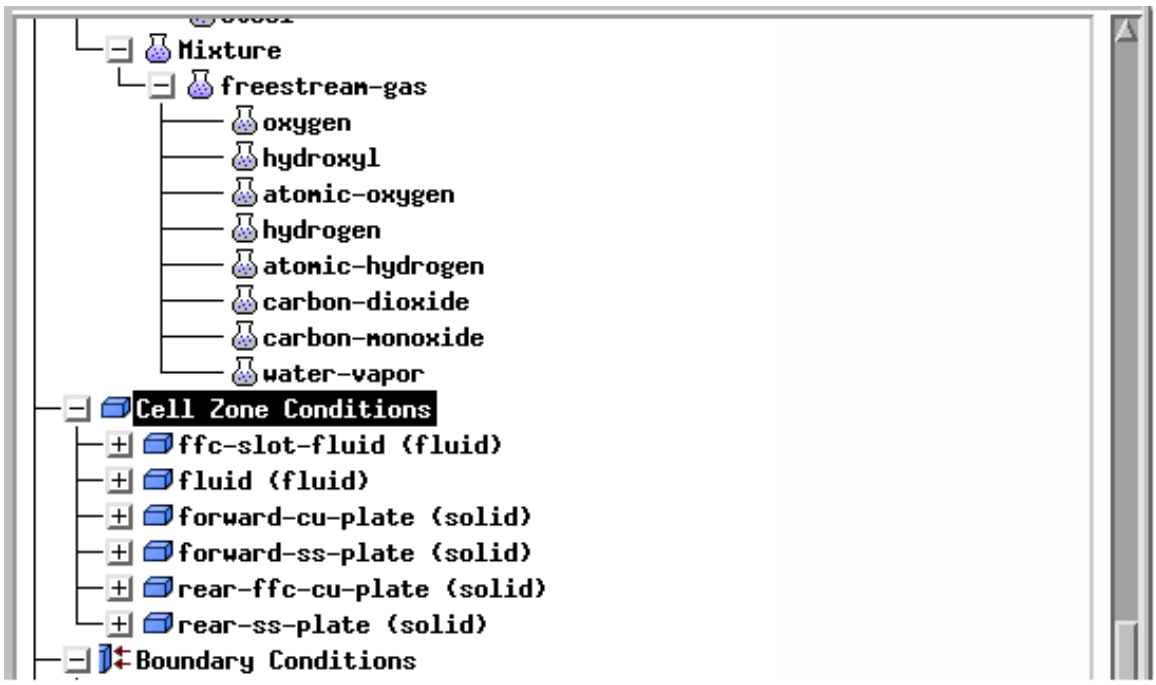

Figure 64. Fluent Settings: Cell Zone Conditions

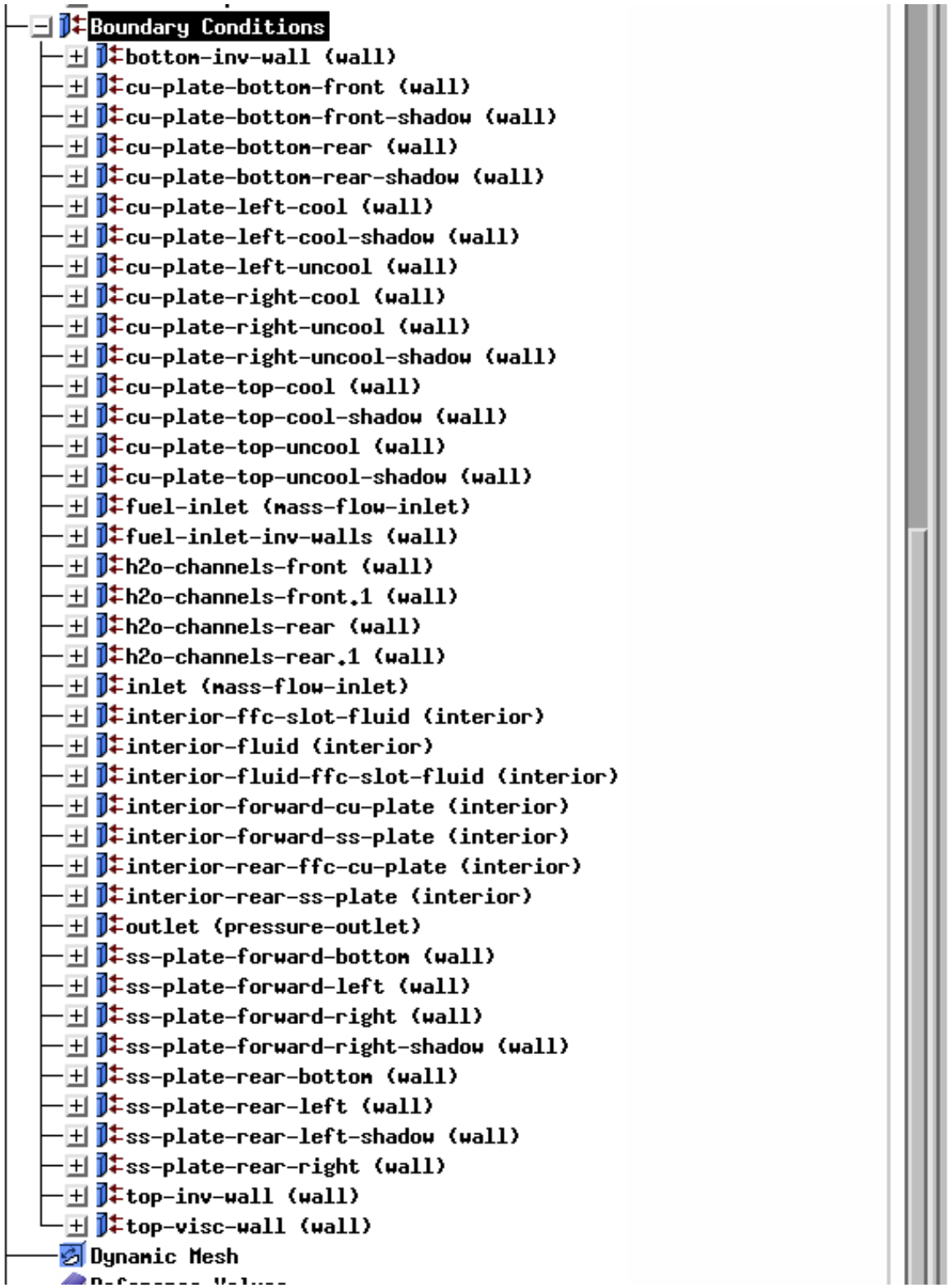

Figure 65. Fluent Settings: Boundary Conditions list. See Appendix C for locations of boundary conditions.

