

Spring 1-24-2019

Adding Semi-Structured Automated Grid Generation and the Menter-Shear Stress Turbulence Transport Model for Internal Combustion Engine Simulations to Novel FEM LANL Combustion Codes

Brad Montgomery Philipbar
University of New Mexico - Main Campus

Follow this and additional works at: https://digitalrepository.unm.edu/me_etds



Part of the [Heat Transfer, Combustion Commons](#)

Recommended Citation

Philipbar, Brad Montgomery. "Adding Semi-Structured Automated Grid Generation and the Menter-Shear Stress Turbulence Transport Model for Internal Combustion Engine Simulations to Novel FEM LANL Combustion Codes." (2019).
https://digitalrepository.unm.edu/me_etds/163

This Thesis is brought to you for free and open access by the Engineering ETDs at UNM Digital Repository. It has been accepted for inclusion in Mechanical Engineering ETDs by an authorized administrator of UNM Digital Repository. For more information, please contact amywinter@unm.edu.

Brad M. Philipbar

Candidate

Mechanical Engineering

Department

This thesis is approved, and it is acceptable in quality
and form for publication:

Approved by the Thesis Committee:

Dr. Sang Lee, Chairperson UNM Mech Eng. Dept

Dr. David Carrington, Advisor LANL-T3: Solid Mechanics and CFD

Dr. Peter Vorobieff, UNM Mech Eng. Dept

**Adding Semi-Structured Automated Grid Generation and
the Menter-Shear Stress Turbulence Transport Model for
Internal Combustion Engine Simulations to Novel FEM
LANL Combustion Codes**

by

Brad M. Philipbar

B.S., Physics, University of New Mexico, 2016

THESIS

Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master's Degree

Mechanical Engineering

The University of New Mexico
Albuquerque, New Mexico

May, 2019

Acknowledgements

I would sincerely like to thank Dr. David Carrington and Dr. Jiajia Waters for their expertise, patience, and insight as well as thank my committee members, Dr. Peter Vorobieff and Dr. Sang Lee. Also, I would like to give special thanks to Dr. Peter Eiseman and Dr. Bradley Knockel for their guidance during this process.

**Adding Semi-Structured Automated Grid Generation and the
Menter-Shear Stress Turbulence Transport Model for Internal
Combustion Engine Simulations to Novel FEM LANL Combustion Codes**

by

Brad M. Philipbar

B.S., Physics, University of New Mexico, 2016

M.S., Mechanical Engineering, University of New Mexico, 2018

ABSTRACT

The addition of GridPro semi-structured, automated generation of grids for complex moving boundaries for combustion engine applications and the Menter Shear Stress Turbulent Transfer (SST) model are being developed by Los Alamos National Laboratory. The software is called Fast, Easy, Accurate, and Robust Continuum Engineering (FEARCE). In addition to improving the time and effort required to build complex grid geometry for turbulent reactive multi-phase flow in internal combustion engines, the SST turbulence model has been programmed into the Predictor Corrector Fractional-Step (PCS) Finite Element Method (FEM) for reactive flow and turbulent incompressible flow regime validation is performed. The Reynolds-Averaged Navier-Stokes finite-element solver without h -adaption is used for validation of the SST turbulence model on two benchmark problems in the subsonic flow regime: (1) 2D duct channel flow, and (2) a 2-D backward-facing step with an applied constant heat flux on the bottom surface downstream of the single-sided sudden expansion of the step. The 2D BFS using the newly installed SST FEARCE code yielded a corresponding $X_{re} = 6.655H$ vs. that as experimentally determined by Vogel and Eaton [1] of $X_{re_{vogel}} = 6.66667H$.

Contents

Acknowledgements	iii
Abstract	iv
Preface	xv
1 Introduction	1
2 Literature Review	3
2.1 Background Introduction	3
2.2 Turbulence Modeling	3
2.3 Convective Heat Transfer Downstream of a Backward Facing Step	5
3 Continuum Equations for Fluids	7
3.1 Governing Equations	9
3.1.1 Derivation of the Instantaneous Conservation of Mass Equation	9
3.1.2 Derivation of the Instantaneous Momentum Equation	10
3.1.3 Stress and Strain Relationships in Fluids	13
3.2 Turbulent Flows and Reynolds Time Averaging	14
3.2.1 Two-Equation Shear-Stress Transport Closure Model	17
3.2.2 Turbulent Energy Transport	20
3.2.3 Derivation of the Enthalpy Transport Equation	21

3.2.4	Derivation of the Turbulent Energy Transport Equation	24
3.3	The Finite Element Method (FEM)	25
3.3.1	The Weighted Residual Method	26
3.3.2	Error and Convergence Estimations (FEM)	29
3.3.3	Galerkin Method of Weighted Residuals	30
3.3.4	FEM Computational Space: Global Coordinates and Natural Coordinates	32
3.3.5	Numerical Integration	34
4	Predictor Corrector Split (PCS) Solution Method	36
4.1	Projection Algorithms	38
4.2	Weak Form of Governing Equations	42
4.3	Matrix Equations	43
4.3.1	Weighted Residual Statement of Velocity under Helmholtz-Hodge Decomposition for $k - \omega$	43
4.3.2	Weighted Residual Statement of the Thermal Energy Equation	44
4.3.3	Weighted Residual Statement of the Turbulent Kinetic Energy for $k - \omega$	44
4.3.4	Weighted Residual Statement of the Specific Dissipation Rate for $k - \omega$	45
4.3.5	Weighted Residual Statement of the Specific Dissipation Rate for MSST	45
4.4	Time Evolution of the Semi-Implicit Scheme for Mass, Momentum, and Energy Equations	47
4.4.1	Momentum in Turbulent Flow with Multi-Species	48
4.4.2	Fractional Split Method	48
4.4.3	Energy Conservation for Multi-Species and Turbulent Energy Transport	52

4.4.4	Species Transport	54
4.4.5	Turbulent Species Transport	54
4.4.6	Petrov-Galerkin Stabilization Scheme	55
4.4.7	Finite Element Matrix Terms for the Mass, Momentum, Energy and Species Equations	56
4.4.8	Turbulent Model Matrices	59
4.5	Boundary Conditions	60
4.5.1	Boundary Conditions for Velocity	60
4.5.2	Viscous and Inviscid Boundary Conditions for Velocity	61
4.5.3	Boundary Conditions for Turbulent Kinetic Energy and Specific Dissipation Rate	61
4.5.4	Artificial Compressibility	64
5	Method Validation	67
5.1	Method Validation	67
6	Conclusions, Solutions and Results for 2D Duct	72
6.1	Solutions and Results for 2D Duct Flow	72
7	Conclusions, Solutions and Results for 2D BFS	76
7.1	Solutions and Results for 2D BFS flow	76
A	Appendix: Coding the Improved MSST	88
A.1	MSST Research Outline	88
A.1.1	Minor Bug Fixes	88
A.1.2	Coding MSST-Step1:	90
A.1.3	Coding MSST-Step 2:	96
A.1.4	Coding MSST-Step 3:	98
A.1.5	Coding MSST-Step 4:	99

<i>CONTENTS</i>	viii
A.1.6 Coding MSST-Step 6	111
B Appendix: GridPro for KIVA and FEARCE	114
References	128

List of Figures

5.1	Schematic of the 2D duct, $H = 1$ cm and $L = 25$ cm.	68
5.2	Schematic of the 2D backward facing step, the step height $H = 3.8$ cm, $L_e = 22.0$ cm and $L_d = 100$ cm.	69
5.3	U -component of velocity [cm/s] streamline plot for the <i>SST</i> FEARCE turbulence model on a 375,000 element, wall bounded 3D GridPro grid at $t = 0.33s$. The solution looks very promising, further study will need to be continued with the addition of the <i>SST</i> model to the parallel FEARCE code.	70
6.1	Duct channel flow $Re=13,750$ comparing the two-equation FEARCE <i>SST</i> code vs. Wilcox $k-\omega$ [2] vs. DNS Mansour et al. [3] at 22 hydraulic diameters from the inlet.	73
6.2	$U+$ vs. $Y+$ comparison of the two-equation FEARCE <i>SST</i> code vs. Wilcox $k-\omega$ [2].	74
6.3	Turbulent energy production $P+$ vs. turbulent energy dissipation $e+$ for duct channel flow at $Re=13,750$ comparing the two-equation FEARCE <i>SST</i> code vs. Wilcox $k-\omega$ [2] vs. DNS Mansour et al. [3].	75

7.1 *U*-velocity [*cm/s*] BFS shown at $Re=28,000$ for the two-equation FEARCE *SST* code at solution time 1.35 [*sec*]. The solution reaches state state appreciably quicker than $k-\omega$ *h*-adapted FEM [4] that reported stable recirculation at around 0.33 [*sec*]. The two-equation FEARCE *SST* code finds stable recirculation at approximately 0.15 [*sec*]. At solution time 1.35 [*sec*] the max L_2 norm was $4.32e-12$, and the FEARCE *SST* code solution was considered converged. The reattachment is 25.29 *cm* corresponding to a $X_{re} = 6.655H$ 78

7.2 *U*-velocity [*cm/s*] BFS reattachment shown at $Re = 28,000$ for the two-equation FEARCE *SST* code at solution time 1.35 [*sec*]. At solution time 1.35 [*sec*] the max L_2 norm was $4.32e-12$, and the FEARCE *SST* code solution was considered converged. The measured point of reattachment was 25.29 *cm* corresponding to a $X_{re} = 6.655 H$ in comparison to the experimentally determined $X_{re_{vogel}} = 6.66667 H$ from Vogel and Eaton [1], it is in considerably good agreement. Typical simulation reattachments for Vogel and Eaton [1] at $Re = 28,000$ have been seen ranging from $X_{re} = 6.8H$ to $7.2H$ 79

7.3 *U*-velocity [*cm/s*] BFS shown at $Re=28,000$ for the two-equation FEARCE *SST* code where the location of slices at $X/H = -3$ and $X/H = 20$ were integrated over to calculate the mass flow rate for Fig. (7.4). The location $X/H = -3$ was chosen for the reference velocity U_{ref} by Vogel and Eaton [1] as the mean velocity for the velocity profile analysis. . . 80

7.4 BFS mass flow rate analysis for comparing the mass flow rates at $X/H = -3$ and $X/H = 20$ at $Re=28,000$ for the two-equation FEARCE *SST* code. 81

7.5 Temperature [*K*] plot for the backward facing step at $Re=28,000$ for the two-equation FEARCE *SST*. 82

7.6	Thermal profiles for the backward facing step for $Re=28,000$ comparing the two-equation FEARCE <i>SST</i> code vs. $k-\omega$ h -adapted FEM [4]. The thermal profiles are also compared to the experimental data from Vogel and Eaton [1] at $X/H = 0.33$	83
7.7	Thermal profiles for the backward facing step for $Re=28,000$ comparing the two-equation FEARCE <i>SST</i> code vs. FEARCE $k-\omega$ non- h -adapted FEM [4]. The thermal profiles are also compared to the experimental data from Vogel and Eaton [1] at $X/H = 4.3$	84
7.8	U -component velocity over U_{ref} velocity profiles for the backward facing step at $Re=28,000$ comparing the two-equation FEARCE <i>SST</i> code on a GridPro grid with 6014 nodes vs. h -adapted $k-\omega$ FEM [4] with a final adapted node count of 153,000 nodes. The velocity profiles are also compared to the experimental data from Vogel and Eaton [1] for $x = -0.67x_{re}$, $x = -0.33x_{re}$, $x = 0.0x_{re}$, and $x = 0.33x_{re}$	85
7.9	Turbulent viscosity plot for the backward facing step at $Re=28,000$ for the two-equation FEARCE <i>SST</i>	86
7.10	Turbulent kinetic energy plot for the backward facing step at $Re=28,000$ for the two-equation FEARCE <i>SST</i>	86
7.11	Turbulent dissipation plot for the backward facing step at $Re=28,000$ for the two-equation FEARCE <i>SST</i>	86
7.12	Turbulent energy dissipation plot for the backward facing step at $Re=28,000$ for the two-equation FEARCE <i>SST</i>	87
7.13	Turbulent energy production plot for the backward facing step at $Re=28,000$ for the two-equation FEARCE <i>SST</i>	87
B.1	Temperature distribution for a KIVA-4mpi turbulent combustion engine simulation with snapping moving parts on a GridPro grid 2 valve, 2 port test system.	115

B.2	Turbulent kinetic energy plot distribution with bead stream-traces of a KIVA-4mpi turbulent combustion engine simulation with snapping moving parts on a GridPro grid 2 valve, 2 port test system.	116
B.3	KIVA-4mpi Snapping System for Moving Immersed Parts (GridPro 2 valve, 2 port idealized test system).	117
B.4	Schematic breakdown of the FEARCE/GridPro moving marker valve overlay 3D port blocking properties.	118
B.5	Schematic breakdown of the FEARCE/GridPro moving marker overlay grid system.	120
B.6	A 4-valve DISI moving marker system FEARCE test simulation on GridPro grid. Bowled piston, valves, and manifold surfaces created in GridPro from x-ray tomography.	121
B.7	A 1.2 million node 4 valve direct injection spark injection (DISI) engine from x-ray tomography, with .stl triangulated (tet) moving marker valves and bowled piston, grid generated by with GridPro. The grid has been sparsened down from 40 million.	122
B.8	Spray-G needle valve injector GridPro grid, the image shows an outer ream of topology nodes along with the surfaces used to create the grid overlaid onto the grid.	123
B.9	Spray-G GridPro injector nozzle grid.	123
B.10	Spray-G GridPro injector nozzle grid with needle valve sleeve.	124
B.11	Spray-G GridPro injector nozzle cell vs skew distribution, exhibiting the high quality found in GridPro's capability to mesh complex components.	124
B.12	The Spray-G GridPro injector was then "modularized" by creating a standalone topology, this module can now be effectively plugged into test engine designs. Surfaces and Topology(left), Grid(right)	125

B.13 A GridPro spark system was then "modularized", this module can also now be plugged into test engine designs. Surfaces and Topology(left), Grid(right) 125

B.14 A GridPro 4V manifold with insert topology for spark and Spray-G modules. 126

B.15 A GridPro 4V manifold with insert topology for spark and Spray-G modules, top down view. 126

B.16 A GridPro 4V manifold with inserted topology spark and Spray-G modules. 127

B.17 A GridPro 4V manifold with inserted topology spark and Spray-G modules(left) and validated grid(right). 127

List of Tables

7.1 Species Composition for Air.	78
--	----

Preface

Nomenclature

\mathbf{A}_u	advection matrix	F_{es}	heat flux (W/m^2)
$\mathbf{K}_{\tau u}$	fluid stress matrix (N/m^2)	$f_{k,j}$	body forces (N/m^3)
$\mathbf{M}u$	mass matrix	F_u	velocity boundary gradient
χ_i^j	molar fraction (<i>moles/totalmoles</i>)	H_j	enthalpy of species j (N/m^2)
\dot{q}	rate of heat flux (W/m^2)	$H_{o,j}$	enthalpy of formation (J)
\dot{q}_d	rate of heat flux from species diffusion divergence of a flux ($\nabla \cdot q$)	K	turbulent kinetic energy (m^2/s^2)
w_{chem}^j	source term - chemical reaction (kg/s)	k	kinetic energy (m^2/s^2)
w_{spray}^j	source term - spray evaporation (kg/s)	N	basis or shape function
ϵ	turbulent dissipation rate (m^2/s^3)	n	current time level
γ	ratio of specific heats	$n + 1$	current time level + dt
γ_i^j	nodal (i) mass fraction species (j) (ρ_k/ρ)	P	pressure (Pa)
γ_k	mass fraction k^{th} species (ρ_k/ρ)	p	pressure (Pa)
$\gamma_j f_j(x_i)$	body force of species (N/m^3)	P_k	turbulent production (kg/ms^3)
$\gamma_j V_j$	species diffusion velocity (m/s)	Pr	Prandtl number (<i>non - dimensional</i>)
$\hat{\phi}$	general <i>approximated</i> dependent variable	Pr_t	turbulent Prandtl number (<i>non - dimensional</i>)
κ	conductance coefficient ($W/m^{\circ}K$)	Q	general source term ($kg/m^3s, J/m^3s$)
κ_{mix}	conductance coefficient for mixture density	q	flux ($W/m^2, kg/m^2$)
λ	second coefficient of viscosity	q_{div}	divergence of flux ($\nabla \cdot q$)
μ	fluid viscosity, aggregate viscosity ($N \cdot s/m^2, kg/m \cdot s, Pa \cdot s$)	R	residual
μ_{mix}	turbulent viscosity ($N \cdot s/m^2, kg/m \cdot s$)	Re	Reynolds number (<i>non - dimensional</i>)
ω	specific dissipation rate ($1/s$)	S_{ij}	strain rate ($N/m^2, kg/ms^2$)
ϕ	general dependent variable	Sc	Schmidt number (<i>non - dimensional</i>)
ρ	density (kg/m^3)	Sc_t	Schmidt number (<i>non - dimensional</i>)
c	speed of sound	T	temperature ($^{\circ}K$) Kelvin
c_p, c_v	specific heat at constant P ($J/kg^{\circ}K$)	t_{ij}	stress tensor ($N/m^2, kg/m^2$)
D	diffusion of species (m^2/s)	U'	fluctuating velocity (m/s)
D_k	turbulent diffusion (m^2/s)	U^*	intermediate specific momentum (kg/ms)
dt	time step size (s)	u_i	velocity component i (m/s)
E	total internal energy (ϕe)(J)	U_{ij}	specific momentum (kg/ms)
e	specific internal energy (J/kg)	W	weight in weighted residual
f_{drop}	drop forces (N/m^2)	w_j	species volumetric source (kg/m^3s)

Chapter 1

Introduction

Numerical modeling of turbulent reactive multi-phase flow in internal combustion engines is a complex multi-scale problem. Adding to the difficulty of modeling the applicable physics is the complex domain with moving immersed bodies and boundaries. A large aspect of modeling internal combustion engines rapidly is providing a quality grid of the engine, and ideally be able to do this with a minimal amount of expensive labor.

This manuscript will discuss new capabilities available in LANL codes, in particular the introduction of the Menter Shear-Stress Transport (SST) model [5] into the the Fast Easy, Accurate and Robust Continuum Engineering (FEARCE) code. Also a new grid capability is introduced that has an important emphasis placed on reducing one of the most time-consuming components for the engine designer and combustion researcher: the ability to generate a high-quality grid quickly. Using GridPro [6] for its advanced meshing algorithm and automated grid generation increases quality and reduces the time and labor needed to run complex engine simulations. The meshes that can now be generated with GridPro benefit both the KIVA-4mpi parallel Finite Volume engine modeling software and newly developed KIVA-hpFE Finite Element code (FEARCE) which will include a Large eddy-simulation (LES) FEM turbulence

model, the existing k - ω two-equation turbulence model, and the newly developed SST turbulence model.

The intricacies of implementing GridPro grid generation are shown in Appendix (B) along with engine simulations using the grids. A large amount of success for this work was due to the early implementation of GridPro for KIVA-4mpi and then FEARCE LANL codes, both implementations and new capabilities are also shown in Appendix (B).

Chapter 2

Literature Review

2.1 Background Introduction

The purpose of this chapter is to introduce a background to contemporary and historical turbulence modeling and computational fluid dynamics (CFD). Following this development with the particular background that is relevant to the separated flow, adverse pressure gradient backward facing step geometry that has an applied a constant heat-flux surface behind the single-sided sudden expansion.

2.2 Turbulence Modeling

The beginning of turbulence studies can be attributed to the time when Boussinesq first introduced the eddy viscosity supplement to molecular viscosity in 1887 [7], followed by Reynolds time averaging, then Prandtl included a mixing length hypothesis helpful in solving the eddy viscosity problem in [8]. Among the core characteristics of Prandtl's initial contribution in 1904 was the revolutionary insight that the effects of friction are somewhat negligible unless fluid is very close to the wall of the object moving through the fluid, thus remedying D'Alembert's hydrodynamic paradox of 1752 [9]. Prandtl later conceptualized that eddy viscosity is the proportionality fac-

tor that describes the turbulent transfer of energy due to moving eddies, giving rise to tangential stresses.

Developed herein, two two-equation models are used in the FEM, one pre-existing introduced by Wilcox [2] or rather more properly by Kolmogorov in 1942, and the other that blends these two most popular models, that of Wilcox's $k-\omega$ and $k-\epsilon$ introduced by Menter [10]. Of the two-equation models $k-\epsilon$ is the most widely used and focuses on aspects of turbulence modeling that affect turbulent kinetic energy.

The $k-\epsilon$ model is specifically tailored to resolve planar shear layers and recirculating flows. A brief history leading up to the standard $k-\epsilon$ model would include the Particle-in-Cell (PIC) method proposed by Harlow in 1957 [11] [12]. The success of Harlow's PIC method gave the early CFD community the ability to solve seemingly *unsolvable* problems and paved the future for the Fluid-in-Cell (FLIC) [13], Marker-in-Cell (MAC) [14], Implicit-Continuous-Fluid-Eulerian (ICE) [15], and SOLA and Reactive flow codes [16] that would lead to the early numerical modeling of turbulence with $k-\epsilon$.

A brief history of the the $k-\omega$ two equation model begins with the turbulent length scale from Kolmogorov in 1941 [17]. Kolmogorov is generally considered to be a founding expert of modern turbulence theory. Kolmogorov provided the relationship of length scale $k^{1/2}/\bar{\omega}$ and time scale $1/\bar{\omega}$ to the $k-\omega$. The insight by Kolmogorov alleviates the need of former knowledge of the flow to get a solution to the two equation $k-\omega$.

In 1994 Menter blended the more popular $k-\epsilon$ two-equation model with that of $k-\omega$ to create what he called the Menter Shear Stress Transport model (SST) [10]. The $k-\epsilon$ like $k-\omega$ is rooted in finding a way to avoid algebraically setting turbulent length scales. Menter simply combined the two models, $k-\omega$ and $k-\epsilon$, so that $k-\omega$ can be used in the inner region local to the boundary layer and cleverly provides a switching mechanism to apply $k-\epsilon$ further off in the free stream shear portion of the flow. The SST model [10]

seeks to offset free-stream flow sensitivity and attempts to improve prediction of adverse pressure gradients, something that could be deemed useful in combustion engine modeling. To my knowledge the SST has not been applied to combustion modeling applications. Many forms of SST exist ranging from the “standard” (SST) model [10], the SST with Vorticity Source Term (SST-V) [18], SST from 2003 (SST-2003) [19], SST with Controlled Decay (SST-sust) [20], SST with Controlled Decay and Vorticity Source Term (SST-Vsust), SST with Rotation/Curvature Correction (SST-RC) [21], and Hellsten’s Simplified Rotation/Curvature Correction (SST-RC-Hellsten) [22]. Because of the various forms of the Menter SST model typical naming conventions have adopted naming individual models as the base model (SST) with the appended variant.

2.3 Convective Heat Transfer Downstream of a Backward Facing Step

Rich flow physics can be garnered from the geometrical simplicity of a backward facing step (BFS) [23]. Both experimental and theoretical investigations of backward-facing step flow are often used to investigate a CFD turbulence models effectiveness. The BFS produces an internally wall bounded flow with an adverse pressure gradient consisting of a separating boundary layer and reattaching shear layer.

In 1983 Armaly et al. [23] investigated BFS flow using laser-doppler measurements of velocity distribution and reattachment lengths were reported downstream for a single BFS mounted in a 2D channel for Reynolds numbers (Re) between $70 < Re < 8000$. Armaly found additional recirculation attachment locations downstream and further reinforced prior investigations by [24], [25], and [26]. Armaly noted high aspect ratio in the test section (1:36) was chosen specifically to ensure the incoming flow was fully developed, yet the experimental results showed 2D characteristics at low and

high Re values only ($Re < 400$ and $Re > 6000$). Within the nice 2D Re regimes, Armaly solved the numerical steady differential equations for mass conservation and momentum and showed meaningful comparative analysis to his experimental results; leaving the regimes that exhibited strong 3D effects to later investigations.

In 1985, Vogel & Eaton [1] developed an experiment to extract useful thermal transfer data from the bed of a backward facing step apparatus. Vogel & Eaton combined heat transfer and fluid dynamic measurements in the separated and reattaching boundary layer, with particular concern for near-wall regions. The Vogel & Eaton study applied a constant heat-flux surface behind a single-sided sudden expansion to produce Stanton number profiles as a function of Re and boundary-layer thickness at the location of the separating shear-wall layer. The study by Vogel & Eaton further emphasized how fluctuations in skin friction values controlled the heat (thermal) transfer rate near the reattachment location, adding that the Reynolds concepts of redeveloping boundary layer begin to display prevalence 2 to 3 step heights downstream of the reattachment location. The work of Vogel & Eaton in 1985 is also of practical interest to the heat transfer and CFD engineer due to the implications a large variance in local heat transfer coefficient can cause, but even more importantly the substantial overall heat augmentation affects prediction of thermal loads near shear reattachment areas [1].

Chapter 3

Continuum Equations for Fluids

The following are the conservative form of the governing equations for fluid flow commonly used in CFD textbooks:

$$(3.1) \quad \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{V}) = 0$$

$$(3.2) \quad \frac{\partial(\rho u)}{\partial t} + \nabla \cdot (\rho u \vec{V}) = -\frac{\partial P}{\partial x} + \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} + \rho f_x$$

$$(3.3) \quad \frac{\partial(\rho v)}{\partial t} + \nabla \cdot (\rho v \vec{V}) = -\frac{\partial P}{\partial y} + \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \tau_{yy}}{\partial y} + \frac{\partial \tau_{zy}}{\partial z} + \rho f_y$$

$$(3.4) \quad \frac{\partial(\rho w)}{\partial t} + \nabla \cdot (\rho w \vec{V}) = -\frac{\partial P}{\partial z} + \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \tau_{zz}}{\partial z} + \rho f_z$$

$$\begin{aligned}
(3.5) \quad \frac{\partial}{\partial t} \left[\rho \left(e + \frac{V^2}{2} \right) \right] + \nabla \cdot \left[\rho \left(e + \frac{V^2}{2} \right) \vec{V} \right] = \\
\rho \dot{q} + \frac{\partial}{\partial x} \left(k \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(k \frac{\partial T}{\partial y} \right) + \frac{\partial}{\partial z} \left(k \frac{\partial T}{\partial z} \right) \\
- \frac{\partial(\mathbf{u}p)}{\partial x} - \frac{\partial(\mathbf{v}p)}{\partial y} - \frac{\partial(\mathbf{w}p)}{\partial z} + \\
\frac{\partial(\mathbf{u}\tau_{xx})}{\partial x} + \frac{\partial(\mathbf{u}\tau_{yx})}{\partial y} + \frac{\partial(\mathbf{u}\tau_{zx})}{\partial z} + \frac{\partial(\mathbf{v}\tau_{xy})}{\partial x} + \\
\frac{\partial(\mathbf{v}\tau_{yy})}{\partial y} + \frac{\partial(\mathbf{v}\tau_{zy})}{\partial z} + \frac{\partial(\mathbf{w}\tau_{xz})}{\partial x} + \\
\frac{\partial(\mathbf{w}\tau_{yz})}{\partial y} + \frac{\partial(\mathbf{w}\tau_{zz})}{\partial z} + \rho \frac{\partial(\rho v)}{\partial t} + \nabla \cdot (\rho v \vec{V}) \\
= -\frac{\partial P}{\partial x} + \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \tau_{yy}}{\partial y} + \frac{\partial \tau_{zy}}{\partial z} + \rho \vec{f} \cdot \vec{V}
\end{aligned}$$

Equations (3.1), (3.2), (3.3), (3.4), and (3.5) describe the motion of fluid flow, and one can assume density does not vary with pressure for an incompressible form [27]. Equations (3.1), (3.2), (3.3), (3.4), and (3.5) also state conservation laws of physics, mass acting on a fluid is conserved (3.1), the time rate of change of momentum equals the sum of the forces acting on the fluid (3.2), (3.3), and (3.4), and the time rate of change of energy is equal to the sum of the rate of heat addition and work done on the fluid (3.5). Equations (3.2), (3.3), and (3.4) are commonly referred to as Newton's second law of motion (or as Navier-Stokes equations), while equation (3.5) is the first law of thermodynamics [27]. The governing equations (3.1), (3.2), (3.3), (3.4), and (3.5) form a coupled system and remain one of the fundamental open problems in non-linear analysis, being included in the Millennium Prize Problems by the Clay Mathematics Institute [28]. In a proper derivation from conservation laws of mass and momentum and principle assumptions relating the stresses to velocity gradients, smoothness of the solution is assumed. There is no self-consistent model of the Navier-Stokes equations, i.e., one that proves singularities in the solution do not develop in finite time from smooth initial data with finite energy [28].