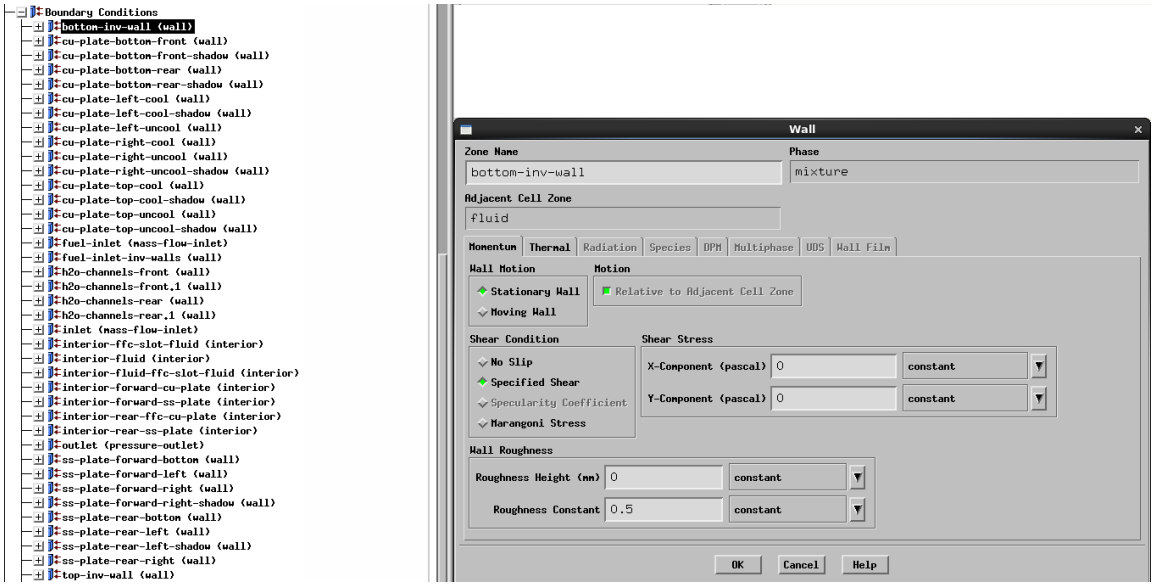

Figure 66. Fluent Settings: bottom-inv-wall momentum settings

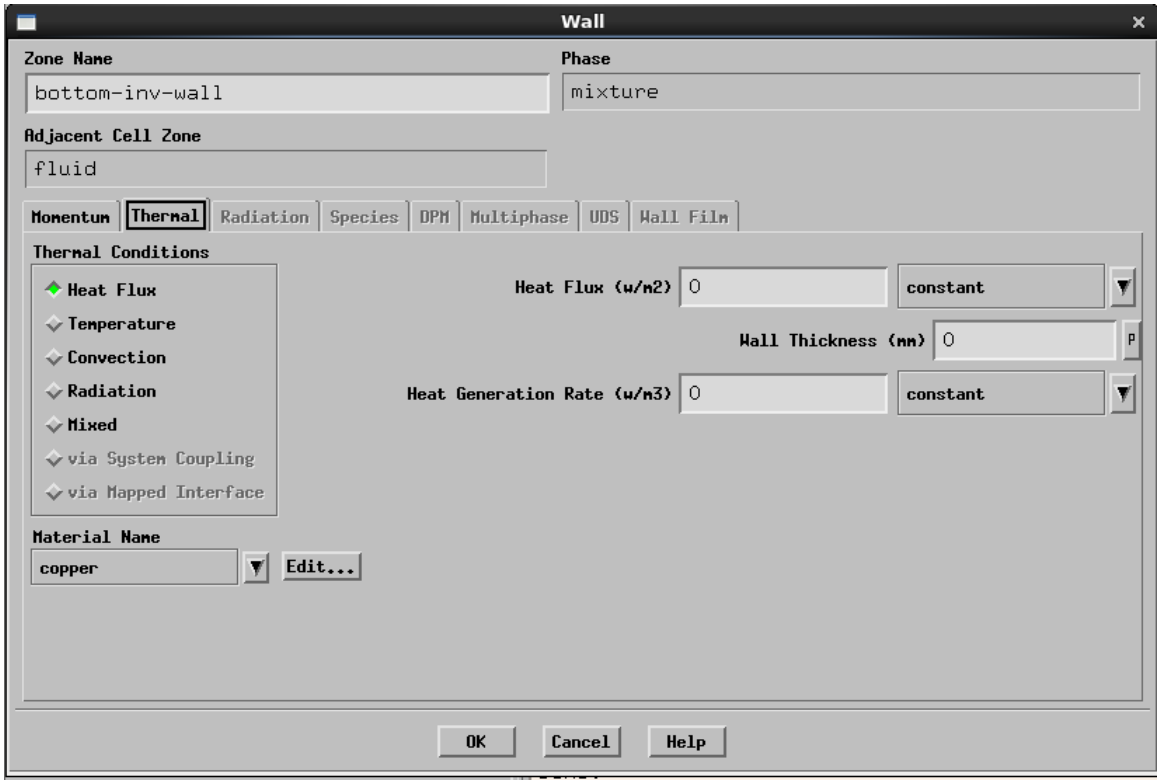

Figure 67. Fluent Settings: bottom-inv-wall thermal settings

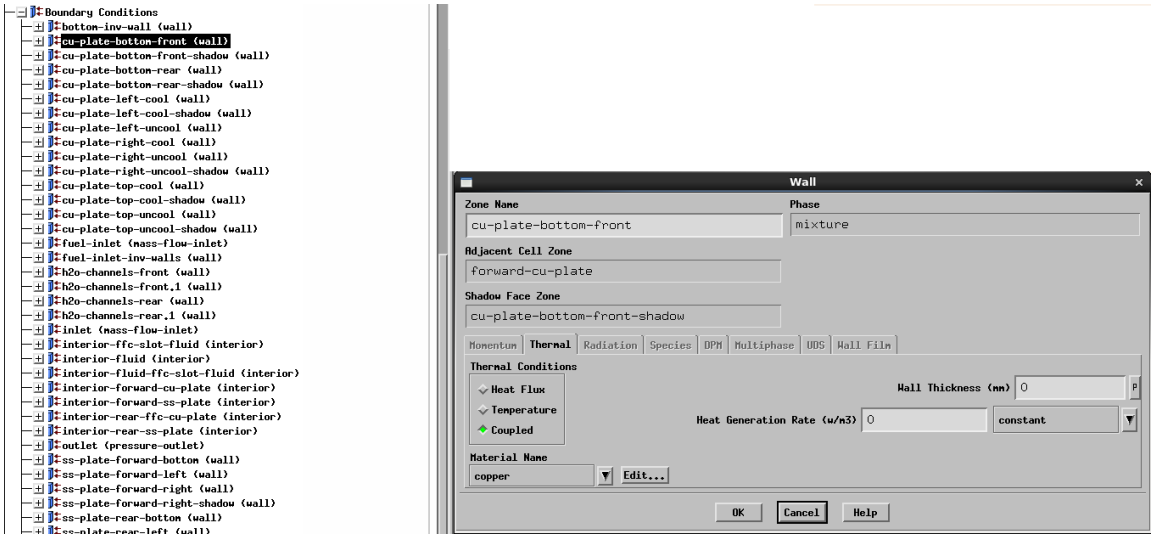

Figure 68. Fluent Settings: cu-plate-bottom-front thermal settings

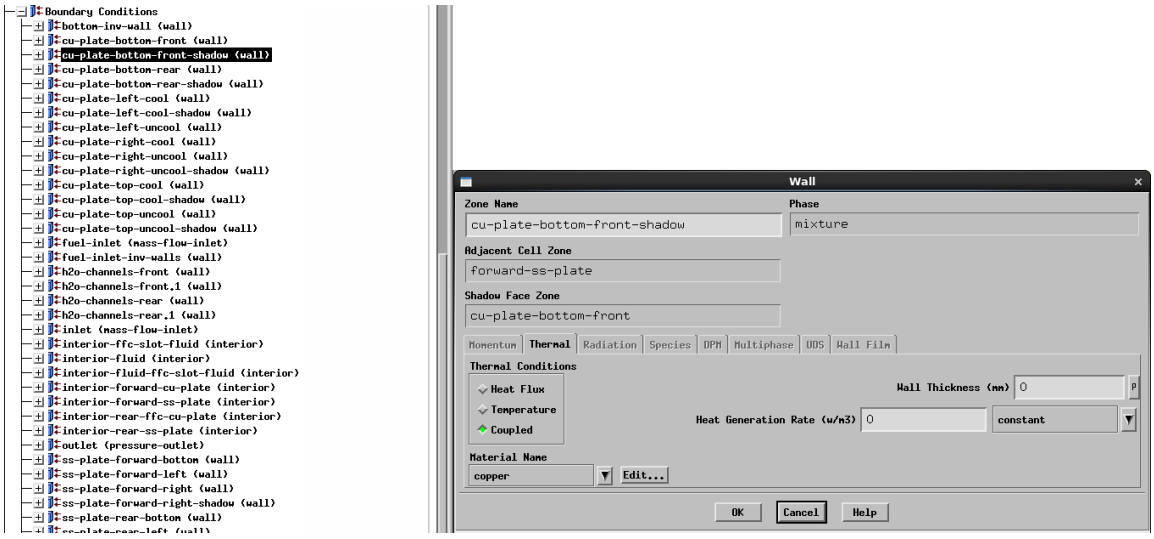

Figure 69. Fluent Settings: cu-plate-bottom-front-shadow settings

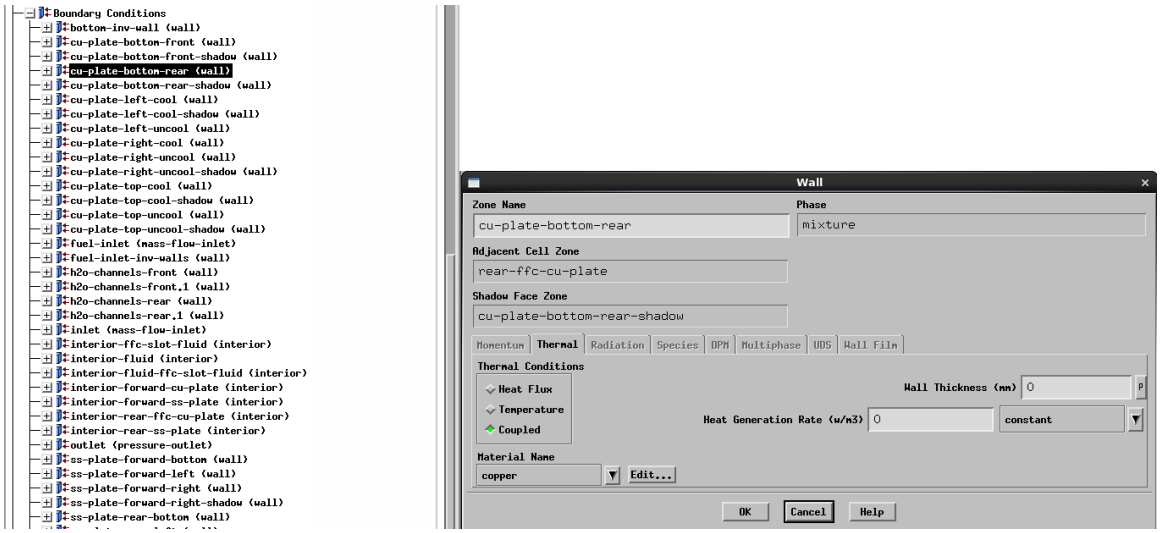

Figure 70. Fluent Settings: cu-plate-bottom-rear thermal settings

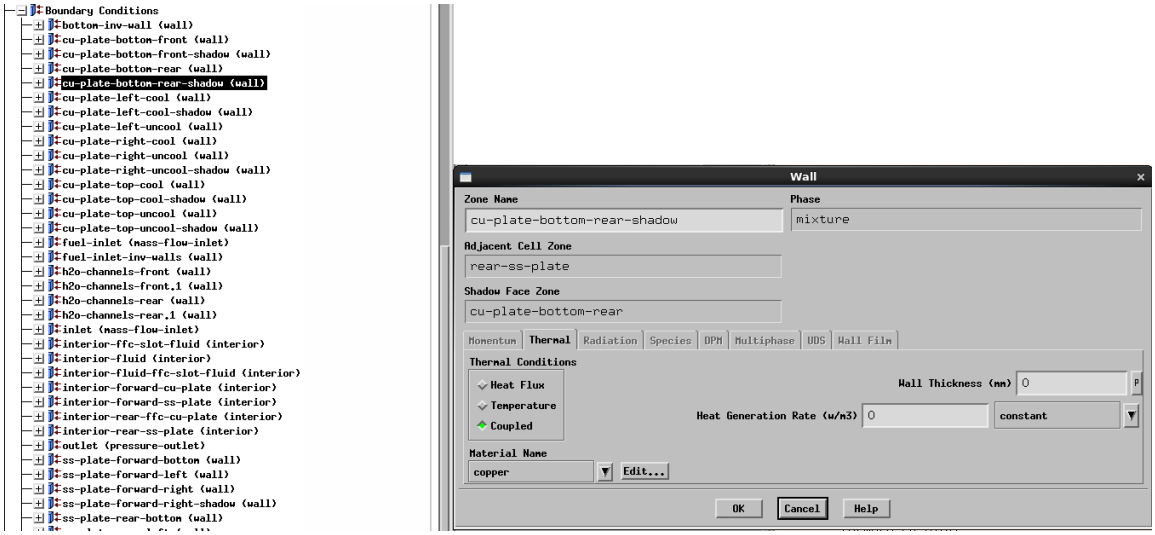

Figure 71. Fluent Settings: cu-plate-bottom-rear-shadow thermal settings

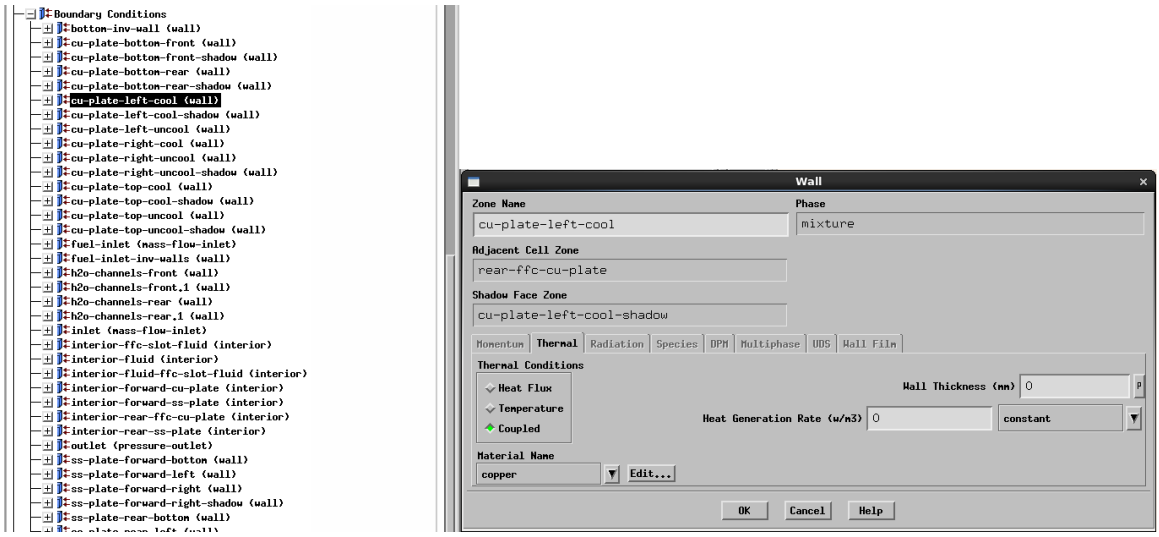

Figure 72. Fluent Settings: cu-plate-left-cool thermal settings

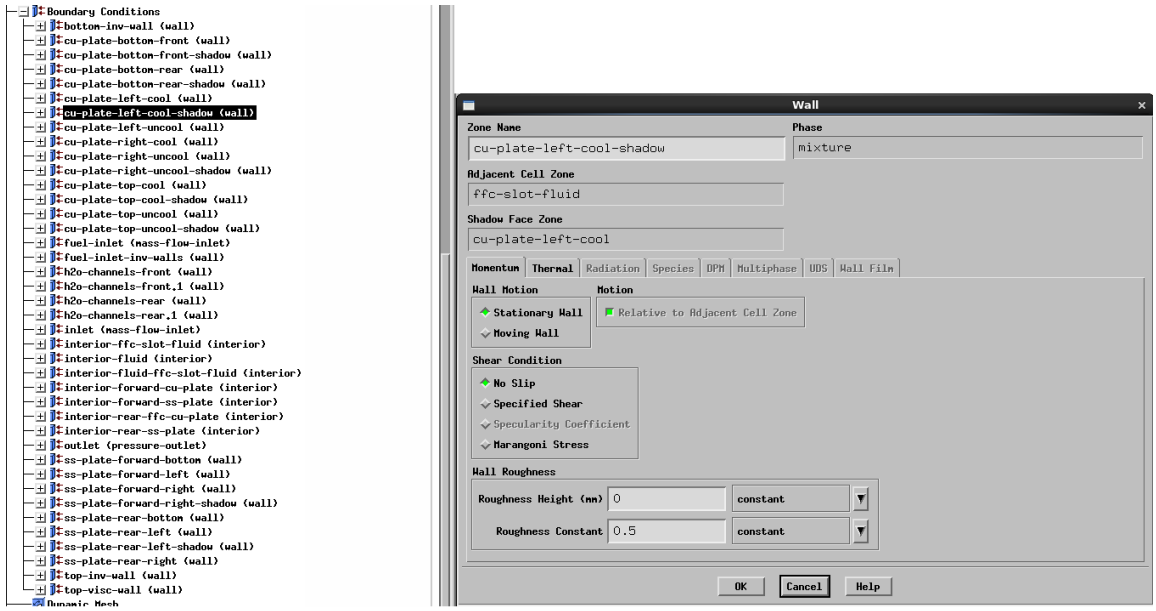

Figure 73. Fluent Settings: cu-plate-left-cool-shadow momentum settings

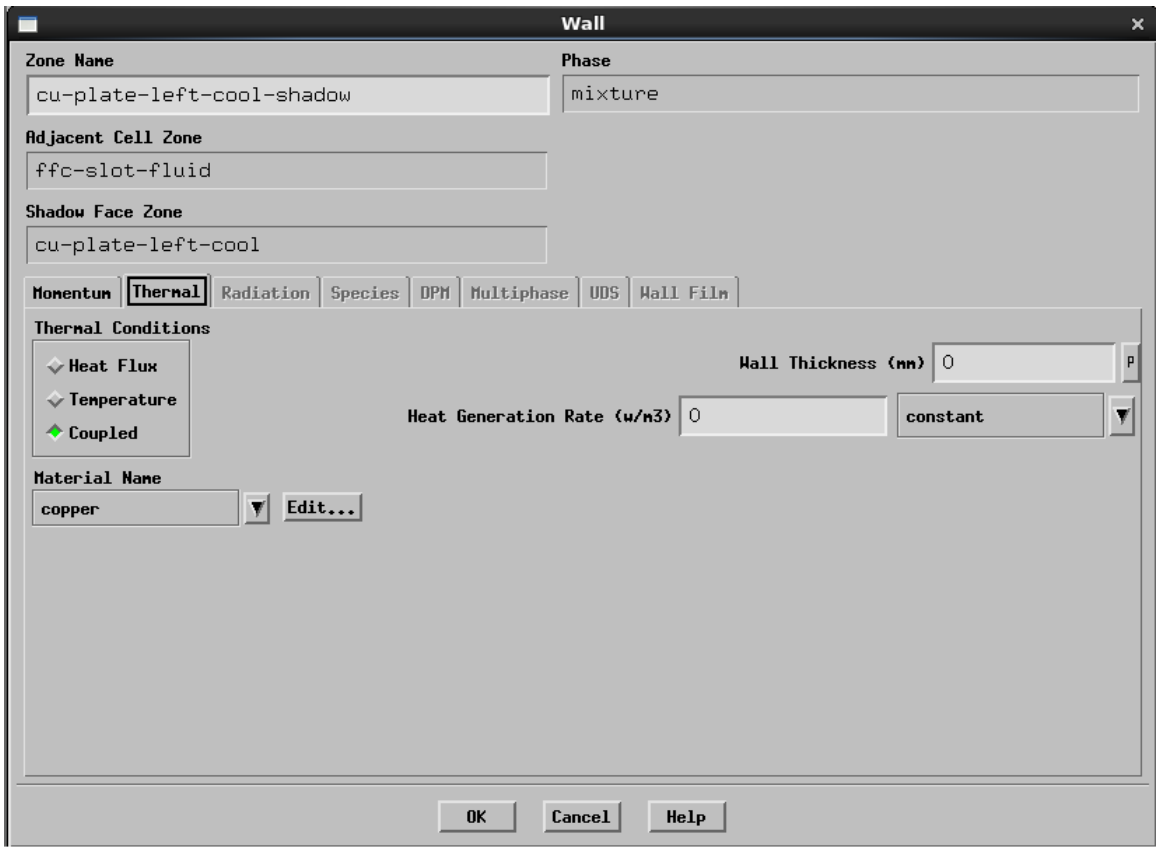

Figure 74. Fluent Settings: cu-plate-left-cool-shadow thermal settings

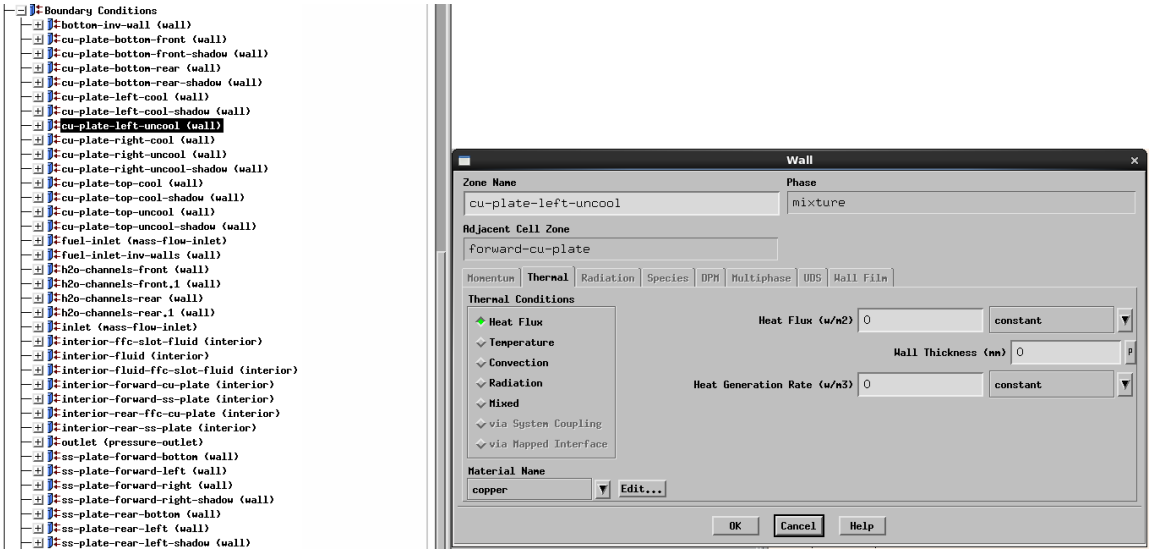

Figure 75. Fluent Settings: cu-plate-left-uncool thermal settings

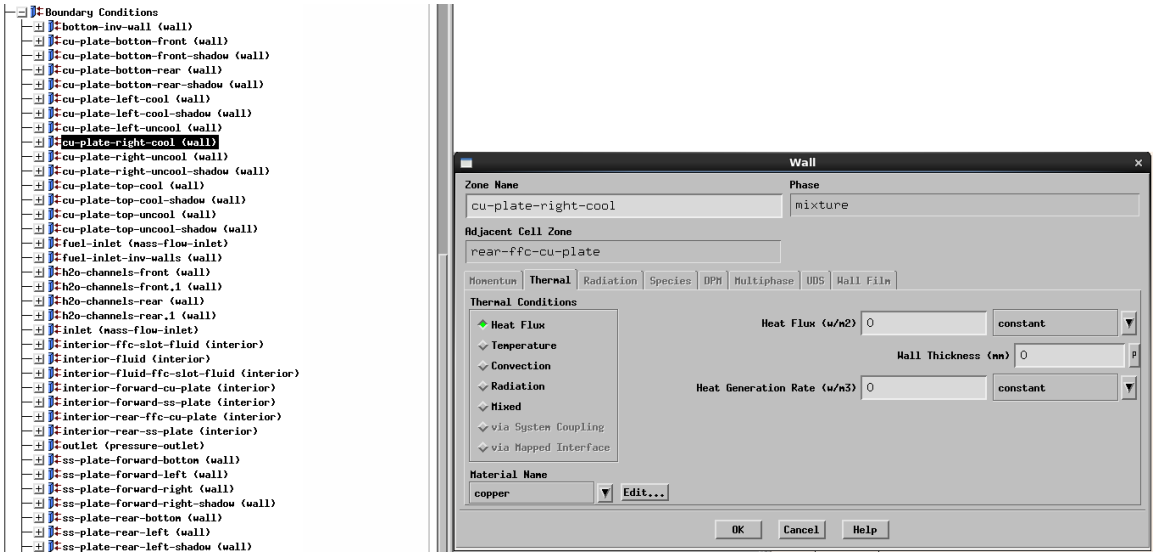

Figure 76. Fluent Settings: cu-plate-right-cool thermal settings

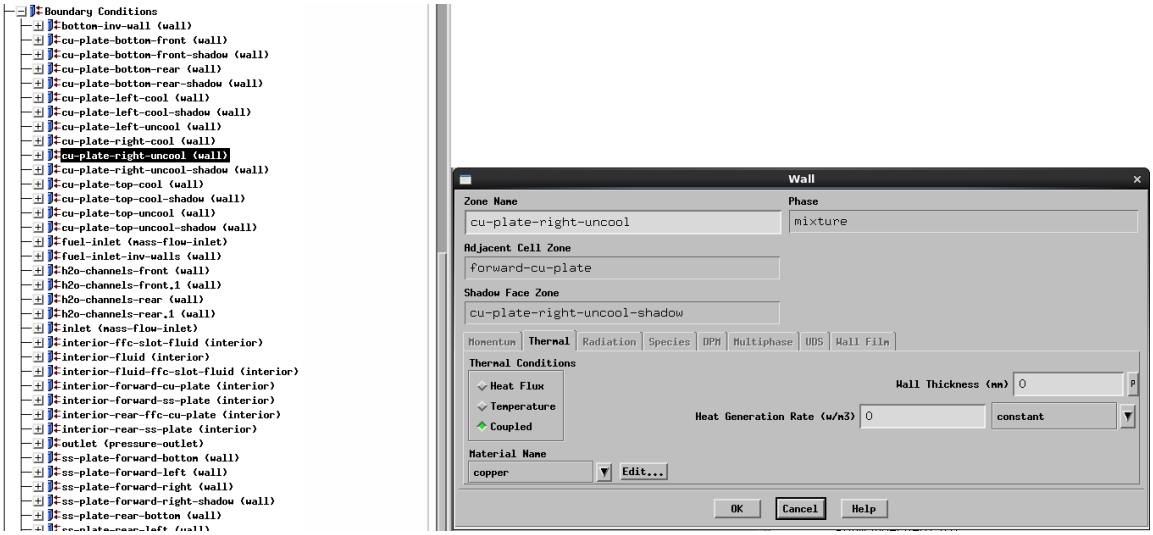

Figure 77. Fluent Settings: cu-plate-right-uncool thermal settings

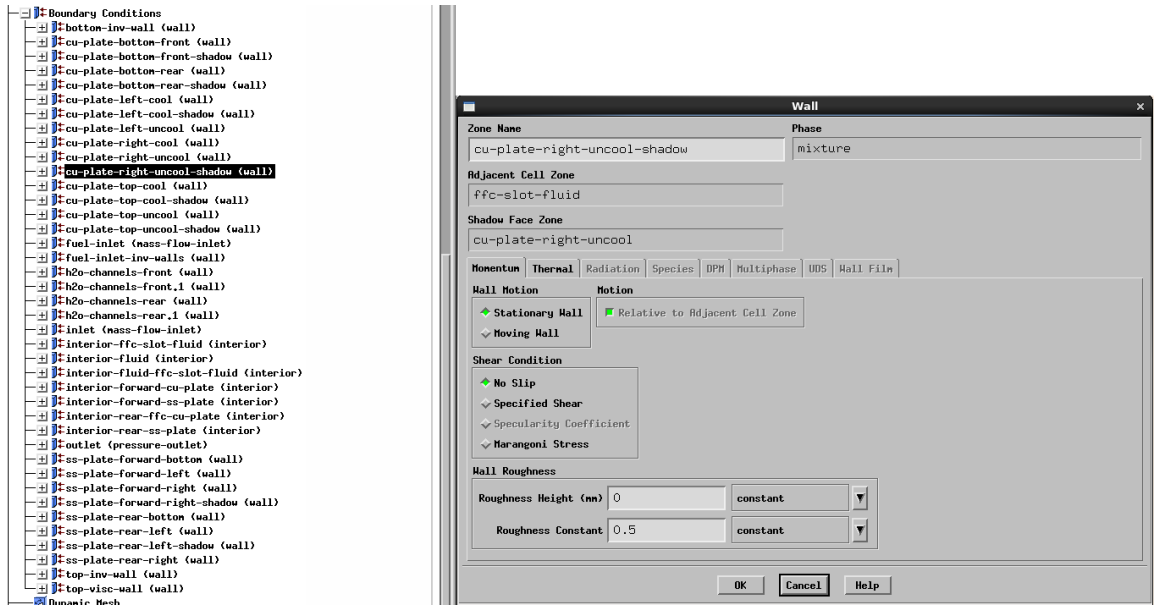

Figure 78. Fluent Settings: cu-plate-right-uncool-shadow momentum settings

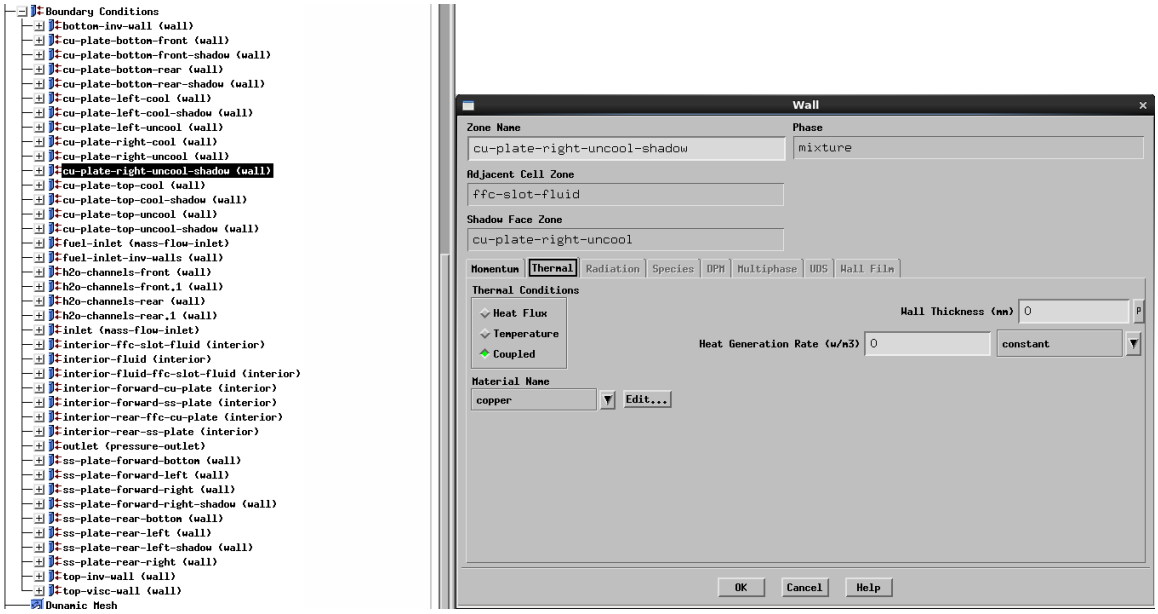

Figure 79. Fluent Settings: cu-plate-right-uncool-shadow thermal settings

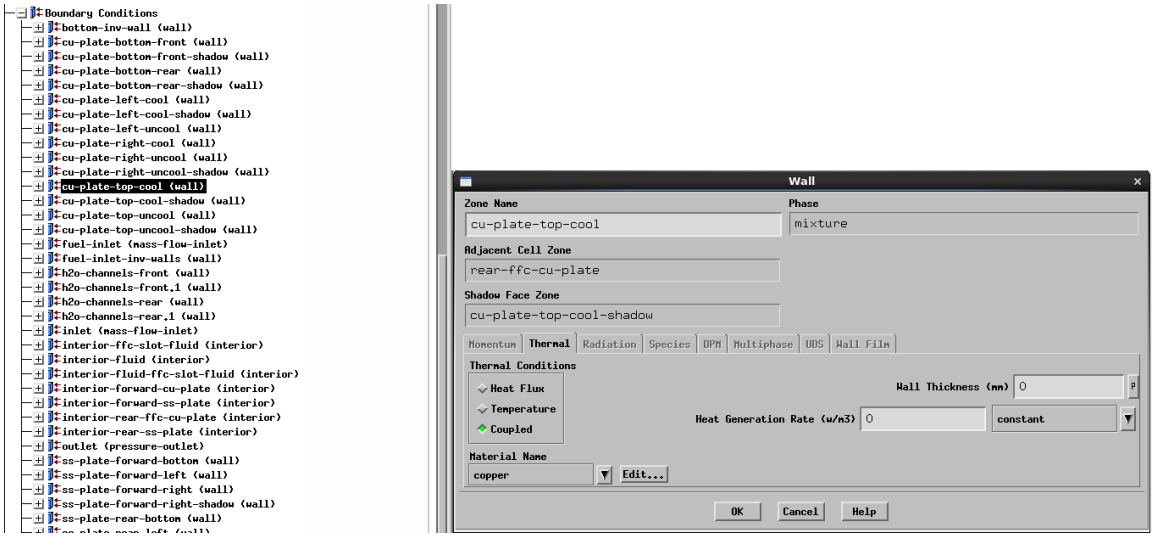

Figure 80. Fluent Settings: cu-plate-top-cool- thermal settings

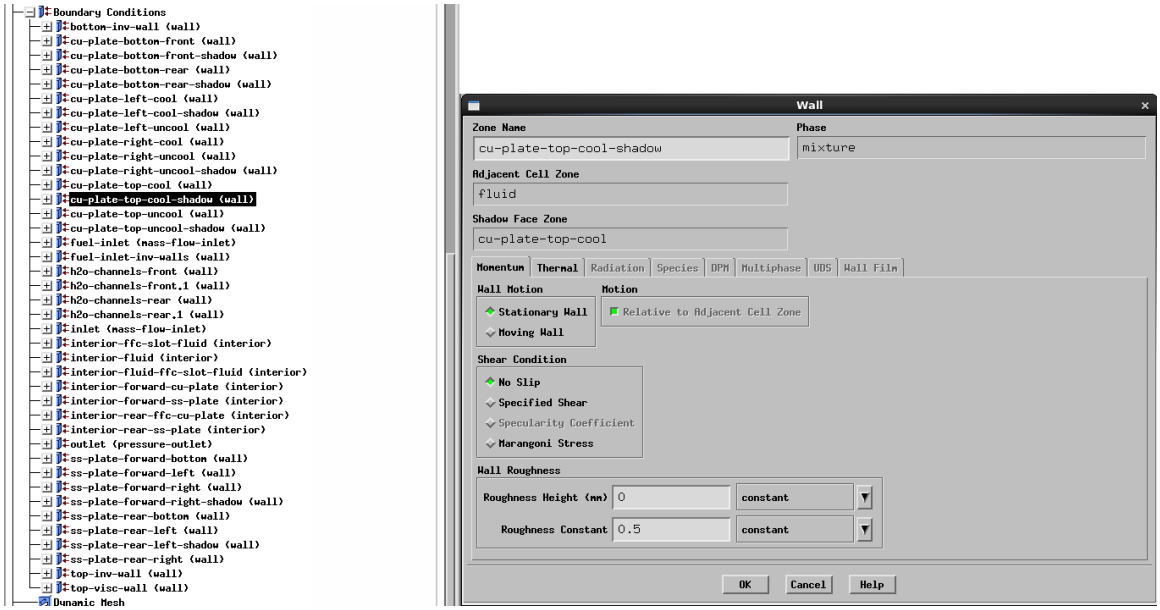

Figure 81. Fluent Settings: cu-plate-top-cool-shadow momentum settings

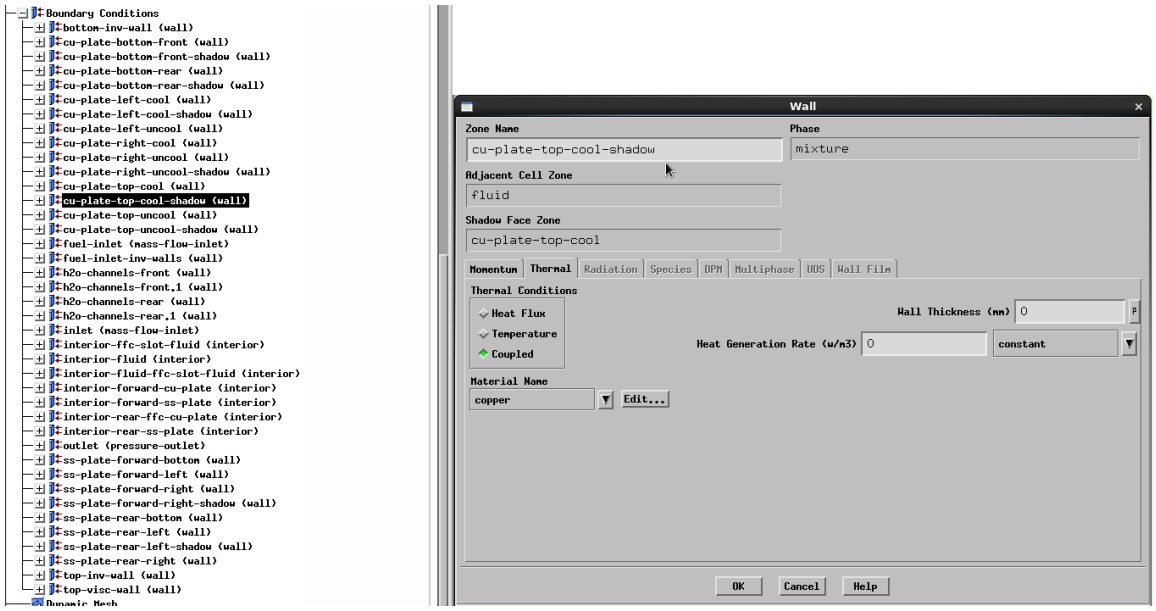

Figure 82. Fluent Settings: cu-plate-top-cool-shadow thermal settings

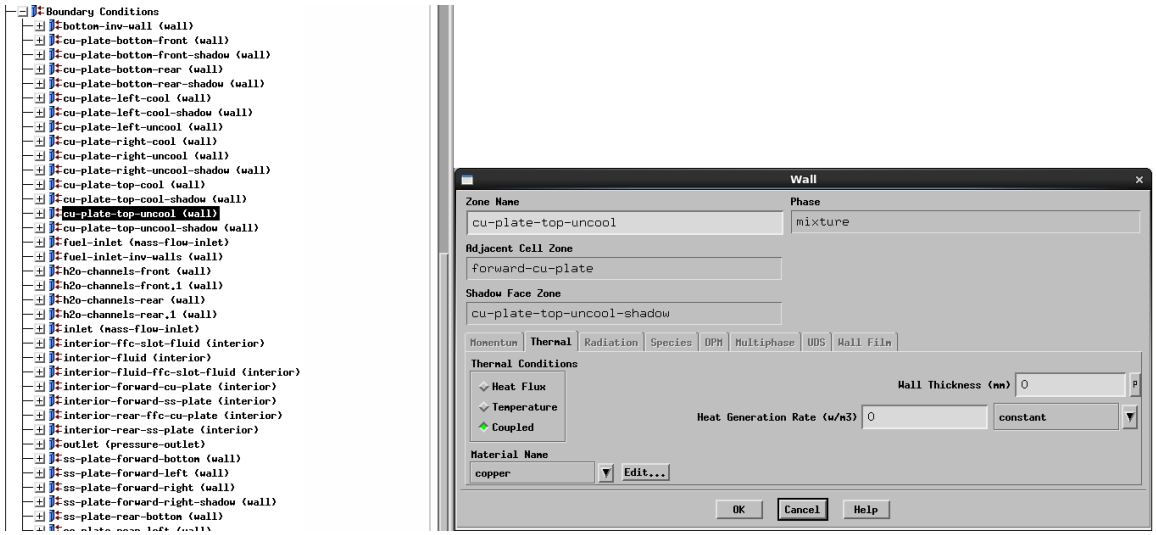

Figure 83. Fluent Settings: cu-plate-top-uncool thermal settings

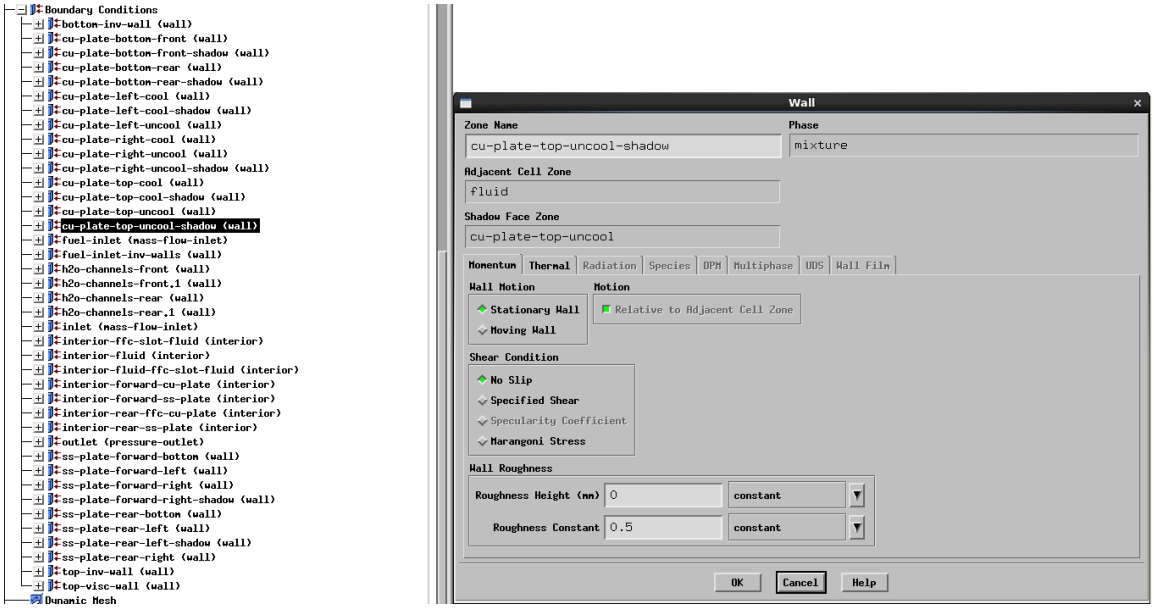

Figure 84. Fluent Settings: cu-plate-top-uncool-shadow momentum settings

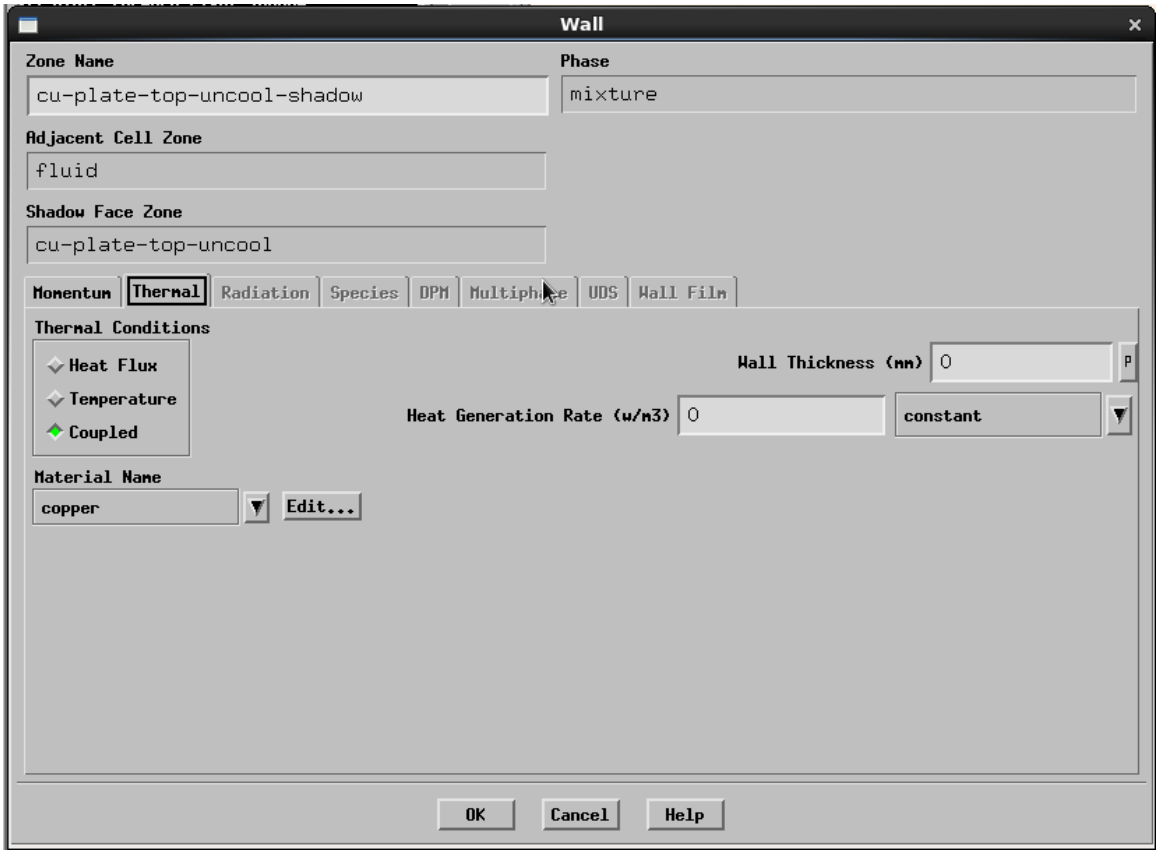

Figure 85. Fluent Settings: cu-plate-top-uncool-shadow thermal settings

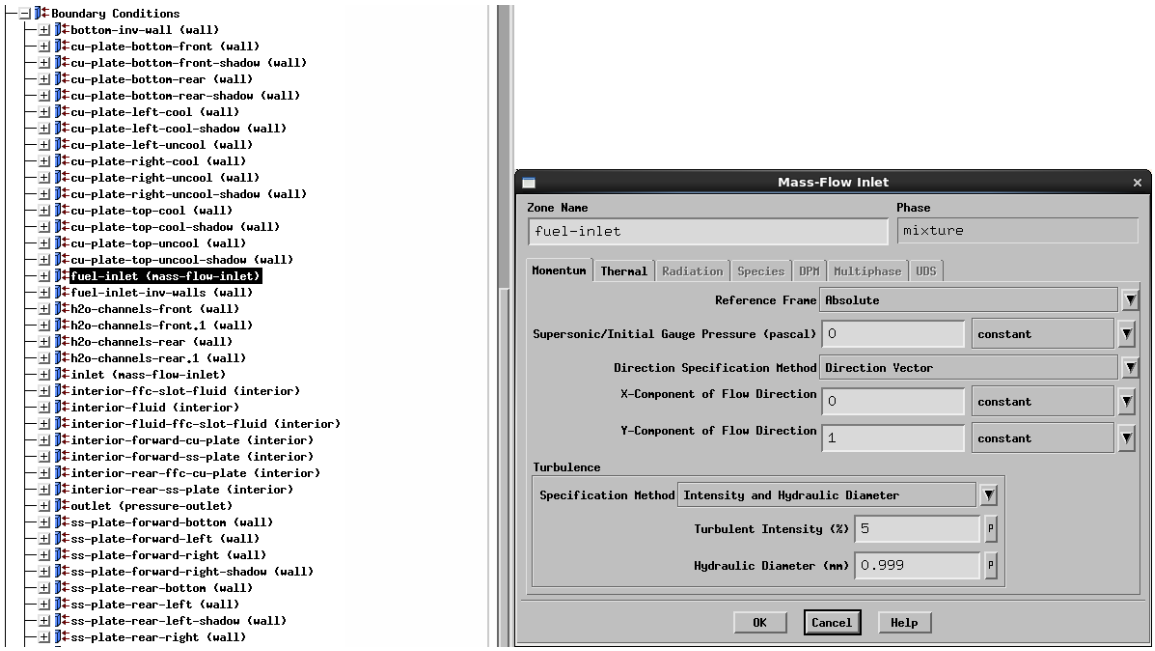

Figure 86. Fluent Settings: fuel-inlet momentum settings (mixture phase)

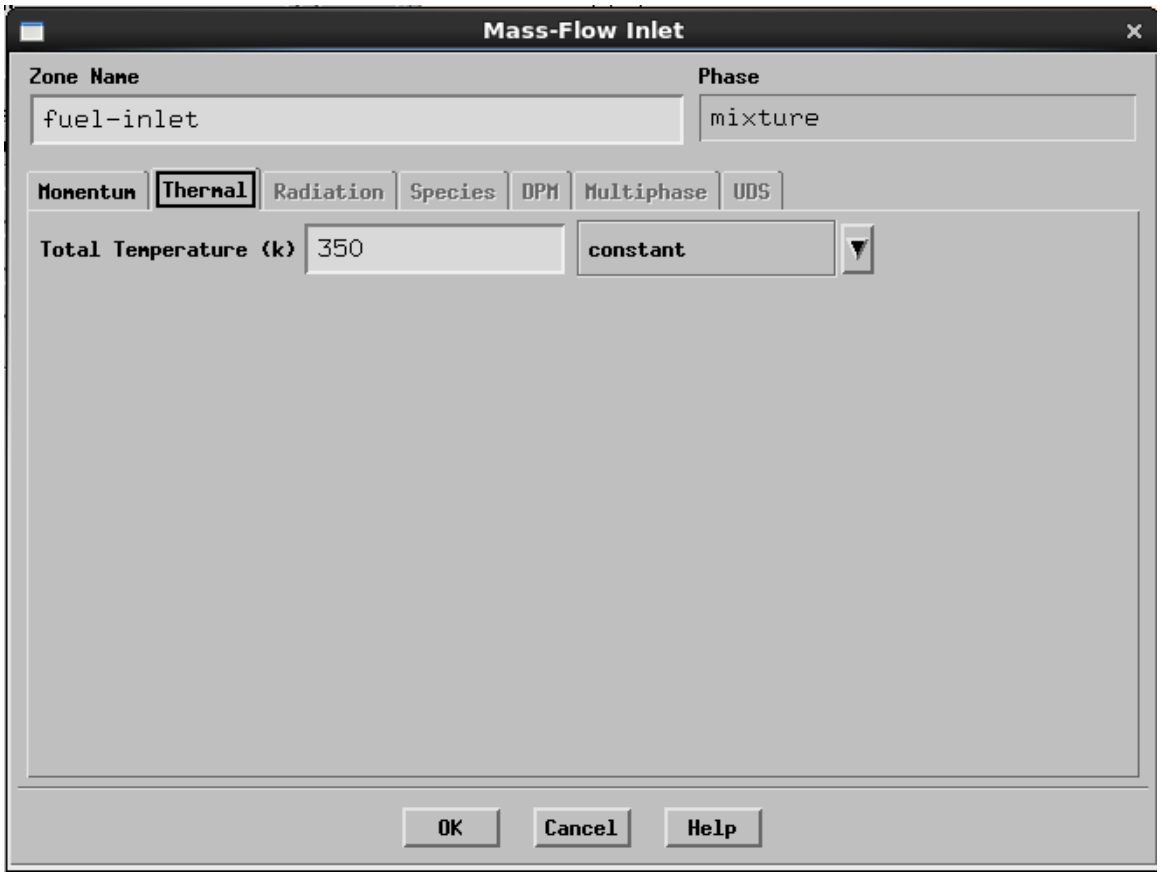

Figure 87. Fluent Settings: fuel-inlet thermal settings (mixture phase)

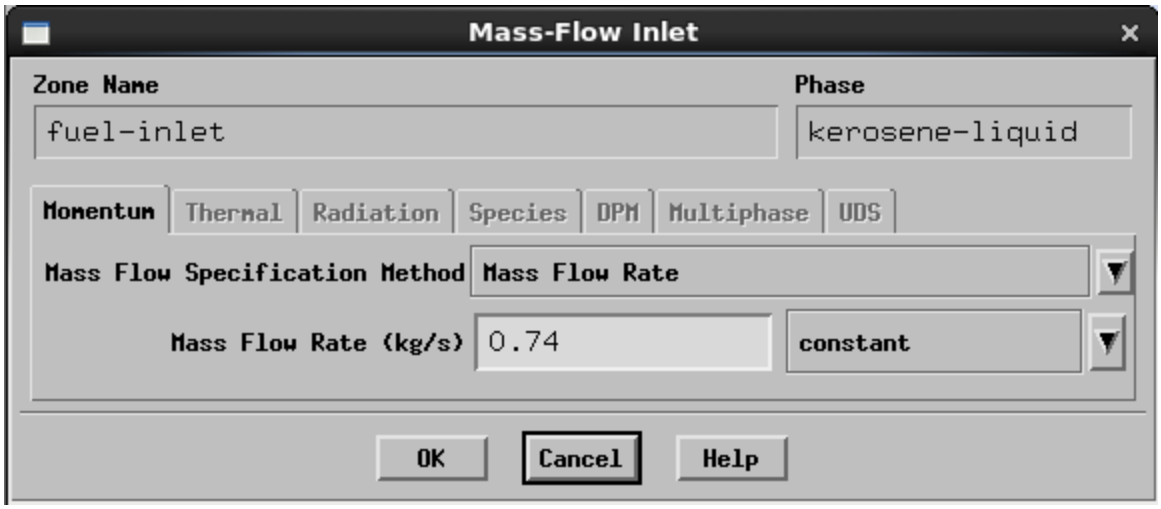

Figure 88. Fluent Settings: fuel-inlet momentum settings (kerosene-liquid phase)

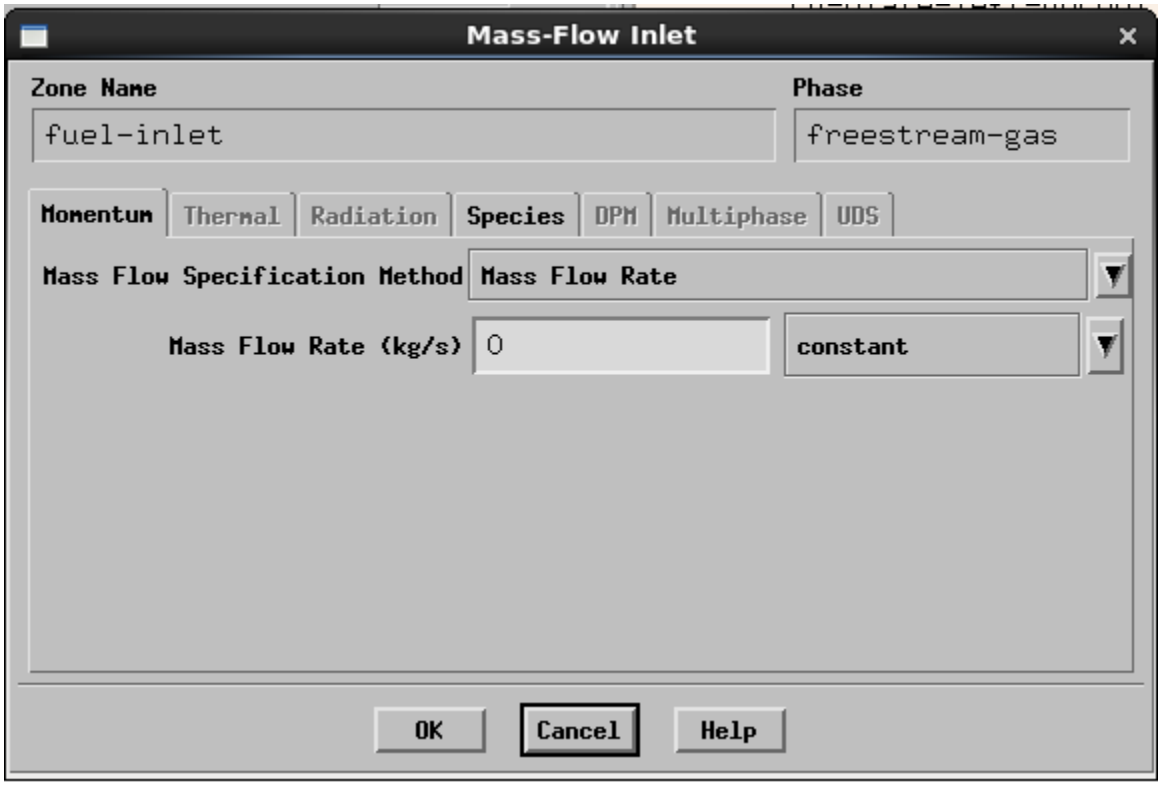

Figure 89. Fluent Settings: fuel-inlet momentum settings (freestream-gas phase)

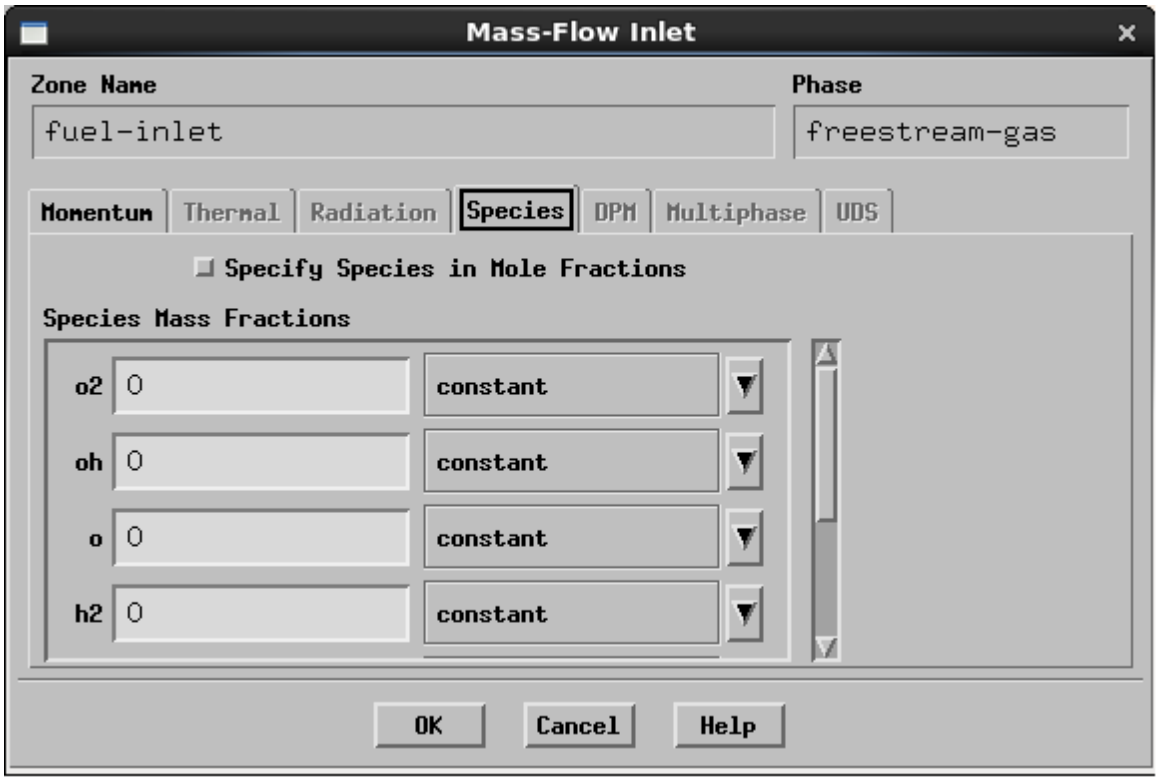

Figure 90. Fluent Settings: fuel-inlet species settings (freestream-gas phase). All species set to 0.