3.1 Governing Equations

The governing relationships shown in equations (3.1), (3.2), (3.3), (3.4), and (3.5) can be averaged in time to produce the Reynolds-Averaged Navier-Stokes equations (RANS) [29]. The instantaneous equations can be represented with the addition of extra terms that arise from the Reynolds decomposition. Reynolds decomposition is a mathematical procedure that separates the expectation value from its fluctuations in an attempt to represent turbulent stresses. A derivation would include the base instantaneous momentum description which allows, with the derivation of the turbulent momentum equations, an approach to satisfy the closure problem. To get equations only containing the mean pressure and velocity the RANS equations must be solved by modeling the Reynolds stress terms as a function of the mean flow. By modeling the Reynolds stresses in this manner we can alleviate the need for referencing the fluctuating terms of the velocity field.

3.1.1 Derivation of the Instantaneous Conservation of Mass Equation

Conservative fields imply the restriction that a quantifiable amount of variable enters a closed system and exits in a similar fashion. Mass consistency can be expressed mathematically as:

$$(3.6) \quad \frac{\partial M_v}{\partial t} = \sum_{inflow} \dot{m} - \sum_{outflow} \dot{m}$$

where the time derivative of mass is the mass flow rate denoted by \dot{m} , which is equal to the density ρ , multiplied by a vector of velocity components \vec{V} . For simplicity a 2D derivation follows that is easily expandable to 3D. The velocity vector $\vec{V} = \{u, v\}^T$ has cartesian velocity components x and y . Using a control surface for the 2D deriva-

tion, equation (3.6) can be thought as a summation of mass-inflow and mass-outflow entering and exiting the domain. The equivalent statement to the total change in mass flow rate is stated in equation (3.7).

$$(3.7) \quad \frac{\partial \rho \Delta x \Delta y}{\partial t} = \rho u \Delta y + \rho v \Delta x - \left[\rho u + \frac{\partial \rho u \Delta x}{\partial x} \right] \Delta y - \left[\rho v + \frac{\partial \rho v \Delta y}{\partial y} \right] \Delta x$$

$$(3.8) \quad \frac{\partial \rho}{\partial t} = \frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} - \left[\frac{\partial \rho u}{\partial x} + \frac{\partial \rho u}{\partial x} \right] - \left[\frac{\partial \rho v}{\partial y} + \frac{\partial \rho v}{\partial y} \right] = - \left[\frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} \right]$$

The control area, similar to volume in 3D, is represented by $\Delta x \Delta y$ in equation (3.7). By dividing out the appropriate quantity and taking the limit as $\Delta x \Delta y$ tends toward dx and dy , we are left with equation (3.8). Equation (3.8) shows the mass flow contributions from the x and y directions, into and out of the control surface from $\rho u \Delta y$ and $\rho v \Delta x$. The common representation of equation (3.8) in 3D cartesian coordinates with vector notation is shown earlier in this chapter as equation (3.1). Distributing the vector differential operator to the second term of equation (3.1) the mass conservation equation is presented in a more familiar form as equation (3.9).

$$(3.9) \quad \frac{\partial \rho}{\partial t} + u \frac{\partial \rho}{\partial x} + v \frac{\partial \rho}{\partial y} + \rho \left[\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right] = 0$$

It is also common to represent the substantial derivative form of equation (3.9) as $D\rho/Dt + \rho \Delta \cdot \vec{V}$, where the substantial derivative $D\rho/Dt$ is clearly equal to the first three terms shown in equation (3.9).

3.1.2 Derivation of the Instantaneous Momentum Equation

Equations (3.2), (3.3), and (3.4) can be derived using the similar control volume technique as in the previous subsection. Newton's second law of motion states the acceleration vector, \mathbf{a} in $\mathbf{F} = m\mathbf{a}$ is applied to any small piece of the fluid and as stated

previously requires the use of a substantial derivative. Both pressure and viscous stresses are relatable to surface forces acting on a fluid according to Newton's second law in classical context. Newton's second law results in a summation of forces equal to the time rate of change of mass multiplied by the velocity vector with gradients of fluxing momenta, body forces, stresses, and pressure summed to impart their contributions. To conserve momentum, the momentum equation is balanced and gives a resulting conservation law of momentum. The term shown in (3.10) is responsible for the x -momentum leaving to the right of the control area and the term (3.11) is responsible for the excess rate of momentum leaving the control area in the y -direction.

$$(3.10) \quad \frac{\partial(\rho u^2)}{\partial x} \Delta x \Delta y$$

$$(3.11) \quad \frac{\partial(\rho v u)}{\partial y} \Delta x \Delta y$$

Summing the components of fluxing momenta in the x -direction returns the momentum flux flowing into the control element per unit area, as stated in equation (3.12).

$$(3.12) \quad \rho u^2 \Delta y + \rho v u \Delta x - \left(\left[\rho u^2 + \frac{\partial \rho u^2}{\partial x} \Delta x \right] \Delta y + \left[\rho v u + \frac{\partial \rho v u}{\partial y} \Delta y \right] \Delta x \right) = \\ - \left[\frac{\partial \rho u^2}{\partial x} \Delta x \Delta y + \frac{\partial \rho v u}{\partial y} \Delta y \Delta x \right]$$

The forces that act on the control area additionally include a normal hydrostatic force. This force must be included in addition to the normal stresses, τ_{xx} , the fluid experiences during motion. The summation in the x -direction elicits pressure forces which are accounted for in equation (3.13). Dividing both sides of equation (3.13) by the area $\Delta x \Delta y$ forms this excess stress in addition to normal stresses. The excess stress is normal to the surface in the x -direction represented mathematically as

$\frac{\partial(\tau_{xx})}{\partial x} \Delta x$ and likewise tangentially as $\frac{\partial(\tau_{yx})}{\partial y} \Delta y$. Gravitational body force affects are accounted for by equation (3.14) with ρg_x and ρg_y being the x and y components respectively.

$$(3.13) \quad p\Delta y - \left(p \frac{\partial p}{\partial x} \right) \Delta y = - \frac{\partial p}{\partial x} \Delta x \Delta y$$

$$(3.14) \quad \rho \vec{F} \Delta x \Delta y = (\rho g_x + \rho g_y) \Delta x \Delta y$$

By neglecting other external physical affects such as radiation pressure and then equating terms, one can derive the momentum change per unit time for the x -component, i.e., the momentum transport equation in the x -direction as equation (3.15).

$$(3.15) \quad \frac{d\rho u}{dt} \Delta x \Delta y + \left(\frac{\partial \rho u^2}{\partial x} \Delta x \Delta y + \frac{\partial \rho v u}{\partial y} \Delta y \Delta x \right) \\ = - \frac{\partial p}{\partial x} \Delta x \Delta y + \frac{\partial \tau_{xx}}{\partial x} \Delta x \Delta y + \rho g_x \Delta x \Delta y$$

In 2D dividing the area through each side of equation (3.15) and taking the limit as both Δx and Δy approach the infinitesimal steps dx , dy reduces equation (3.15) to that of equation (3.16). Taking the partial derivative and rearranging terms in equation (3.16) yields equation (3.17). The fourth term on the left hand side of equation (3.17) is zero via the conservation of mass.

$$(3.16) \quad \frac{d\rho u}{dt} + \left(\frac{\partial \rho u^2}{\partial x} + \frac{\partial \rho v u}{\partial y} \right) = - \frac{\partial p}{\partial x} + \frac{\partial(\tau_{xx})}{\partial x} + \frac{\partial(\tau_{yx})}{\partial y} + \rho g_x$$

$$(3.17) \quad \rho \frac{du}{dt} + \rho \frac{\partial \rho u^2}{\partial x} + \rho \frac{\partial \rho v u}{\partial y} + u \left(-\frac{d\rho}{dt} + \frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} \right) \\ = -\frac{\partial p}{\partial x} + \frac{\partial(\tau_{xx})}{\partial x} + \frac{\partial(\tau_{yx})}{\partial y} + \rho g_x$$

Equation (3.18) is the final form of the x -component of conservation of instantaneous momentum. Equations (3.2), (3.3), and (3.4) constitute the more general cartesian 3D forms of (3.18) for the x , y , and z -components of the conservation of instantaneous momentum. The 3D forms introduced in (3.2), (3.3), and (3.4) are derived similarly with simple extension from the 2D derivation that was just performed, by including the z -spatial gradient terms.

$$(3.18) \quad \rho \frac{du}{dt} + \rho \frac{\partial u^2}{\partial x} + \rho \frac{\partial v u}{\partial y} = -\frac{\partial p}{\partial x} + \frac{\partial(\tau_{xx})}{\partial x} + \frac{\partial(\tau_{yx})}{\partial y} + \rho g_x.$$

3.1.3 Stress and Strain Relationships in Fluids

The stress exerted on a fluid is related to the strain rate and for Newtonian fluids this is a linear relationship stemming from the stress tensor provided in equation (3.19). The partial derivative $\frac{\partial u_k}{\partial x_k}$ sums the velocity derivatives over the index k and is multiplied by $(2/3\mu - \kappa)$ defined as u' in equation (3.19).

$$(3.19) \quad \tau_{ij} = -\mu' \frac{\partial u_k}{\partial x_k} \delta_{ij} + \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$$

The bulk viscosity is taken as zero, for ideal monotonic gases, therefore equation (3.18) can be expressed in more detailed form as (3.20). For incompressible flows mass conservation implies $\nabla \cdot \vec{V} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$, in 2D.

$$(3.20) \quad \rho \frac{du}{dt} + \rho \frac{\partial u^2}{\partial x} + \rho \frac{\partial v u}{\partial y} = -\frac{\partial p}{\partial x} + 2 \frac{\partial}{\partial x} \mu' \frac{\partial u_i}{\partial x_j} + \\ \frac{\partial}{\partial y} \mu' \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) - \frac{\partial}{\partial x} \mu' \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) + \rho g_x$$

$$(3.21) \quad \rho \frac{\partial \vec{V}}{\partial t} + \rho \overbrace{\vec{V}(\nabla \cdot \vec{V})}^{V_j \frac{\partial V_i}{\partial x_j}} = \rho \vec{F} - \nabla p - \nabla \cdot \mu \nabla \vec{V}$$

Using vector notation the conservation of instantaneous momentum for an incompressible fluid is shown in equation (3.21) and with the common substantial derivative in (3.22)

$$(3.22) \quad \rho \frac{D\vec{V}}{Dt} = \rho \vec{F} - \nabla p + \nabla \cdot (\mu \nabla \vec{V}).$$

3.2 Turbulent Flows and Reynolds Time Averaging

The realistic modeling of fluid flow requires an explanation of turbulence. Turbulent fluids are governed or approximated by complicated coupled systems of non-linear partial differential equations. The inherently 3D nature of turbulence is visible in fluid structures well known to be notoriously difficult to model. Topological fluid dynamics are highly observed at the kinematic level where the common continuous deformation of transported scalar and vector fields are prevalent [30], i.e., problems where stirring and mixing occur. The regional locations of viscous pockets in adequately high Reynolds number flows produces violent cascades of turbulent eddies. For these appreciably high Reynolds number flows there exists instabilities in the shear layer of the flow that are responsible for turbulent fluctuations. As result of these turbulent fluctuations a cascade forms various length scale eddies. The smaller eddies are responsible for the dissipation of turbulent kinetic energy and the larger scale eddies are more likened to scales similar in length to that of the mean flow. While the governing equations give rise to this turbulence, solving them directly is another field called Direct Numerical Simulations (DNS).

Reynolds was the first to derive the system of averaged turbulent-motion equations in 1895 [29]. The Reynolds averaging of Navier-Stokes equations is a process of separating the dynamical fields into slowly varying mean fields and rapidly changing turbulent components [29]. Turbulence in the case of dynamical systems is theorized to exhibit a single strange attractor that is fractal in structure [31]. The process of time averaging transient turbulent information numerically is representable as an average with some small error. Unfortunately, there is no way to find the analytical value of an upper bound on this error. By splitting the instantaneous variable of interest into the constitutive mean and fluctuating components, i.e., $\frac{\partial(\bar{u}_i+u'_i)}{\partial x_i} = 0$.

In summary, the observed behavior of turbulent conditions exhibit period doubling behavior, i.e., chaotic motion, thought to only have one strange attractor [32], although this is yet to be proven. Solving turbulent systems elicit the problem that the present determines the future, but the present does not always dictate the future in a non-chaotic way [33]. In response the engineer can choose alternative methods to solve turbulent systems, such as the solving the RANS equations. The equations of motion for cartesian coordinates representing an incompressible fluid, the RANS set of equations, are shown in (3.23), (3.24), and (3.25).

$$(3.23) \quad \frac{\partial \rho}{\partial t} + \frac{\partial \rho \bar{u}_i}{\partial x_i} = 0$$

$$(3.24) \quad \frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho \bar{u}_i u_j}{\partial x_j} = \frac{\partial}{\partial x_j} \left(\mu \frac{\partial s_{ij}}{\partial x_j} \right) - \frac{\partial \bar{p}}{\partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j}$$

$$(3.25) \quad \rho \frac{\partial \bar{u}_i}{\partial t} + \rho \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j} (\mu s_{ij} - \overline{\rho u'_i u'_j})$$

Equation (3.25) is the incompressible form of the momentum equation where s_{ij} is the strain rate tensor that accounts for the affects of molecular viscosity defined in equation (3.26) and the Reynolds stress tensor τ_{ij} defined by equation (3.27).

$$(3.26) \quad s_{ij} \equiv \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right)$$

$$(3.27) \quad \tau_{ij} \equiv \overline{\rho \bar{u}_i \bar{u}_j} = \overline{\rho (\bar{u}_i + u'_i) (\bar{u}_j + u'_j)} = \overline{\rho (\bar{u}_i \bar{u}_j + \bar{u}_i u'_j + u'_i \bar{u}_j + u'_i u'_j)} = \rho (\bar{u}_i \bar{u}_j + \overline{u'_i u'_j})$$

The system of equations now has 6 new unknowns introduced by the addition of the Reynolds stress tensor. For the 3D system there are now 10 unknowns. The closure problem arises from the introduction of the Reynolds stress tensor. Two-equation turbulence models take higher order moments to allow for closure coefficients, determined by analysis, to *close* the system. Considering the turbulent kinetic energy and taking the trace of Reynolds stress gives the form of equation (3.28). The turbulent viscosity term, $\mu_t = \rho c_u k^2 / \epsilon$ is introduced where ϵ is the rate of turbulent dissipation. A full derivation is not included but is given by time averaging the moment of velocity of the governing equations and subtracting the kinetic energy of the mean flow, this leaves only the turbulent kinetic energy.

$$(3.28) \quad \tau_{ij} \equiv \mu_t \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) - \frac{2}{3} \delta_{ij} k$$

The transport of turbulent kinetic requires terms that both produce and dissipate turbulent effects. The Boussinesq eddy viscosity conjecture of 1887 [34], states that the momentum transfer caused by turbulent eddies can be modeled with an eddy viscosity and effectively the Reynolds stress tensor is proportional to the trace-less mean strain rate tensor, equation (3.29). Equation (3.30) now assumes all velocities

are averaged, except where noted, and c_2 is a closure coefficient. The production term P_k in (3.30), is given by (3.31). The turbulent diffusion is $D_k = -\frac{\rho\mu_t}{\sigma_k} \frac{\partial k}{\partial x_j}$ and $\epsilon = \mu \overline{\frac{\partial u'_i}{\partial x_j} \frac{\partial u'_i}{\partial x_j}}$. The units of turbulent kinetic energy are length squared over time squared (l^2/t^2), and the units for specific dissipation are length squared over time cubed (l^2/t^3).

$$(3.29) \quad \tau_{ij} = \frac{2}{3} K \delta_{ij} - 2\mu s_{ij}$$

$$(3.30) \quad \rho \frac{\partial \vec{k}}{\partial t} + \rho(\vec{u} \cdot \nabla) \vec{k} = c_2 \mu_t \nabla^2 \vec{k} + \vec{P}_k - \nabla \vec{D}_k - \vec{\epsilon}$$

$$(3.31) \quad P_k = \tau_{ij} \frac{\partial u_i}{\partial x_j} = \left[\mu_t \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) - \delta_{ij} \frac{2}{3} \rho k \right] \frac{\partial u_i}{\partial x_j}$$

3.2.1 Two-Equation Shear-Stress Transport Closure Model

The improved Menter two-equation turbulence model or improved two-equation k - ω turbulence model [5] introduced the new Baseline (BSL) model. The BSL model was designed to give results similar to Wilcox k - ω without the strong dependency on arbitrary free-stream values [5]. The BSL model is identical to that of the Wilcox k - ω in the innermost 50% of the boundary layer and changes gradually to the Jones-launders k - ϵ model [35] in a blended k - ω formulation closer to the boundary-layer edge [5]. The second version of the model Menter introduced is based upon the BSL model, the SST model, and has the additional ability to account for the transport of the principal shear-stress in adverse pressure gradient boundary layers [5].

The original Jones-launders k - ϵ is transformed into the k - ω formulation. The difference between the original Jones-launders k - ϵ and the k - ω model is the additional cross diffusion terms appear in the ω equation and that closure constants are differ-

ent. Equations (3.32) and (3.33) are the original formulation of the k - ω model and equations (3.34) and (3.35) are the transformed k - ϵ equations. Equations (3.32) and (3.33) are multiplied by a blending function F_1 and equations (3.34) and (3.35) are multiplied by $(1 - F_1)$ and the corresponding equations of each set are added together to yield the new model, equations (3.36) and (3.37).

$$(3.32) \quad \frac{\partial \rho k}{\partial t} + \frac{\partial \rho u_j k}{\partial x_j} = P_k - \beta^* \rho \omega k + \frac{\partial}{\partial x_j} \left[(\mu + \sigma_{k_1} \mu_t) \frac{\partial k}{\partial x_j} \right]$$

$$(3.33) \quad \frac{\partial \rho \omega}{\partial t} + \frac{\partial \rho u_j \omega}{\partial x_j} = \gamma_1 P_\omega - \beta_1 \rho \omega^2 + \frac{\partial}{\partial x_j} \left[(\mu + \sigma_{\omega_1} \mu_t) \frac{\partial \omega}{\partial x_j} \right]$$

$$(3.34) \quad \frac{\partial \rho k}{\partial t} + \frac{\partial \rho u_j k}{\partial x_j} = P_k - \beta^* \rho \omega k + \frac{\partial}{\partial x_j} \left[(\mu + \sigma_{k_2} \mu_t) \frac{\partial k}{\partial x_j} \right]$$

$$(3.35) \quad \frac{\partial \rho \omega}{\partial t} + \frac{\partial \rho u_j \omega}{\partial x_j} = \gamma_2 P_\omega - \beta_2 \rho \omega^2 + 2\rho \sigma_{\omega^2} \frac{1}{\omega} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j} + \frac{\partial}{\partial x_j} \left[(\mu + \sigma_{\omega_2} \mu_t) \frac{\partial \omega}{\partial x_j} \right]$$

$$(3.36) \quad \frac{\partial \rho k}{\partial t} + \frac{\partial \rho u_j k}{\partial x_j} = P_k - \beta^* \rho \omega k + \frac{\partial}{\partial x_j} \left[(\mu + \sigma_k \mu_t) \frac{\partial k}{\partial x_j} \right]$$

$$(3.37) \quad \frac{\partial \rho \omega}{\partial t} + \frac{\partial \rho u_j \omega}{\partial x_j} = \gamma P_\omega - \beta \rho \omega^2 + 2\rho(1 - F_1) \sigma_{\omega^2} \frac{1}{\omega} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j} + \frac{\partial}{\partial x_j} \left[(\mu + \sigma_\omega \mu_t) \frac{\partial \omega}{\partial x_j} \right]$$

The equations (3.32) through equation (3.37) reference the Lagrangian derivative which is not identical to Menter's original paper [5], as his paper suffered typos. The equations (3.32) through equation (3.37) are been shown above in proper conservation form, this is consistent with Wilcox's versioning [2]. A blending function Φ is defined in equation (3.38), where $\Phi_1 \equiv$ is any constant in the original model, i.e., (σ_{k_1}, \dots) ,

$\Phi_2 \equiv (\sigma_{k_2}, \dots)$ being any constant in the transformed model, and $\Phi \equiv (\sigma_k, \dots)$ any constant in the new model.

$$(3.38) \quad \Phi = F_1 \Phi_1 + (1 - F_1) \Phi_2$$

Equation's (3.39), (3.40), (3.41) follow similarly to previous and the turbulent eddy viscosity is computed from equation (3.42).

$$(3.39) \quad P = \tau_{ij} \frac{\partial u_i}{\partial x_j}$$

$$(3.40) \quad \tau_{ij} = \mu_t \left(2S_{ij} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) - \frac{2}{3} \rho k \delta_{ij}$$

$$(3.41) \quad S_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$$

$$(3.42) \quad \mu_t = \frac{\rho \alpha_1 k}{\max(\alpha_1 \omega, \Omega F_2)}$$

The set of constants corresponding to the Wilcox equations are, Set 1 (Wilcox): $\sigma_{k_1} = 0.5$, $\sigma_{\omega_1} = 0.5$, $\beta_1 = 0.0750$, $\beta^* = 0.09$, $\kappa = 0.41$, $\gamma_1 = \beta/\beta^* - \sigma_{k_2} \kappa^2 / \sqrt{\beta^*}$ and Set 2 (Jones-Launder): $\sigma_{k_2} = 1.0$, $\sigma_{\omega_1} = 0.856$, $\beta_2 = 0.0828$, $\beta^* = 0.09$, $\kappa = 0.41$, and $\gamma_2 = \beta/\beta^* - \sigma_{\omega_2} \kappa^2 / \sqrt{\beta^*}$. Each of the constants is a blend of an inner and outer constant, blended with equation (3.38).

The additional functions are provided in equations (3.43) through (3.47), where ρ is the density, $\nu_t = \mu_t/\rho$ is the turbulent kinematic viscosity, and μ is the molecular dynamic viscosity. Of particular interest and one of the more difficult aspects of integrating the SST model into an FEM code, is d the distance from the field point

to the nearest wall used in equations (3.43) through (3.47).

$$(3.43) \quad F_1 = \tanh(\arg_1^4)$$

$$(3.44) \quad \arg_1 = \min \left[\max \left(\frac{\sqrt{k}}{\beta^* \omega d}, \frac{500\nu}{d^2 \omega} \right), \frac{4\rho\sigma_{\omega^2} k}{CD_{k\omega} d^2} \right]$$

$$(3.45) \quad CD_{k\omega} = \max \left[2\rho\sigma_{\omega^2} \frac{1}{\omega} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j}, 10^{-20} \right]$$

$$(3.46) \quad F_2 = \tanh(\arg_2^2)$$

$$(3.47) \quad \arg_2 = \max \left(2 \frac{\sqrt{k}}{\beta^* \omega d}, \frac{500\nu}{d^2 \omega} \right)$$

Here, $\Omega = \sqrt{2W_{ij}W_{ij}}$ is the vorticity magnitude, with W_{ij} defined in (3.48) below.

$$(3.48) \quad W_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} \right)$$

The boundary conditions are implemented the same as the existing k - ω and will be shown in the following sections herein that develop the PCS solution method.

3.2.2 Turbulent Energy Transport

In computational fluid dynamics the turbulent kinetic energy (TKE) is the mean kinetic energy per unit mass associated with the eddies that develop under turbulent conditions [36]. Assuming incompressible flow the TKE equation is given in (3.49). It is common to define a variable k , the average fluctuating kinetic energy per unit mass. Each term in equation (3.49) has a distinct role in the overall kinetic energy balance.

It is important to understand which mechanisms are relevant to certain flow regimes. Similar to how the transport of fluctuating momenta influences the transport of mean momentum, so too does the mean flow feed kinetic energy to the turbulent field [37].