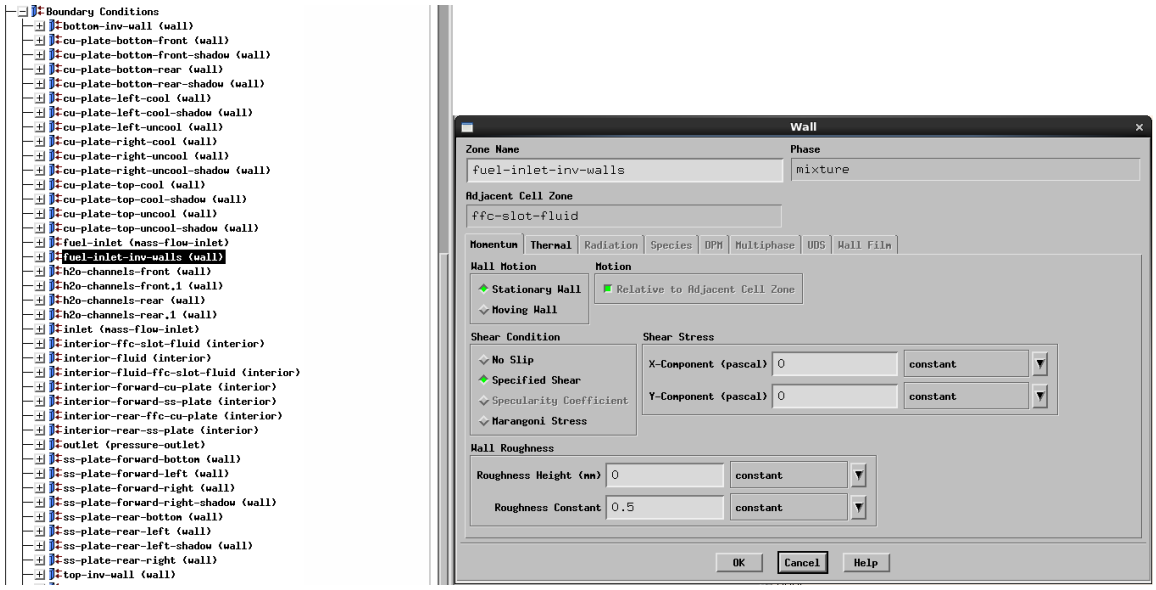

Figure 91. Fluent Settings: fuel-inlet-inv-walls momentum settings

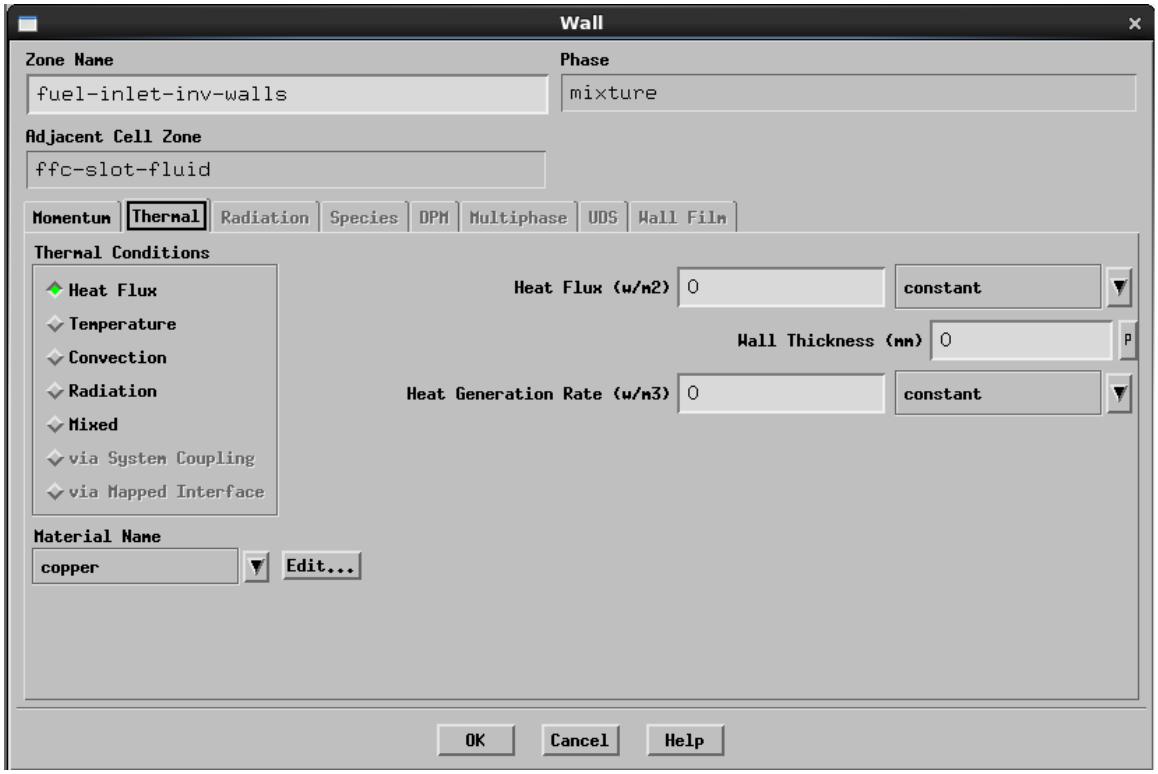

Figure 92. Fluent Settings: fuel-inlet-inv-walls thermal settings

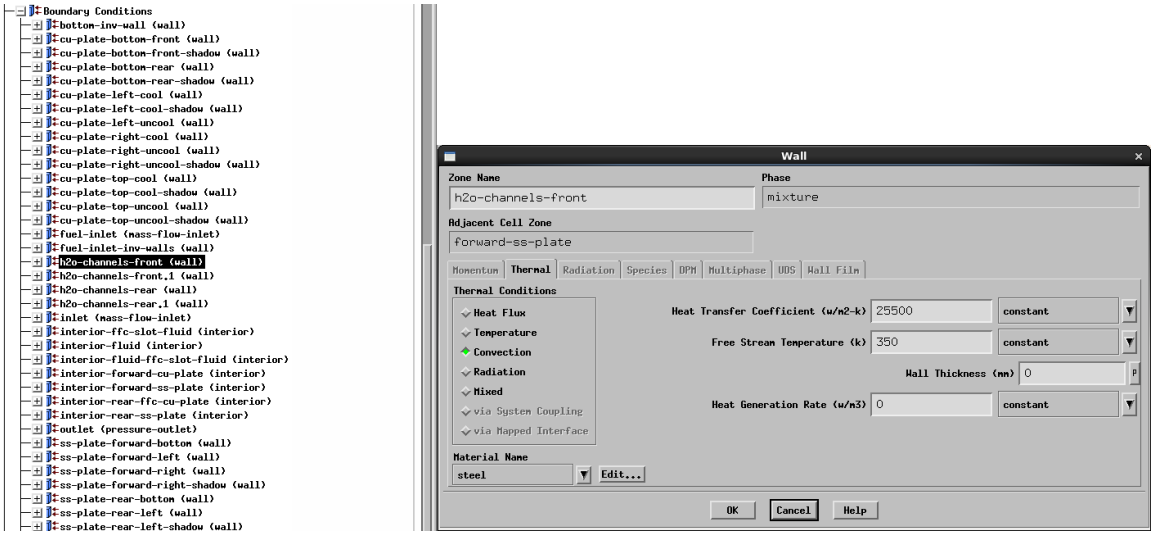

Figure 93. Fluent Settings: h2o-channels-front thermal settings

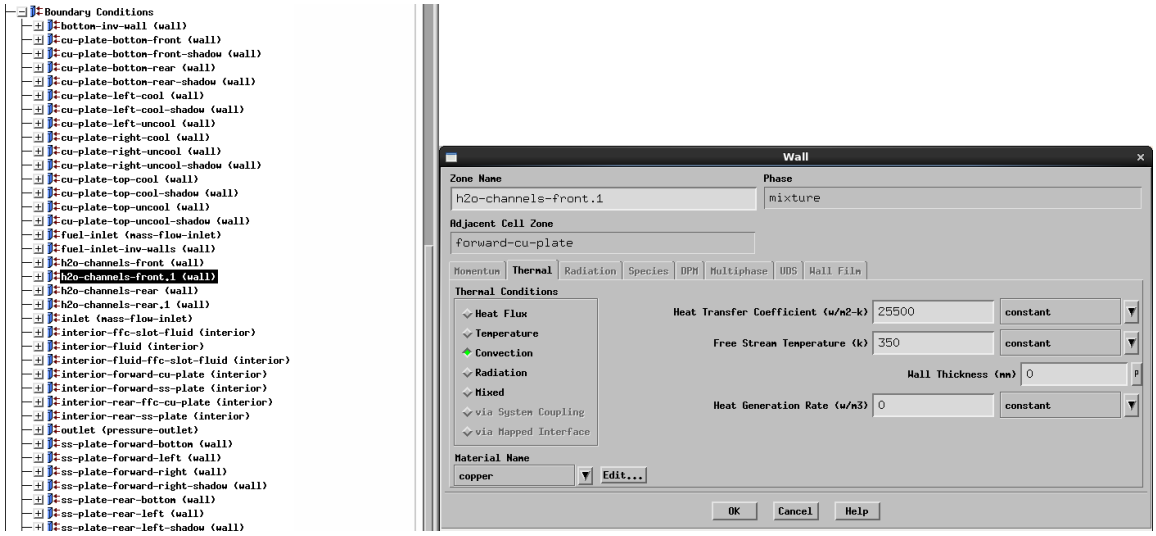

Figure 94. Fluent Settings: h2o-channels-front.1 thermal settings

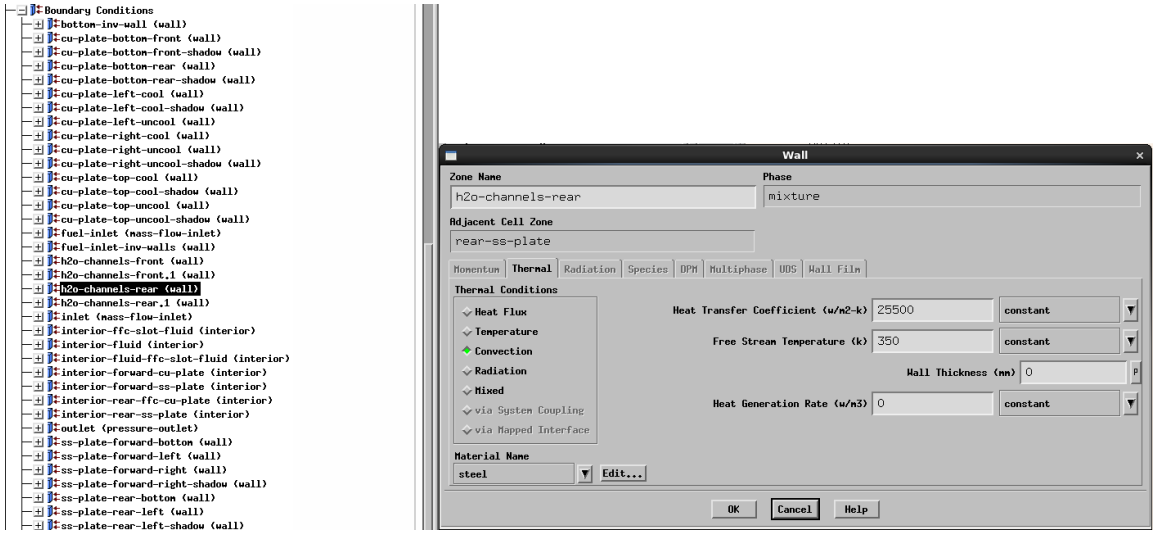

Figure 95. Fluent Settings: h2o-channels-rear thermal settings

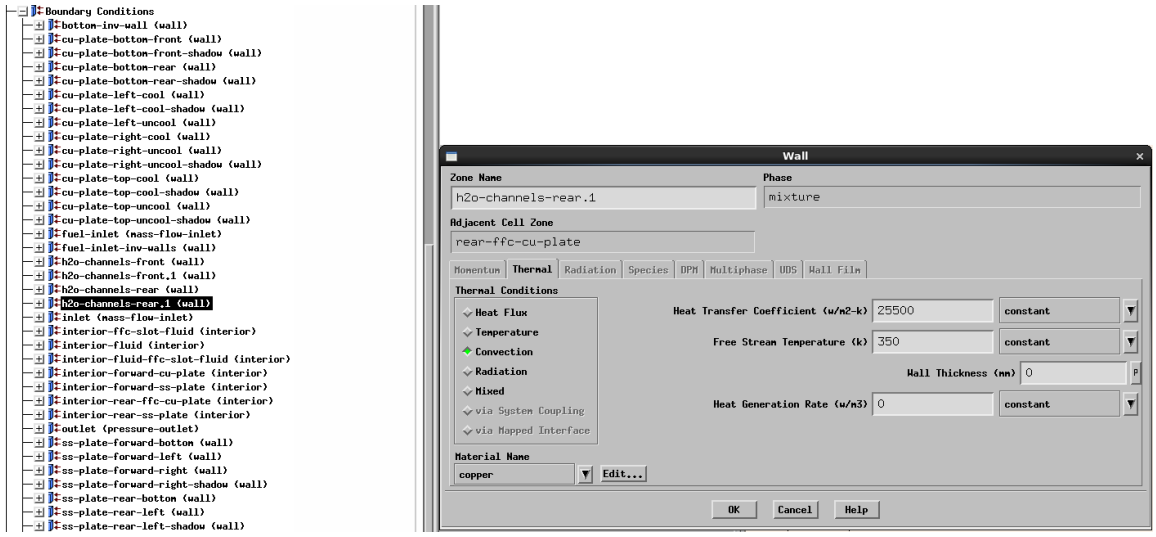

Figure 96. Fluent Settings: h2o-channels-rear.1 thermal settings

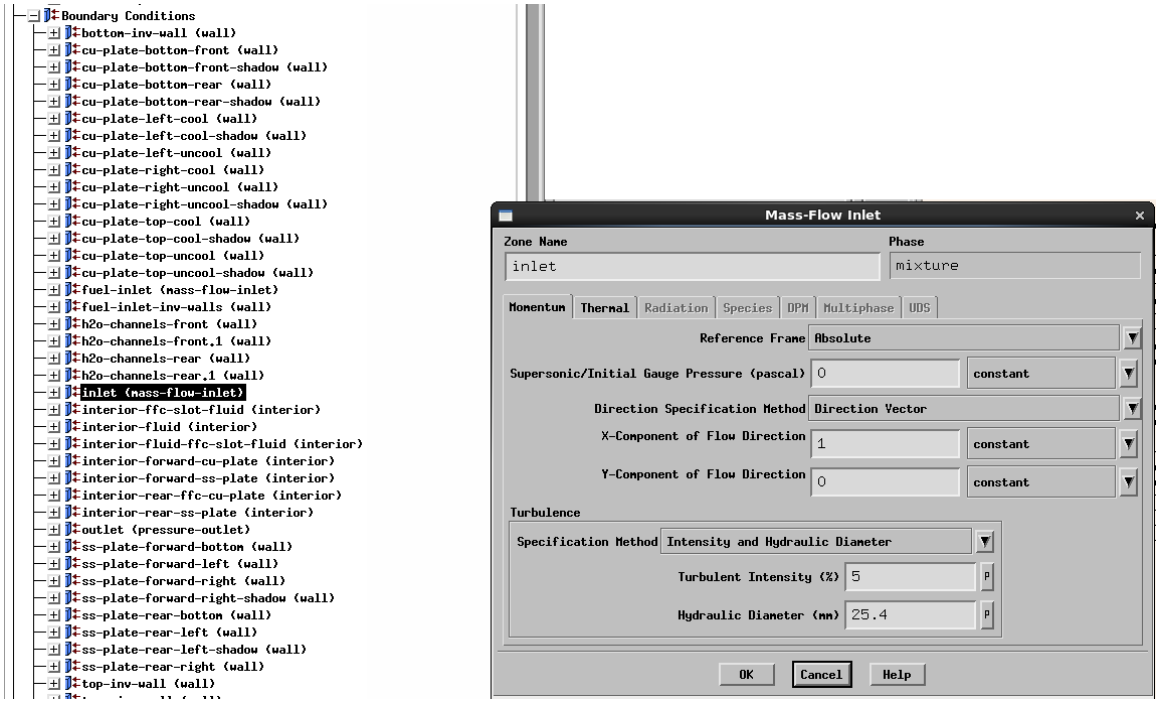

Figure 97. Fluent Settings: inlet momentum settings (mixture phase)

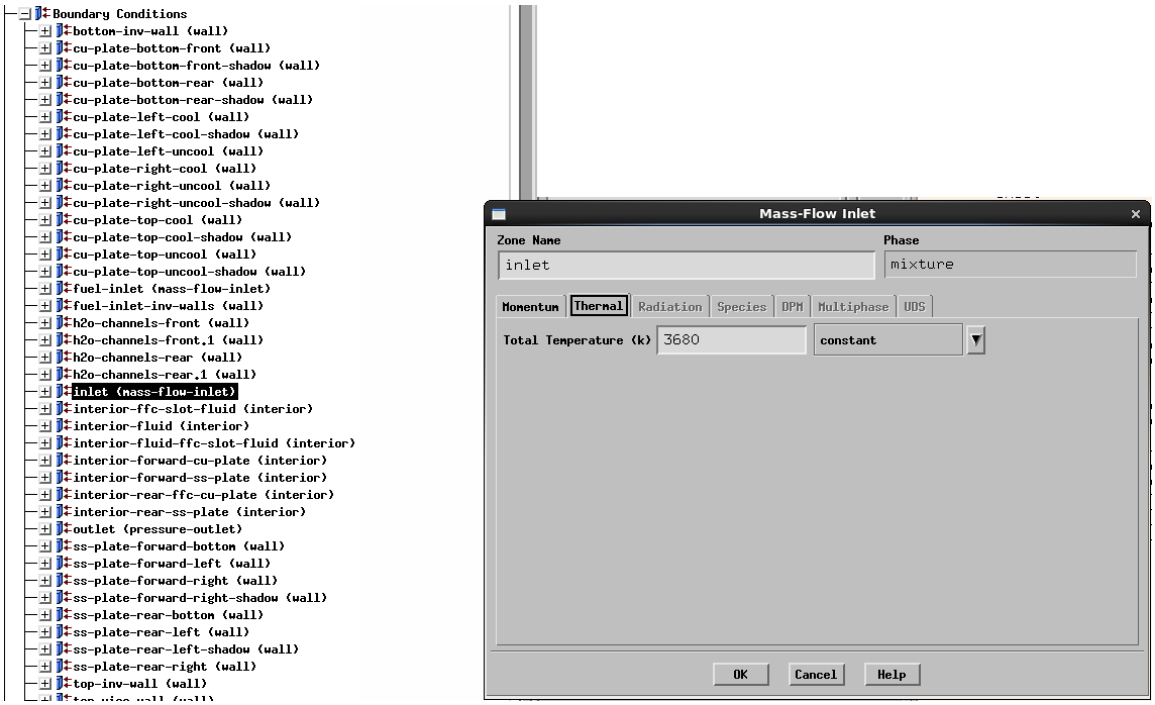

Figure 98. Fluent Settings: inlet thermal settings (mixture phase)

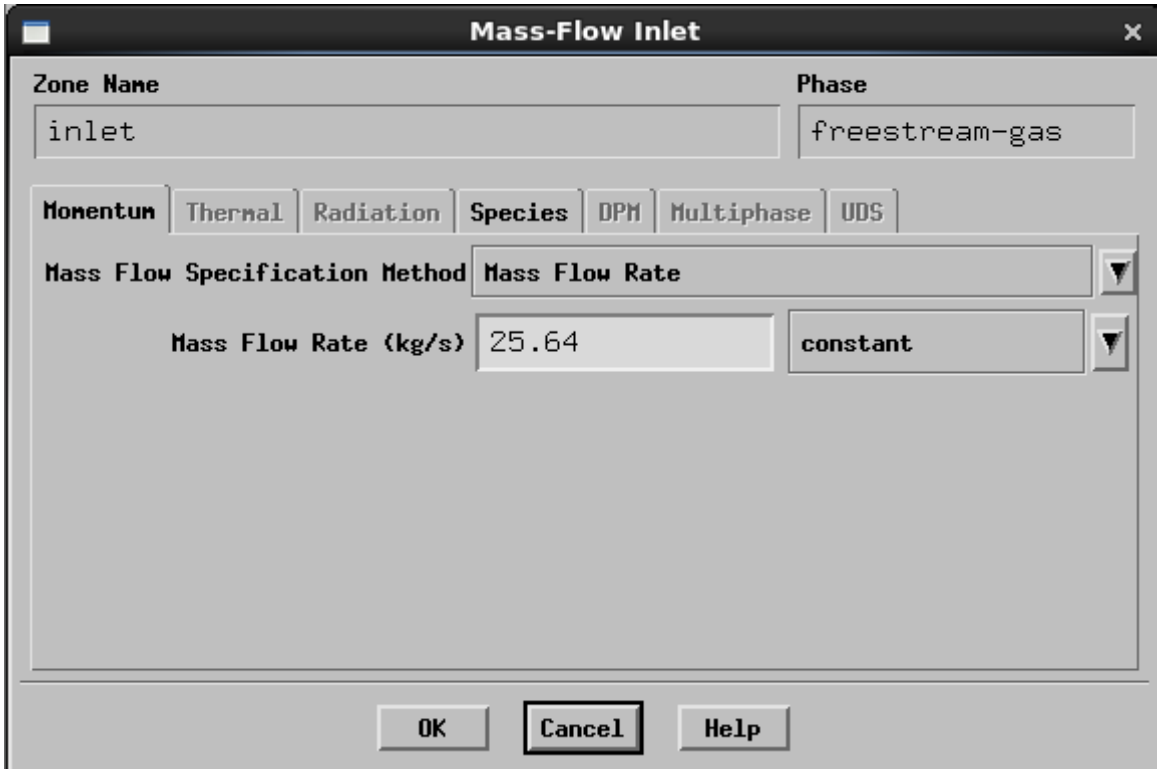

Figure 99. Fluent Settings: inlet momentum settings (freestream-gas phase)

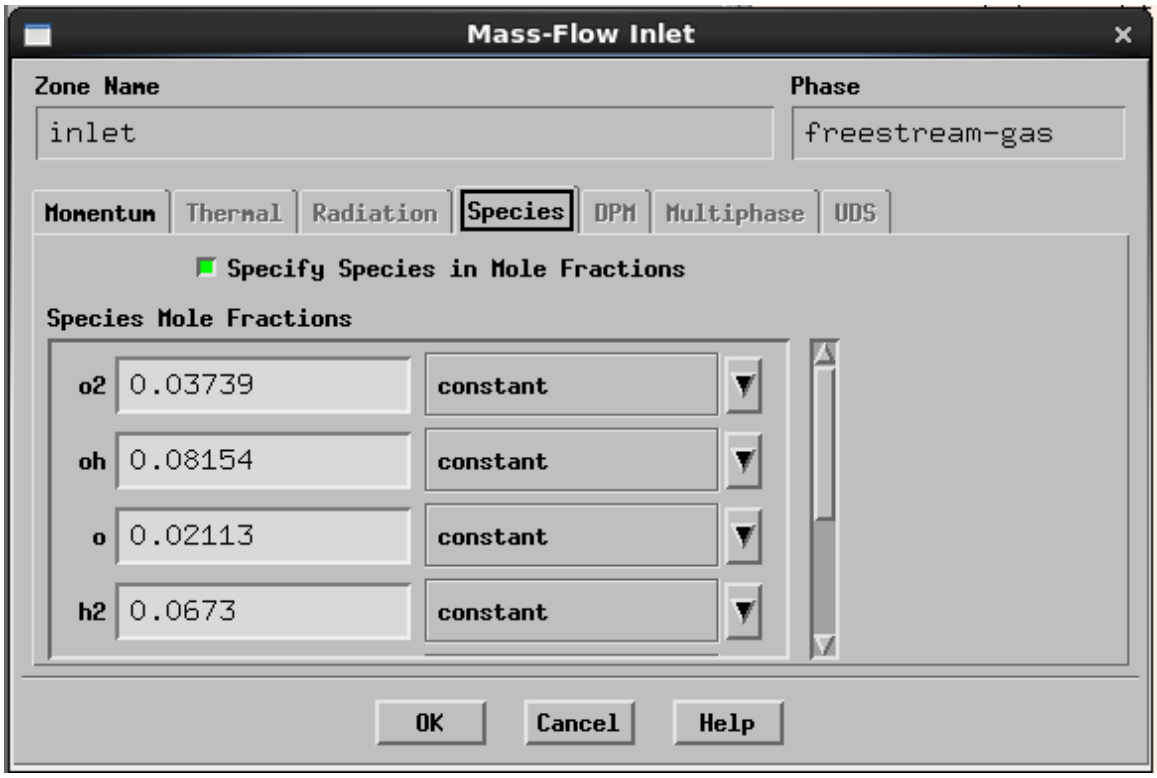

Figure 100. Fluent Settings: inlet species settings (freestream-gas phase)

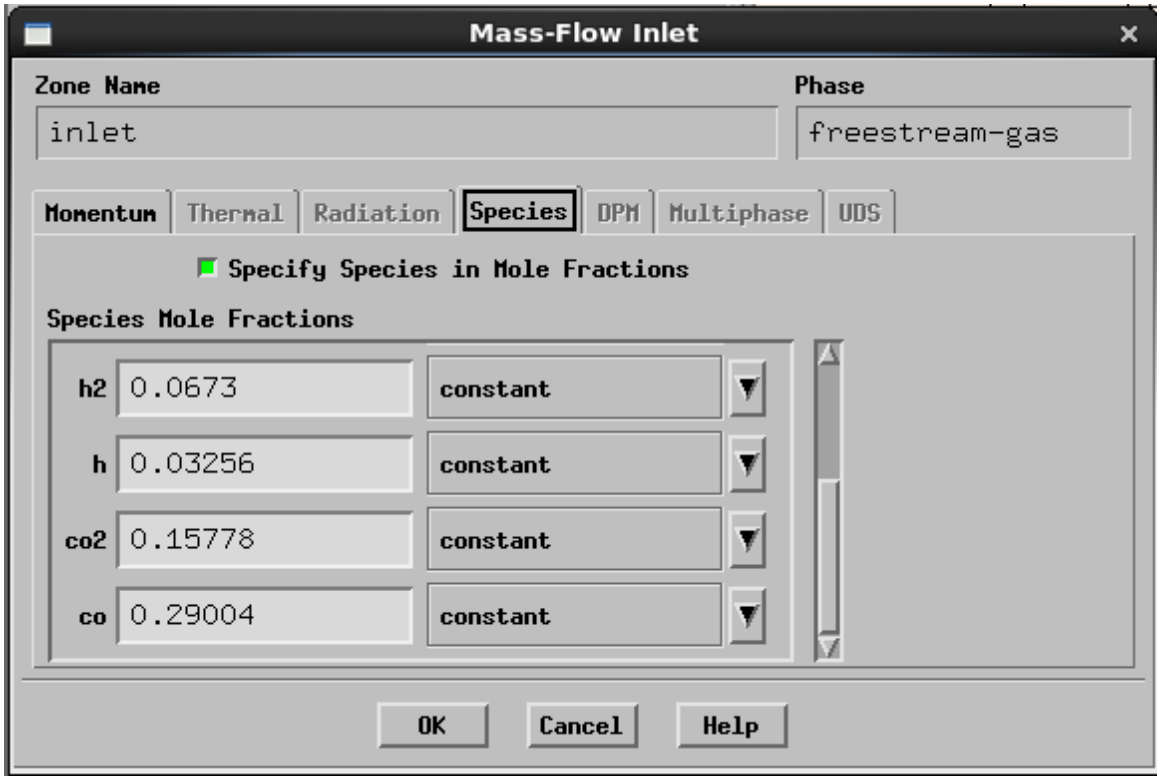

Figure 101. Fluent Settings: inlet species settings continued (freestream-gas phase)

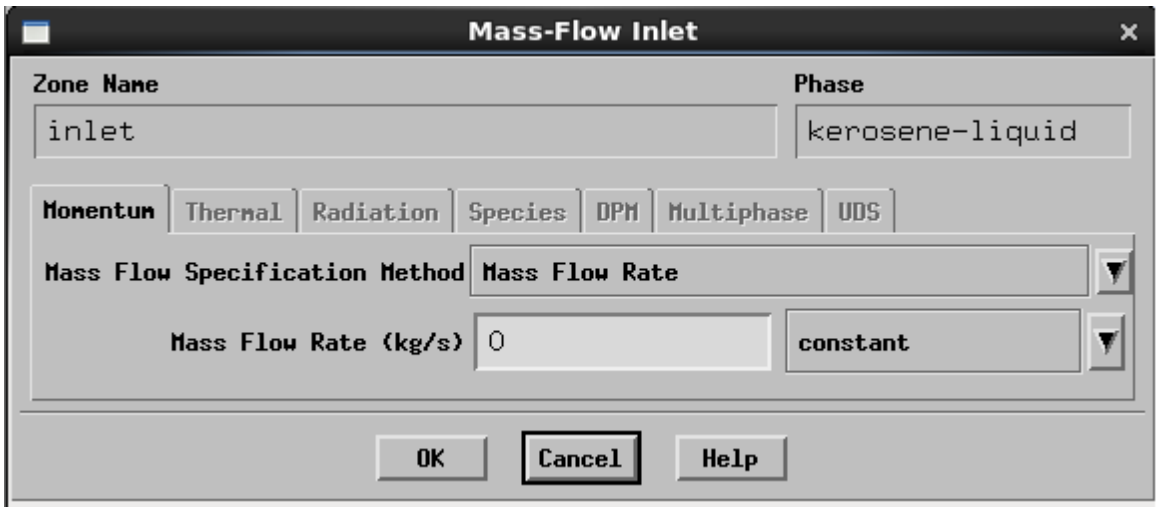

Figure 102. Fluent Settings: inlet momentum settings (kerosene-liquid phase)

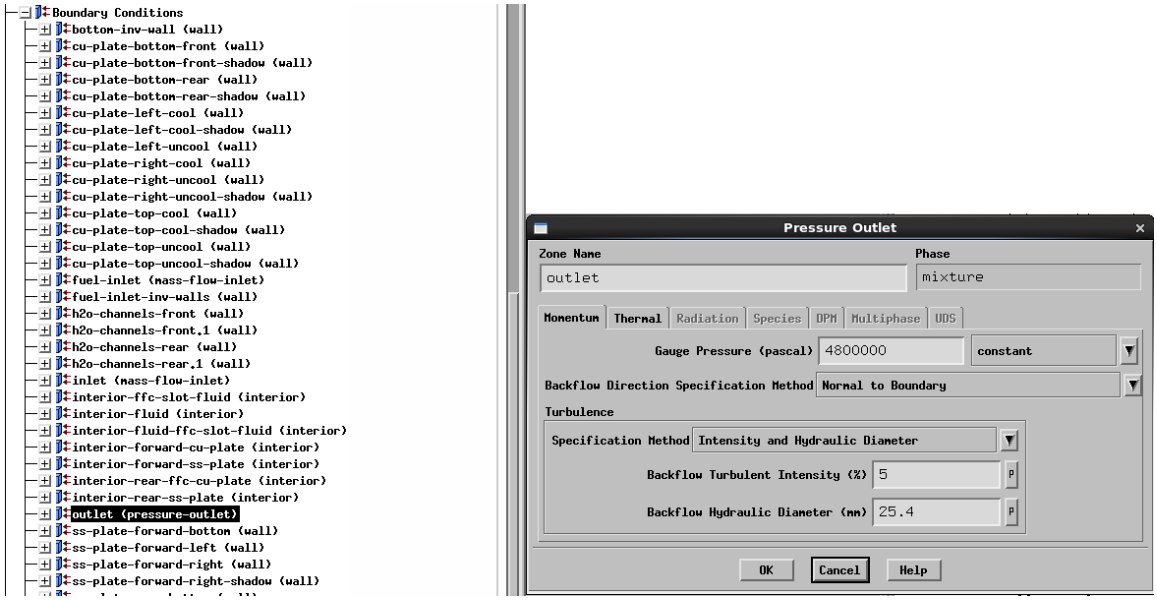

Figure 103. Fluent Settings: outlet momentum settings (mixture phase)

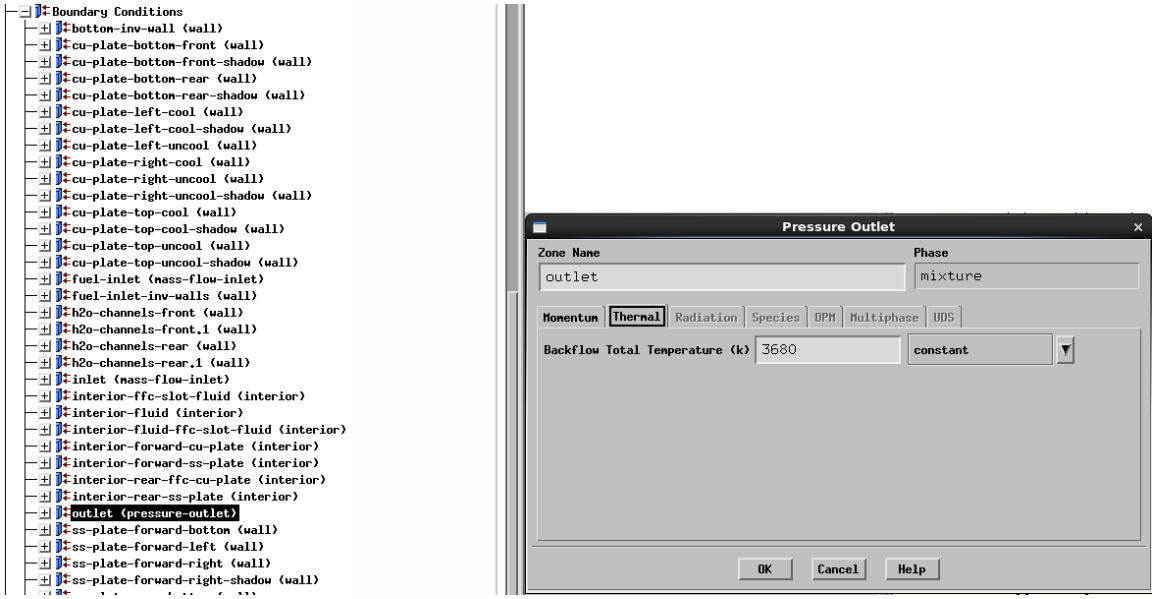

Figure 104. Fluent Settings: outlet thermal settings (mixture phase)

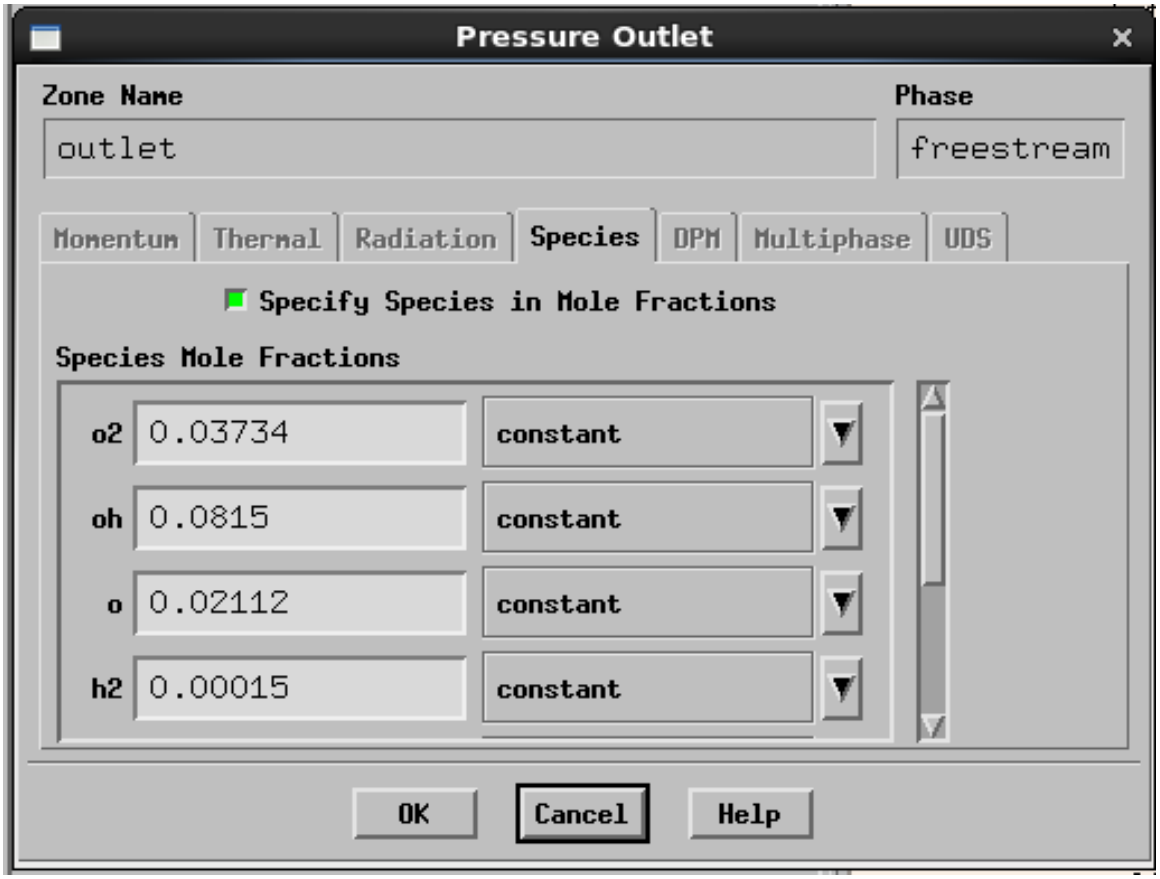

Figure 105. Fluent Settings: outlet species settings (freestream-gas phase)

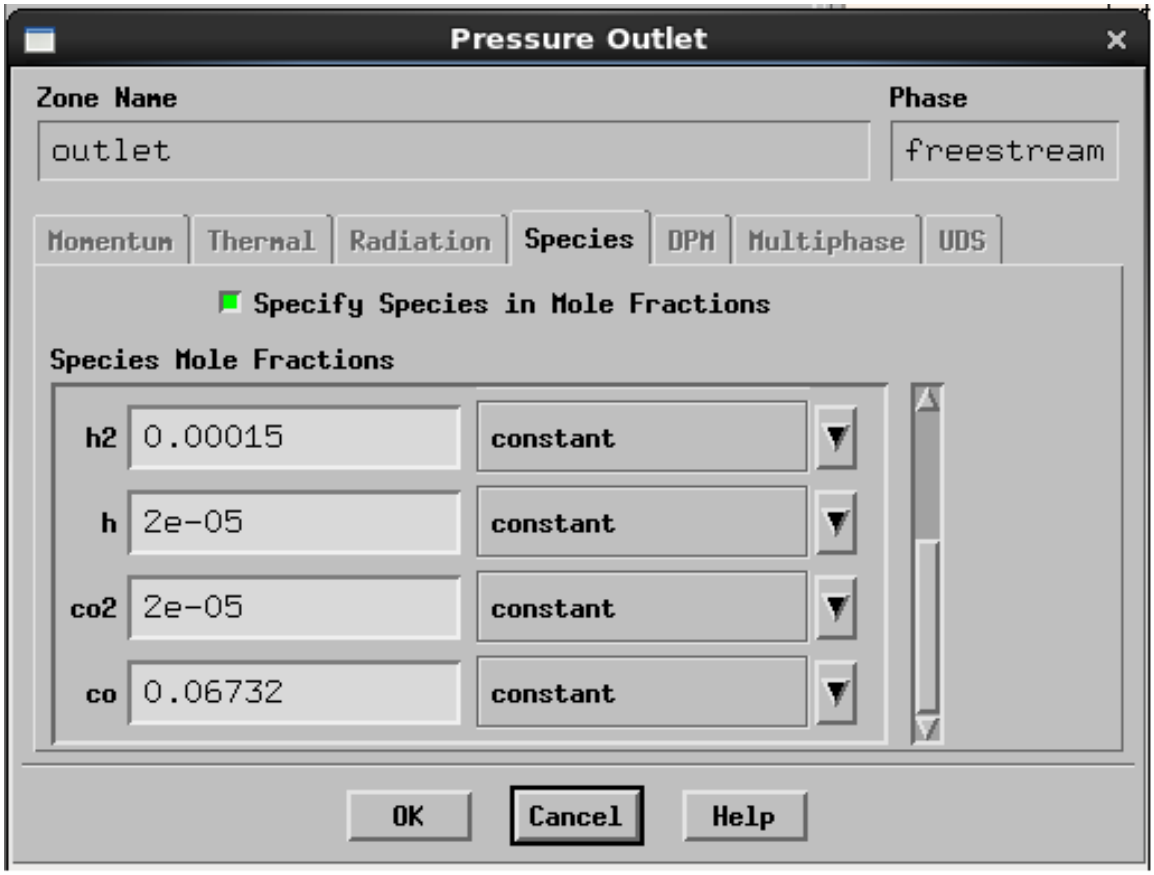

Figure 106. Fluent Settings: outlet species settings continued (freestream-gas phase)

Figure 107. Fluent Settings: outlet multiphase settings (kerosene-liquid phase)

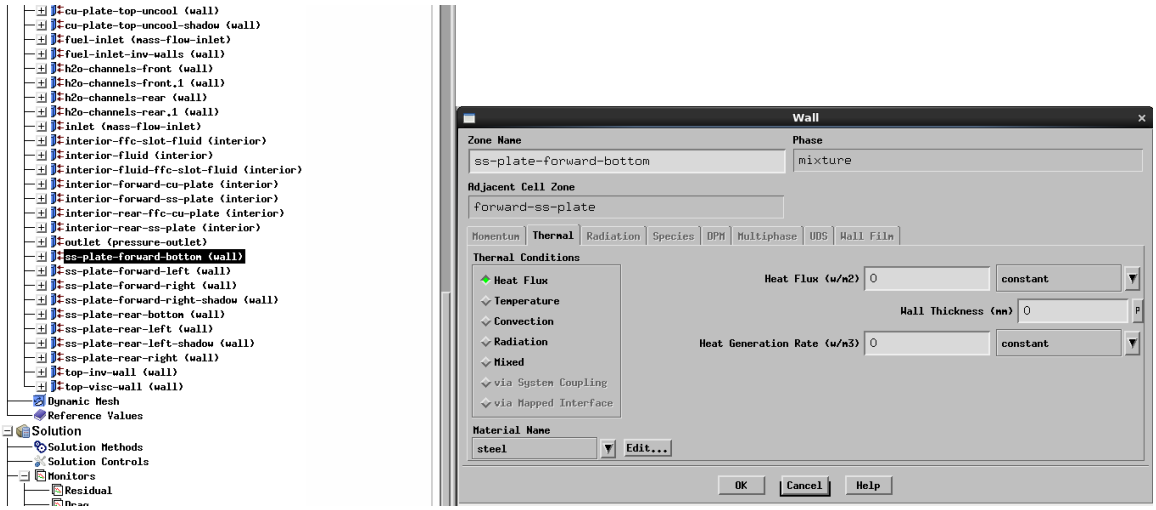

Figure 108. Fluent Settings: ss-plate-forward-bottom thermal settings

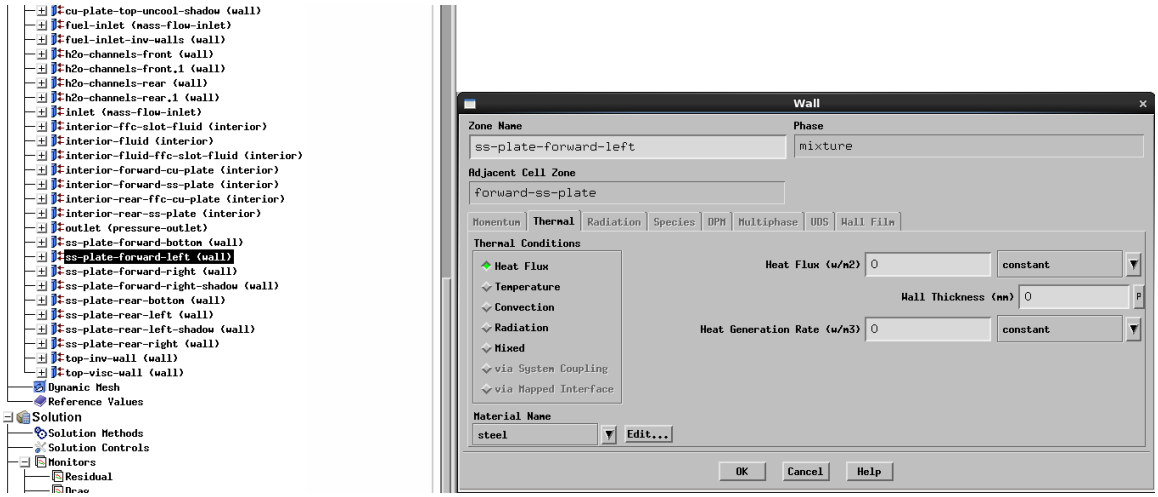

Figure 109. Fluent Settings: ss-plate-forward-left thermal settings

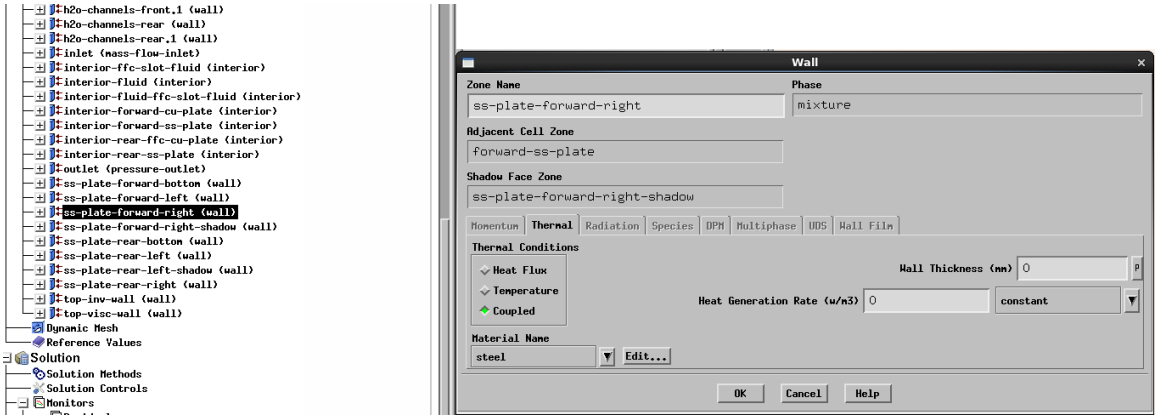

Figure 110. Fluent Settings: ss-plate-forward-right thermal settings

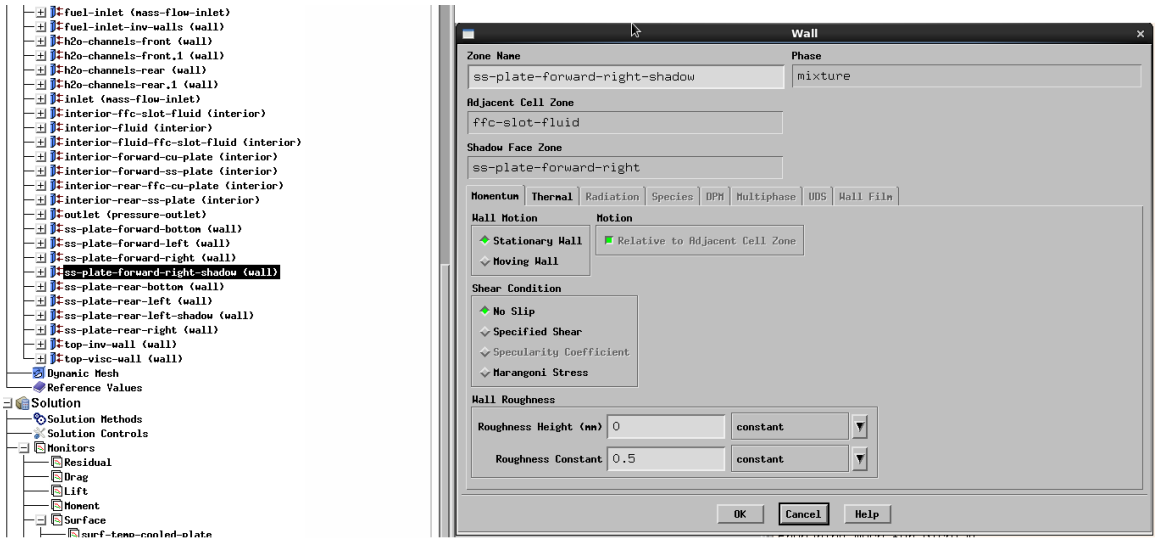

Figure 111. Fluent Settings: ss-plate-forward-right-shadow momentum settings

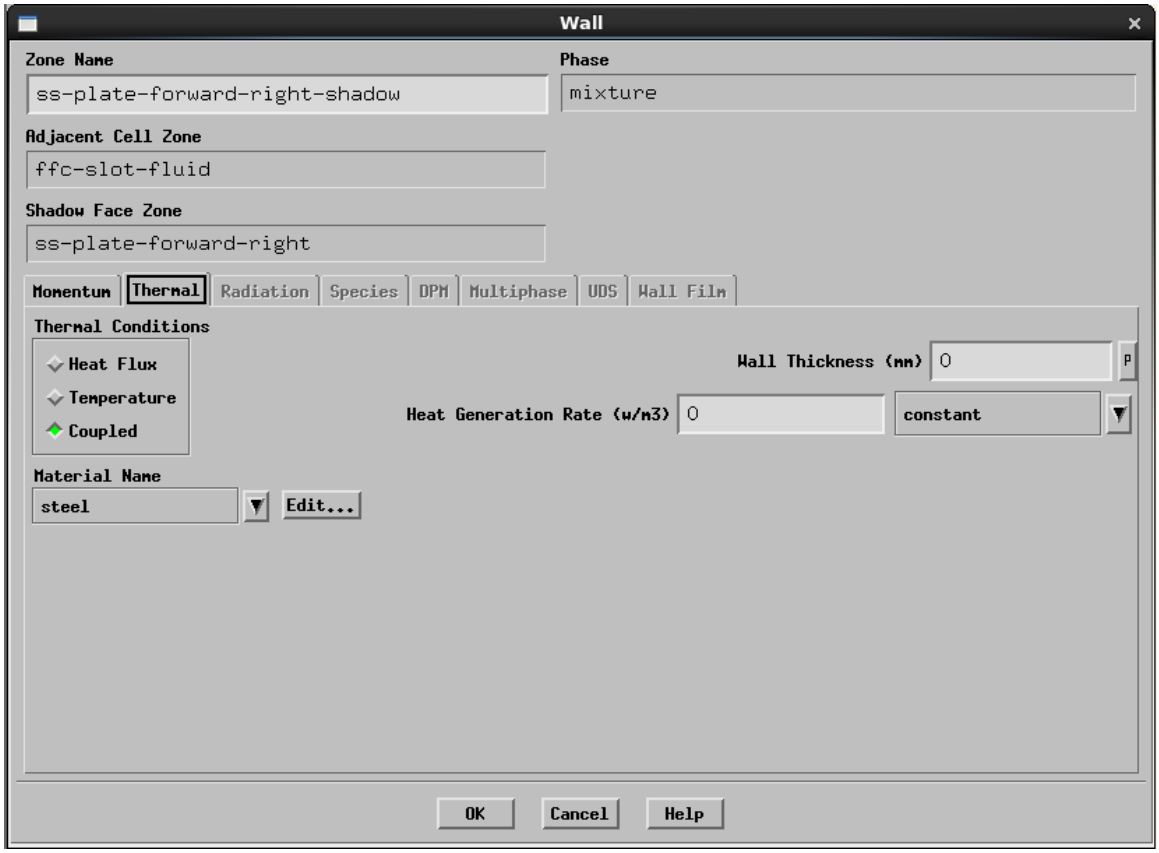

Figure 112. Fluent Settings: ss-plate-forward-right-shadow thermal settings

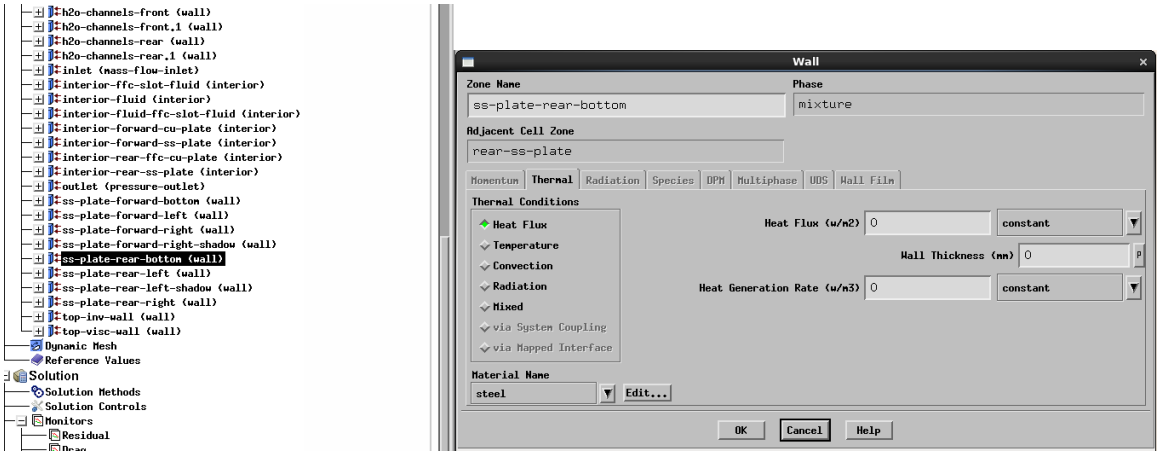

Figure 113. Fluent Settings: ss-plate-rear-bottom thermal settings

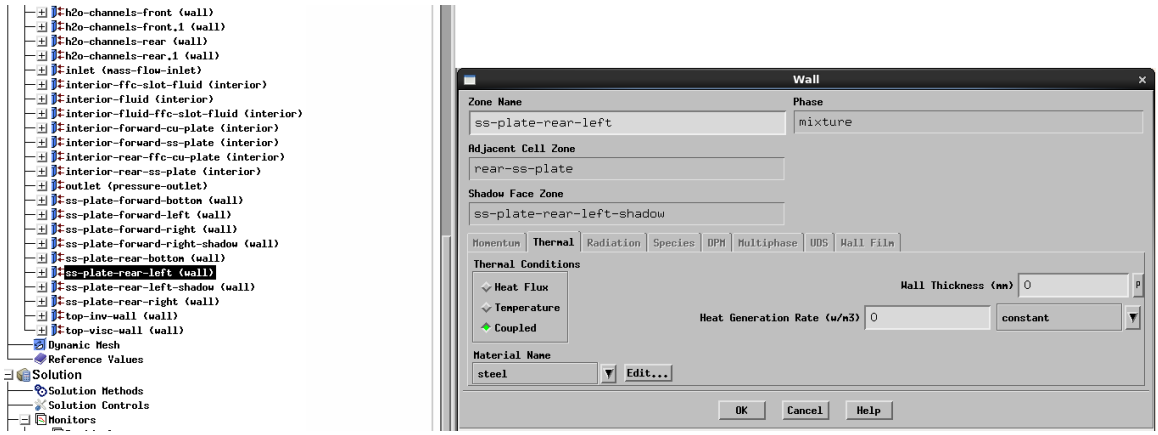

Figure 114. ss-plate-rear-left thermal settings

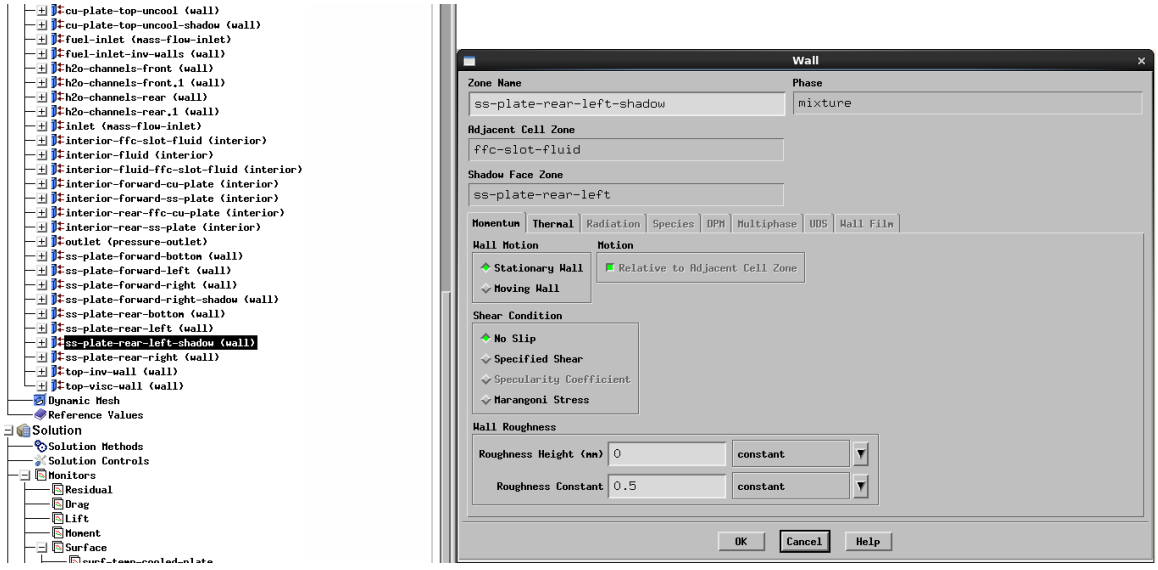

Figure 115. Fluent Settings: ss-plate-rear-left-shadow momentum settings

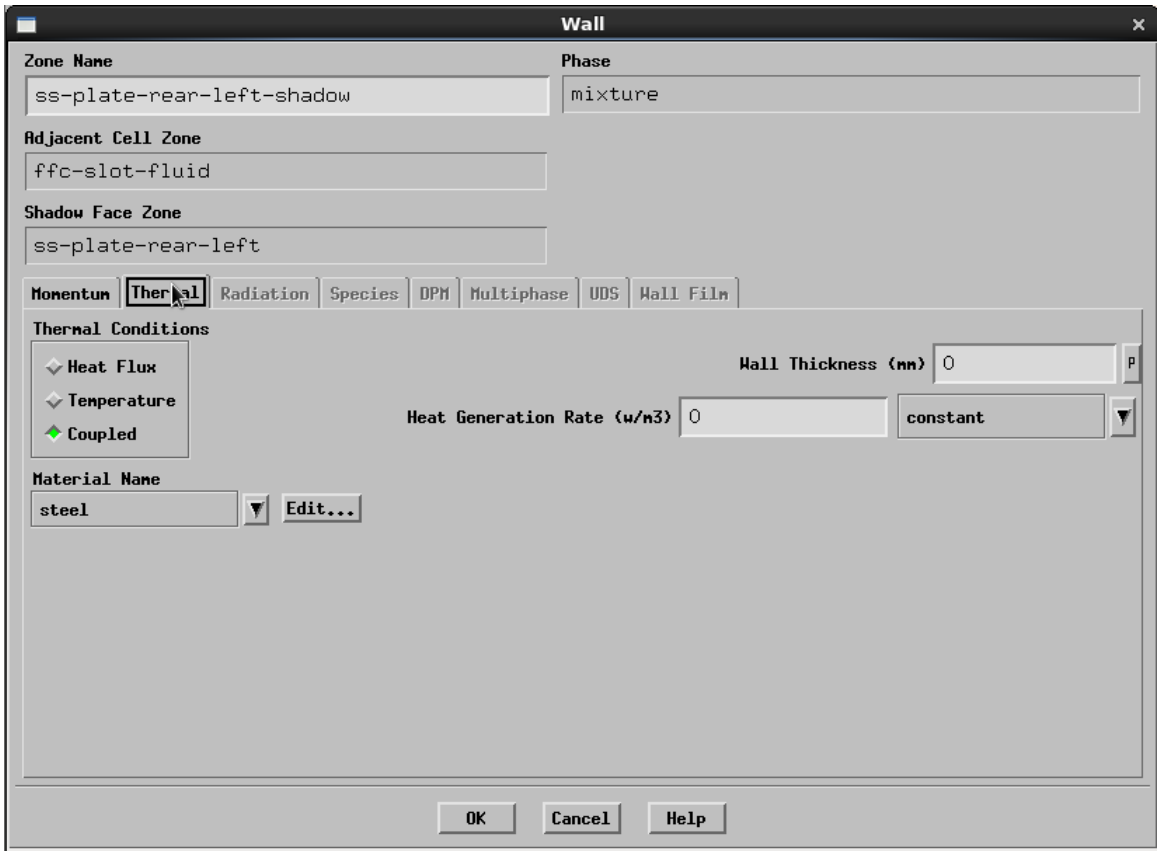

Figure 116. Fluent Settings: ss-plate-rear-left-shadow thermal settings

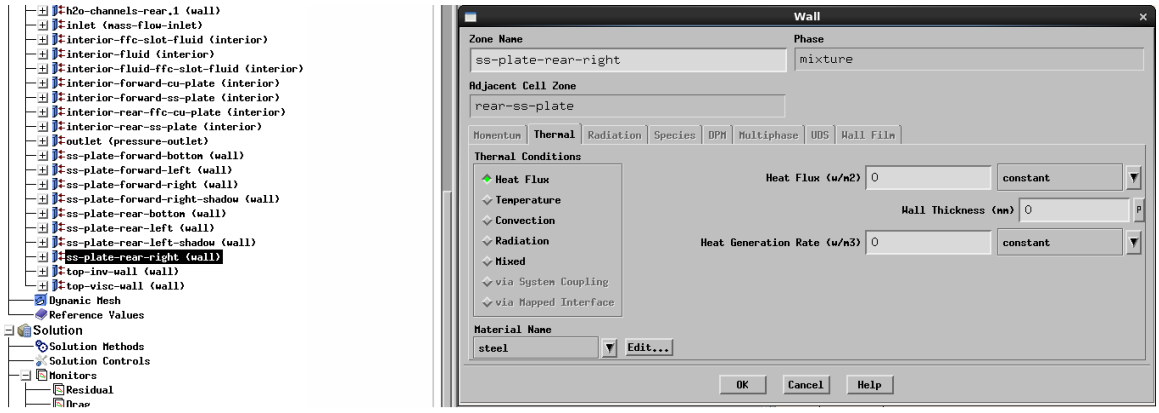

Figure 117. Fluent Settings: ss-plate-rear-right thermal settings

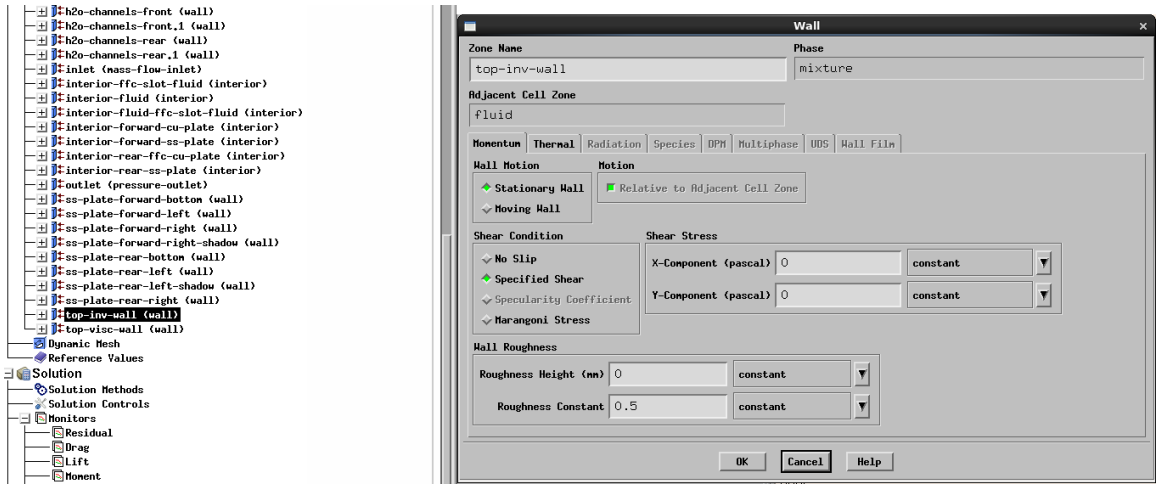

Figure 118. Fluent Settings: top-inv-wall momentum settings

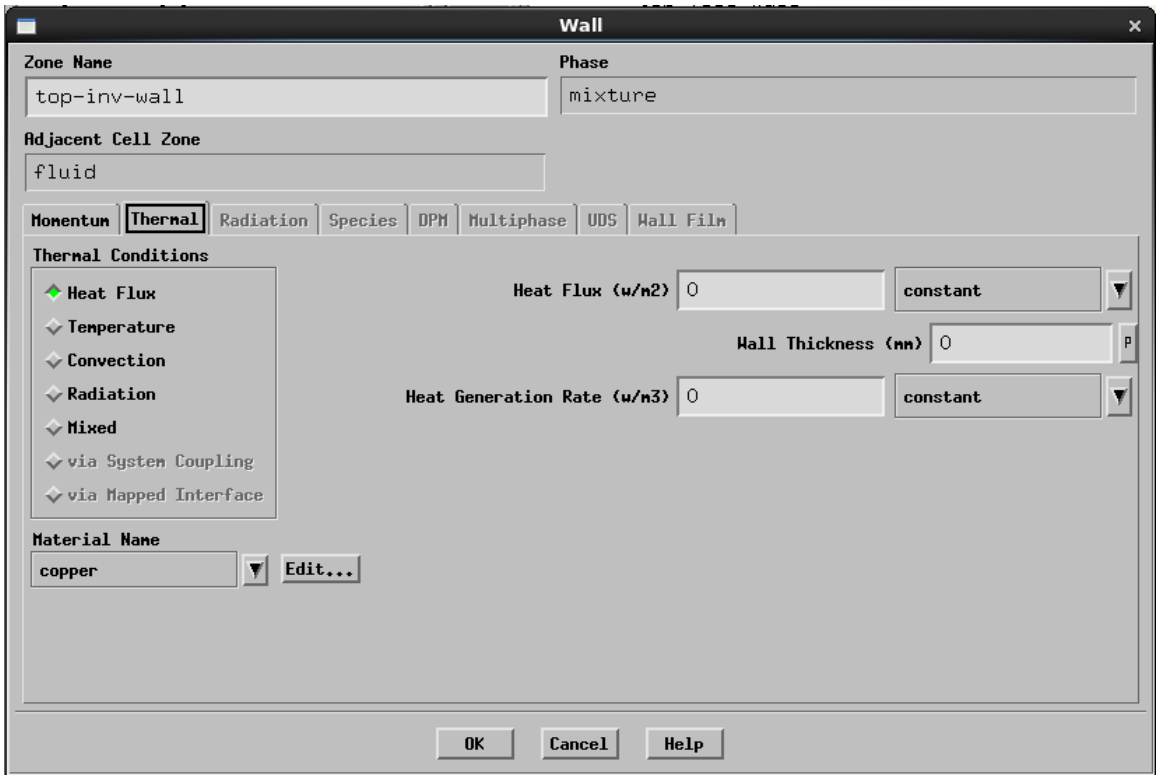

Figure 119. Fluent Settings: top-inv-wall thermal settings

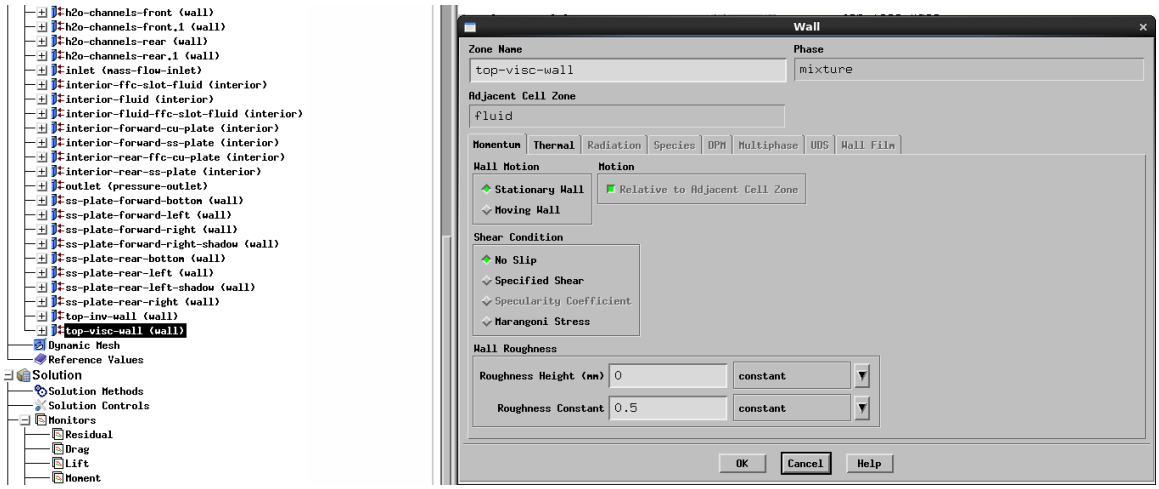

Figure 120. Fluent Settings: top-visc-wall momentum settings

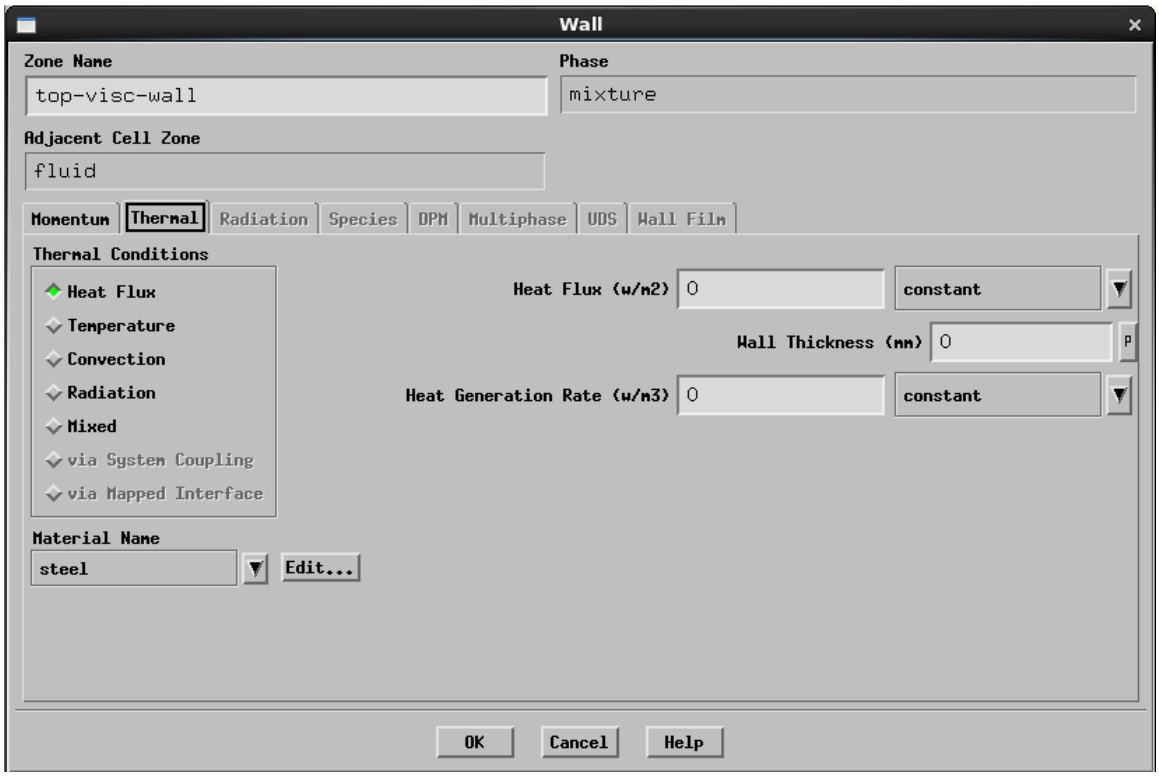

Figure 121. Fluent Settings: top-visc-wall thermal settings

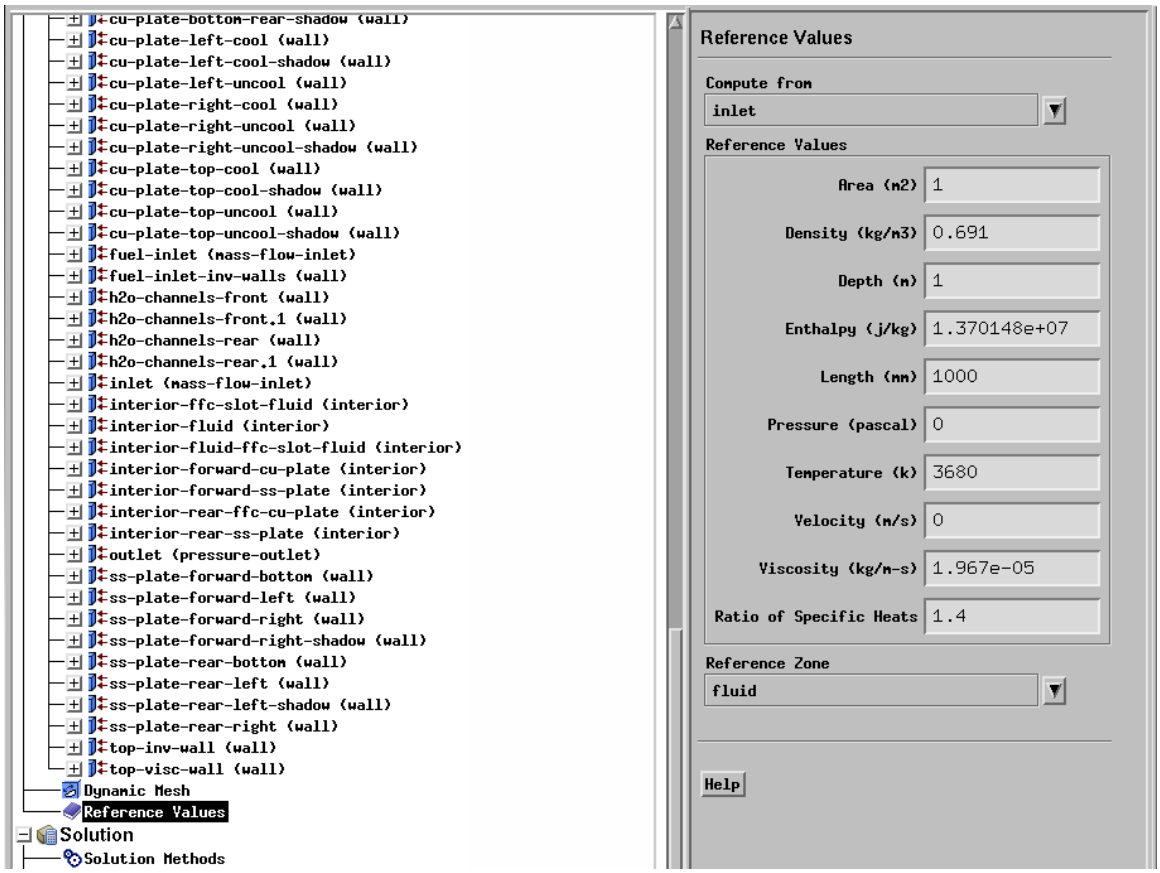

Figure 122. Fluent Settings: Reference Values

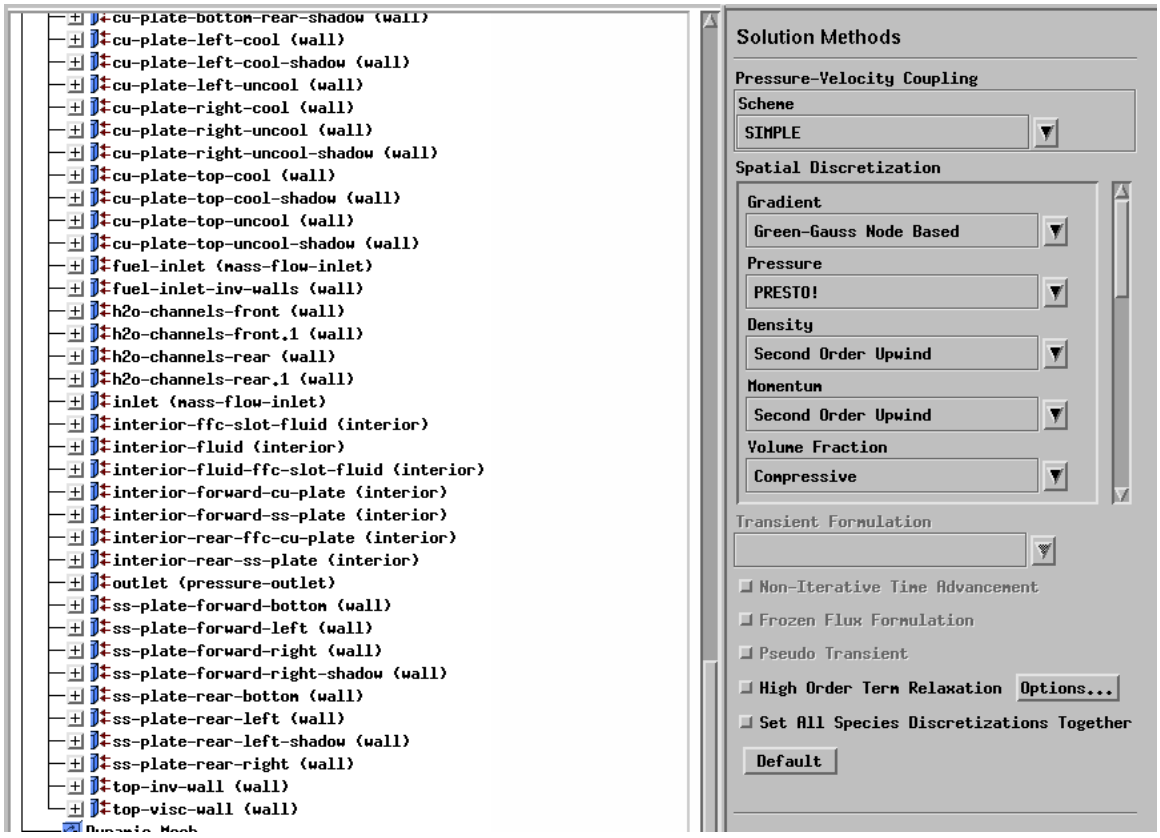

Figure 123. Fluent Settings: Solution methods for "Scheme", "Gradient" and "Volume Fraction". All other solution methods were set to SOU for final test cases.