(3.49)

$$\underbrace{\frac{\partial k}{\partial t}}_{\text{Local derivative}} + \underbrace{\bar{u}_j \frac{\partial k}{\partial x_j}}_{\text{Advection}} = - \underbrace{\frac{1}{\rho_o} \frac{\partial \overline{u'_i p'}}{\partial x_i}}_{\text{Pressure diffusion}} - \underbrace{\frac{1}{2} \frac{\partial \overline{u'_j u'_j u'_i}}{\partial x_i}}_{\text{Turbulent transport}} + \underbrace{\nu \frac{\partial^2 k}{\partial x_j^2}}_{\text{Molecular viscous transport}} - \underbrace{\overline{u'_i u'_j} \frac{\partial \bar{u}_i}{\partial x_j}}_{\text{Production}} - \underbrace{\nu \frac{\partial \overline{u'_i} \partial \overline{u'_i}}{\partial x_j \partial x_j}}_{\text{Dissipation}} - \underbrace{\frac{g}{\rho_o} \overline{\rho' u'_i} \delta_{i3}}_{\text{Buoyancy flux}}$$

Conservation of energy is commonly described by equation (3.50), relating the time rate of change of energy to work done on the system, the rate of advected energy through the boundaries, and the diffused energy into the system via molecular diffusion or conduction: *system*, in this description is referring to fluid control area or volume in 2D or 3D, respectively.

$$(3.50) \quad d\dot{E}_i = d\dot{W} + d\dot{E}_c + d\dot{E}_d$$

3.2.3 Derivation of the Enthalpy Transport Equation

Summing contributions into and out of the control volume, in the x -direction from internal energy, kinetic energy, body forces, pressure, heat flux or radiation energy density changes, advection, and stress forces, similar to the derivation of the momentum transport equation, one can derive an equation for the total enthalpy or energy transported into and out of the control volume [38]. The net rate of internal energy flowing into the control volume is equal to $u \frac{\partial \rho e}{\partial x} + u \frac{\partial \rho e}{\partial y}$. Equation (3.51) shows the

total change in energy flux in the control volume.

$$(3.51) \quad \left(\frac{\partial u \rho \left(e + \frac{\widehat{V^2}}{2} \right)}{\partial x} + \frac{\partial v \rho \left(e + \frac{V^2}{2} \right)}{\partial y} \right) = \frac{\partial u_i \rho \left(e + \frac{V^2}{2} \right)}{\partial x_i} = \frac{u_i \rho e}{\partial x_i} + \frac{\partial u_i \rho \frac{V^2}{2}}{\partial x_i}$$

Because the work done on the fluid includes pressure forces, body forces and stress forces the rate of work on the control volume is given by equation (3.52). The rate of stress forces contributing to work done on the fluid acting in the direction normal to the surface area of the volume are shown in tensor form as equation (3.52) and similarly in the tangential direction by equation (3.53). The rate of work due to stress is incorporated into the dissipation term in equation (3.49) as shown in (3.54), and for a Newtonian fluid in equation (3.55).

$$(3.52) \quad - \left(\frac{\partial(pu)}{\partial x} + \frac{\partial(pv)}{\partial y} \right) = - \left(u \frac{\partial p}{\partial x} + v \frac{\partial p}{\partial y} \right) - \left(p \frac{\partial u}{\partial x} + p \frac{\partial v}{\partial y} \right)$$

$$(3.53) \quad - \frac{\partial(u_i \sigma_{ij} \delta_{ij})}{\partial x} = - \frac{\partial(u \sigma_{xx})}{\partial x} - \frac{\partial(v \sigma_{yy})}{\partial x}$$

$$(3.54) \quad \phi = \sigma_{ij} \frac{\partial u_i}{\partial x_j} = \sigma_{xx} \frac{\partial u}{\partial x} + \sigma_{xy} \frac{\partial u}{\partial y} + \sigma_{yx} \frac{\partial v}{\partial x} + \sigma_{yy} \frac{\partial v}{\partial y}$$

$$(3.55) \quad \phi = 2\mu \left(\frac{\partial u}{\partial x} \right)^2 + 2\mu \left(\frac{\partial v}{\partial y} \right)^2 + \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)^2 + \left(\beta - \frac{2}{3}\mu \right) \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right)^2$$

The heat flux diffused per unit area into the control volume is $-\left(\frac{\partial \dot{q}_x}{\partial x} + \frac{\partial \dot{q}_y}{\partial y} \right)$ and the rate of work done due to gravitational body force is $\rho f \cdot V dx dy = \rho u g_x + \rho v g_y$. By Fourier's Law for heat flux, i.e., $\dot{q} = \kappa \frac{\partial T}{\partial x}$, and adding equations (3.51), (3.52), (3.53), (3.54), (3.55), while neglecting radiation energy density, and the divergence of radiation flux

one derives the transport total energy equation (3.56). Applying the rule for taking derivatives of a product to the second term on the right side of equation (3.56) and applying mass conservation the total energy equation is (3.57).

$$(3.56) \quad \frac{d\rho E_t}{dt} = -\left(\frac{\partial\rho e}{\partial x} + v\frac{\partial\rho e}{\partial y}\right) - \frac{\partial u\rho\frac{V^2}{2}}{\partial x} - \frac{\partial v\rho\frac{V^2}{2}}{\partial y} + \frac{\partial}{\partial x}\left(\kappa\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(\kappa\frac{\partial T}{\partial y}\right) \\ - p\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) - \left(u\frac{\partial p}{\partial x} + v\frac{\partial p}{\partial y}\right) + \phi + \rho u g_x + \rho v g_y$$

$$(3.57) \quad \rho\frac{DE_t}{Dt} + p\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) + \left(u\frac{\partial p}{\partial x} + v\frac{\partial p}{\partial y}\right) = \\ \frac{\partial}{\partial x}\left(\kappa\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(\kappa\frac{\partial T}{\partial y}\right) + \phi + \rho u g_x + \rho v g_y$$

With the thermodynamic property for enthalpy defined as the internal energy of the system plus the product of pressure and volume $h = e + p/\rho$ [39] the rate of total enthalpy change per unit time is shown as equation (3.58). Applying the relationship for mass consistency, $D\rho/Dt = \rho\partial v_i/\partial x_i$, to equation (3.58) we get the enthalpy equation (3.59).

$$(3.58) \quad \frac{Dh}{Dt} = \frac{De}{Dt} - \frac{Dp}{Dt} + \frac{p}{\rho^2}\frac{D\rho}{Dt}$$

$$(3.59) \quad \rho\frac{Dh}{Dt} = \left[\left(\kappa\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(\kappa\frac{\partial T}{\partial y}\right)\right] + \frac{\partial p}{\partial t} + \phi + \rho u g_x + \rho v g_y$$

From the symmetry of second derivatives and application of Euler's reciprocity relation to the thermodynamic characteristic functions [38], Maxwell derived his relation $dh = c_p dT + (1/\rho)(1 - \beta T)dp$ where β is the coefficient of thermal expansion we can derive equation (3.60). Equation (3.60) can be simplified to equation (3.61) assuming incompressible flow using the ideal gas law, $h = \rho c_p T$, providing the final form of the

conservation of energy equation (3.62).

$$(3.60) \quad \frac{c_p dT}{dt} + \frac{\beta T}{\rho} \frac{dp}{dt} = \left(u \frac{\partial e}{\partial x} + v \frac{\partial e}{\partial y} \right) + \frac{1}{\rho} \left[\left(\kappa \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(\kappa \frac{\partial T}{\partial y} \right) \right] + \frac{1}{\rho} \left(u \frac{\partial p}{\partial x} + v \frac{\partial p}{\partial y} \right) + \frac{1}{\rho} \phi$$

$$(3.61) \quad \rho c_p \frac{\partial T}{\partial t} + \rho c_p u_j \frac{\partial T}{\partial x_j} + (1 - \beta T) \left[\frac{\partial P}{\partial t} + u_j \frac{\partial P}{\partial x_j} \right] = \frac{\partial}{\partial x_j} \left(\kappa \frac{\partial T}{\partial x_i} \right)$$

$$(3.62) \quad \rho c_p \frac{\partial T}{\partial t} + \rho c_p u_j \frac{\partial T}{\partial x_j} = \frac{\partial}{\partial x_j} \left(\kappa \frac{\partial T}{\partial x_i} \right)$$

3.2.4 Derivation of the Turbulent Energy Transport Equation

The instantaneous rate of change for thermal energy of an incompressible fluid provided in equation (3.62) can be separated into mean and fluctuating components similar to the derivation of the momentum equation [40]. Upon averaging an equation for thermal transport can be shown in equation (3.63) where we introduce the turbulent heat flux as the second term on the right hand side. The representation of turbulent heat flux is provided in similar manner to that of the turbulent inertial term shown in equation (3.64), where ϵ_h is defined as the turbulent eddy viscosity for enthalpy transport and Pr_t is introduced as the turbulent Prandtl number [38]. Plugging equation (3.64) into (3.63) and assuming averaged values for temperature and velocity we can drop the average over bar notation to get equation (3.65).

$$(3.63) \quad \rho c_p \frac{\partial \bar{T}}{\partial t} + \rho c_p u_j \frac{\partial \bar{T}}{\partial x_j} = \frac{\partial}{\partial x_j} \left(\kappa \frac{\partial \bar{T}}{\partial x_i} \right) - \frac{\partial \bar{T} \bar{u}'}{\partial x_i}$$

$$(3.64) \quad \frac{\partial \bar{T} \bar{u}_{j'}}{\partial x_j} = \epsilon_h \frac{\partial \bar{T}}{\partial x_j} = \frac{\mu_t}{Pr_t} \frac{\partial \bar{T}}{\partial x_j}$$

$$(3.65) \quad \rho c_p \frac{\partial T}{\partial t} + \rho c_p u_j \frac{\partial T}{\partial x_j} = \frac{\partial}{\partial x_j} \left(\left(\frac{\mu_t}{Pr_t} + \kappa \right) \frac{\partial T}{\partial x_j} \right)$$

$$(3.66) \quad \frac{de}{dt} = - \left(u \frac{\partial e}{\partial x} + v \frac{\partial e}{\partial y} \right) + \frac{1}{\rho} \left[\frac{\partial}{\partial x} \left(\left(\frac{\mu_t}{Pr_t} + \kappa \right) \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(\left(\frac{\mu_t}{Pr_t} + \kappa \right) \frac{\partial T}{\partial y} \right) \right] + \frac{1}{\rho} \left(u \frac{\partial p}{\partial x} + v \frac{\partial p}{\partial y} \right) + \frac{1}{\rho} \phi$$

Equation (3.65) is the final form of the turbulent energy transport equation, and equation (3.66) is the equivalent internal energy only equation.

3.3 The Finite Element Method (FEM)

The finite element method (FEM) is a proven numerical method for solving many types of problems in engineering and useful for application to mathematical methods in physics. The FEM segments a domain into elements, then connects the elements with nodes which result in solvable system of simultaneous algebraic equations. The term finite in FEM is in reference to the number of degrees of freedom, which is *-finite-* unlike the continuous infinite representation. The FEM is rooted in applications to solid mechanics problems, yet the FEM process for applications to fluid dynamics began with the mathematical development of the weighted-residual method (WRM) form of variational statements to non-linear equations describing fluid flow. It is the relationship that arose from the WRM that the foundational set of mathematics the FEM relies on began. The WRM was by introduced by Galerkin in 1915 [41]. The FEM is widely regarded as clever and works well, producing accurate solutions to difficult partial differential equations. This section outlines the

basics of WRM and the specifics of the Galerkin method with the specific application to solving Navier-Stokes equations for viscous incompressible fluid flows in 3D. The base LANL FEARCE code allows for h -adaptation and runs in parallel with multiple processors using message-passing-interface (MPI) on decomposed computational domains. To simplify the implementation of the SST turbulence model into the existing FEARCE code a serial version without h -adaptation was used. FEARCE employs a Petrov-Galerkin up-winding time increment adjustment for numerical stability and Legendre integration that will be developed later in subsequent sections. What follows is the Galerkin WRM FEM solution of the Navier-Stokes and species transport equations for turbulent incompressible fluid flows. Specifically, the time dependent Poisson formulation for the pressure form if the governing equation is solved without h -adaptation.

3.3.1 The Weighted Residual Method

The WRM, norms and bilinear formulation are developed in this section.

Often solutions to structural mechanics problems often rely on the Raleigh-Ritz variational process that introduces the functional. The functional is a minimization of energy applied where the integrand is the potential for stress [42]. The WRM also includes another process that finds the orthogonal projection of the residual equation. Functional analysis includes the subset of these type of projection techniques with goal of WRM to find an easy method for developing the variational statement, or often called weak statement.

Typically FEM solutions to second order partial differential equations start with an application of Green's theorem [43] which gives a way to produce the weak formulation of the governing residual equation by reducing a second order partial differential equation to a first order equation. Depending on the problem, Green's theorem relates a line integral around a closed curve and a double integral over some plane region

bounded by the closed curve or volume integrals involving the Laplacian to surface integrals. For someone who is versed in field equations, vector calculus or has worked in applied electrodynamics Green's contributions are very familiar. However, it is interesting to note Green's original essay [43] did not actually derive Green's Theorem, rather he derived a form of the divergence theorem and only later did Cauchy actually publish, "Green's theorem", albeit without proof [44].

Consider a problem domain that is a discrete system. Then the FEM seeks to minimize the residual error over the domain of this discrete system. Using a Taylor series to expand the heat conduction equation with constant conductivity with an internal heat source in an isotropic material we can derive concepts needed for WRM. The residual \mathbf{R} is shown in (3.67) for the heat equation. The approximated temperature can be represented by the function \hat{T} shown in (3.68) as a polynomial expansion of order n . The terms N_i and T_i in equation (3.68) are the Galerkin weights and nodal values of the trial temperature, respectively.

$$(3.67) \quad \mathbf{R}(\hat{T}, x_i) \equiv -K\nabla^2\hat{T} - Q$$

$$(3.68) \quad \hat{T}(x_i) = \sum_{i=1}^n T_i N_i$$

The FEM seeks to minimize equation (3.67) over the domain [45]. Equation (3.67) is multiplied by a weighting function, \mathbf{W}_i so that on average it will be zero when integrated over the domain. Equation (3.69) shows the integral of the inner product of the weighting function and residual over the domain Ω which is defined as a Hilbert space projection.

$$(3.69) \quad \int_{\Omega} \mathbf{W}_i \cdot \mathbf{R}(\hat{T}, x_i) d\Omega = 0$$

In the orthogonal projection the Galerkin approximation is a weak formulation which can be stated mathematically as $P : \text{find } u \in V, \text{ such that } a(u, v) = f(v), \forall v \in V$, where V is a Hilbert space, $a(\cdot, \cdot)$ is an inner-product in V , and $f(\cdot)$ is a continuous linear form in V . The problem can be stated for computational purposes as the need to discretize the continuous problem P to a finite dimensional problem. In this development P would be the heat equation and on a subspace of dimension $N : V_h \subset V$, and the Galerkin WRM restricts P to V_h , and leads to a finite dimensional problem $P_h : \text{to find } u_h \in V_h, \text{ such that } a(u_h, v_h) = f(v_h), \forall u_h \in V_h \subset V$. There is an important convergence method built into equation (3.69). The convergence method can be seen by letting $V_h \rightarrow V$ increase the dimension N of V_h , such that the convergence of $P_h \rightarrow P$ would be expected to produce the optimal approximation to P . Therefore, equation (3.69) can be stated in more general mathematical terms as $a(u - u_h, h) = 0, \forall u_h \in V_h$. This implies the approximation u_h , often denoted by a hat (\hat{u}), is nothing more than the orthogonal projection of u on to the subspace V_h with respect to the inner-product $a(\cdot, \cdot)$. The FEM solution $u_h \in V_h \subset V$ is the projection of $u \in V$ in the bilinear form a onto the subspace V_h and is because of this, the optimal solution approximation of u in V_h . This makes u_h the optimal approximation of u because $\|u - h_h\|_a = \min_{v_h \in V_h} \|u - v_h\|_a$ is the closest distance wherein the energy norm is $\|u\|_a = \sqrt{a(u, u)}$.