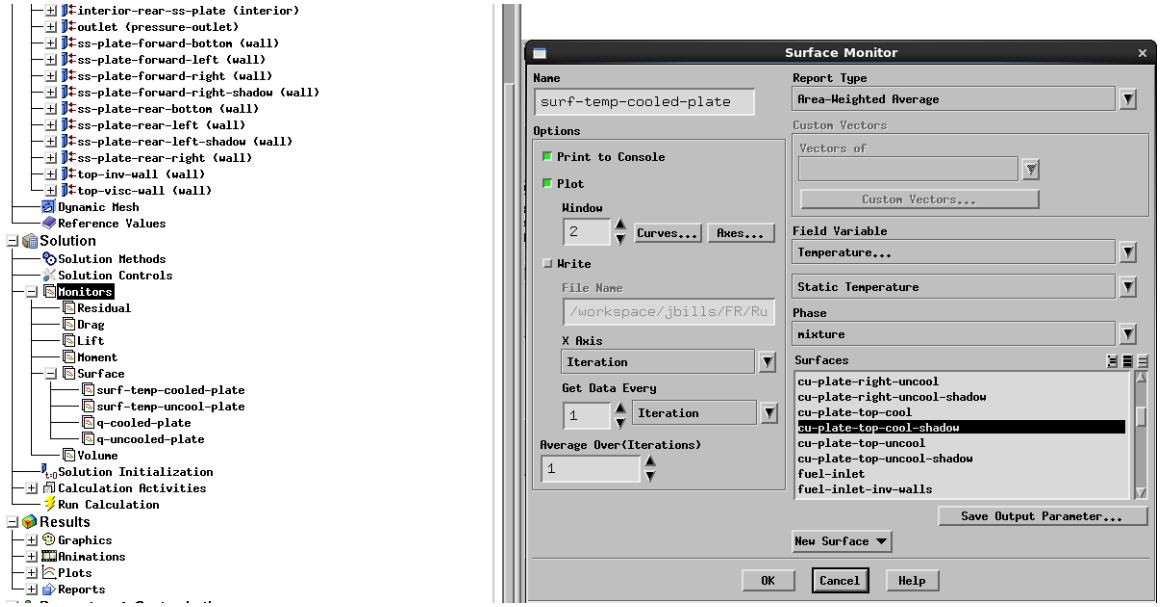

Figure 124. Fluent Settings: Surface monitor for $\overline{T}_s$ on the LFFC surface.

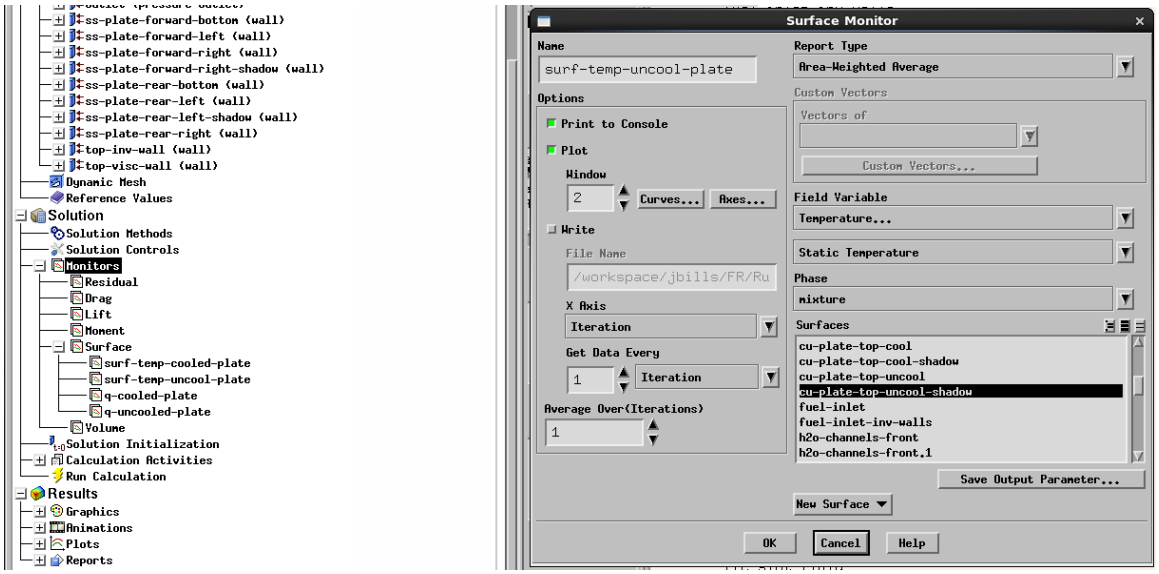

Figure 125. Fluent Settings: Surface monitor for $\overline{T}_s$ on the non-LFFC surface.

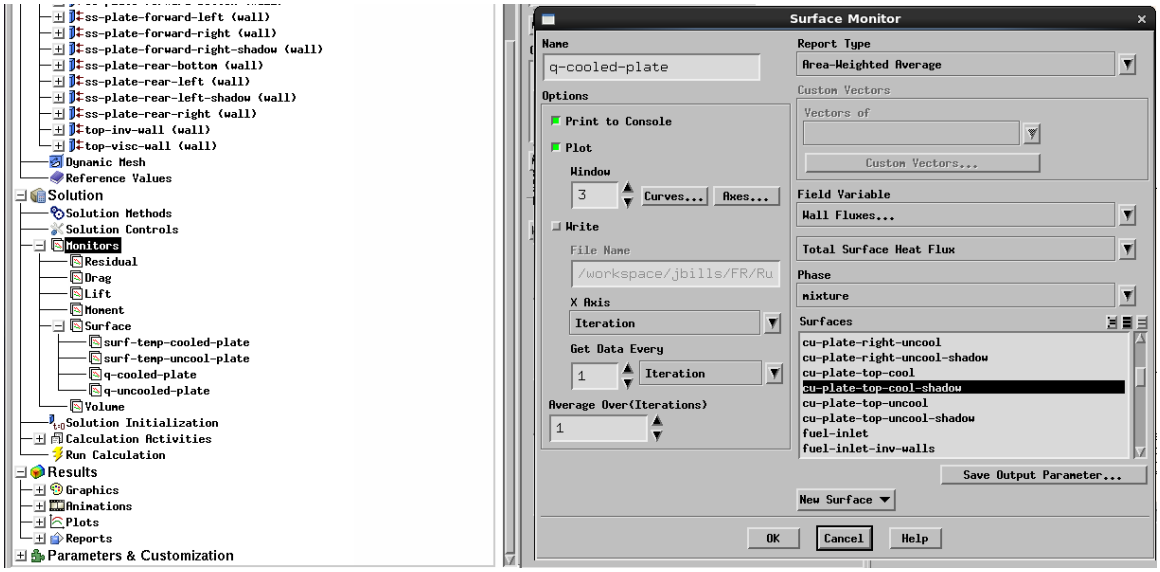

Figure 126. Fluent Settings: Surface monitor for $\overline{q''}_s$ on the LFFC surface.

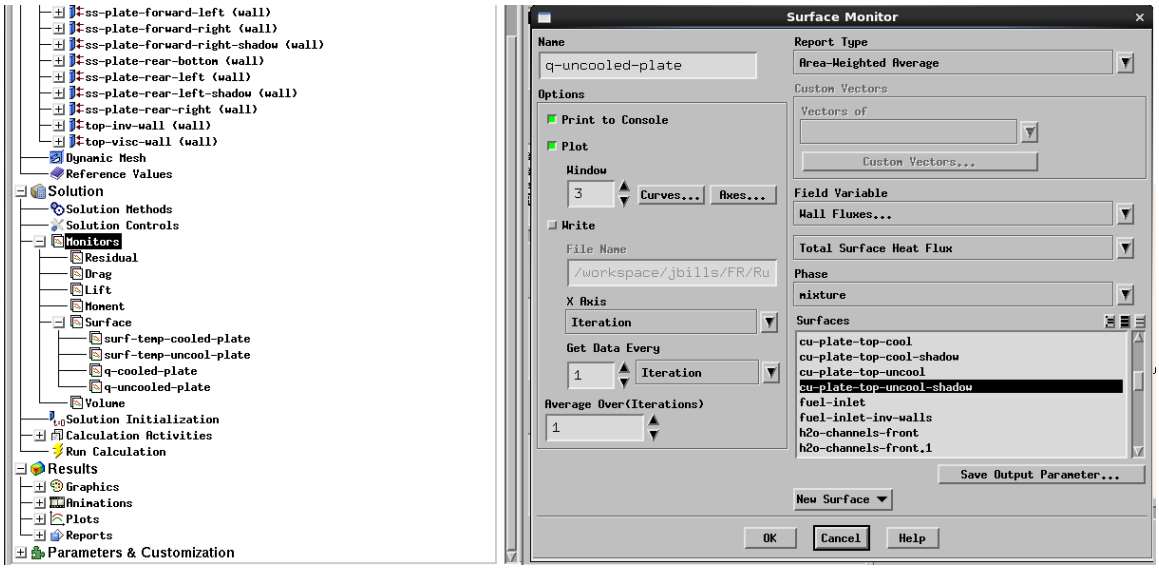

Figure 127. Fluent Settings: Surface monitor for $\overline{q''}_s$ on the non-LFFC surface.

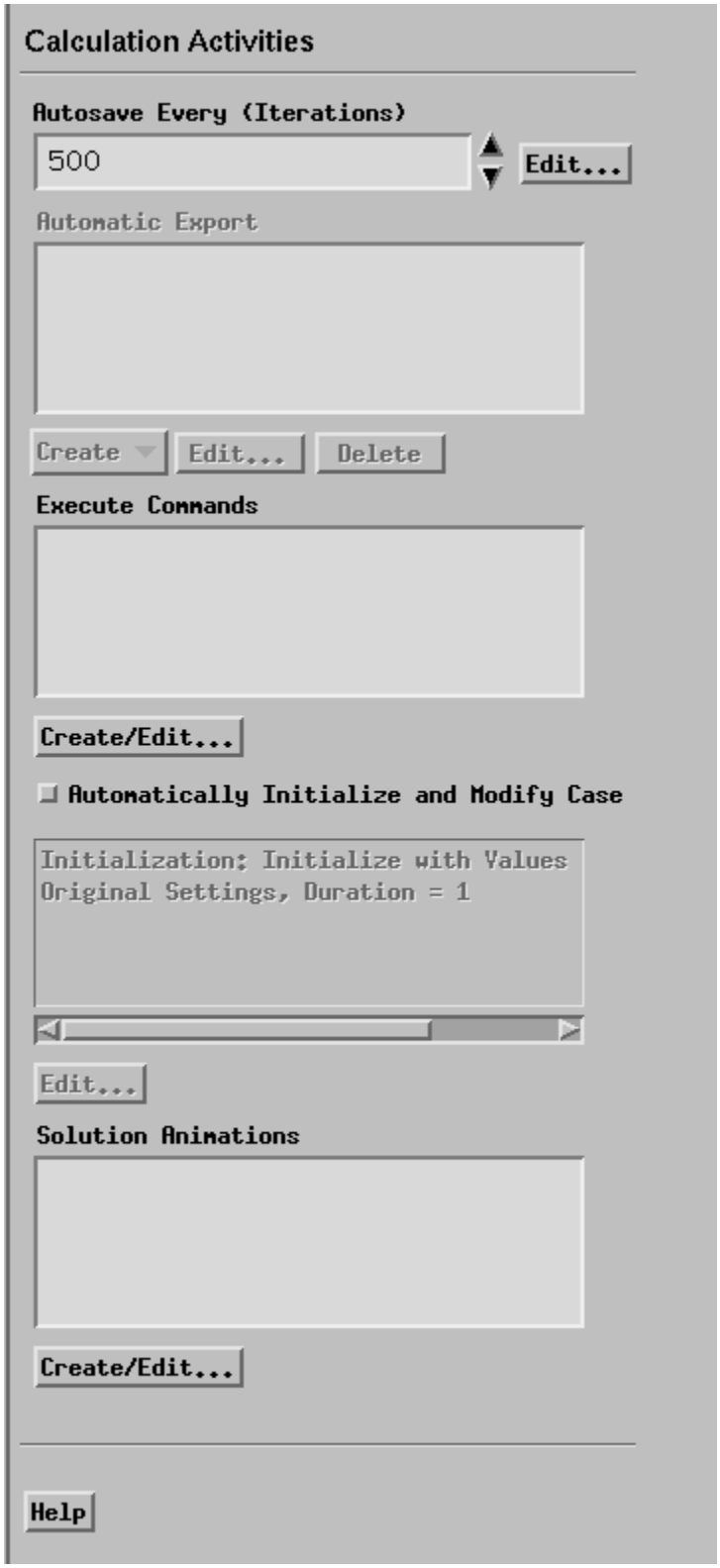

Figure 128. Fluent Settings: Calculation Activities

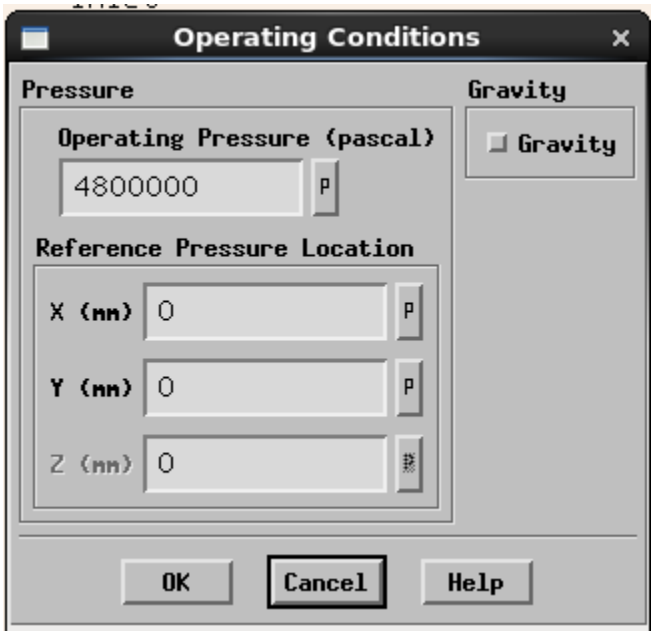

Figure 129. Fluent Settings: Operating Conditions. Set to outlet pressure to ensure non-dimensional pressures in domain are order 1.

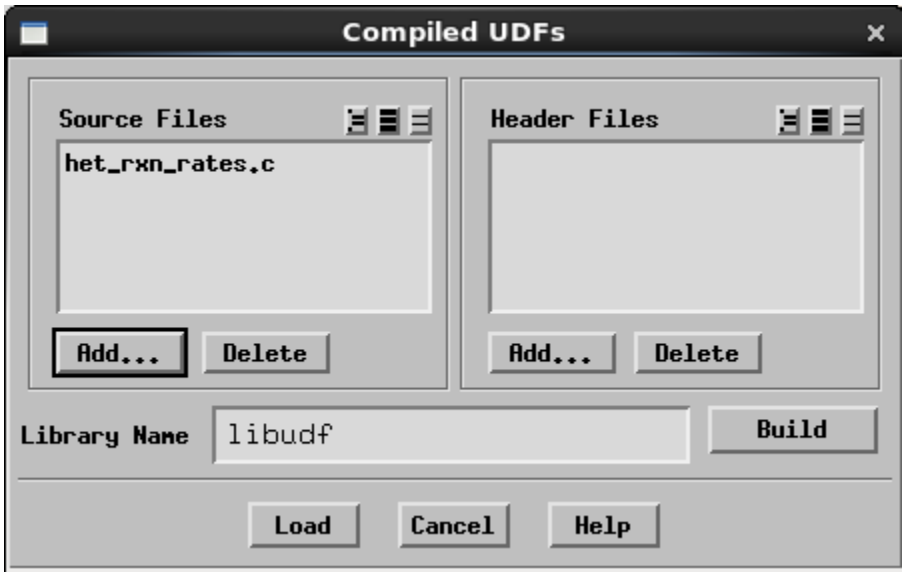

Figure 130. Fluent Settings: Compiled UDFs dialogue box. Used to compile UDF and hook to baseline Fluent code.

Appendix C. Numerical Model Boundary Conditions

Diagrams of the boundary condition names used for the numerical model within Fluent. Note that all boundary conditions with "shadow" appended on the end (see Figure 65) are co-located with their non-shadow counterpart, and thus are not directly called out in the diagrams. The boundary conditions for the cooling channels (h2o-channels) are only shown on a single channel but are repeated for all respective channels (either upstream or downstream of fuel injection slot). Figures begin on next page.

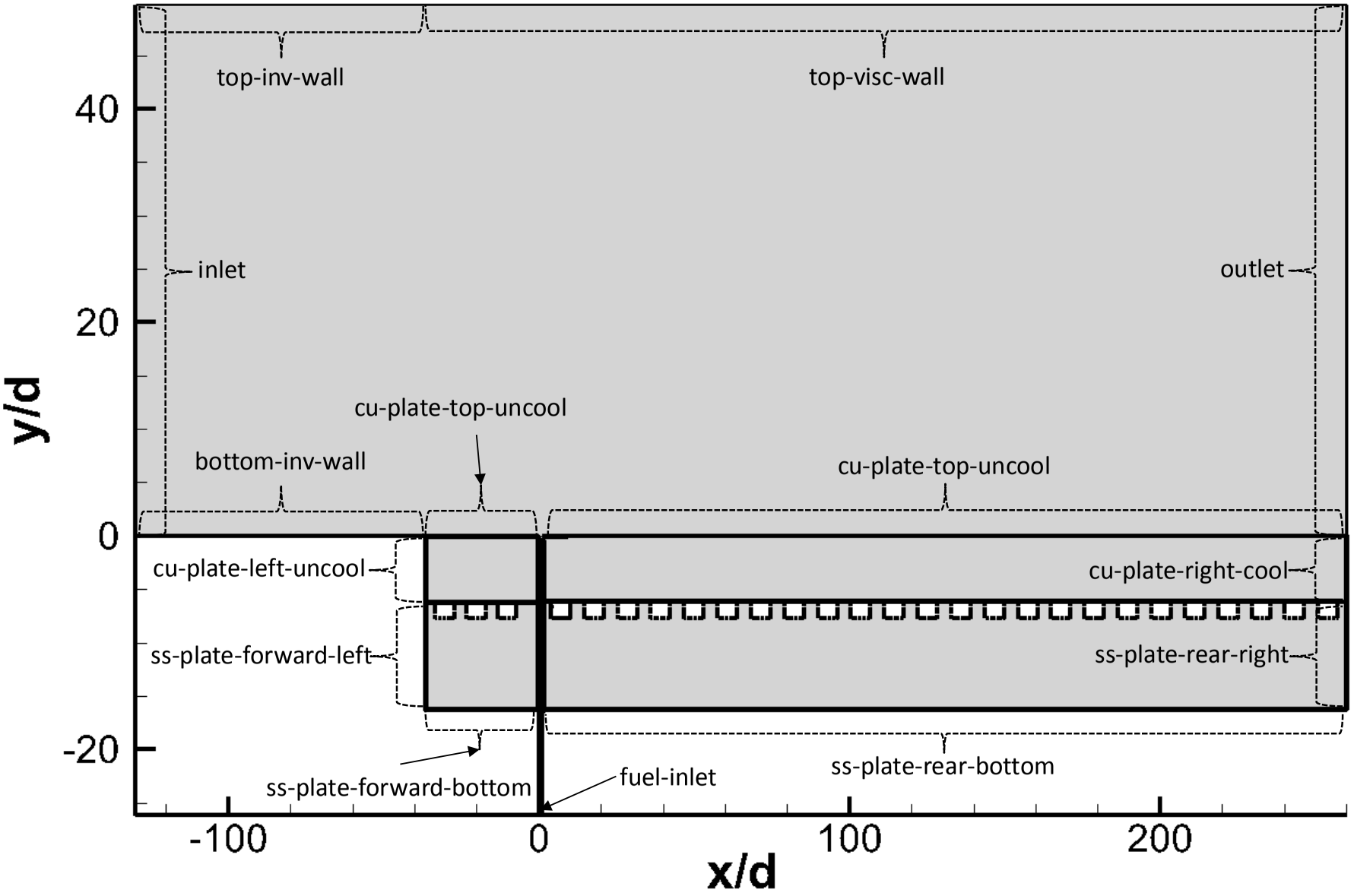

Figure 131. Diagram of boundary condition names (1/3). See Figure 65 for list.

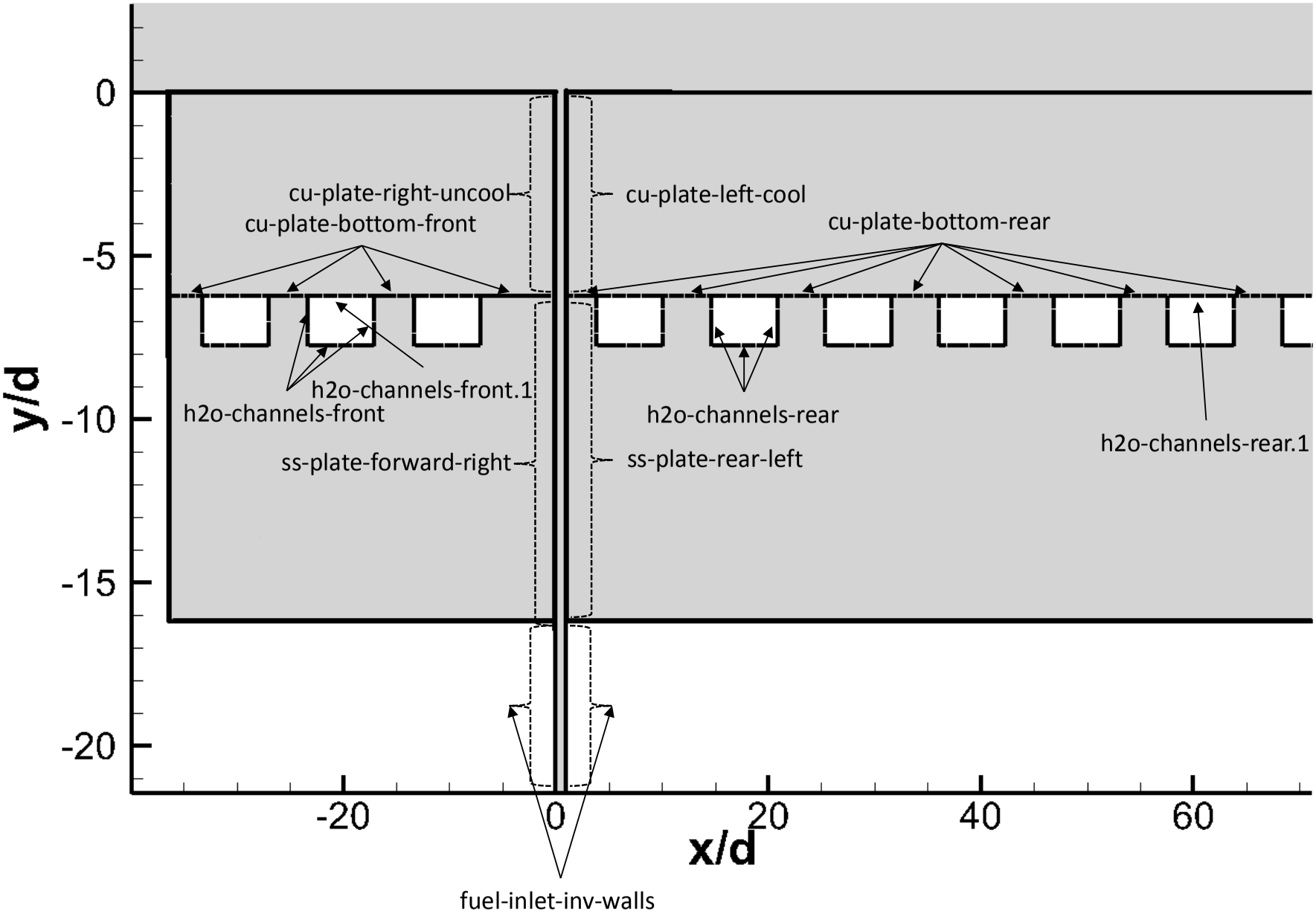

Figure 132. Diagram of boundary condition names continued (2/3). See Figure 65 for list.

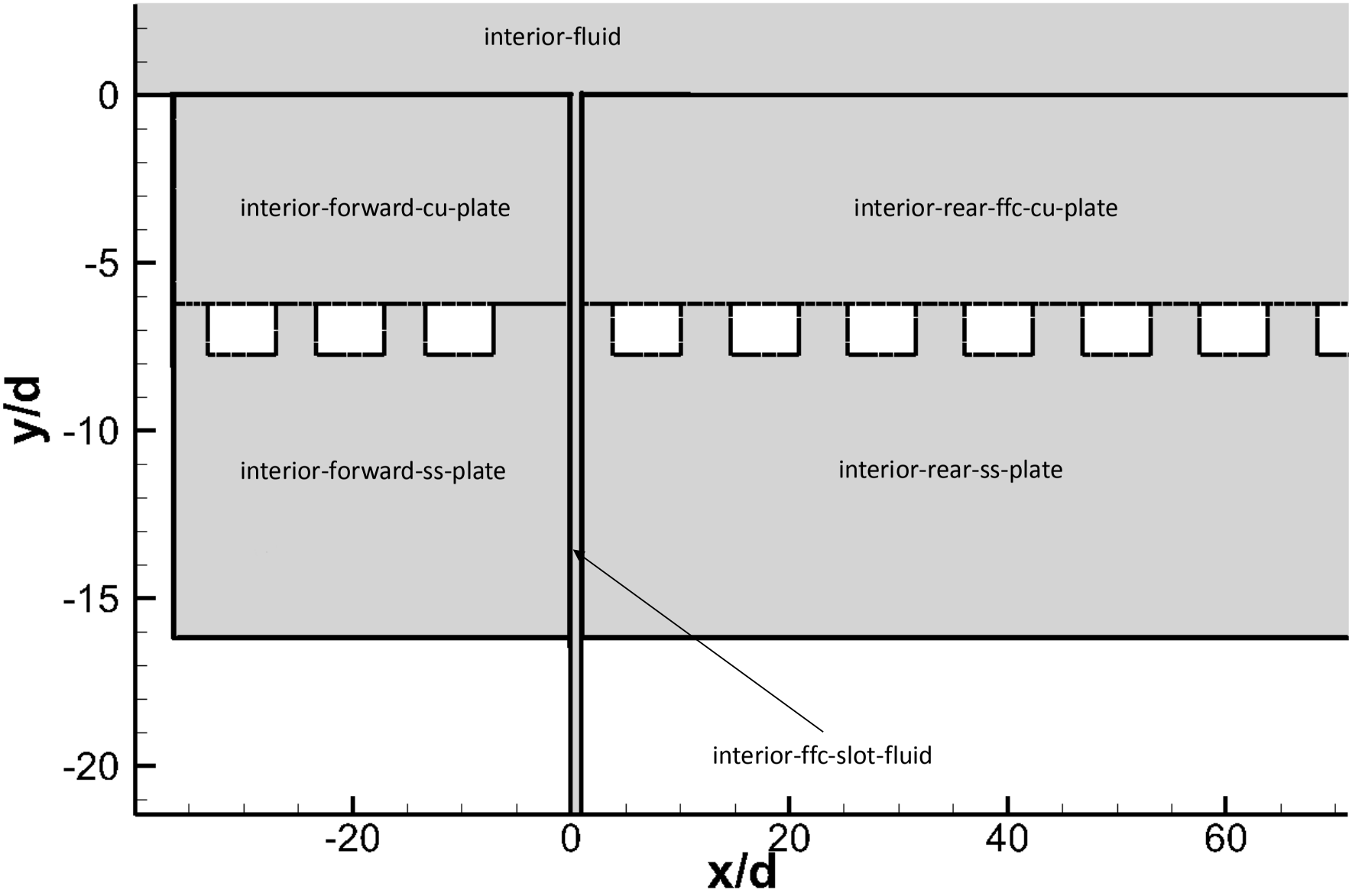

Figure 133. Diagram of boundary condition names continued (3/3). See Figure 65 for list.

Appendix D. CEA Results: RP-1 and O_2 Combustion

Inputs and results used to generate the mixture products from the combustion of RP-1 and O_2 with a mixture ratio of 2.8. Printout starts on next page.

```
 ******************************************************************************

         NASA-GLENN CHEMICAL EQUILIBRIUM PROGRAM CEA2, MAY 21, 2004
                   BY  BONNIE MCBRIDE AND SANFORD GORDON
      REFS: NASA RP-1311, PART I, 1994 AND NASA RP-1311, PART II, 1996

 ******************************************************************************



 prob
 o/f=2.8 tp t,k=3680 p,psia=700
 react
   fuel=RP-1 wt=1 t,k=298
   oxid=O2(L) wt=1 t,k=98
 only
   H H2 H2O O O2 OH CO CO2
 end

 OPTIONS: TP=T  HP=F  SP=F  TV=F  UV=F  SV=F  DETN=F  SHOCK=F  REFL=F  INCD=F
 RKT=F  FROZ=F  EQL=F  IONS=F  SIUNIT=T  DEBUGF=F  SHKDBG=F  DETDBG=F  TRNSPT=F

 T,K =  3680.0000

 TRACE= 0.00E+00  S/R= 0.000000E+00  H/R= 0.000000E+00  U/R= 0.000000E+00

 P,BAR =    48.263113

 NOTE! REACTANT O2(L)          HAS BEEN DEFINED FOR TEMPERATURE   90.17K ONLY.
 YOUR TEMPERATURE ASSIGNMENT   98.00 IS NOT = BUT <10 K FROM THIS VALUE. (REACT)

    REACTANT          WT.FRAC   (ENERGY/R),K   TEMP,K  DENSITY
        EXPLODED FORMULA
 F: RP-1             1.000000  -0.297284E+04   298.00  0.0000
          C  1.00000  H  1.95000
 O: O2(L)            1.000000  -0.156101E+04    90.17  0.0000
          O  2.00000

  SPECIES BEING CONSIDERED IN THIS SYSTEM
 (CONDENSED PHASE MAY HAVE NAME LISTED SEVERAL TIMES)
  LAST thermo.inp UPDATE:    9/09/04

  tpis79  *CO              g 9/99  *CO2             g 6/97  *H
  tpis78  *H2              g 8/89  H2O              g 5/97  *O
  g 4/02  *OH              tpis89  *O2

 O/F =   2.800000

                       EFFECTIVE FUEL     EFFECTIVE OXIDANT        MIXTURE
 ENTHALPY                   h(2)/R             h(1)/R               h0/R
 (KG-MOL)(K)/KG       -0.21270751E+03     -0.48783267E+02     -0.91921225E+02

 KG-FORM.WT./KG             bi(2)              bi(1)              b0i
  C                    0.71550294E-01      0.00000000E+00      0.18829025E-01
  *H                   0.13952307E+00      0.00000000E+00      0.36716598E-01
  *O                   0.00000000E+00      0.62502344E-01      0.46054359E-01
```

POINT	ITN	T	C	H	O
1	6	3680.000	-15.654	-10.206	-15.259

THERMODYNAMIC EQUILIBRIUM PROPERTIES AT ASSIGNED

TEMPERATURE AND PRESSURE

CASE =

	REACTANT	WT FRACTION (SEE NOTE)	ENERGY KJ/KG-MOL	TEMP K
FUEL	RP-1	1.0000000	-24717.700	298.000
OXIDANT	O2(L)	1.0000000	-12979.000	90.170

O/F= 2.80000 %FUEL= 26.315789 R,EQ.RATIO= 1.216309 PHI,EQ.RATIO= 1.216309

THERMODYNAMIC PROPERTIES

P, BAR	48.263
T, K	3680.00
RHO, KG/CU M	3.7515 0
H, KJ/KG	-471.04
U, KJ/KG	-1757.56
G, KJ/KG	-41986.2
S, KJ/(KG)(K)	11.2813
M, (1/n)	23.783
(dLV/dLP)t	-1.05299
(dLV/dLT)p	1.9071
Cp, KJ/(KG)(K)	7.4667
GAMMAs	1.1329
SON VEL,M/SEC	1207.3

MOLE FRACTIONS

*CO	0.29004
*CO2	0.15778
*H	0.03256
*H2	0.06730
H2O	0.31227
*O	0.02113
*OH	0.08154
*O2	0.03739

* THERMODYNAMIC PROPERTIES FITTED TO 20000.K

PRODUCTS WHICH WERE CONSIDERED BUT WHOSE MOLE FRACTIONS
WERE LESS THAN 5.000000E-06 FOR ALL ASSIGNED CONDITIONS

NOTE. WEIGHT FRACTION OF FUEL IN TOTAL FUELS AND OF OXIDANT IN TOTAL OXIDANTS

Appendix E. Journal Files

```
;Author: Jacob Bills
;Fuses double-sided walls exported as two separate boundaries from Pointwise
;and changes the "type" to "wall".

/define/boundary-conditions/modify-zones/fuse-face-zones cu-plate-bottom-
front cu-plate-bottom-front.1 () cu-plate-bottom-front
/define/boundary-conditions/modify-zones/zone-type cu-plate-bottom-front wall

/define/boundary-conditions/modify-zones/fuse-face-zones cu-plate-bottom-
rear cu-plate-bottom-rear.1 () cu-plate-bottom-rear
/define/boundary-conditions/modify-zones/zone-type cu-plate-bottom-rear wall

/define/boundary-conditions/modify-zones/fuse-face-zones cu-plate-left-
cool cu-plate-left-cool.1 () cu-plate-left-cool
/define/boundary-conditions/modify-zones/zone-type cu-plate-left-cool wall

/define/boundary-conditions/modify-zones/fuse-face-zones cu-plate-right-
uncool cu-plate-right-uncool.1 () cu-plate-right-uncool
/define/boundary-conditions/modify-zones/zone-type cu-plate-right-uncool wall

/define/boundary-conditions/modify-zones/fuse-face-zones cu-plate-top-
cool cu-plate-top-cool.1 () cu-plate-top-cool
/define/boundary-conditions/modify-zones/zone-type cu-plate-top-cool wall

/define/boundary-conditions/modify-zones/fuse-face-zones cu-plate-top-
```

```
uncool cu-plate-top-uncool.1 () cu-plate-top-uncool
/define/boundary-conditions/modify-zones/zone-type cu-plate-top-uncool wall

/define/boundary-conditions/modify-zones/fuse-face-zones ss-plate-rear-
left ss-plate-rear-left.1 () ss-plate-rear-left
/define/boundary-conditions/modify-zones/zone-type ss-plate-rear-left wall

/define/boundary-conditions/modify-zones/fuse-face-zones ss-plate-forward-
right ss-plate-forward-right.1 () ss-plate-forward-right
/define/boundary-conditions/modify-zones/zone-type ss-plate-forward-right wall
```

Appendix F. User Defined Function: Heterogeneous Reaction Rates

```
#include "udf.h"
/*Reaction rates for heterogeneous reactions*/
/*Taken from Thermophsics Characterization of Kerosene Combustion by Ten-See Wang*/
/*Author of this script: Jacob Bills*/
/*November, 2015*/

/*Constants defined bellow. Set DEBUG to TRUE to print values to TUI.*/
/*DEBUG only recomended on coarse grids.*/

#define DEBUG FALSE
#define B 1.0
#define pexp 0.3
#define C12H24exp 0.5

/*Constants*/
#define mwC12H23 167.31 /*MW of C12H23 g/mol*/
#define mwO2 32.0 /*MW of O2 g/mol*/
#define prim_index 0 /*index of primary phase*/
#define sec_index 1 /*index of secondary phase*/

/************************************************************/
/* UDF for calculating the reaction rate for the Paraffin Global Step */
/************************************************************/
DEFINE_HET_RXN_RATE(paraffin_global_step, c, t, hr, mw, yi, rr, rr_t)
```

```
{
Thread **pt = THREAD_SUB_THREADS(t);
Thread *tp = pt[0];
Thread *ts = pt[1];
real K_f, c_O2, c_C12H23, pval, var1, var2, rate ;
real temp = C_T(c,t); /*calls current cell temperature [K]*/
real pres = C_P(c,t); /*calls current cell pressure [Pa]*/
real vol_check;
real A = 3.888E4;   /*Pre-exponential factor from Table 3*/
real E_R = 1.22E4;   /*E/R value from Table 3*/

if(Data_Valid_P())   /*Checks for valid data in cell before continuing*/
{
/*Checks volume of fluid value for fuel in cell*/
vol_check = C_VOF(c,ts);

/*Convert pressure to atm*/
pres = pres/101325.0;

/*p^0.3*/
pval = pow(pres, pexp);

/*-E/RT*/
var1 = -1.0*E_R/temp;

/*e^(-E/RT)*/
```

```
var2 = exp(var1);

/*Kf = A*T^B*exp(-E/RT)*/
K_f = A*temp*var2;

/*calculates concentration of O2 [gmol/cm^3]*/
c_O2 = (yi[0][0]*C_R(c,t)/mwO2)*(pow(10.0, 3)/pow(10.0, 6));

/*calculates concentration of C12H23 [gmol/cm^3]*/
c_C12H23 = yi[1][0]*C_R(c,t)/mwC12H23*(pow(10.0, 3)/pow(10.0, 6));

/*RXN rate in kmol/(m^3*s)*/
rate = (K_f*pval*sqrt(c_C12H23)*c_O2)*(pow(10.0, -3)/pow(10.0, -6)) ;

/*Checks that volume fraction greater than 0*/
if(vol_check > 0) {
*rr = rate;
}
else {
*rr = 0;
}

#if DEBUG
Message("temp=%f\n",temp);
Message("pres=%f\n",pres);
Message("p^0.3=%f\n",pval);
```

```
Message("var1=%f\n",var1);
Message("var2=%f\n",var2);
Message("k=%f\n",K_f);
Message("[O2]=%f\n",c_O2);
Message("[C12H23]=%f\n",c_C12H23);
Message("MFC12H23=%f\n",yi[1][0]);
Message("MFO2=%f\n",yi[0][0]);
Message("rho=%f\n",C_R(c,t));
Message("rr=%f\n",rate);
#endif
}
}

/************************************************************/
/* UDF for calculating the reaction rate for the Napthene Global Step */
/************************************************************/
DEFINE_HET_RXN_RATE(napthene_global_step, c, t, hr, mw, yi, rr, rr_t)
{

Thread **pt = THREAD_SUB_THREADS(t);
Thread *tp = pt[0];
Thread *ts = pt[1];
real K_f, c_O2, c_C12H23, pval, var1, var2, rate ;
real temp = C_T(c,t); /*calls current cell temperature [K]*/
real pres = C_P(c,t); /*calls current cell pressure [Pa]*/
real vol_check;
```

```
real A = 2.312E7;   /*Pre-exponential factor from Table 3*/
real E_R = 1.965E4;   /*E/R value from Table 3*/

if(Data_Valid_P())   /*Checks for valid data in cell before continuing*/
{
/*Checks volume of fluid value for fuel in cell*/
vol_check = C_VOF(c,ts);

/*Convert pressure to atm*/
pres = pres/101325.0;

/*p^0.3*/
pval = pow(pres, pexp);

/*-E/RT*/
var1 = -1.0*E_R/temp;

/*e^(-E/RT)*/
var2 = exp(var1);

/*Kf = A*T^B*exp(-E/RT)*/
K_f = A*temp*var2;

/*calculates concentration of O2 [gmol/cm^3]*/
c_O2 = (yi[0][0]*C_R(c,t)/mwO2)*(pow(10.0, 3)/pow(10.0, 6));
```

```
/*calculates concentration of C12H23 [gmol/cm^3]*/
c_C12H23 = yi[1][0]*C_R(c,t)/mwC12H23*(pow(10.0, 3)/pow(10.0, 6));

/*Rxn rate in kmol/(m^3*s)*/
rate = (K_f*pval*sqrt(c_C12H23)*c_O2)*(pow(10.0, -3)/pow(10.0, -6)) ;

/*Checks that volume fraction greater than 0*/
if(vol_check > 0) {
*rr = rate;
}
else {
*rr = 0;
}

#if DEBUG
Message("temp=%f\n",temp);
Message("pres=%f\n",pres);
Message("p^0.3=%f\n",pval);
Message("var1=%f\n",var1);
Message("var2=%f\n",var2);
Message("k=%f\n",K_f);
Message("[O2]=%f\n",c_O2);
Message("[C12H23]=%f\n",c_C12H23);
Message("MFC12H23=%f\n",yi[1][0]);
Message("MFO2=%f\n",yi[0][0]);
Message("rho=%f\n",C_R(c,t));
```

```
Message("rr=%f\n",rate);
#endif
}
}
```

Appendix G. MATLAB Solution Convergence Script

This MATLAB code demonstrates the method used to validate the solution convergence of the final test cases.

Contents


```
%iterconvg_FR.m
%Checking iterative convergence for the Final Runs (FR)
%Runs Grid-Convg-Run13, FR Runs: 8, 11, 13, 15b
%Author: Jacob Bills
%Jan. 2016
clc; clear; close all;
Fname = 'Iter_history.mat';
load(Fname);
```

User Inputs

```
%Select which Run to analyze: Run8, GCRun13, Run11, Run13, Run15
% 2.9 RXN = 4, 2.9 no RXN = 5, 1.6 RXN = 2, 1.6 no RXN = 3, no LFFC = 1
title_var = 6;
```

```
if title_var == 4
    Run = Run8;
    Run.history.cool.q = -Run.history.cool.q;
    titleT = 'Area-averaged surface temperature: LFFC surface, 2.9\% LFFC
        RXN';
    titleq = 'Area-averaged heat flux: LFFC surface, 2.9\% LFFC RXN';
    %Controling sample bar and text locations on plots
    line_locq = 0.15; %Moves sample lines up/down
    text_locq= 0.05; %Moves sample text up(-)/down(+)
    text_startq = 2.5;  %Moves sample text left/right
    ax_onq = 0;

    line_locT = 0.025;
    text_locT= 0.013;
    text_start_T = 2.5;
    ax_onT = 0;
elseif title_var == 5
    Run = GCRun13;
    Run.history.cool.q = -Run.history.cool.q;
    titleT = 'Area-averaged surface temperature: LFFC surface, 2.9\% LFFC
        No RXN';
    titleq = 'Area-averaged heat flux: LFFC surface, 2.9\% LFFC No RXN';
    %Controling sample bar and text locations on plots
    line_locq = 0.12; %Moves sample lines up/down
    text_locq= 0.05; %Moves sample text up/down
    text_startq = 2.5;  %Moves sample text left/right
    ax_onq = 0;
```

```
    line_locT = 0.01;
    text_locT= 0.005;
    text_start_T = 2.5;
    ax_onT = 1;
    Tmin = 795;
    Tmax = 860;
elseif title_var == 2
    Run = Run11;
    Run.history.cool.q = -Run.history.cool.q;
    titleT = 'Area-averaged surface temperature: LFFC surface, 1.6\% LFFC
        RXN';
    titleq = 'Area-averaged heat flux: LFFC surface, 1.6\% LFFC RXN';
    %Controling sample bar and text locations on plots
    line_locq = 0.04; %Moves sample lines up/down
    text_locq= 0.02; %Moves sample text up/down
    text_startq = 2.5;  %Moves sample text left/right
    ax_onq = 1;
    qmin = 0.93e7;
    qmax = 1.2e7;

    line_locT = 0.01;
    text_locT= 0.005;
    text_start_T = 2.5;
    ax_onT = 0;
    Tmin = 795;
    Tmax = 860;
elseif title_var == 3
```

```
    Run = Run13;
    Run.history.cool.q = -Run.history.cool.q;
    titleT = 'Area-averaged surface temperature: LFFC surface, 1.6\% LFFC
       No RXN';
    titleq = 'Area-averaged heat flux: LFFC surface, 1.6\% LFFC No RXN';
    %Controling sample bar and text locations on plots
    line_locq = 0.04; %Moves sample lines up/down
    text_locq= 0.02; %Moves sample text up/down
    text_startq = 2.5;  %Moves sample text left/right
    ax_onq = 1;
    qmin = 8e6;
    qmax = 11e6;

    line_locT = 0.013;
    text_locT= 0.0037;
    text_start_T = 2.5;
    ax_onT = 1;
    Tmin = 855;
    Tmax = 920;
elseif title_var == 1
    Run = Run15;
    titleT = 'Area-averaged surface temperature: LFFC surface, No LFFC';
    titleq = 'Area-averaged heat flux: LFFC surface, No LFFC';
    %Controling sample bar and text locations on plots
    line_locq = 0.05; %Moves sample lines up/down
    text_locq= 0.01; %Moves sample text up/down
    text_startq = 2.5;  %Moves sample text left/right
    ax_onq = 0;
```

```
    line_locT = 0.01;
    text_locT= 0.003;
    text_start_T = 2.5;
    ax_onT = 0;

elseif title_var == 6
    Run = Run17;
    Run.history.cool.q = -Run.history.cool.q;
    titleT = 'Area-averaged surface temperature: LFFC surface, 2.9\% LFFC
        No RXN, Adiabatic S.S. plate';
    titleq = 'Area-averaged heat flux: LFFC surface, 2.9\% LFFC No RXN,
        Adiabatic S.S. plate';
    %Controling sample bar and text locations on plots
    line_locq = 0.1; %Moves sample lines up/down
    text_locq= 0.05; %Moves sample text up/down
    text_startq = 2.5;  %Moves sample text left/right
    ax_onq = 0;

    line_locT = 0.01;
    text_locT= 0.004;
    text_start_T = 2.5;
    ax_onT = 1;
    Tmin = 890;
    Tmax = 1000;
end
%2.5 percentage error bars
err_bar = 0.025;
```

```
plotsize = [.1 .1 .4 .4];

%Sets heat flux convention of positive for heat flux into plate
Run.history.cool.q = -Run.history.cool.q;
```

Iterative convergence for heat flux on FFC plate

```
sample_range = 15000;

samp1_e = 100000;
samp1_s = samp1_e - sample_range;
samp2_e = (samp1_e-samp1_s)/2+samp1_s;
samp2_s = samp2_e - sample_range;
samp3_e = (samp2_e-samp2_s)/2+samp2_s;
samp3_s = samp3_e - sample_range;

std1 = std(Run.history.cool.q(samp1_s:samp1_e));
std2 = std(Run.history.cool.q(samp2_s:samp2_e));
std3 = std(Run.history.cool.q(samp3_s:samp3_e));

mean1 = mean(Run.history.cool.q(samp1_s:samp1_e));
mean2 = mean(Run.history.cool.q(samp2_s:samp2_e));
mean3 = mean(Run.history.cool.q(samp3_s:samp3_e));

%If std_means is less than mean_stds than the solution is itterativley
%converged
std_means = std([mean1, mean2, mean3]);
mean_stds = mean([std1, std2, std3]);
```

```
fprintf('For Heat Flux:\n')
fprintf('The standard dev. of the means is %1.3e.\n',std_means)
fprintf('The mean of the standard devs. is %1.3e.\n',mean_stds)
fprintf('\nIf %1.3e is < %1.3e then solution\n is iteratively converged for
     heat flux.\n'...
    ,std_means,mean_stds)
```

Find closest iter for solution

```
%q
meanq = mean(Run.history.cool.q(samp3_s:samp1_e));
iv = 1;
if title_var == 6
    step_size = 1000;
else
    step_size = 500;
end
for indx = samp3_s:step_size:100000
    qdiff(iv) = abs(abs(Run.history.cool.q(indx)-meanq)/meanq);
    iv = iv + 1;
end
qdiff_min = min(qdiff);
q_indx_min = find(qdiff==qdiff_min);
index_useq = step_size*(q_indx_min-1) + samp3_s;
if title_var == 1
    index_useq = 100000;
end
fprintf('\nThe iteration with the smallest difference in q''''\n')
fprintf('from the overall mean is iteration %5.0i.\n',index_useq)
```

```
iter_spot = index_useq;


%T
meanT = mean(Run.history.cool.T(samp3_s:samp1_e));
iv = 1;
for indx = samp3_s:step_size:100000
    Tdiff(iv) = abs(abs(Run.history.cool.T(indx)-meanT)/meanT);
    iv = iv + 1;
end


Tdiff_min = min(Tdiff);
T_indx_min = find(Tdiff==Tdiff_min);
index_useT = step_size*(T_indx_min-1) + samp3_s;
if title_var == 1
    index_useT = 100000;
end
```

Checking iterative convergence for the no film cooling case

```
if title_var == 1
   Diffq = zeros(1,length(samp3_s:samp1_e)-1);
   j = 1;
   for i = samp3_s:(samp1_e-1)
       Diffq(j) = abs(Run.history.cool.q(j)-Run.history.cool.q(j+1))/Run.
           history.cool.q(j);
       j = j + 1;
   end
   maxDiffq = max(Diffq((end-10000):end));
   fprintf('\nThe maximum difference in the last 10k iters\n')
```

```
   fprintf('for the non-LFFC case q'''' is %2.2i percent.\n\n',maxDiffq
       *100)

   DiffT = zeros(1,length(samp3_s:samp1_e)-1);
   j = 1;
   for i = samp3_s:(samp1_e-1)
       DiffT(j) = abs(Run.history.cool.T(j)-Run.history.cool.T(j+1))/Run.
           history.cool.T(j);
       j = j + 1;
   end
   maxDiffT = max(DiffT((end-10000):end));
   fprintf('\nThe maximum difference in the last 10k iters\n')
   fprintf('for the non-LFFC case T is %2.2i percent.\n\n',maxDiffT*100)
end
```

plot for selecting iteration to use

```
%q
figure('units','normalized','position',plotsize)
x = Run.history.cool.iter;
y = Run.history.cool.q;
ymax = max(Run.history.cool.q(samp3_s:end));
ymin = min(Run.history.cool.q(samp3_s:end));
y1 = mean3+0.01*mean3;
y2 = y1+0.01*mean3;
y3 = y2+0.01*mean3;
main_mean = mean(Run.history.cool.q(samp3_s:samp1_e));
er_bar_up = main_mean + err_bar*main_mean;
er_bar_down = main_mean - err_bar*main_mean;
```

```
if er_bar_up > ymax || er_bar_down < ymin
    ymax = er_bar_up + er_bar_up*0.001;
    ymin = er_bar_down - er_bar_down*0.001;
end

orng = [1 .5 0];  %For plotting with orange
ml = '--';
mlc = 'k';
mlw = 2;
el = '-.';
elc = 'r';
elw = 0.9;
msize = 6;

subplot(2,1,1);
q_legend = plot(x(samp3_s:samp1_e),y(samp3_s:samp1_e),'-b','LineWidth',0.9)
    ;
hold on


%mean
subplot(2,1,1);
mean_legend = line([samp3_s samp1_e],[meanq meanq],'Color',mlc,'LineWidth',
    mlw,'LineStyle',ml);

subplot(2,1,1);
error_legend = line([samp3_s samp1_e],[er_bar_up er_bar_up],'Color',elc,'
    LineWidth',elw,'LineStyle',el);
```

```
subplot(2,1,1);
line([samp3_s samp1_e],[er_bar_down er_bar_down],'Color',elc,'LineWidth',
    elw,'LineStyle',el)
subplot(2,1,1);
iterdot = plot(iter_spot,Run.history.cool.q(iter_spot),'o','MarkerFaceColor
    ',elc,'MarkerEdgeColor',elc,'MarkerSize',msize);
diff_iter = abs(abs(Run.history.cool.q(iter_spot)-meanq)/meanq);
fprintf('The selected iteration average heat flux\n')
fprintf('differs by %1.4f.\n',diff_iter);

title(titleq,'interpreter','latex')
xlabel('Iteration','interpreter','latex')
ylabel('$\overline{q\prime\prime}_s$ $(W/m^2)$','interpreter','latex')
axis([samp3_s samp1_e ymin ymax])
l = legend([q_legend,mean_legend, error_legend, iterdot],'$\overline{q\
    prime\prime}_s$','Mean','2.5\% Margin','Selected Iteration');
set(l,'interpreter','latex')
set(gcf, 'renderer', 'zbuffer');

%Iterative convergence for T on FFC plate
std1 = std(Run.history.cool.T(samp1_s:samp1_e));
std2 = std(Run.history.cool.T(samp2_s:samp2_e));
std3 = std(Run.history.cool.T(samp3_s:samp3_e));

mean1 = mean(Run.history.cool.T(samp1_s:samp1_e));
mean2 = mean(Run.history.cool.T(samp2_s:samp2_e));
mean3 = mean(Run.history.cool.T(samp3_s:samp3_e));
```

```
%If std_means is less than mean_stds than the solution is itterativley
%converged
std_means = std([mean1, mean2, mean3]);
mean_stds = mean([std1, std2, std3]);
fprintf('\n\nFor Surface Temperature:\n')
fprintf('The standard dev. of the means is %2.2f.\n',std_means)
fprintf('The mean of the standard devs. is %2.2f.\n',mean_stds)
fprintf('\nIf %2.2f is < %2.2f then solution\n is iteratively converged for
     surface temperature.\n'...
    ,std_means,mean_stds)

%plot
%figure
y1 = mean3-0.005*mean3;
y2 = y1-0.005*mean3;
y3 = y2-0.005*mean3;
x = Run.history.cool.iter;
y = Run.history.cool.T;
ymax = max(Run.history.cool.T(samp3_s:end));
ymin = min(Run.history.cool.T(samp3_s:end));
main_mean = mean(Run.history.cool.T(samp3_s:samp1_e));
er_bar_up = main_mean + err_bar*main_mean;
er_bar_down = main_mean - err_bar*main_mean;
if er_bar_up > ymax || er_bar_down < ymin
    ymax = er_bar_up + er_bar_up*0.001;
    ymin = er_bar_down - er_bar_down*0.001;
end
```

```
subplot(2,1,2);
T_legend = plot(x(samp3_s:samp1_e),y(samp3_s:samp1_e),'-b','LineWidth',0.9)
    ;
hold on

%mean
subplot(2,1,2);
mean_legend = line([samp3_s samp1_e],[meanT meanT],'Color',mlc,'LineWidth',
    mlw,'LineStyle',ml);
subplot(2,1,2);

subplot(2,1,2);
error_legend = line([samp3_s samp1_e],[er_bar_up er_bar_up],'Color',elc,'
    LineWidth',elw,'LineStyle',el);
subplot(2,1,2);
line([samp3_s samp1_e],[er_bar_down er_bar_down],'Color',elc,'LineWidth',
    elw,'LineStyle',el)
subplot(2,1,2);
iterdot = plot(iter_spot,Run.history.cool.T(iter_spot),'o','MarkerFaceColor
    ',elc,'MarkerEdgeColor',elc,'MarkerSize',msize);
diff_iter = abs(abs(Run.history.cool.T(iter_spot)-main_mean)/main_mean);
fprintf('The selected iteration average surfface temperature\n')
fprintf('differs by %1.4f.\n',diff_iter);

title(titleT,'interpreter','latex')
xlabel('Iteration','interpreter','latex')
ylabel('$\overline{T}_{s}$ (K)','interpreter','latex')
axis([samp3_s samp1_e ymin ymax])
```

```
l = legend([T_legend,mean_legend,error_legend,iterdot],'$\overline{T}_{s}$
    ','Mean','2.5\% Margin','Selected Iteration');
set(l,'interpreter','latex')
set(gcf, 'renderer', 'zbuffer');

fprintf('\nThe iteration with the smallest difference in T\n')
fprintf('from the overall mean is iteration %5.0i.\n',index_useT)
```

Plot showing sample sets and sample set means

```
%-------------------
%q
%-------------------
mean1 = mean(Run.history.cool.q(samp1_s:samp1_e));
mean2 = mean(Run.history.cool.q(samp2_s:samp2_e));
mean3 = mean(Run.history.cool.q(samp3_s:samp3_e));

x = Run.history.cool.iter;
y = Run.history.cool.q;
ymax = max(Run.history.cool.q(samp3_s:end));
ymin = min(Run.history.cool.q(samp3_s:end));
y1 = mean3-line_locq*mean3;
y2 = y1-line_locq*mean3;
y3 = y2-line_locq*mean3;
main_mean = mean(Run.history.cool.q(samp3_s:samp1_e));
er_bar_up = main_mean + err_bar*main_mean;
er_bar_down = main_mean - err_bar*main_mean;

figure('units','normalized','position',plotsize)
```

```
subplot(2,1,1);
q_legend = plot(x(samp3_s:samp1_e),y(samp3_s:samp1_e),'-b','LineWidth',0.9)
    ;
hold on

subplot(2,1,1);
samp_legend = line([samp1_s samp1_e],[y1 y1],'Color',orng,'LineWidth',0.9);
subplot(2,1,1);
plot(x(samp1_s),y1,'d','MarkerFaceColor',orng,'MarkerEdgeColor','none')
subplot(2,1,1);
plot(x(samp1_e),y1,'d','MarkerFaceColor',orng,'MarkerEdgeColor','none')

subplot(2,1,1);
line([samp2_s samp2_e],[y2 y2],'Color',orng,'LineWidth',0.9)
subplot(2,1,1);
plot(x(samp2_s),y2,'d','MarkerFaceColor',orng,'MarkerEdgeColor','none')
subplot(2,1,1);
plot(x(samp2_e),y2,'d','MarkerFaceColor',orng,'MarkerEdgeColor','none')

subplot(2,1,1);
line([samp3_s samp3_e],[y3 y3],'Color',orng,'LineWidth',0.9)
subplot(2,1,1);
plot(x(samp3_s),y3,'d','MarkerFaceColor',orng,'MarkerEdgeColor','none')
subplot(2,1,1);
plot(x(samp3_e),y3,'d','MarkerFaceColor',orng,'MarkerEdgeColor','none')

%mean
subplot(2,1,1);
```

```
line([samp1_s samp1_e],[mean1 mean1],'Color',mlc,'LineWidth',mlw,'LineStyle
    ',ml)
subplot(2,1,1);
plot(x(samp1_s),mean1,'d','MarkerFaceColor',mlc,'MarkerEdgeColor','none')
subplot(2,1,1);
plot(x(samp1_e),mean1,'d','MarkerFaceColor',mlc,'MarkerEdgeColor','none')

subplot(2,1,1);
line([samp2_s samp2_e],[mean2 mean2],'Color',mlc,'LineWidth',mlw,'LineStyle
    ',ml)
subplot(2,1,1);
plot(x(samp2_s),mean2,'d','MarkerFaceColor',mlc,'MarkerEdgeColor','none')
subplot(2,1,1);
plot(x(samp2_e),mean2,'d','MarkerFaceColor',mlc,'MarkerEdgeColor','none')

subplot(2,1,1);
line([samp3_s samp3_e],[mean3 mean3],'Color',mlc,'LineWidth',mlw,'LineStyle
    ',ml)
subplot(2,1,1);
plot(x(samp3_s),mean3,'d','MarkerFaceColor',mlc,'MarkerEdgeColor','none')
subplot(2,1,1);
plot(x(samp3_e),mean3,'d','MarkerFaceColor',mlc,'MarkerEdgeColor','none')


subplot(2,1,1);
text(samp1_s+(samp1_e-samp1_s)/text_startq,y1-text_locq*mean3,'Sample 1','
    interpreter','latex')
subplot(2,1,1);
```

```
text(samp2_s+(samp2_e-samp2_s)/text_startq,y2-text_locq*mean3,'Sample 2','
    interpreter','latex')
subplot(2,1,1);
text(samp3_s+(samp3_e-samp3_s)/text_startq,y3-text_locq*mean3,'Sample 3','
    interpreter','latex')



title(titleq,'interpreter','latex')
xlabel('Iteration','interpreter','latex')
ylabel('$\overline{q\prime\prime}_s$ $(W/m^2)$','interpreter','latex')
if ax_onq == 1;
    axis([samp3_s samp1_e qmin qmax])
end
l = legend([q_legend,samp_legend,mean_legend],'$\overline{q\prime\prime}_s$
    ','Sample Range','Mean');
set(l,'interpreter','latex')
set(gcf, 'renderer', 'zbuffer');

%-------------------
%T
%-------------------
mean1 = mean(Run.history.cool.T(samp1_s:samp1_e));
mean2 = mean(Run.history.cool.T(samp2_s:samp2_e));
mean3 = mean(Run.history.cool.T(samp3_s:samp3_e));

y1 = mean3-line_locT*mean3;
y2 = y1-line_locT*mean3;
y3 = y2-line_locT*mean3;
```

```
x = Run.history.cool.iter;
y = Run.history.cool.T;
ymax = max(Run.history.cool.T(samp3_s:end));
ymin = min(Run.history.cool.T(samp3_s:end));
main_mean = mean(Run.history.cool.T(samp3_s:samp1_e));
er_bar_up = main_mean + err_bar*main_mean;
er_bar_down = main_mean - err_bar*main_mean;
if er_bar_up > ymax || er_bar_down < ymin
    ymax = er_bar_up + er_bar_up*0.001;
    ymin = er_bar_down - er_bar_down*0.001;
end

subplot(2,1,2);
T_legend = plot(x(samp3_s:samp1_e),y(samp3_s:samp1_e),'-b','LineWidth',0.9)
    ;
hold on

subplot(2,1,2);
samp_legend = line([samp1_s samp1_e],[y1 y1],'Color',orng,'LineWidth',0.9);
subplot(2,1,2);
plot(x(samp1_s),y1,'d','MarkerFaceColor',orng,'MarkerEdgeColor','none')
subplot(2,1,2);
plot(x(samp1_e),y1,'d','MarkerFaceColor',orng,'MarkerEdgeColor','none')

subplot(2,1,2);
line([samp2_s samp2_e],[y2 y2],'Color',orng,'LineWidth',0.9)
subplot(2,1,2);
plot(x(samp2_s),y2,'d','MarkerFaceColor',orng,'MarkerEdgeColor','none')
```

```
subplot(2,1,2);
plot(x(samp2_e),y2,'d','MarkerFaceColor',orng,'MarkerEdgeColor','none')

subplot(2,1,2);
line([samp3_s samp3_e],[y3 y3],'Color',orng,'LineWidth',0.9)
subplot(2,1,2);
plot(x(samp3_s),y3,'d','MarkerFaceColor',orng,'MarkerEdgeColor','none')
subplot(2,1,2);
plot(x(samp3_e),y3,'d','MarkerFaceColor',orng,'MarkerEdgeColor','none')
%
%mean
subplot(2,1,2);
line([samp1_s samp1_e],[mean1 mean1],'Color',mlc,'LineWidth',mlw,'LineStyle
    ',ml)
subplot(2,1,2);
plot(x(samp1_s),mean1,'d','MarkerFaceColor',mlc,'MarkerEdgeColor','none')
subplot(2,1,2);
plot(x(samp1_e),mean1,'d','MarkerFaceColor',mlc,'MarkerEdgeColor','none')

subplot(2,1,2);
line([samp2_s samp2_e],[mean2 mean2],'Color',mlc,'LineWidth',mlw,'LineStyle
    ',ml)
subplot(2,1,2);
plot(x(samp2_s),mean2,'d','MarkerFaceColor',mlc,'MarkerEdgeColor','none')
subplot(2,1,2);
plot(x(samp2_e),mean2,'d','MarkerFaceColor',mlc,'MarkerEdgeColor','none')

subplot(2,1,2);
```

```
line([samp3_s samp3_e],[mean3 mean3],'Color',mlc,'LineWidth',mlw,'LineStyle
    ',ml)
subplot(2,1,2);
plot(x(samp3_s),mean3,'d','MarkerFaceColor',mlc,'MarkerEdgeColor','none')
subplot(2,1,2);
plot(x(samp3_e),mean3,'d','MarkerFaceColor',mlc,'MarkerEdgeColor','none')
%--------------------

subplot(2,1,2);
text(samp1_s+(samp1_e-samp1_s)/text_start_T,y1-text_locT*mean3,'Sample 1','
    interpreter','latex')
subplot(2,1,2);
text(samp2_s+(samp2_e-samp2_s)/text_start_T,y2-text_locT*mean3,'Sample 2','
    interpreter','latex')
subplot(2,1,2);
text(samp3_s+(samp3_e-samp3_s)/text_start_T,y3-text_locT*mean3,'Sample 3','
    interpreter','latex')

title(titleT,'interpreter','latex')
xlabel('Iteration','interpreter','latex')
ylabel('$\overline{T}_{s}$ (K)','interpreter','latex')
if ax_onT == 1;
    axis([samp3_s samp1_e Tmin Tmax])
end
l = legend([T_legend,samp_legend,mean_legend],'$\overline{T}_{s}$','Sample
    Range','Mean');
set(l,'interpreter','latex')
set(gcf, 'renderer', 'zbuffer');
```

Check iterative convg. using variance and mean

```
recompute = 0;
if recompute == 1
    [ rho_s8,s,History8 ] = Autocorr( samp3_s,samp1_e,Run8 );
    [ rho_sGC13,s,HistoryGC13 ] = Autocorr( samp3_s,samp1_e,GCRun13 );
    [ rho_s11,s,History11 ] = Autocorr( samp3_s,samp1_e,Run11 );
    [ rho_s13,s,History13 ] = Autocorr( samp3_s,samp1_e,Run13 );
    [ rho_s15,s,History15 ] = Autocorr( samp3_s,samp1_e,Run15 );
    [ rho_s17,s,History17 ] = Autocorr( samp3_s,samp1_e,Run17 );
    save('Autocorr.mat', 'rho_s8','rho_sGC13','rho_s11','rho_s13','rho_s15
        ','rho_s17','s'...
        ,'History8','HistoryGC13','History11','History13','History15','
            History17');
else
    load('Autocorr.mat')
end
```

Plot Means and Varriance

```
sz = 2.3; %Line width for rxn
lw = 0.9;  %Line width for no rxn
orng = [1 .5 0];  %For plotting with orange

crs = '-.r';
med = '--b';
fine = '-g';
efine = ':k';
```

```
ffc29rxn = ':b'; %FR-Run-8

ffc29 = '-b'; %Grid-Convg-13

ffc16rxn = ':g'; %FR-Run-11

ffc16 = '-g'; %FR-Run-13

ffc00 = '-r'; %FR-Run-15

rxn = ':k';   %For showing rxns in legend

norxn = '-k'; %For showing non-rxns in legend


ffc29sym = 'ob';

ffc16sym = '*g';

ffc00sym = '<r';

adbsym = 's';


skp = 2500;

IS = 2;

IE = 1;


figure('units','normalized','position',plotsize)

subplot(2,1,1)

plot(Run8.history.cool.iter,History8.mean.q,ffc29rxn,'LineWidth',sz);

hold on

f1=plot(Run8.history.cool.iter(1:skp:end),History8.mean.q(1:skp:end),
   ffc29sym);

plot(GCRun13.history.cool.iter,HistoryGC13.mean.q,ffc29,'LineWidth',lw)

plot(GCRun13.history.cool.iter(1:skp:end),HistoryGC13.mean.q(1:skp:end),
   ffc29sym)

plot(Run11.history.cool.iter,History11.mean.q,ffc16rxn,'LineWidth',sz);
```

```
f2=plot(Run11.history.cool.iter(1:skp:end),History11.mean.q(1:skp:end),
    ffc16sym);
plot(Run13.history.cool.iter,History13.mean.q,ffc16,'LineWidth',lw)
plot(Run13.history.cool.iter(1:skp:end),History13.mean.q(1:skp:end),
    ffc16sym)
plot(Run15.history.cool.iter,History15.mean.q,ffc00,'LineWidth',lw)
f3=plot(Run15.history.cool.iter(1:skp:end),History15.mean.q(1:skp:end),
    ffc00sym);
f4=plot(Run17.history.cool.iter(1:skp:end),History17.mean.q(1:skp:end),
    adbsym,'Color',orng);
plot(Run17.history.cool.iter,History17.mean.q,'Color',orng,'LineWidth',lw);
rxnleg = plot(Run8.history.cool.iter(1),History8.mean.q(1),rxn,'LineWidth',
    sz);
norxnleg = plot(GCRun13.history.cool.iter(1),HistoryGC13.mean.q(1),norxn,'
    LineWidth',lw);
title('Moving average of area-averaged heat flux','interpreter','latex')
xlabel('Iteration','interpreter','latex')
ylabel('$\overline{q\prime\prime}_s$ $(W/m^2)$','interpreter','latex')
grid on
l = legend([f1 f2 f3 f4 rxnleg norxnleg],'2.9\% LFFC','1.6\% LFFC','No LFFC
    ','Adiabatic','RXN','No RXN');
set(l,'interpreter','latex')
set(gcf, 'renderer', 'zbuffer');

subplot(2,1,2)
plot(Run8.history.cool.iter,History8.mean.T,ffc29rxn,'LineWidth',sz);
hold on
```

```
f1=plot(Run8.history.cool.iter(1:skp:end),History8.mean.T(1:skp:end),
    ffc29sym);
plot(GCRun13.history.cool.iter,HistoryGC13.mean.T,ffc29,'LineWidth',lw)
plot(GCRun13.history.cool.iter(1:skp:end),HistoryGC13.mean.T(1:skp:end),
    ffc29sym)
plot(Run11.history.cool.iter,History11.mean.T,ffc16rxn,'LineWidth',sz);
f2=plot(Run11.history.cool.iter(1:skp:end),History11.mean.T(1:skp:end),
    ffc16sym);
plot(Run13.history.cool.iter,History13.mean.T,ffc16,'LineWidth',lw)
plot(Run13.history.cool.iter(1:skp:end),History13.mean.T(1:skp:end),
    ffc16sym)
plot(Run15.history.cool.iter,History15.mean.T,ffc00,'LineWidth',lw)
f3=plot(Run15.history.cool.iter(1:skp:end),History15.mean.T(1:skp:end),
    ffc00sym);
f4=plot(Run17.history.cool.iter(1:skp:end),History17.mean.T(1:skp:end),
    adbsym,'Color',orng);
plot(Run17.history.cool.iter,History17.mean.T,'Color',orng,'LineWidth',lw);
rxnleg = plot(Run8.history.cool.iter(1),History8.mean.T(1),rxn,'LineWidth',
    sz);
norxnleg = plot(GCRun13.history.cool.iter(1),HistoryGC13.mean.T(1),norxn,'
    LineWidth',lw);
title('Moving average of area-averaged surface temperature','interpreter','
    latex')
xlabel('Iteration','interpreter','latex')
ylabel('$\overline{T}_{s}$ (K)','interpreter','latex')
grid on
l = legend([f1 f2 f3 f4 rxnleg norxnleg],'2.9\% LFFC','1.6\% LFFC','No LFFC
    ','Adiabatic S.S.','RXN','No RXN');
```

```
set(l,'interpreter','latex')
set(gcf, 'renderer', 'zbuffer');
```

Appendix H. MATLAB Grid Convergence Script

This MATLAB code demonstrates the method used to evaluate grid convergence for the fluid domain. The solid domain code is not included as it used duplicate calculations just with different data files and grid values.