The Hilbert space as defined in equation (3.69) is a complete inner-product vector space and as such includes piecewise differentiable functions on the order of the weights, W . In other words when the integrand of equation (3.69) is set to zero it is an orthogonal projection of the residual that seeks to find the values of T that meet satisfy the equivalence statement. When equation (3.69) is applied over a domain that is discretized into finite control volumes the resulting set of algebraic equations are solved for the unknown variables, for the heat equation example these would be the values of T and the nodes of the $\{T_i\}$.

3.3.2 Error and Convergence Estimations (FEM)

The error and convergence estimations of the inner product projections of equation (3.69) are calculated using a vector space of functions equipped with a norm that is a combination of L^p -norms of the given function and its derivatives up to some specified order referred to as a Sobolev space [46]. In the generalization of the concept of a derivative of a function, the weak derivatives are understood to make the space complete, a Banach space. Sobolev spaces are used in the FEM to define function spaces with derivatives involving integral forms such that sufficiently many derivatives can be used for some application domain. The Sobolev space of order m consists of the functions that are square integrable, e.g, if there is no derivative then a relevant Sobolev space would be $H^0(\Omega) = L^2(\Omega)$. The functions that are belonging to $L^2(\Omega)$ including their partial derivatives to order m that have an inner product are also included in this Sobolev space. This implies the Sobolev spaces are subsets of the Hilbert spaces [47], $H^m(\Omega) \subset L^2(\Omega)$.

The Sobolev spaces can be defined mathematically as in equation (3.70). The derivative terms $D^\alpha u$ in (3.70) are the weak derivatives or variational derivatives, while the Sobolev inner product space is defined by equation (3.71) and subspace of the Sobolev space $H_o^m(\Omega)$ defined by equation (3.72). The Sobolev space $H_o^m(\Omega)$ defined by equation (3.72) contains derivatives that vanish on the boundaries found in boundary value problems commonly solved for momentum and heat transfer problems. Equation (3.72) allows for semi-norms commonly associated with weak topology. Poincare'-Friedrichs inequality [48] provides the equivalency of norms in the subspaces $H_o^m(\Omega)$ and $H^m(\Omega)$ [45] as shown mathematically in equation (3.73) where the semi-norm $|\cdot|_m^2$ is defined as $\sum_{|\alpha|=m} \int_{\Omega} |D^\alpha u| d\Omega$ and the norm, $\|\cdot\|_{L^2}^2$ is defined as $\|u\|_{H_o}^2 = \|u\|_{L^2}^2 = \int_{\Omega} u^2 d\Omega$ with C being dependent on the discretization of the domain Ω along with the curvature of the exact solution and the degree of the

polynomial interpolation [45].

$$(3.70) \quad H^m(\Omega) = \{u \in L^2(\Omega) : D^\alpha u \in L^2(\Omega) \forall \alpha \text{ such that } |\alpha| \leq m\}$$

$$(3.71) \quad (u, v)_{H^m} = \int_{\Omega} \sum_{|\alpha| \leq m} (D^\alpha u)(D^\alpha v) dx \quad \forall u, v \in H^m(\Omega)$$

$$(3.72) \quad H_o^m(\Omega) = \{u \in H^m(\Omega) : u = 0 \in \Gamma\}$$

$$(3.73) \quad \|u\|_{L^2(\Omega)}^2 \leq C|u|_m^2$$

To find the exactness of an approximated solution mathematical procedures must be employed as part of variational methods with the associated norms to determine the exactness of a solution. The solution to any projection is as exact as the polynomial's interpolation will allow [45]. For example, the exactness of the equation 3.67 and it's bilinear representation are defined by the norm $\|T - \hat{T}_h\|_{H^0} \leq ch^p$, where c is a constant dependent on the domain, h is the spacing, length or size of an element, and p is the order of the approximating polynomial.

3.3.3 Galerkin Method of Weighted Residuals

By choice the interpolating functions can be made to equal the weight as in equation (3.74), a process referred to as the Galerkin weighted residual method (WRM). Choosing the weights to equal the interpolating functions in this way allows for one set of basis functions. The problem domain is integrated over the integral equations

particular to the elements comprising the domain, producing a system of equations.

$$(3.74) \quad \mathbf{W}_i = \mathbf{N}_i$$

Using Green's theorem we can show the variational form of the integral equations that weaken the original heat equation example into first order differentials, equation (3.75). Surface fluxes are expressed automatically by this process, as shown in equation (3.76), where the flux of energy (q) is passed through the boundary surfaces of the domain. The first term on the left hand side of equation (3.75) is the bilinear form of the governing equation and the inner product it produces allows for the determination of a weak statement that includes the surface fluxes at the boundaries, equation (3.77).

$$(3.75) \quad \int_{\Omega} \frac{\partial \mathbf{W}_i}{\partial x_i} \left(k \frac{\partial \hat{\mathbf{T}}}{\partial x_i} \right) d\Omega - \int_{\Omega} \mathbf{W}_i Q d\Omega + \int_{\Gamma} \mathbf{W}_i \left(-k \frac{\partial \hat{\mathbf{T}}}{\partial \mathbf{n}} \right) d\Gamma = 0$$

$$(3.76) \quad q = \int_{\Gamma} \mathbf{W}_i \left(-k \frac{\partial \hat{\mathbf{T}}}{\partial \mathbf{n}} \right) d\Gamma$$

$$(3.77) \quad b(T, W) = \int_{\Omega} \frac{\partial \mathbf{W}_i}{\partial x_i} \left(k \frac{\partial \hat{\mathbf{T}}}{\partial x_i} \right) d\Omega$$

Four popular textbooks used to introduce the Galerkin WRM can be outlined. The first by Reddy [49], makes use of the weighted residuals via variational calculus and development of the strong to weak formulation, but seems to lack some important aspects of WRM and there is no clear connection made between direct stiffness methods, shape function utilization and WRM. Logan [50], develops a method of analysis using sparse explanation of WRM and similar to Reddy does not draw clear connections to shape function utilization and limits the problem scope to 1D bar elements. Two

series of texts by Pepper and Heinrich, an introductory text [51], and an intermediate text [52], do well in developing a presentation of the WRM and Galerkin formulation, trial and test function spaces, error estimates and the stability analysis necessary for applications to fluid flow and heat transfer problems.

3.3.4 FEM Computational Space: Global Coordinates and Natural Coordinates

The weakened derivative form of the dependent variable still requires the choice of an appropriate weight function. The first order Lagrangian interpolating polynomial can be used as the eigenfunction, and the process of interpolating functions and relating the natural coordinates to the computational space follows in this section. Using normalized transformations for interpolating functions along with the approximation of the Galerkin WRM also requires a choice in elements. Isoparametric elements or brick rectangular elements offer advantages over triangular elements, but can be restrictive in application. To build quadrilateral elements that can handle curved boundaries, one must overcome two critical difficulties [52] 1) finding shape functions for general quadrilateral elements in a global frame of reference, because the shape functions will no longer be bilinear and 2) integrating over arbitrary quadrilateral regions. These two difficulties can be overcome by using transformations of coordinates. The amazing uniqueness FEM offers is that coordinate transformations can be performed locally on arbitrary geometries, but we will also need numerical integration to do so. The use of numerical integration is necessary because the integrals found in the weak WRM are no longer simple polynomials [52]. The linear system of equations is assembled from individual elements in the problems geometric domain, which is transformed to an element of normalized length in the computational domain. A one to one and onto mapping is performed between the geometric and computational domain. This transformation allows one to use generic interpolating functions, there are commonly

referred to as shape function in the FEM. The shape functions for isoparametric tri-linear hexahedral elements are shown in equation (3.78). The derivatives of the shape functions are shown in equation (3.79), and are obtained by simple application of derivative rules. The derivatives of a global quantity in the computational domain can be found from the shape functions and a matrix of all first-order partial derivatives, the Jacobian, equation (3.80). The Jacobian and it's application are shown in equation (3.81).

$$(3.78) \quad \begin{bmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \\ N_5 \\ N_6 \\ N_7 \\ N_8 \end{bmatrix} = \frac{1}{8} \begin{bmatrix} (1 - \xi)(1 - \eta)(1 - \zeta) \\ (1 + \xi)(1 - \eta)(1 - \zeta) \\ (1 + \xi)(1 + \eta)(1 - \zeta) \\ (1 - \xi)(1 + \eta)(1 - \zeta) \\ (1 - \xi)(1 - \eta)(1 + \zeta) \\ (1 + \xi)(1 - \eta)(1 + \zeta) \\ (1 + \xi)(1 + \eta)(1 + \zeta) \\ (1 - \xi)(1 + \eta)(1 + \zeta) \end{bmatrix}$$

$$(3.79) \quad \frac{\partial N_i}{\partial \xi} = \frac{\partial N_i}{\partial x_j} \frac{\partial x_j}{\partial \xi} \delta_{ij}$$

$$(3.80) \quad \mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix}$$

$$(3.81) \quad \begin{bmatrix} \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} \\ \frac{\partial}{\partial \zeta} \end{bmatrix} = \mathbf{J} \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{bmatrix}$$

The global and computational domain derivatives of a function are given by equation (3.82). Equation (3.82) is a one-to-one and onto the natural coordinate domain, and notably a non-singular transformation. Each element and that elements corresponding derivatives of variables, otherwise known as trial functions in the global domain, are shown in equation (3.83), and the inverse Jacobian is shown in equation (3.84).

$$(3.82) \quad \begin{bmatrix} \frac{\partial N_j}{\partial \xi} \\ \frac{\partial N_j}{\partial \eta} \\ \frac{\partial N_j}{\partial \zeta} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \begin{bmatrix} \frac{\partial N_j}{\partial x} \\ \frac{\partial N_j}{\partial y} \\ \frac{\partial N_j}{\partial z} \end{bmatrix}$$

$$(3.83) \quad \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \sum f_i \frac{\partial N_j}{\partial \xi} \\ \sum f_i \frac{\partial N_j}{\partial \eta} \\ \sum f_i \frac{\partial N_j}{\partial \zeta} \end{bmatrix}$$

$$(3.84) \quad \mathbf{J}^{-1} = \frac{1}{|\mathbf{J}|} \begin{bmatrix} \frac{\partial x}{\partial \xi} & -\frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ -\frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & -\frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & -\frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix}$$

3.3.5 Numerical Integration

The nodes that comprise the computational domain are sampling points for integration. The transformation to a normalized computational domain and use of Gauss-Legendre quadrature for numerical integration is applied to the integral equations [51].

Integration is commonly referred to as quadrature in 1D and cubature for higher dimensions, FEM common terminology is the use of quadrature to refer to numerical approximations. The limits are not changed and in 3D the general integration is performed by the algorithm presented in equation (3.85), where w_i are the Gauss weights evaluated at Gauss points ξ, η, ζ . In 3D eight weighting points are applied and higher accuracy is possible by use of more integrating points.

$$(3.85) \quad \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 f(\xi, \eta, \zeta) |\mathbf{J}| d\xi = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n w_i w_j w_k f(\xi, \eta, \zeta) \mathbf{J}(\xi, \eta, \zeta)$$

It should be noted the use of isoparametric elements is essential to making the FEM a useful numerical tool. Because the transformations are local, i.e. element-by element they free one from having to deal with the domain geometry globally and allow the discretization of a geometry with any degree of complexity.

Chapter 4

Predictor Corrector Split (PCS)

Solution Method

A Predictor-Corrector Split (PCS) projection method in a FEM frame work has been developed for application to combustion modeling, Carrington [53]. The software and the algorithms the software uses are part of the newly invented LANL KIVA development code named Finite Elements for Accurate, Robust, Continuum, Engineering (FEARCE). FEARCE is combined with KIVA spray and chemistry models and uses a moving marker Arbitrary-Lagrangian Eulerian (ALE) type algorithm, and is considered the next version of software from LANL in a long history of codes dating back to the 1980's. The FEARCE code allows for both h -adaptation and parallel execution on decomposed computational domains. The specifics of h -adaptation and parallel execution or compressible solver options included in the FEARCE code suite are not included here, rather implementation of the SST turbulence model that has been added to the existing and installed k - ω and Large-Eddy Simulation (LES) turbulence models will be explained and developed. The serial PCS solution method for turbulent reactive incompressible flow regimes and the newly installed SST turbulence model using the Navier-Stokes solver follows in subsequent sections of this chapter.

[53]

The PCS solver formulation follows similarly to two other general algorithm formulations for incompressible flow by Zienkiewicz and Codina [54] and Zienkiewicz et al. [55]. Both [54] and [55] make use of a characteristic FEM with a fractional-step method. The primary difference between these systems is the use of the Petrov-Galerkin (P-G) weighting for advection [56], that is similar to the Streamline Upwinding Petrov-Galerkin (SUPG) [57], in place of the characteristic terms shown in [55]. The modified method now works particularly good for changes in implicitness that range from closely implicit to full explicit. When used in full explicit mode the code is optimal for parallel high performance computing (HPC) on multi-core processors. The parallelized code has been shown to have super-linear scaling in the characteristic based split (CBS) stabilization.