Contents

Grid Convg Check

Checks the grid convergence/independence for final fluid domain using the non-reacting freestream settings (Grid-Convg-Runs 9-11 extended and Run 13) Jacob Bills

```
clc; clear; close all; format compact;
```

User Inputs

```
%Set to 1 to re-read in the data and save a new .mat file.
%Set to 0 to load previous .mat file.
read_data = 0;

%Range over which to check for convergence
range = (80000:1:90000);

%Plot size
plotsize = [.1 .1 .4 .4];

%For plotting purposes:
labelyT = '$\overline{T}_{s}$ (K)';
labelyq = '$\overline{q\prime\prime}_s$ $(W/m^2)$';
legT = '$\overline{T}_{s}$';
legq = '$\overline{q\prime\prime}_s$';
```

Read Data

```
if read_data == 1
    %Run9
    Fname = 'Run9\Extended\q-cooled-plate-1.out';
    M = csvread(Fname,2);
    Run9.history.cool.iter = M(:,1); %Iteration number
    Run9.history.cool.q = M(:,2);    %Area averaged temp [K]
    Fname = 'Run9\Extended\q-uncooled-plate-1.out';
    M = csvread(Fname,2);
    Run9.history.uncool.iter = M(:,1); %Iteration number
    Run9.history.uncool.q = M(:,2);    %Area averaged temp [K]
```

```
Fname = 'Run9\Extended\surf-temp-cooled-plate-1.out';
M = csvread(Fname,2);
Run9.history.cool.T = M(:,2);    %Area averaged temp [K]
Fname = 'Run9\Extended\surf-temp-uncool-plate-1.out';
M = csvread(Fname,2);
Run9.history.uncool.T = M(:,2);    %Area averaged temp [K]



%Run10
Fname = 'Run10\Extended\q-cooled-plate-1.out';
M = csvread(Fname,2);
Run10.history.cool.iter = M(:,1); %Iteration number
Run10.history.cool.q = M(:,2);    %Area averaged temp [K]
Fname = 'Run10\Extended\q-uncooled-plate-1.out';
M = csvread(Fname,2);
Run10.history.uncool.iter = M(:,1); %Iteration number
Run10.history.uncool.q = M(:,2);    %Area averaged temp [K]

Fname = 'Run10\Extended\surf-temp-cooled-plate-1.out';
M = csvread(Fname,2);
Run10.history.cool.T = M(:,2);    %Area averaged temp [K]
Fname = 'Run10\Extended\surf-temp-uncool-plate-1.out';
M = csvread(Fname,2);
Run10.history.uncool.T = M(:,2);    %Area averaged temp [K]
Run10.history.uncool.T = Run10.history.uncool.T(1:end-1);

%Run11
Fname = 'Run11\Extended\q-cooled-plate-1.out';
```

```
M = csvread(Fname,2);
Run11.history.cool.iter = M(:,1); %Iteration number
Run11.history.cool.q = M(:,2);   %Area averaged temp [K]
Fname = 'Run11\Extended\q-uncooled-plate-1.out';
M = csvread(Fname,2);
Run11.history.uncool.iter = M(:,1); %Iteration number
Run11.history.uncool.q = M(:,2);   %Area averaged temp [K]

Fname = 'Run11\Extended\surf-temp-cooled-plate-1.out';
M = csvread(Fname,2);
Run11.history.cool.T = M(:,2);   %Area averaged temp [K]
Fname = 'Run11\Extended\surf-temp-uncool-plate-1.out';
M = csvread(Fname,2);
Run11.history.uncool.T = M(:,2);   %Area averaged temp [K]

%Run13
Fname = 'Run13\q-cooled-plate-1.out';
M = csvread(Fname,2);
Run13.history.cool.iter = M(:,1); %Iteration number
Run13.history.cool.q = M(:,2);   %Area averaged temp [K]
Fname = 'Run13\q-uncooled-plate-1.out';
M = csvread(Fname,2);
Run13.history.uncool.iter = M(:,1); %Iteration number
Run13.history.uncool.q = M(:,2);   %Area averaged temp [K]
ind = (69000:1:69128);             %Removing duplicate entries
Run13.history.uncool.iter(ind) = [];
Run13.history.uncool.q(ind) = [];
```

```
    Fname = 'Run13\surf-temp-cooled-plate-1.out';
    M = csvread(Fname,2);
    Run13.history.cool.T = M(:,2);   %Area averaged temp [K]
    Run13.history.cool.T(ind) = [];
    Fname = 'Run13\surf-temp-uncool-plate-1.out';
    M = csvread(Fname,2);
    Run13.history.uncool.T = M(:,2);   %Area averaged temp [K]
    Run13.history.uncool.T(ind) = [];

    save('gridconvg_fluid.mat')

else
    load('gridconvg_fluid.mat')

end
```

Calcs

```
%Grid Cells
C3 = 139443;  %Coarse
C2 = 520130;  %Medium
C1 = 2033174; %Fine
C0 = 4548562; %Extra Fine

%Refinement Selection: Select which three grids
%to use, note N3 must be the coarsest grid selected and N1 the finest.
N1 = C0;
N2 = C1;
N3 = C2;
```

```
%Refinement Ratios (From Roache 2008)
A = 0.0254*0.19902+(0.00826+0.00508)*0.00051+0.00318*0.01854...
    +0.00508*0.01854+0.00826*0.13233-27*0.00076*0.00318; %Area of grid
        domain [m^2]
h1 = (A/N1)^(1/2);
h2 = (A/N2)^(1/2);
h3 = (A/N3)^(1/2);
r32 = h3/h2;
r21 = h2/h1;
```

F values (Figure of merit)

```
%q FFC plate
F1 = abs(mean(Run11.history.cool.q(range(1):range(end)))); %Fine
F2 = abs(mean(Run10.history.cool.q(range(1):range(end)))); %Medium
F3 = abs(mean(Run9.history.cool.q(range(1):range(end)))); %Coarse
F0 = abs(mean(Run13.history.cool.q(range(1):range(end)))); %Extra Fine
%T_surf FFC plate
F1_T = abs(mean(Run11.history.cool.T(range(1):range(end)))); %Fine
F2_T = abs(mean(Run10.history.cool.T(range(1):range(end)))); %Medium
F3_T = abs(mean(Run9.history.cool.T(range(1):range(end)))); %Coarse
F0_T = abs(mean(Run13.history.cool.T(range(1):range(end)))); %Extra Fine
```

Calc relative errors

```
%q FFC plate (2008 Roache eqn 5)
e32 = abs((F2-F3)/F2);
e21 = abs((F1-F2)/F1);
e10 = abs((F0-F1)/F0);
```

```
fprintf('The q difference between the fine\n and extra fine grids is %1.3f
    .\n', e10);
fprintf('The q difference between the medium\n and fine grids is %1.3f.\n',
     e21);
fprintf('The q difference between the coarse\n and medium grids is %1.3f.\n
    ', e32);

%T_surf FFC plate (2008 Roache eqn 5)
e32_T = abs((F2_T-F3_T)/F2_T);
e21_T = abs((F1_T-F2_T)/F1_T);
e10_T = abs((F0_T-F1_T)/F0_T);

fprintf('\n\nThe T difference between the fine\n and extra fine grids is
    %1.4f.\n', e10_T);
fprintf('The T difference between the medium\n and fine grids is %1.4f.\n',
     e21_T);
fprintf('The T difference between the coarse\n and medium grids is %1.4f.\n
    \n', e32_T);
```

Calculate the order of accuracy for q-cool

```
%Order of Accuracy for q

eps23 = abs(F1 - F2);
eps12 = abs(F0 - F1);
syms p
%Roache Eqn 13
p = double(solve(eps23/(r32^p-1) == r21^p*(eps12/(r21^p-1)),p));
```

```
%Roache Eqns 5 and 9 combined, eqn 7 in 2008 paper
GCI12 = 1.25*e21/(r32^p-1);
GCI01 = 1.25*e10/(r21^p-1);
fprintf('GCI from medium to fine for q is %1.4e.\n',GCI12);
fprintf('GCI from fine to extra fine for q is %1.4e.\n\n',GCI01);

%Alternate method to calc p (eqns 3a etc in roache 2008). Give same as
%first method to calc p. Just here for sanity check.
syms p2
eps23_2 = F2 - F1;
eps12_2 = F1 - F0;
s = 1*sign(eps23_2/eps12_2);
q = log((r21^p2-s)/(r32^p2-s));
p2 = double(solve(p2 == (1/log(r21))*abs(log(abs(eps23_2/eps12_2))+q)));
Fext_2 = (r21^p2*F0-F1)/(r21^p2-1);
```

Order of accuracy for T-cool

```
%Order of Acc. for F_T

eps23_T = abs(F1_T - F2_T);
eps12_T = abs(F0_T - F1_T);
syms p_T
%Roache Eqn 13
p_T = double(solve(eps23_T/(r32^p_T-1) == r21^p_T*(eps12_T/(r21^p_T-1)),p_T
    ));
%Roache Eqns 5 and 9 combined, eqn 7 in 2008 paper
GCI12_T = 1.25*e21_T/(r32^p_T-1);
GCI01_T = 1.25*e10_T/(r21^p_T-1);
```

```
fprintf('GCI from medium to fine for T is %1.4e.\n',GCI12_T);
fprintf('GCI from fine to extra fine for T is %1.4e.\n\n',GCI01_T);

%Alternate method to calc p (eqns 3a etc in roache 2008). Give same as
%first method to calc p. Just here for sanity check.
syms p2_T
eps23_2_T = F2_T - F1_T;
eps12_2_T = F1_T - F0_T;
s_T = 1*sign(eps23_2_T/eps12_2_T);
q_T = log((r21^p2_T-s_T)/(r32^p2_T-s_T));
p2_T = double(solve(p2_T == (1/log(r21))*abs(log(abs(eps23_2_T/eps12_2_T))+
    q_T)));
Fext_2_T = (r21^p2_T*F0_T-F1_T)/(r21^p2_T-1);
```

Plot Info

```
iter1 = range(1);
iter2 = range(end);
sz = 2; %Line width for Run13
lw = 0.9;  %Line width for all other plots

crs = '-.r';
med = '--b';
fine = '-g';
efine = ':k';
```

Final Plots

```
figure('units','normalized','position',plotsize)
subplot(2,1,1)
```

```
c = plot(Run9.history.cool.iter(range),Run9.history.cool.q(range),crs,'
    LineWidth',lw);
hold on
subplot(2,1,1)
m = plot(Run10.history.cool.iter(range),Run10.history.cool.q(range),med,'
    LineWidth',lw);
subplot(2,1,1)
f = plot(Run11.history.cool.iter(range),Run11.history.cool.q(range),fine,'
    LineWidth',lw);
subplot(2,1,1)
ef = plot(Run13.history.cool.iter(range),Run13.history.cool.q(range),efine
    ,'LineWidth',sz);
xlabel('Iteration Number','interpreter','latex')
ylabel(labelyq,'interpreter','latex')
title('Area-averaged surface heat flux vs iteration: LFFC surface','
    interpreter','latex')
axis([iter1 iter2 7.8e6 12e6])
l = legend([c,m,f,ef],'Coarse','Medium','Fine','Extra Fine');
set(l,'interpreter','latex')
set(gcf, 'renderer', 'zbuffer');


subplot(2,1,2)
c = plot(Run9.history.cool.iter(range),Run9.history.cool.T(range),crs,'
    LineWidth',lw);
hold on
subplot(2,1,2)
m = plot(Run10.history.cool.iter(range),Run10.history.cool.T(range),med,'
    LineWidth',lw);
```

```
subplot(2,1,2)
f = plot(Run11.history.cool.iter(range),Run11.history.cool.T(range),fine,'
    LineWidth',lw);
subplot(2,1,2)
ef = plot(Run13.history.cool.iter(range),Run13.history.cool.T(range),efine
    ,'LineWidth',sz);
xlabel('Iteration Number','interpreter','latex')
ylabel(labelyT,'interpreter','latex')
title('Area-averaged surface temperature vs iteration: LFFC surface','
    interpreter','latex')
axis([iter1 iter2 800 950])
l = legend([c,m,f,ef],'Coarse','Medium','Fine','Extra Fine');
set(l,'interpreter','latex')
set(gcf, 'renderer', 'zbuffer');
```

Appendix I. Solution Convergence Plots: Fluid Grid Convergence Study

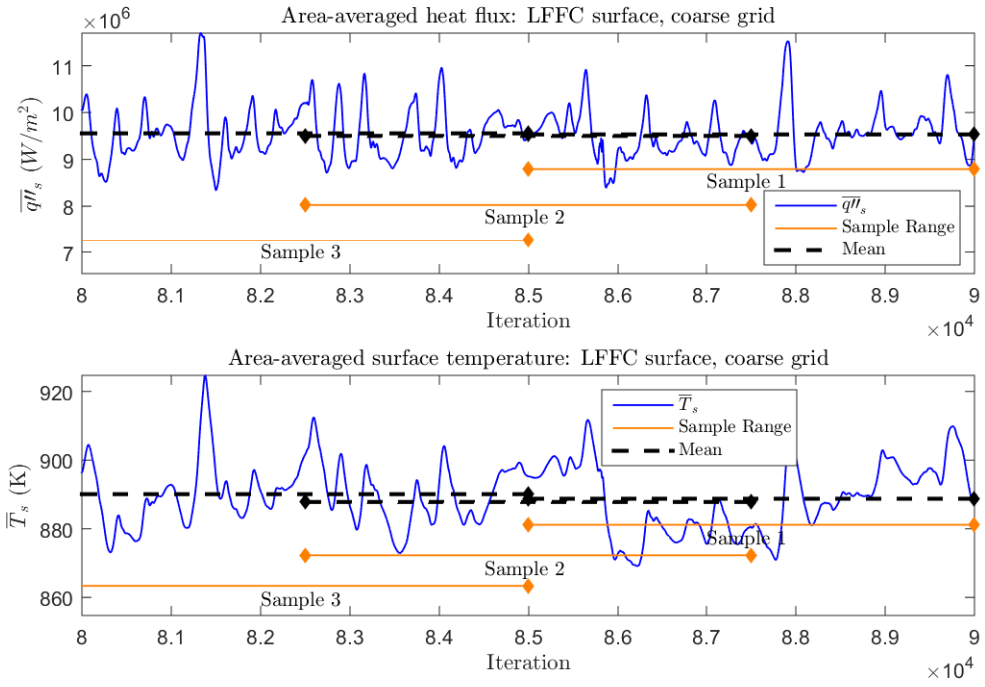

Figure 134. Solution convergence plot for coarse grid.

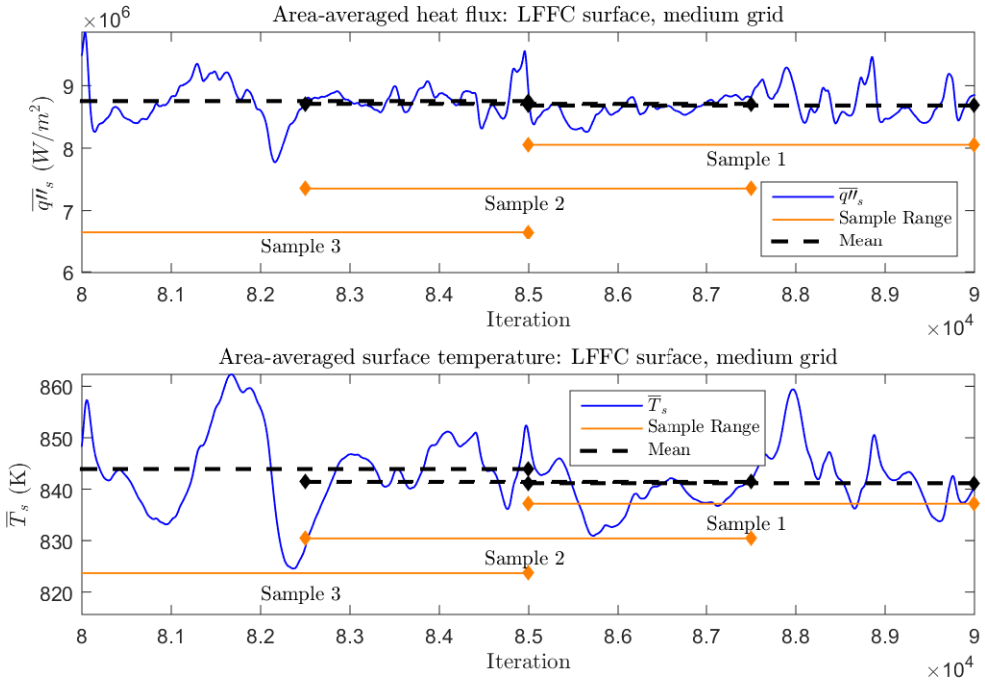

Figure 135. Solution convergence plot for medium grid.

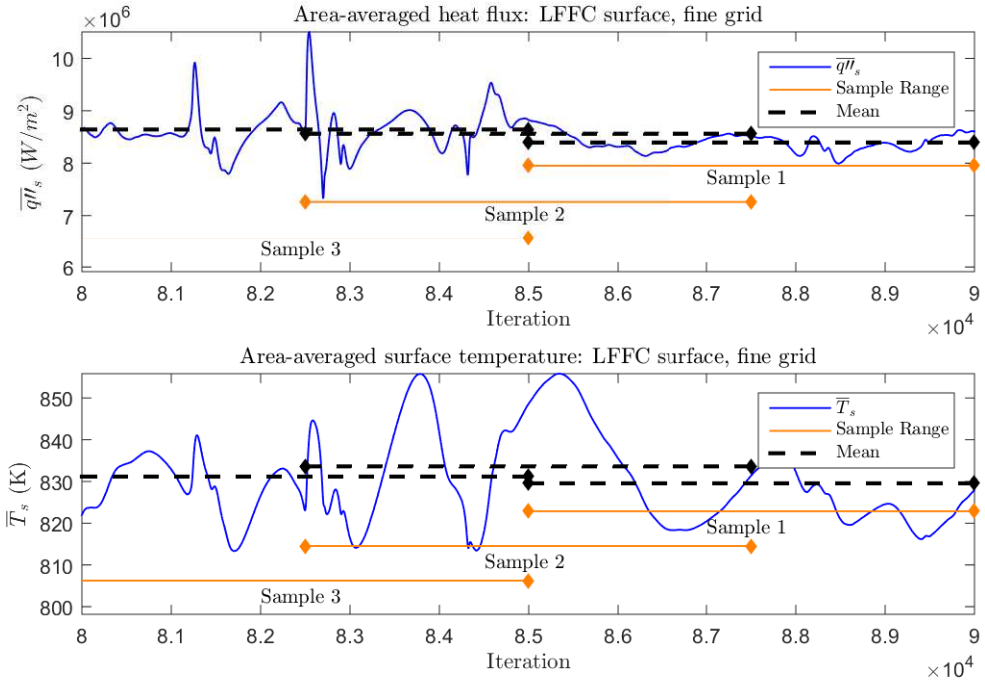

Figure 136. Solution convergence plot for fine grid.

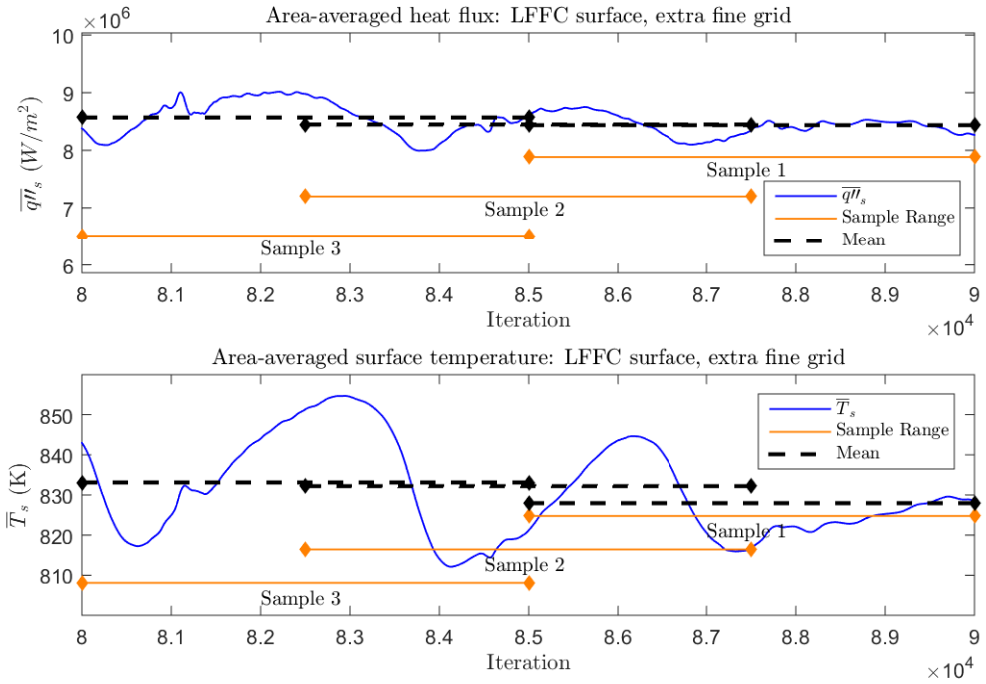

Figure 137. Solution convergence plot for extra fine grid.

Appendix J. Iteration History: Test Case 6

Iteration history for Test Case 6, No-RXN 2.9% LFFC with an adiabatic stainless steel plate.

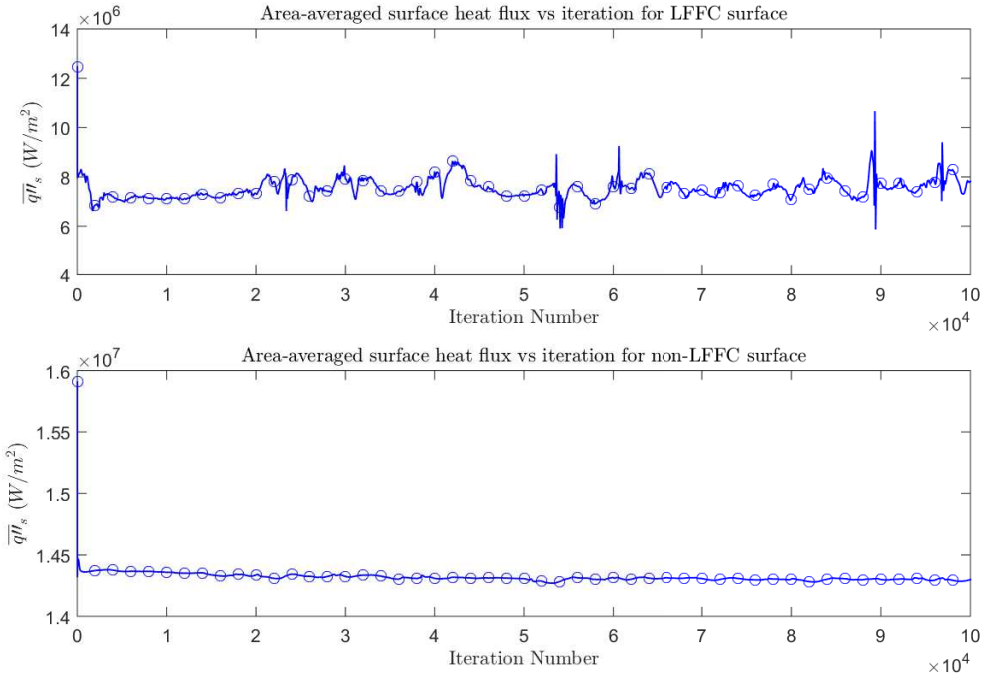

Figure 138. Solution history of $\overline{q''}_s$ for TC6.

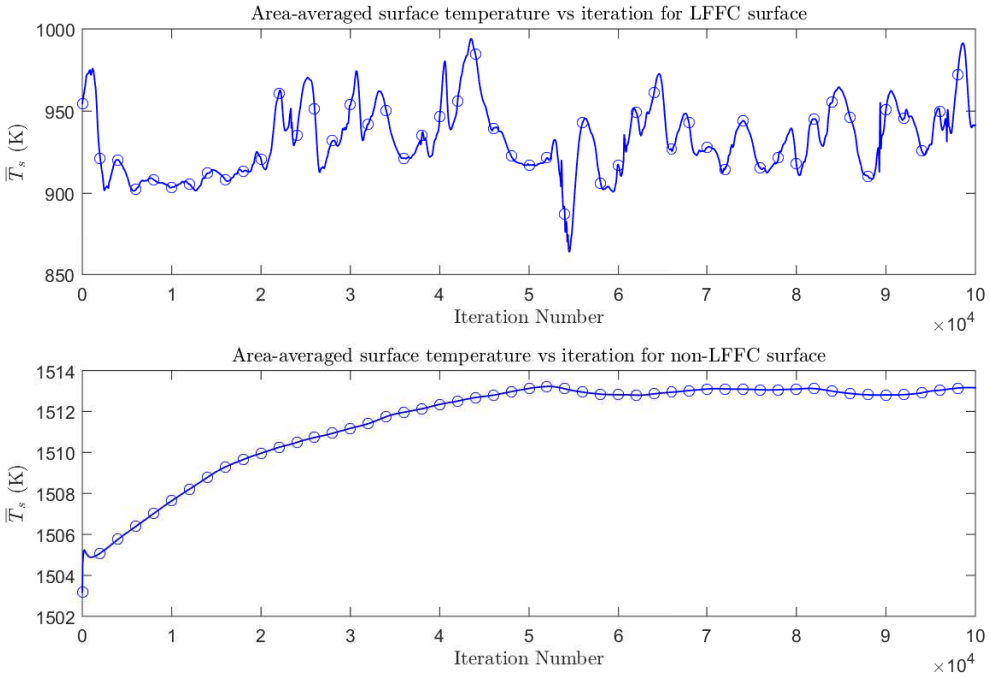

Figure 139. Solution history of $\overline{T}_s$ for TC6.

Appendix K. Solution Convergence Plots: Final Test Cases

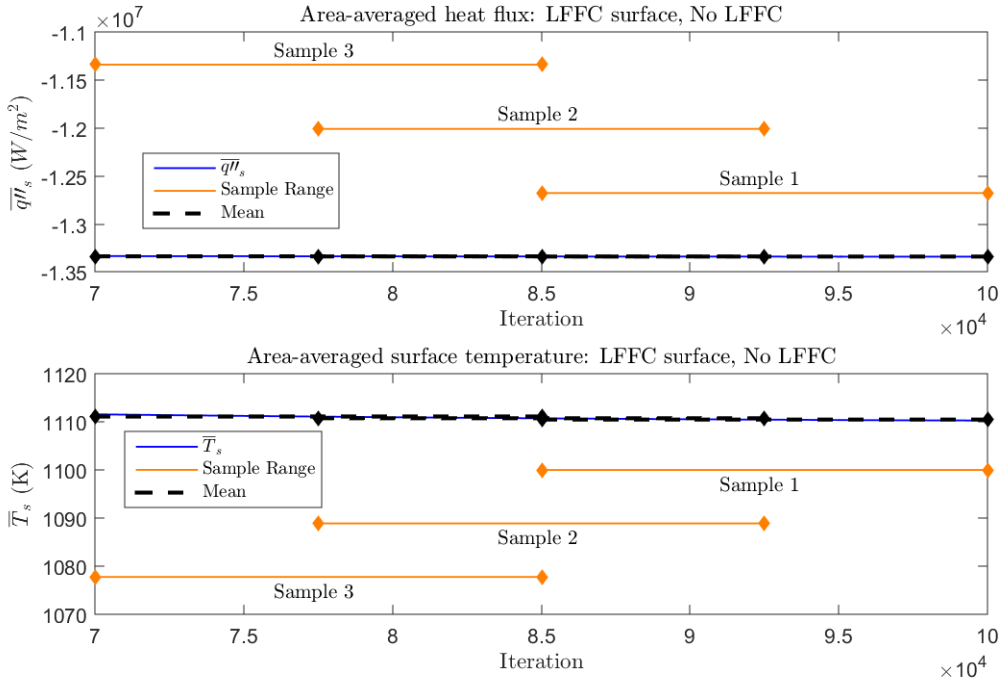

Figure 140. Solution convergence for Test Case 1.

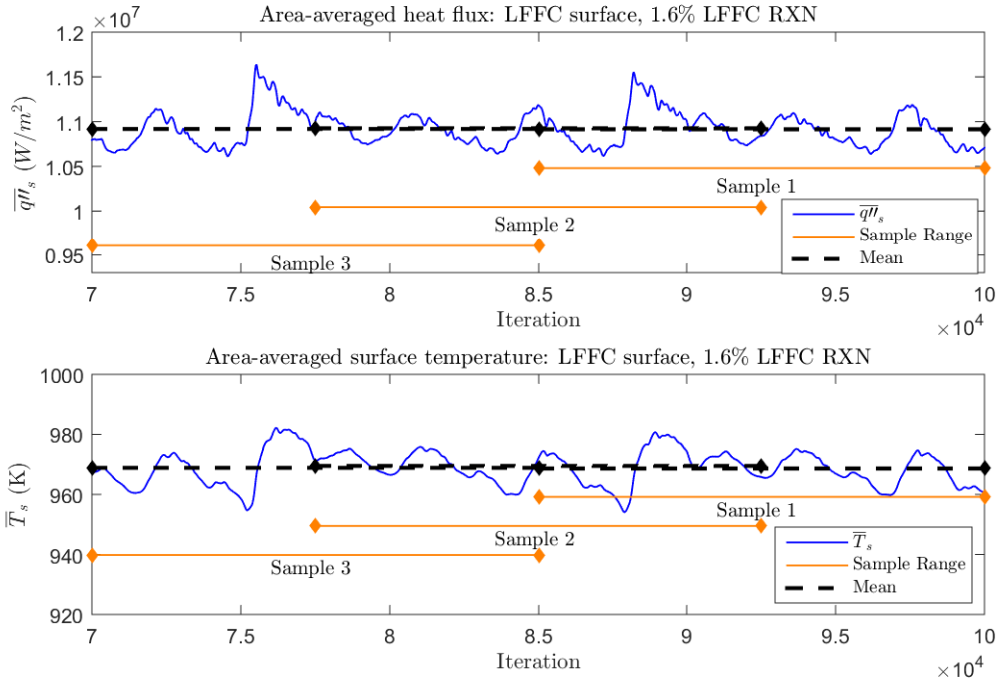

Figure 141. Solution convergence for Test Case 2.

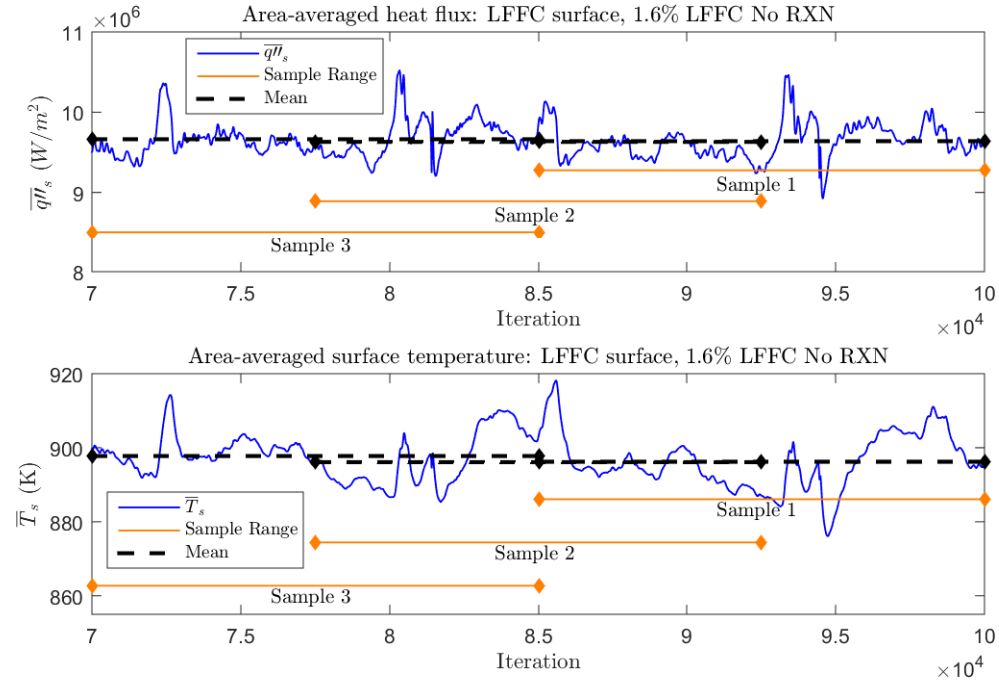

Figure 142. Solution convergence for Test Case 3.

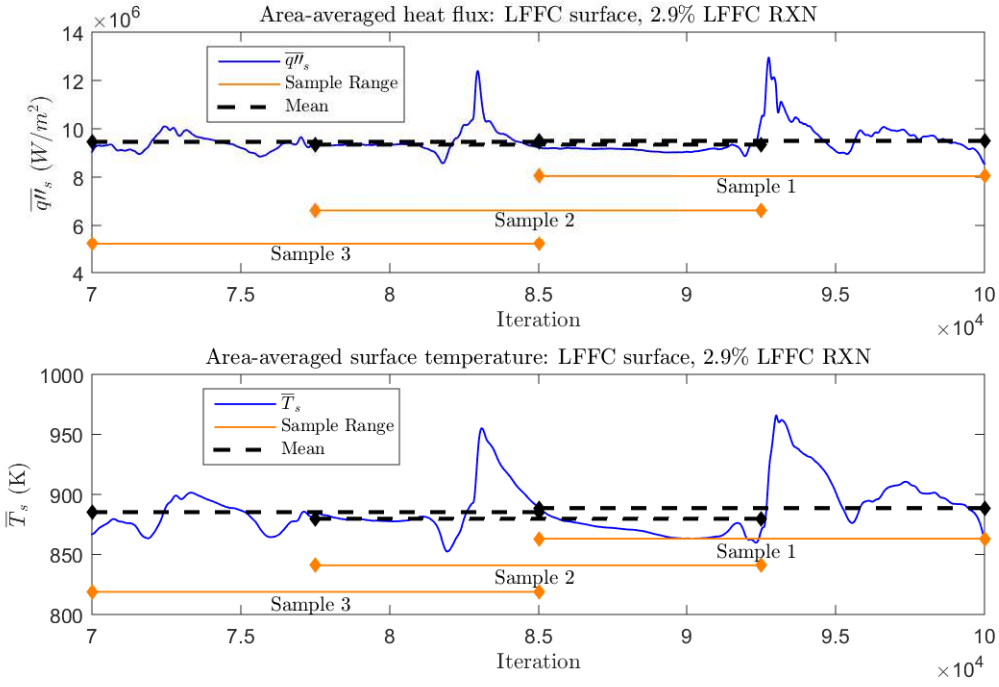

Figure 143. Solution convergence for Test Case 4.

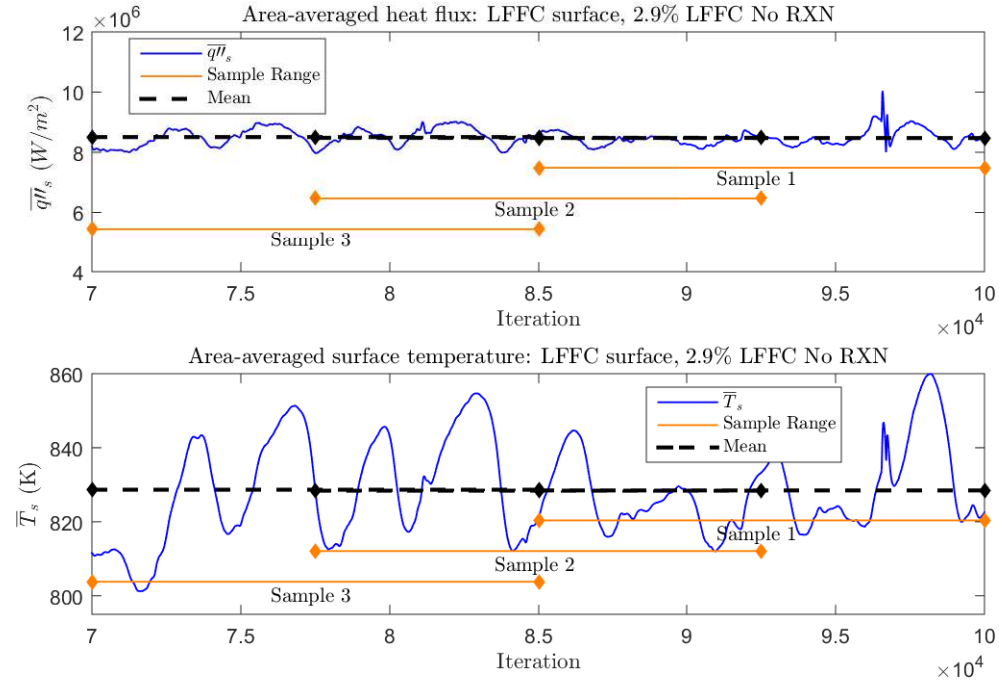

Figure 144. Solution convergence for Test Case 5.

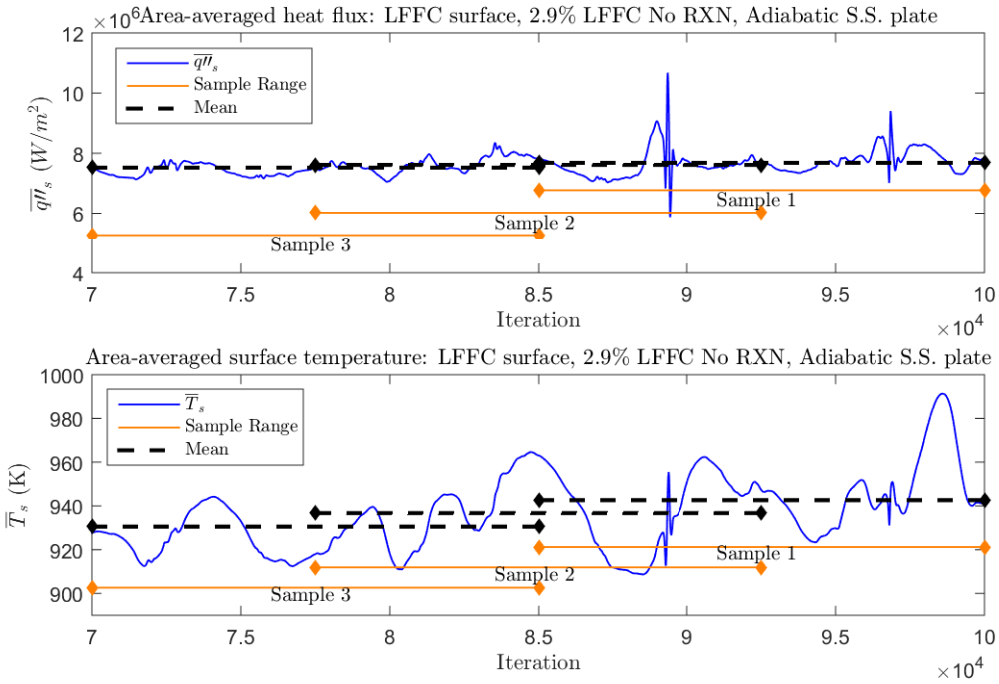

Figure 145. Solution convergence for Test Case 6.