The discretization of the PCS system is conservative for both the compressible and incompressible momentum transport equations and the other transport equations used for reactive flow [53]. Having the governing equations in conservative form for the incompressible solver is not common and is a direct result of designing the system to handle all flow regimes. The PCS method used is more commonly used in a semi-implicit fashion [53]. If a semi-implicit solve is wanted, the pressure will range from implicit to explicit. The FEM method, in general for equal-order type projection approximation is outlined in detail in Pepper and Carrington [58]. The solution to the Navier-Stokes equations is similar to Carrington et. al [4] and Wang et. al [59], however the equation set is altered to span all flow regimes in the PCS algorithm, as described in Carrington [60]. The PCS system solves the set of turbulent Navier-stokes equations in a multi-component formulation also as described in [60].

4.1 Projection Algorithms

A major difficulty for the numerical simulation of incompressible flows is that the velocity and the pressure are coupled by the incompressibility constraint [61]. Projection methods are used to solve Navier-Stokes equations, and are known as an effective way of numerically solving time-dependent incompressible CFD problems. Although many have contributed to the advancement of projection type algorithms or methods the concept was originally introduced by Chorin in 1967 [62] and also Témam [63]. Later Gresho [64], and others [65], [66] continued to refine the formulation and address concerns on the validity of excluding the pressure gradient term from the auxiliary velocity, i.e. uncoupling the velocity from the pressure. The uncoupling of these two variables is what gives the projection method efficiency and allows the system of equations to be separately solved. In 3D the system is effectively 6th order, Gresho [64] states, “the velocity and pressure are not really meant to be uncoupled for viscous incompressible flow”. Gresho is describing how a flow is driven by advective and viscous terms in the momentum equation and how the velocity and pressure co-evolve such that the velocity remains divergence free. However, in projection methods an auxiliary velocity is used that is allowed to evolve absent of the influence of a contemporaneous pressure gradient [67]. By decoupling the velocity and pressure fields in Navier-stokes equations a self-adjoint system is created, and allows for a Helmholtz-Hodge decomposition [68] of the momentum equation into a curl-free and divergence-free part. The process performs the projection of a predicted velocity onto a divergence free space such that one can maintain a divergence free velocity field. More specifically it can be described as the Helmholtz-Hodge decomposition of a vector field into a solenoidal and irrotational part, a two part algorithmic process. The first part performed by computing the intermediate velocity at each time step. This intermediate velocity, as initially computed, lacks the ability to meet the incompressibility constraint. It is used in the second step where the pressure projects the

intermediate velocity onto the the divergence-free velocity field. Thus, giving at the next time step the updated velocity and pressure [69]. The obvious advantage of this type of projection algorithm is a more efficient solving of the incompressible Navier-Stokes equations, yet the primary reason remains in the decoupling of the velocity and pressure fields.

The projection algorithm or projection step method that is used herein, is based on [70] and [71]. We can begin to derive the projection algorithm by application of the previously mentioned Helmholtz-Hodge decomposition of a vector field in domain Ω . Equation (4.1) is uniquely decomposed where \mathbf{U} is the divergence free velocity. A divergence free velocity implies $\nabla \cdot \mathbf{U} = 0$ in Ω , while $\mathbf{U} \cdot \mathbf{n} = 0$ on the boundary Γ . Equation (4.1) can now be clearly referenced as the segment of the decomposition that is responsible for the projection onto the divergence-free field.

$$(4.1) \quad \mathbf{V} = \mathbf{U} + \nabla P$$

The curl of a vector function that is a function of the gradient of a scalar is always irrotational and curl free. The projection of the velocity \mathbf{V} in (4.1) has the benefit of the gradient having zero curl under the decomposition as $\nabla \times \nabla P = 0$. The projection seeks to find a proper P such that we have equation (4.2). By taking the divergence of (4.2) we see the resulting equation (4.3).

$$(4.2) \quad \nabla P = \mathbf{V} + \mathbf{U}$$

$$(4.3) \quad \nabla \cdot \nabla P = \nabla^2 P = \nabla \cdot (\mathbf{V} + \mathbf{U}) = \nabla \cdot (\mathbf{V}) + \nabla \cdot (\mathbf{U}) = \nabla \cdot (\mathbf{V})$$

The linear orthogonal projection operator L is applied to the incompressible Navier-Stokes vector field as shown in equation (4.4) since L is a linear operator it has the

quality, $L(\nabla P) = 0$ resulting in the pressure being removed from the set of equations. Thus, the projection under the linear operator L is given by equation (4.5) with \mathbf{U} being the divergence-free averaged time velocity.

$$(4.4) \quad L\left(\frac{\partial \mathbf{U}}{\partial t} + \nabla P\right) = L(-(\mathbf{U} \cdot \nabla)\rho\mathbf{U} + \mu\nabla^2\mathbf{U})$$

$$(4.5) \quad \frac{\partial \mathbf{U}}{\partial t} = -(\mathbf{U} \cdot \nabla)\rho\mathbf{U} + \mu\nabla^2\mathbf{U}$$

$$(4.6) \quad \rho\frac{\mathbf{V}_{n+1}^* - \mathbf{V}_n}{dt} + \rho\mathbf{V}_n \cdot \nabla\mathbf{V}_n = \mu\nabla^2\mathbf{V}_n$$

The fractional split is accomplished by splitting the velocity into two averaged components \mathbf{V}^* and \mathbf{V} , in fractionally splitting the momentum equation (4.5) under the linear orthogonal projection applied via equation (4.4) we are able derive equation (4.6). In equation (4.6) the velocity components of \mathbf{V} are from either the initial guess or from the previously computed time step. This process still requires the acquisition of a divergence-free velocity through a proper P or more precisely ∇P . The advanced velocity is marched forward explicitly and a velocity \mathbf{V} is sought that satisfies the continuity equation (3.23). One seeks to find the projection of a perturbed velocity \mathbf{V}^* onto the divergence free space to complete the calculation of the velocities that are subject to the incompressibility constraint. Under the decomposition of the vector field $L(\mathbf{V}^*)$ a projection is made producing equation (4.7). The pressure Poisson equation is now easily derive by applying the gradient operator to both sides of equation (4.7) giving equation (4.8). Equation (4.8) can be represented in discretized finite element form by (4.9), where M is the mass matrix. The Euler-Lagrange equations are formed from equation (4.7) and equation (4.8) and are used to derive equation (4.9) where C is defined as the gradient operator. The equation (4.10) is subject to the

continuity constraint $C^T \mathbf{V} = 0$, the system then solved by sequentially creating a diagonal form of the mass matrix which is a process commonly referred to as lumping and produces the lumped mass matrix by multiplying by it's inverse matrix and taking applying the gradient operator to both sides all while enforcing continuity we derive equation (4.12).

$$(4.7) \quad \mathbf{V}^* = \mathbf{V} + dt \nabla P \quad s.t \quad \nabla \cdot \mathbf{V} = 0$$

$$(4.8) \quad \nabla^2 P = \frac{-\nabla \cdot \mathbf{V}^*}{dt}$$

$$(4.9) \quad M \frac{(\mathbf{V} - \mathbf{V}^*)}{dt} + \nabla P = 0$$

$$(4.10) \quad \frac{M}{dt} (\mathbf{V} - \mathbf{V}^*) = -CP$$

$$(4.11) \quad C^T M^{-1} CP = -C^T \mathbf{V}^*$$

$$(4.12) \quad \mathbf{V} = \mathbf{V}^* - dt M^{-1} CP$$

Equation (4.9) is subject to the continuity constraint $C^T \mathbf{V} = 0$, the system is then solved by sequentially creating a diagonal form of the mass matrix which is a process commonly referred to as lumping and produces the lumped mass matrix [53]. By multiplying the lumped mass matrix by the inverse lumped mass matrix, applying the gradient operator to both sides, and finally enforcing the continuity requirement we derive equation (4.12).

4.2 Weak Form of Governing Equations

In the following equations, $[]$ is used to refer to a matrix and $\{ \}$ will be used to refer to a column vector. The weak statement for the projection algorithm or more commonly referred to as the Euler-Lagrange variational is shown in equation (4.13). The summation on the left hand side of equation (4.13) creates the diagonalized mass matrix, which can be solved for the averaged velocity \mathbf{V} with the help of the weighted residual statement, equation (3.69), produces the divergence free velocity shown in equation (4.13).

$$(4.13) \quad \int_{\Omega} \left[\frac{\partial N_j}{\partial x_j} \right] \{N_k\} \sum_{l=1}^n [N_l] \left\{ \frac{\partial N_i}{\partial x_i} \right\} \{P_i\} d\Omega = \int_{\Omega} \left[\frac{\partial N_j}{\partial x_j} \right] \{\mathbf{V}_i^*\} \frac{d\Omega}{dt}$$

$$(4.14) \quad \{\mathbf{V}_i\} = \{\mathbf{V}_i^*\} - dt \left[\{N_k\} \sum_{i=1}^n [N_1] d\Omega \right]^{-1} \left[\int_{\Omega} [N_j] \{P_j\} d\Omega \right]$$

Once the inverse matrix for the pressure field is created, next is the most computationally intensive and time-consuming part of the process, computing the solution of the Euler-Lagrange equation enforcing mass matrix [53].

The application of the FEM to the solution of governing equations requires the weak statement formulations derived first then coded. In this section a derivation of the weak form of the energy and mass transport equations is performed before deriving the weak form of the SST turbulence model. The MWR is applied to the weak statements to show the representation of the governing equations. The dependent variables are replaced with their trial functions as in equation (4.15), where Z_i are the dependent variables, $[N_j]$ are the shape or basis functions for the element, $[]$ is a row matrix or row vector, and $\{ \}$ is a column vector.

$$(4.15) \quad Z_i = \sum_{i=1}^n \phi_n(x) z_i^n(t) = [N_j] \{Z_i\}$$

4.3 Matrix Equations

The boundary terms and their evaluation will be described in subsequent sections after showing the matrix forms of the equations. The boundary terms become the load vector in the weak form of the matrix statement. By substituting the dependent variable trial functions into the governing equations the following sets of integral ordinary differential equations are produced.

4.3.1 Weighted Residual Statement of Velocity under Helmholtz-Hodge Decomposition for $k - \omega$

The term $f(x_i)$ in equation (4.16) is the body force per unit mass that is typically prescribed as the force of gravity. If the fluid of interest is slightly compressible needing the Boussinesq approximation for density change as function of temperature then this body force term $f(x_i)$ is modeled as the difference in gravity and buoyant forces, i.e. $(\rho_o - \rho)g_{x_i}$.

$$\begin{aligned}
 (4.16) \quad & \left(\rho \int_{\Omega} \{N_k\} \sum_{l=1}^n [N_l] d\Omega \right) \{\dot{\mathbf{V}}_i\} + \rho \left(\int_{\Omega} \{N_i\} (N_k \mathbf{V}_k) \left[\frac{\partial N_j}{\partial x_i} \right] d\Omega \right) \{\mathbf{V}_i\} \\
 & + \int_{\Omega} \{N_i\} \left[\frac{\partial N_j}{\partial x_i} \right] \{P_i\} d\Omega + \int_{\Omega} \{N_i\} \frac{2}{3} \left[\frac{\partial N_j}{\partial x_i} \right] d\Omega \{k_i\} \\
 & - \int_{\Omega} \{N_i\} \left[\frac{\partial N_j}{\partial x_i} \right] \{\mu_t\} \left[\frac{\partial N_j}{\partial x_i} \right] d\Omega \{\mathbf{V}_i\} \\
 & + \left(\int_{\Omega} [\mu + \mu_t] \left\{ \frac{\partial N_j}{\partial x_i} \right\} \left[\frac{\partial N_j}{\partial x_i} \right] d\Omega \right) \{\mathbf{V}_i\} \\
 & + \rho \int_{\Omega} f(x_i) \{N_i\} d\Omega - \int_{\Gamma} \{N_i\} [\mu + \mu_t] \left[\frac{\partial N_j}{\partial x_i} \right] \{\mathbf{V}_i\} \hat{n}_i d\Gamma = 0
 \end{aligned}$$

4.3.2 Weighted Residual Statement of the Thermal Energy Equation

$$\begin{aligned}
(4.17) \quad & \left(\rho c_p \int_{\Omega} \{N_k\} \sum_{l=1}^n [N_l] d\Omega \right) \{\dot{T}_i\} + \left(\rho c_p \int_{\Omega} \{N_i\} (N_k \mathbf{V}_k) \left[\frac{\partial N_j}{\partial x_i} \right] d\Omega \right) \{T_i\} \\
& - \int_{\Omega} \{N_i\} \left(\left[\frac{\partial N_j}{\partial x_i} \right] \left[\frac{\mu_t}{Pr_t} \right] \left[\frac{\partial N_j}{\partial x_i} \right] d\Omega \right) \{T_i\} + \\
& \left(\int_{\Omega} \left[\kappa + \frac{\mu_t}{Pr_t} \right] \left\{ \frac{\partial N_j}{\partial x_i} \right\} \left[\frac{\partial N_j}{\partial x_i} \right] d\Omega \right) \{T_i\} \\
& - \left\{ \int_{\Omega} \{N_i\} \{Q_i\} d\Omega \right\} - \left(\int_{\Gamma} \{N_i\} \{q_i\} \Gamma d\Gamma \right) = 0
\end{aligned}$$