Appendix L. Temperature Contours: Final Test Cases

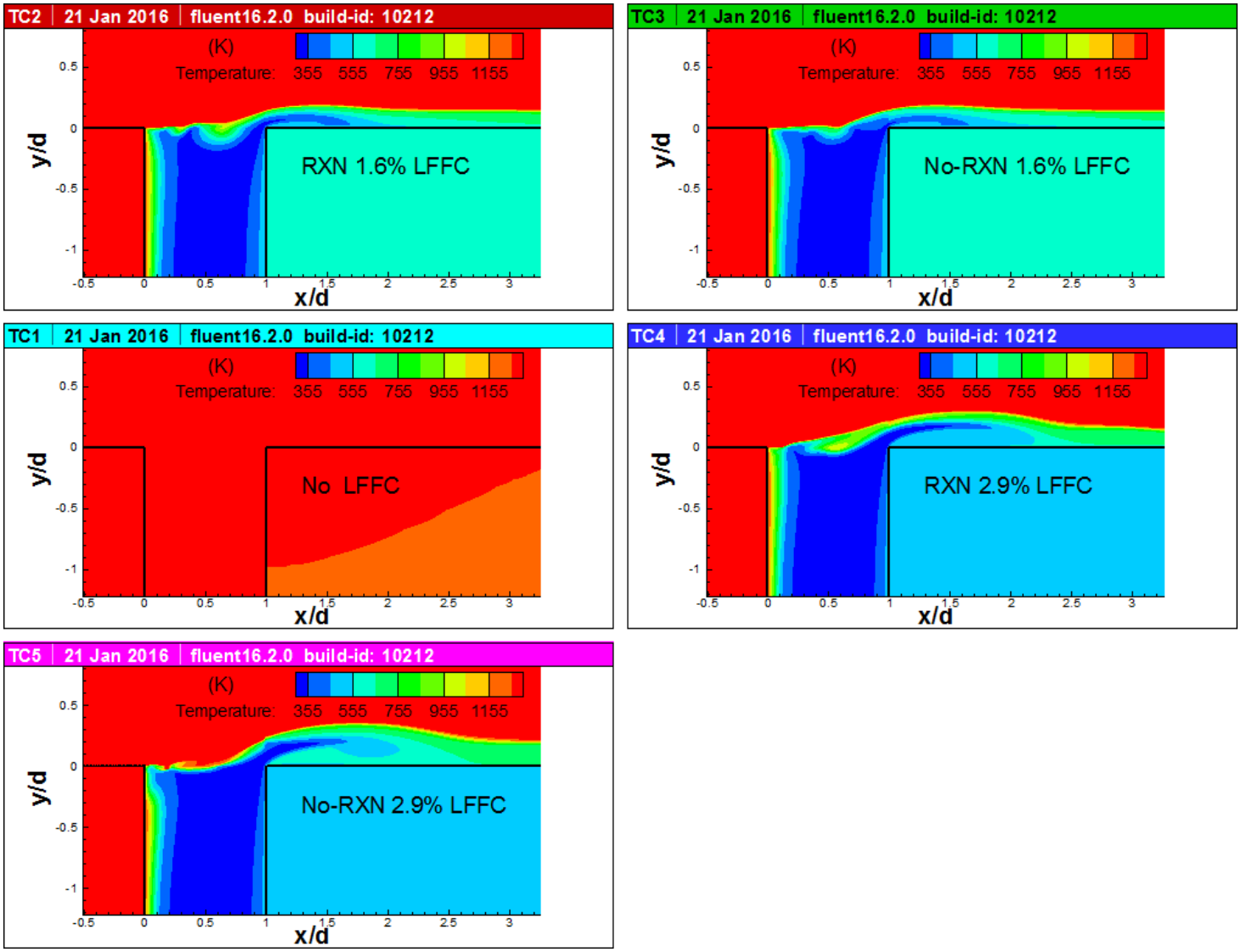

Figure 146. Temperature contours in the fuel injection slot region for Test Cases 1–5.

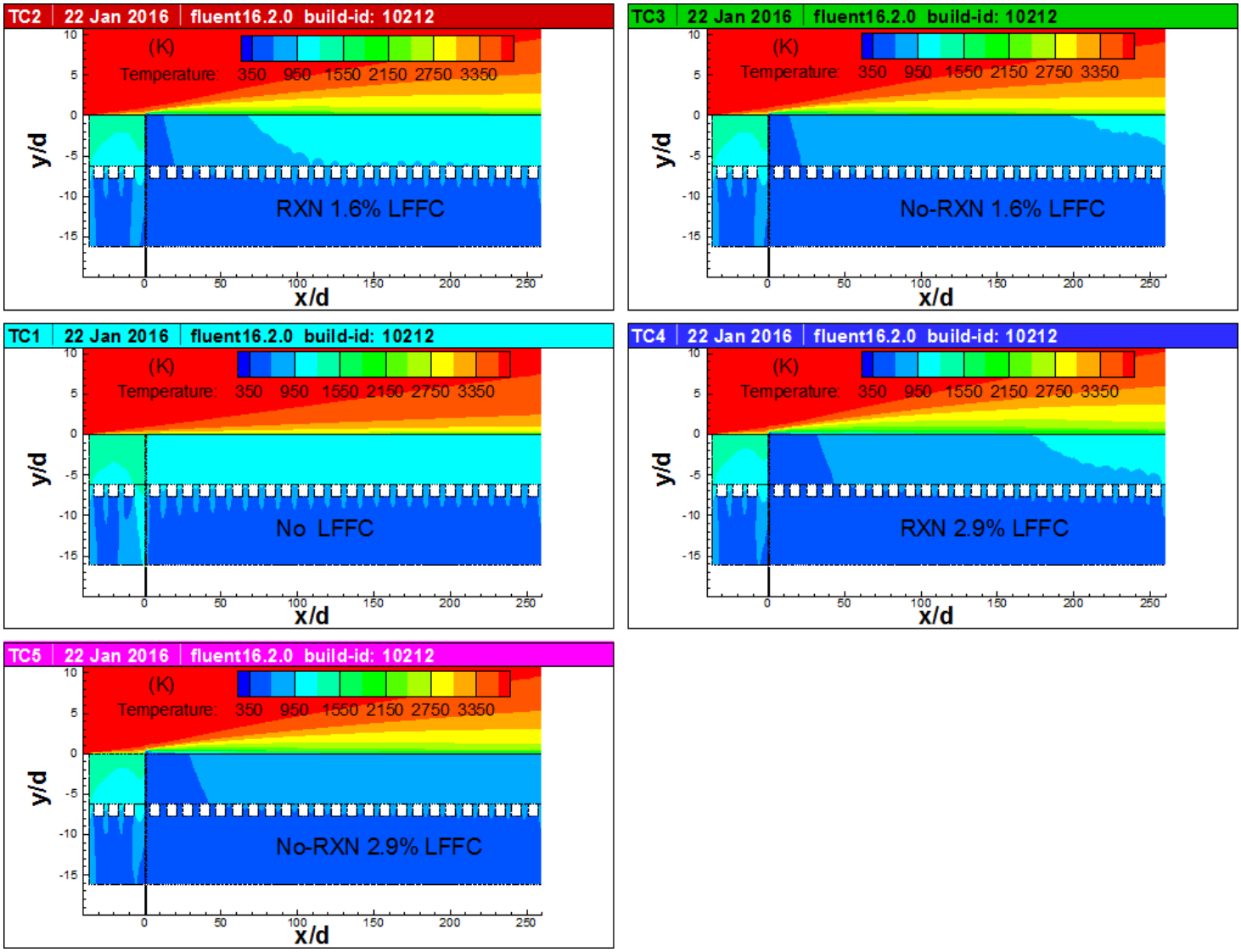

Figure 147. Temperature contours of full solid domain Test Cases 1–5. Aspect ratio is not 1:1 to allow for the full solid domain to be visible.

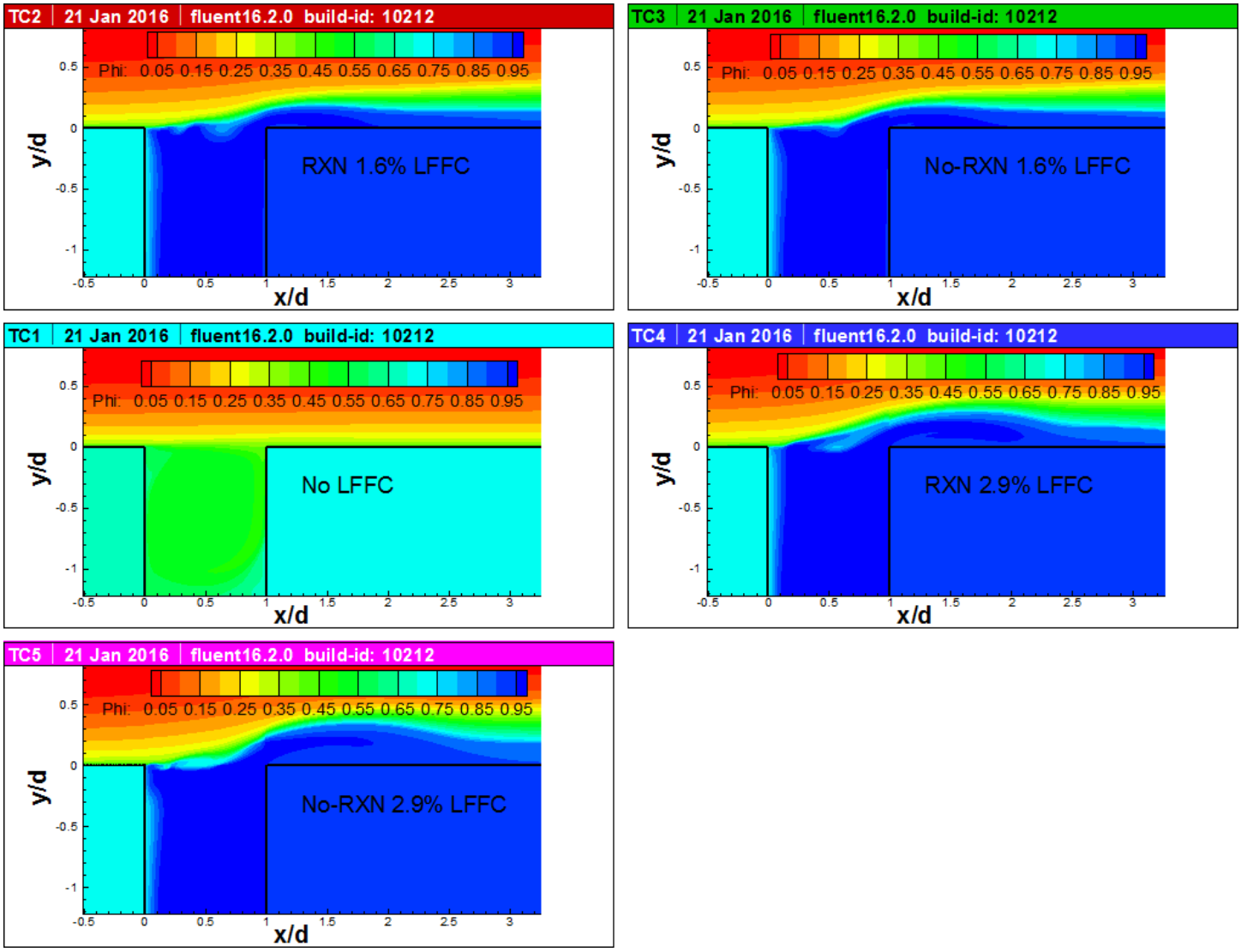

Figure 148. Overall effectiveness (Φ)—non-dimensional temperature—contours for Test Cases 1–5 in the fuel injection slot region.

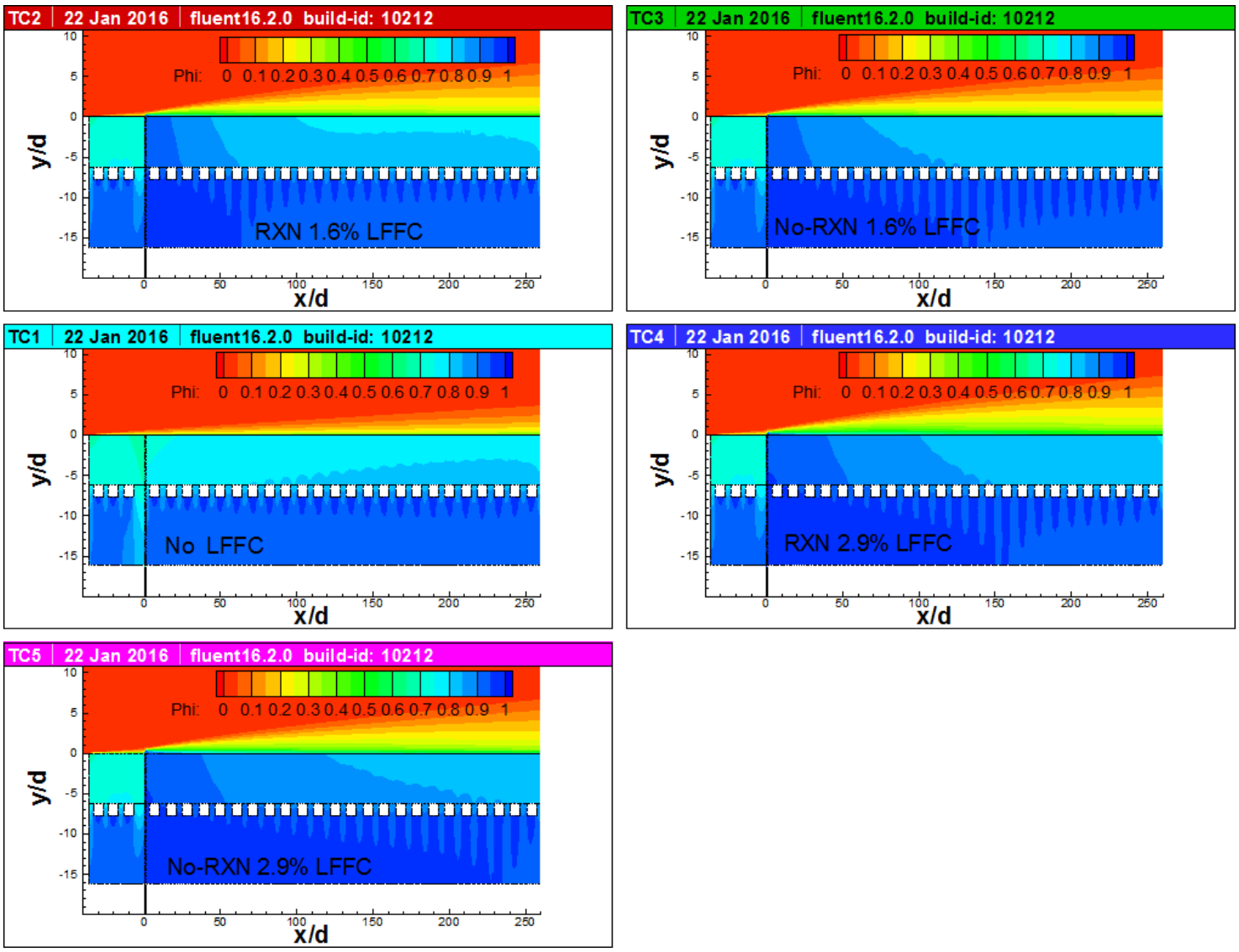

Figure 149. Overall effectiveness (Φ)—non-dimensional temperature—contours for Test Cases 1–5 for the full solid domain.

Appendix M. Volume of Fluid Contours: Final Test Cases

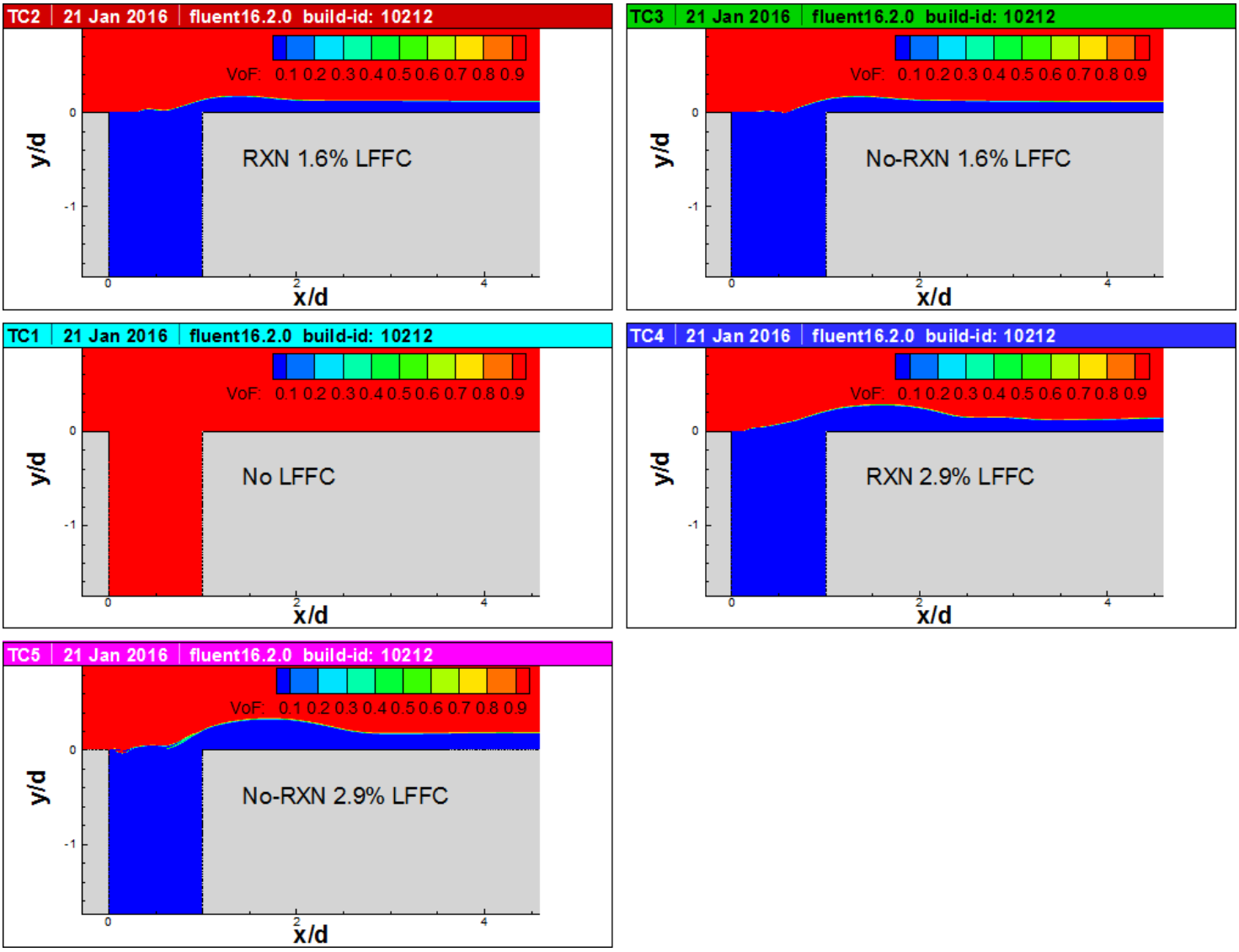

Figure 150. Volume of Fluid contours in the fuel injection slot region for Test Cases 1–5. A VoF of 1 indicates pure gaseous freestream while a value of 0 indicates pure liquid kerosene.

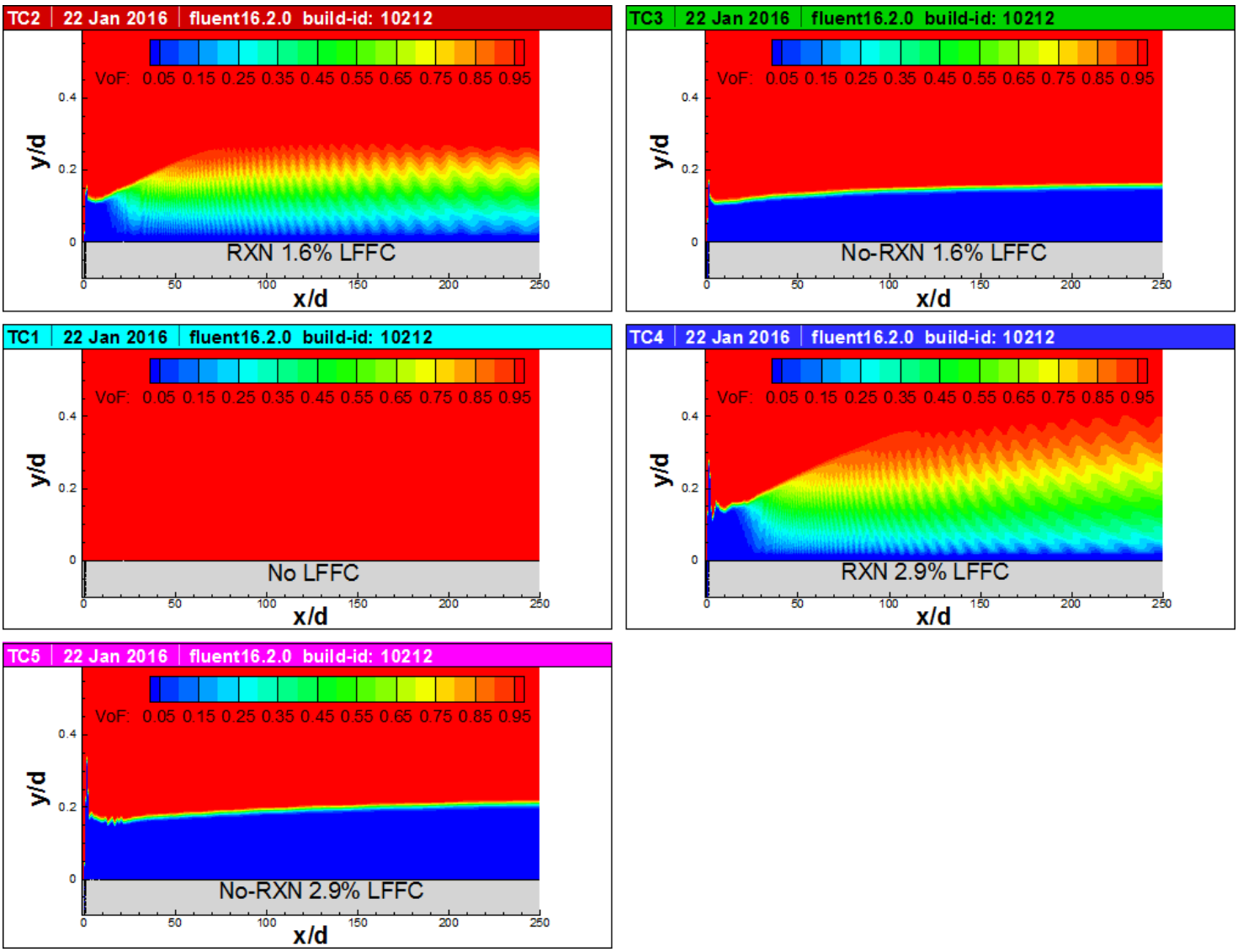

Figure 151. Volume of Fluid contours along full non-LFFC surface for Test Cases 1–5. A VoF of 1 indicates pure gaseous freestream while a value of 0 indicates pure liquid kerosene. Note the aspect ratio is not 1:1 in order for the full profile to be visible in a single image.

Appendix N. Density Contours: Final Test Cases

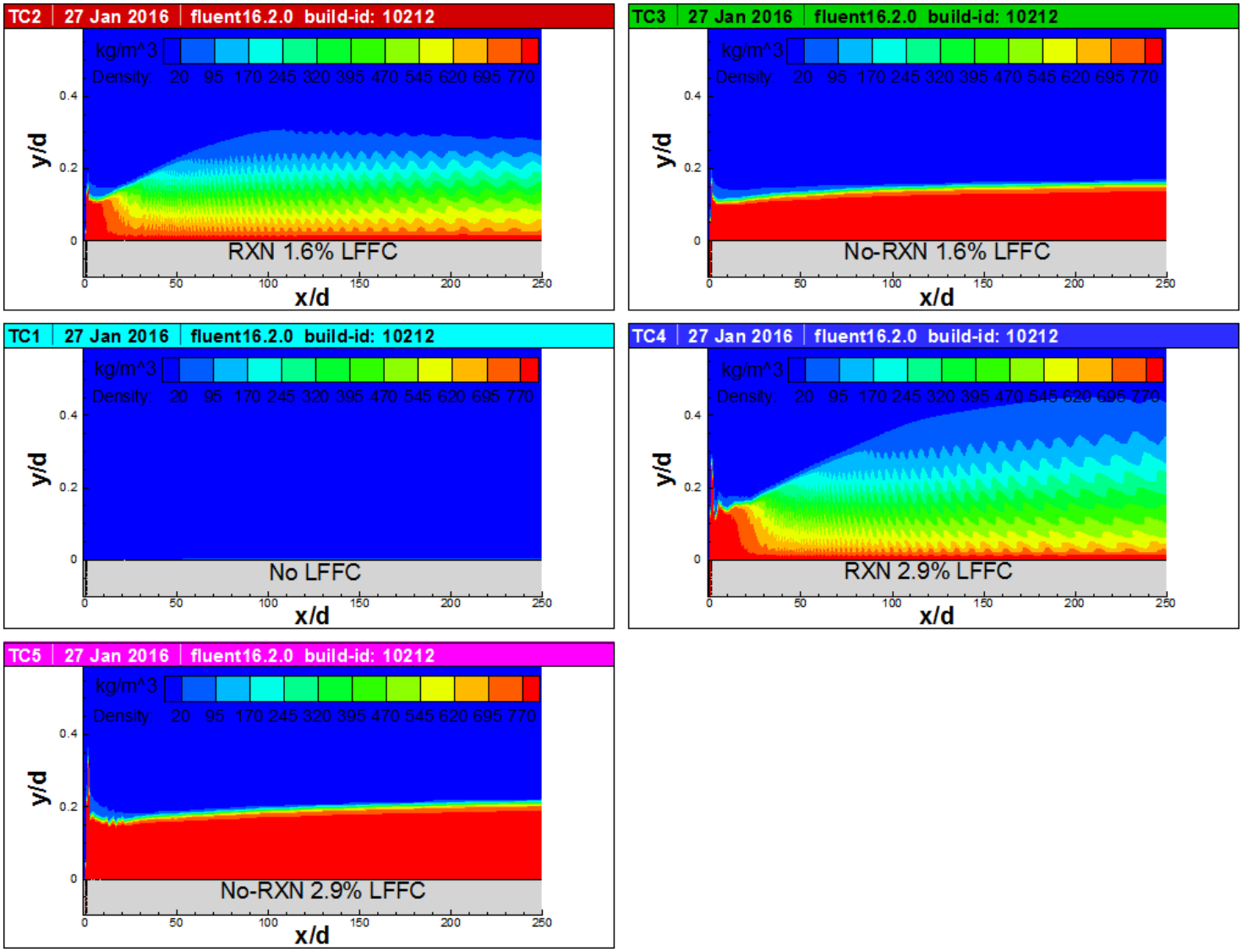

Figure 152. Density contours of Test Cases 1 through 5.

Appendix O. Reaction Rate Contours: Test Cases 2 & 4

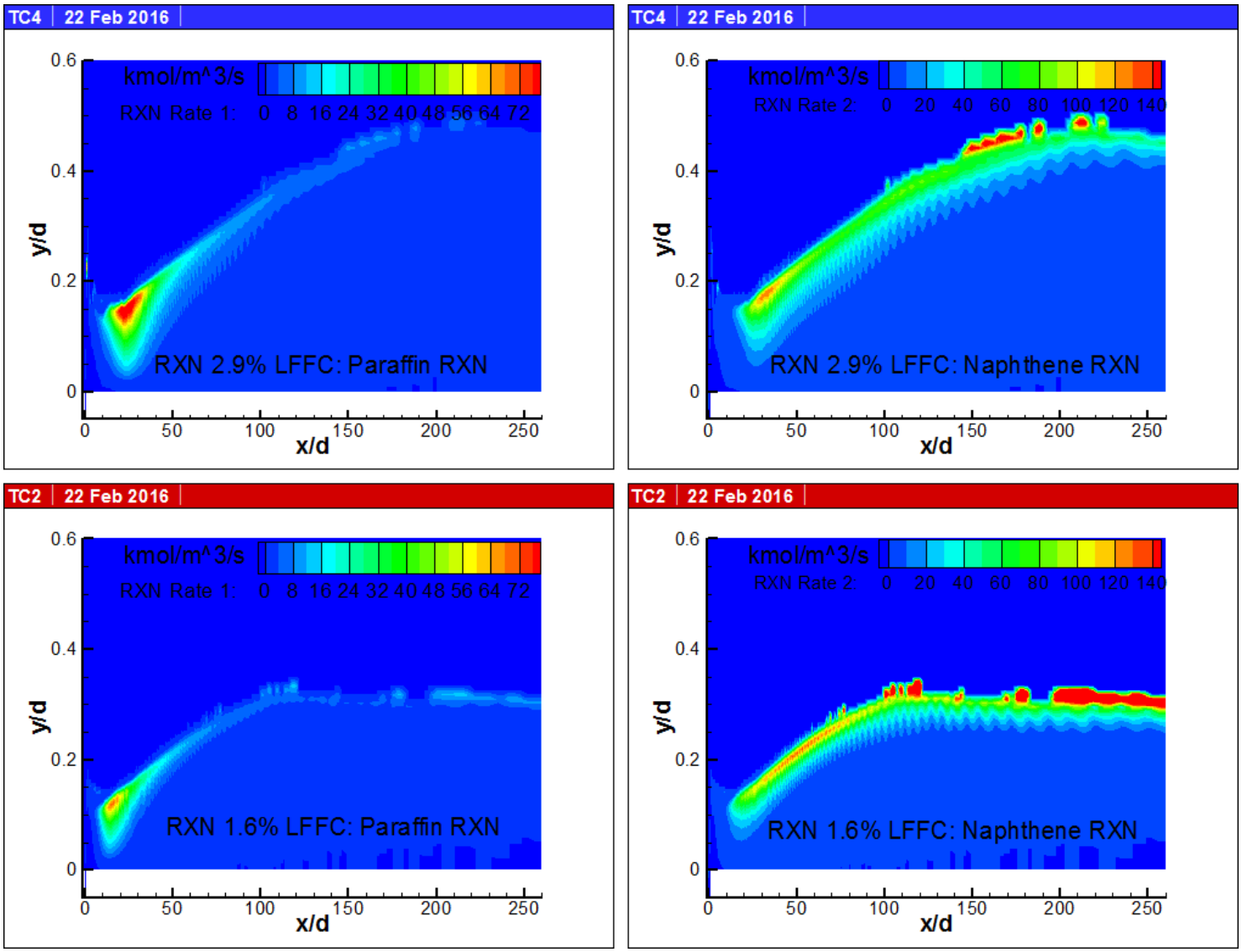

Figure 153. Reaction rates (kmol/m^3/s) for both heterogeneous reactions for the two reacting test cases (2 & 4).

Bibliography

1. *ANSYS Fluent Theory Guide*. Canonsburg, PA: ANSYS, Inc., 2012.

2. Arnold, Richard, et al. "Circumferential Film Cooling Effectiveness in a LOX/H2 Subscale Combustion Chamber." *47th AIAA Aerospace Sciences Meeting*. Orlando, FL: American Institute of Aeronautics and Astronautics, 2009.

3. Bills, Jacob, et al. "Modeling Fuel Film Cooling on Rocket Engine Walls." *54th AIAA Aerospace Sciences Meeting*. 1–11. San Diego, CA: American Institute of Aeronautics and Astronautics, 2016.

4. Blazek, Jiri. *Computational Fluid Dynamics: Principles and Applications* (2nd Edition). San Diego, CA: Elsevier Inc., 2005.

5. Celik, Ismail B., et al. "Procedure for Estimation and Reporting of Uncertainty Due to Discretization in CFD Applications," *Journal of Fluids Engineering*, *130*(7) (2008).

6. Cohn, Richard, "Hydrocarbon Boost Technology for Future Spacelift,", Air Force Research Laboratory, Presentation, 2012. `http://www.dtic.mil/dtic/tr/fulltext/u2/a556159.pdf`.

7. Gater, R. A., et al. *Liquid-film Cooling Its Physical Nature and Theoretical Analysis*. Technical Report, West Lafayette, IN: Purdue University, 1965.

8. Himansu, Ananda, et al. "Modeling of Fuel Film Cooling on Chamber Hot Wall." *50th AIAA/ASME/SAE/ASEE Joint Propulsion Conference*. 2014.

9. Humble, Ronald, et al. *Space Propulsion Analysis and Design*. New York: McGraw-Hill, 1995.

10. Huzel, Dieter K. and David H. Huang. *Design of Liquid Propellant Rocket Engines*. Washington DC: American Insitute of Aeronautics and Astronautics, 1992.

11. Incropera, Frank, et al. *Fundamentals of Heat and Mass Transfer* (6th Edition). Hoboken, NJ: John Wiley & Sons, Inc, 2007.

12. Kirchberger, C, et al. "Experimental Investigation on Film Cooling in a Hydrocarbon / GOX Rocket Combustion Chamber." *48th AIAA/ASME/SAE/ASEE Joint Propulsion Conference*. 1–14. 2012.

13. Knuth, Eldon L. *The Mechanics of Film Cooling*. Dissertation, California Institute of Technology, 1954.

14. Menter, F. R. "Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications," *AIAA Journal*, *32*(8):1598–1605 (1994).

15. Na, S, et al. "CFD Analysis of Film Cooling." *44th Aerospace Sciences Meeting*. Reno, NV: American Institute of Aeronautics and Astronautics, 2006.

16. Patankar, Suhas. *Numerical Heat Transfer and Fluid Flow*. Hemisphere, 1980.

17. Pletcher, Richard H, et al. *Computational Fluid Mechanics and Heat Transfer* (3rd Edition). CRC Press, 2013.

18. Roache, P. J. "Quantification of Uncertainty in Computational Fluid Dynamics," *Annual Review of Fluid Mechanics*, *29*:123–60 (1997).

19. Roache, Patrick J. "Conservatism of the Grid Convergence Index in Finite Volume Computations on Steady-State Fluid Flow and Heat Transfer," *Journal of Fluids Engineering*, *125*(4):731 (2003).

20. Shine, S. R., et al. "A new generalised model for liquid film cooling in rocket combustion chambers," *International Journal of Heat and Mass Transfer*, *55*(19-20):5065–5075 (2012).

21. Slater, John W., "Examining Spatial (Grid) Convergence,", NASA, Web Page [Accessed: 2016-02-10], 2008. `http://www.grc.nasa.gov/WWW/wind/valid/tutorial/spatconv.html`.

22. Snyder, Christopher, "Chemical Equilibrium with Applications,", NASA, Web Page [Accessed: 2015-01-01]. `http://www.grc.nasa.gov/WWW/CEAWeb/ceaHome.htm`.

23. Sutton, George P. and Oscar Biblarz. *Rocket Propulsion Elements*. Wiley, 2010.

24. Tu, Jiyuan, et al. *Computational Fluid Dynamics: A Practical Approach* (2nd Edition). Waltham, MA: Butterworth-Heinemann, 2013.

25. Turns, Stephen. *An Introduction to Combustion Concepts and Applications* (3rd Edition). McGraw Hill, 2012.

26. Wang, Ten-See. "Thermophysics Characterization of Kerosene Combustion," *Journal of Thermophysics and Heat Transfer*, *15*(2):140–147 (2001).

27. White, Frank M. *Viscous Fluid Flow* (3rd Edition). New York: McGraw-Hill, 2006.

28. Wilcox, David, C. *Turbulence Modeling for CFD* (3rd Edition). DCW Industries, Inc., 2010.

29. Winterton, R. "Where did the Dittus and Boelter equation come from?," *International Journal of Heat and Mass Transfer*, *41*:809 (1998).

30. Yang, Wei and Bing Sun. "Numerical Simulation of Liquid Film in a Liquid Oxygen/Rocket Propellant 1 Liquid Rocket," *Journal of Thermophysics and Heat Transfer*, *26*(2):328–336 (2012).

31. Yang, Wei and Bing Sun. "Numerical simulation of liquid film and regenerative cooling in a liquid rocket," *Applied Thermal Engineering*, *54*(2):460–469 (2013).

32. Zhang, H.W., et al. "Numerical study of liquid film cooling in a rocket combustion chamber," *International Journal of Heat and Mass Transfer*, *49*(1-2):349–358 (2006).

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
24–03–2016	Master's Thesis	Sept 2014 — Mar 2016

4. TITLE AND SUBTITLE

Liquid Fuel Film Cooling:
a CFD Analysis With Hydrocarbon Fuel

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Jacob D. Bills, Capt, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENY-16-M-199

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Dr. Edward Coy
Building: 8451
Edwards AFB, CA 93524
DSN: 525-5219
Email: edward.coy@us.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/RQRC

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Cooling of liquid rocket engine combustion chambers and nozzles is a critical component to liquid rocket engine design. A common method of cooling is liquid fuel film cooling. Liquid fuel is injected along the surface of the wall to act as a barrier against the core combustion gases. A numerical model is developed for simulating liquid fuel film cooling in a rocket engine using a hydrocarbon fuel. The model incorporates turbulent multiphase flow with species transport within the commercial ANSYS® Fluent CFD software. Conjugate heat transfer is simulated through walls containing embedded cooling channels. A novel User Defined Function is written to incorporate heterogeneous chemical reactions between the liquid fuel and the freestream gases. Comparisons are made between simulations with and without the heterogeneous reactions as well as with simplifications to the cooling channel geometry. It is found that simplifications to the cooling geometry can artificially reduce cooling performance while neglecting the heterogeneous reactions can artificially increase cooling performance.

15. SUBJECT TERMS

Computational Fluid Dynamics, Liquid Fuel Film Cooling, Multiphase Flow, Heterogeneous Reactions, Liquid Rocket Engines, Conjugate Heat Transfer

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Darrell S. Crowe, Maj, USAF
U	U	U	U	253	19b. TELEPHONE NUMBER *(include area code)* (937)255-3636 x4204