4.3.3 Weighted Residual Statement of the Turbulent Kinetic Energy for $k - \omega$

$$\begin{aligned}
(4.18) \quad & \left(\rho \int_{\Omega} \{N_k\} \sum_{l=1}^n [N_l] d\Omega \right) \{\dot{k}_i\} + \left(\rho \int_{\Omega} \{N_i\} (N_k \mathbf{V}_k) \left[\frac{\partial N_j}{\partial x_i} \right] d\Omega \right) \{k_i\} \\
& - \left(\int_{\Omega} \{N_i\} \left[\frac{\partial N_j}{\partial x_i} \right] \{\sigma^* \mu_t\} \left[\frac{\partial N_j}{\partial x_i} \right] d\Omega \right) \{k_i\} \\
& + \left(\int_{\Omega} \left[\mu + \sigma^* \mu_t \right] \left\{ \frac{\partial N_j}{\partial x_i} \right\} \left[\frac{\partial N_j}{\partial x_i} \right] d\Omega \right) \{k_i\} \\
& + \left(\rho \beta^* \int_{\Omega} \left(\{N_i\} [N_j] \{k_i\} [N_j] \{\tilde{\omega}_i\} d\Omega \right) - \{P_k\} - \left(\int_{\Gamma} \{N_i\} q_k \Gamma d\Gamma \right) \right) = 0
\end{aligned}$$

where:

$$\begin{aligned}
(4.19) \quad \{P_k\} = & \int_{\Omega} \left[\{N_i\} [\mu_t] \left(\left[\frac{\partial N_j}{\partial x_i} \right] \{u_i\} + \left[\frac{\partial N_i}{\partial x_j} \right] \{u_i\} - \frac{2}{3} \left[\frac{\partial N_k}{\partial x_k} \right] \{u_k\} \delta_{ij} \right) \right. \\
& \left. - \delta_{ij} \frac{2}{3} \rho [N_j] \{k_i\} \right] d\Omega
\end{aligned}$$

4.3.4 Weighted Residual Statement of the Specific Dissipation Rate for $k - \omega$

$$\begin{aligned}
(4.20) \quad & \left(\rho \int_{\Omega} \{N_k\} \sum_{l=1}^n [N_l] d\Omega \right) \{\dot{\omega}_i\} + \left(\rho \int_{\Omega} \{N_i\} (N_k \mathbf{V}_k) \left[\frac{\partial N_j}{\partial x_i} \right] d\Omega \right) \{\dot{\omega}_i\} \\
& - \left(\int_{\Omega} \{N_i\} \left[\frac{\partial N_j}{\partial x_i} \right] \{\sigma \mu_t\} \left[\frac{\partial N_j}{\partial x_i} \right] d\Omega \right) \{\tilde{\omega}_i\} \\
& + \left(\int_{\Omega} [\mu + \sigma \mu_t] \left\{ \frac{\partial N_j}{\partial x_i} \right\} \left[\frac{\partial N_j}{\partial x_i} \right] d\Omega \right) \{\tilde{\omega}_i\} \\
& + \left(\rho \beta \int_{\Omega} \left(\{N_i\} ([N_j] \{\tilde{\omega}_i\})^2 d\Omega \right) - \{P_{\omega}\} d\Omega - \left(\int_{\Gamma} \{N_i\} q_{\tilde{\omega}} \Gamma d\Gamma \right) \right) = 0
\end{aligned}$$

where:

$$\begin{aligned}
(4.21) \quad \{P_{\tilde{\omega}}\} = & \alpha \int_{\Omega} \left[\{N_i\} [\mu_{t_i}] \left(\left[\frac{\partial N_j}{\partial x_i} \right] \{u_i\} + \left[\frac{\partial N_i}{\partial x_j} \right] \{u_i\} - \frac{2}{3} \left[\frac{\partial N_k}{\partial x_k} \right] \{u_k\} \delta_{ij} \right) \right. \\
& \left. - \delta_{ij} \frac{2}{3} \rho [N_j] \{k_i\} \right] \left(\frac{[N_j] \{\tilde{\omega}_i\}}{[N_j] \{k_i\}} \right) d\Omega
\end{aligned}$$

4.3.5 Weighted Residual Statement of the Specific Dissipation Rate for MSST

For convenience the two-equation SST closure model presented in section 3.2.1 is restated here as equation (4.22) through equation (4.28) with closure constants and the blending procedure. The weak form derivation immediately follows after after the closure constants are re-shown.

$$(4.22) \quad \frac{\partial \rho k}{\partial t} + \frac{\partial \rho u_j k}{\partial x_j} = P_k - \beta^* \rho \omega k + \frac{\partial}{\partial x_j} \left[(\mu + \sigma k_1 \mu_t) \frac{\partial k}{\partial x_j} \right]$$

$$(4.23) \quad \frac{\partial \rho \omega}{\partial t} + \frac{\partial \rho u_j \omega}{\partial x_j} = \gamma_1 P_\omega - \beta_1 \rho \omega^2 + \frac{\partial}{\partial x_j} \left[(\mu + \sigma_{\omega_1} \mu_t) \frac{\partial \omega}{\partial x_j} \right]$$

$$(4.24) \quad \frac{\partial \rho k}{\partial t} + \frac{\partial \rho u_j k}{\partial x_j} = P_k - \beta^* \rho \omega k + \frac{\partial}{\partial x_j} \left[(\mu + \sigma_{k_2} \mu_t) \frac{\partial k}{\partial x_j} \right]$$

$$(4.25) \quad \frac{\partial \rho \omega}{\partial t} + \frac{\partial \rho u_j \omega}{\partial x_j} = \gamma_2 P_\omega - \beta_2 \rho \omega^2 + 2\rho \sigma_{\omega^2} \frac{1}{\omega} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j} + \frac{\partial}{\partial x_j} \left[(\mu + \omega_2 \mu_t) \frac{\partial \omega}{\partial x_j} \right]$$

$$(4.26) \quad \frac{\partial \rho k}{\partial t} + \frac{\partial \rho u_j k}{\partial x_j} = P_k - \beta^* \rho \omega k + \frac{\partial}{\partial x_j} \left[(\mu + \sigma_k \mu_t) \frac{\partial k}{\partial x_j} \right]$$

$$(4.27) \quad \frac{\partial \rho \omega}{\partial t} + \frac{\partial \rho u_j \omega}{\partial x_j} = \gamma P_\omega - \beta \rho \omega^2 + 2\rho(1 - F_1) \sigma_{\omega^2} \frac{1}{\omega} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j} + \frac{\partial}{\partial x_j} \left[(\mu + \omega \mu_t) \frac{\partial \omega}{\partial x_j} \right]$$

In the new model a blending function Φ is defined in equations (4.28), where $\Phi_1 \equiv$ is any constant in the original model, i.e., (σ_{k_1}, \dots) , $\Phi_2 \equiv (\sigma_{k_2}, \dots)$ being any constant in the transformed model, and $\Phi \equiv (\sigma_k, \dots)$ any constant in the new model.

$$(4.28) \quad \Phi = F_1 \Phi_1 + (1 - F_1) \Phi_2$$

The set of constants corresponding to the Wilcox equations are, Set 1 (Wilcox): $\sigma_{k_1} = 0.5$, $\sigma_{\omega_1} = 0.5$, $\beta_1 = 0.0750$, $\beta^* = 0.09$, $\kappa = 0.41$, $\gamma_1 = \beta/\beta^* - \sigma_{k_2} \kappa^2 / \sqrt{\beta^*}$ and Set 2 (Jones-Launder): $\sigma_{k_2} = 1.0$, $\sigma_{\omega_1} = 0.856$, $\beta_2 = 0.0828$, $\beta^* = 0.09$, $\kappa = 0.41$, and $\gamma_2 = \beta/\beta^* - \sigma_{\omega_2} \kappa^2 / \sqrt{\beta^*}$.

The weak form of the k - ω turbulent dissipation equation, equations 4.20 and 4.19, are modified for the SST turbulent dissipation equation, shown in equations 4.29 and 4.30. The SST turbulent dissipation equations 4.29 and 4.30 are modified in such a way to account for the constants and blending that occur in the SST model.

$$\begin{aligned}
(4.29) \quad & \left(\rho \int_{\Omega} \{N_k\} \sum_{l=1}^n [N_l] d\Omega \right) \{\dot{\tilde{\omega}}_i\} + \left(\rho \int_{\Omega} \{N_i\} (N_k \mathbf{V}_k) \left[\frac{\partial N_j}{\partial x_i} \right] d\Omega \right) \{\dot{\tilde{\omega}}_i\} \\
& - \left(\int_{\Omega} \{N_i\} \left[\frac{\partial N_j}{\partial x_i} \right] \{\sigma \mu_t\} \left[\frac{\partial N_j}{\partial x_i} \right] d\Omega \right) \{\tilde{\omega}_i\} \\
& + \left(\int_{\Omega} \left[\mu + [N_j] \{\tilde{\omega}_i\} \mu_t \right] \left\{ \frac{\partial N_j}{\partial x_i} \right\} \left[\frac{\partial N_j}{\partial x_i} \right] d\Omega \right) \{\tilde{\omega}_i\} - 2\rho\beta([N_j] \{\tilde{\omega}_i\})^2 \\
& + 2\rho(1 - F_1)\sigma_{\omega^2} \left(\int_{\Omega} \left(\frac{1}{[N_l] \{\tilde{\omega}_l\}} \{N_i\} \left[\frac{\partial N_i}{\partial x_j} \right] \{k_j\} \left[\frac{\partial N_i}{\partial x_j} \right] \{\tilde{\omega}_j\} \right) d\Omega \right. \\
& \quad \left. - \gamma \{P_{\omega}\} d\Omega - \left(\int_{\Gamma} \{N_i\} q_{\tilde{\omega}} \Gamma d\Gamma \right) \right) = 0
\end{aligned}$$

where:

$$\begin{aligned}
(4.30) \quad \{P_{\tilde{\omega}}\} = & \alpha \int_{\Omega} \left[\{N_i\} [\mu_{ti}] \left(\left[\frac{\partial N_j}{\partial x_i} \right] \{u_i\} + \left[\frac{\partial N_i}{\partial x_j} \right] \{u_i\} - \frac{2}{3} \left[\frac{\partial N_k}{\partial x_k} \right] \{u_k\} \delta_{ij} \right) \right. \\
& \left. - \delta_{ij} \frac{2}{3} \rho [N_j] \{k_i\} \right] \left(\frac{[N_j] \{\tilde{\omega}_i\}}{[N_j] \{k_i\}} \right) d\Omega
\end{aligned}$$

4.4 Time Evolution of the Semi-Implicit Scheme for Mass, Momentum, and Energy Equations

Time stepping of the semi-implicit scheme for the mass, momentum and energy equations is performed by integrating over each element in the computational domain and adding each contribution from individual elements to the common nodes of adjacent elements [53]. Integrating in this manner forms a matrix equation which is solved for dependent nodal values. This forms matrix equations and is commonly referred to as assembly. The integration is what forms the assembled matrix equations by summing each elements nodal value contributions. The matrix equations are currently solved using either a direct method, i.e. Cholesky or Krylov iterative method with various

options like the gradient residual method or left or right preconditioning. The iterative method requires some form of preconditioning. Krylov iterative methods are very simple to thread and or vectorize. The algorithm lends itself to parallel scaling and future efforts in exascale computing.

4.4.1 Momentum in Turbulent Flow with Multi-Species

Equations (4.31), and (4.32) are for applications to Newtonian and non-Newtonian fluids and also for incompressible solids and fluid structure interaction problems. Equations (4.31), and (4.32) are the form used by the fractional-step FEM method developed in [53] and [72]. Equations (4.31), and (4.32) are the Reynolds mass and momentum averaged equations [2].

$$(4.31) \quad \frac{\partial \rho}{\partial t} + \frac{\partial (\rho \bar{u}_i)}{\partial x_i} = 0$$

$$(4.32) \quad \frac{\partial (\rho \bar{u}_i)}{\partial t} + \frac{\partial (\bar{u}_i \bar{u}_j)}{\partial x_j} = \frac{\partial \bar{t}_{ij}}{\partial x_j} - \frac{\partial \bar{p}}{\partial x_i} + \frac{\partial \bar{\tau}_{ij}}{\partial x_j} + \bar{f}_{\text{drop}} + \rho \sum_{k=1}^{\text{NumSpecies}} \bar{\gamma}_k f_{k,j}$$

where τ_{ij} is the same Reynolds stress tensor as in (3.1), (3.2), (3.3), (3.4), and (3.5). In equation (4.32) the fluctuating components of the body forces from species $\rho' \sum_{k=1}^{\text{NumSpecies}} \gamma'_k f_{k,j}$ are neglected along with the terms f'_{drop} , and f_{drop} . It can be noted that the contribution from the terms f'_{drop} , and f_{drop} impart momentum exchanges of spray droplets into the gas fluid when injection of particles or droplets or in combustion simulations fuel spray. The f'_{drop} , and f_{drop} are excluded from the simulations used in this thesis study.

4.4.2 Fractional Split Method

The creation of a fractional split method begins with the initial guess of velocity that is marched forward in time for prediction [53] [72]. The weak variational form

for the predictor velocity march is shown in matrix form, given by equation (4.33). The individual matrices shown in equation (4.33) are the mass matrix $[\mathbf{M}_v]$, the advection matrix $[\mathbf{A}_u]$, and the fluid stress matrix $[\mathbf{K}_{\tau u}]$, lastly the vector $\{\mathbf{F}_{v_i}\}$ being the velocity boundary gradient.

$$(4.33) \quad \{\Delta \mathbf{U}_i^*\} = -\Delta t \left[[\mathbf{M}_v^{-1}] [\mathbf{A}_u] \{\mathbf{U}_i\} + [\mathbf{K}_{\tau u}] \{\mathbf{U}_i\} - \{\mathbf{F}_{v_i}\} \right]^n$$

where

$$(4.34) \quad \{\Delta U_i^*\} = \{U_i^*\} - \{U_i^n\}$$

The pressure and density calculations are determined by solving the implicit Pressure Poisson equation. The previous section on projection methods addressed the solution process for the implicit Pressure Poisson equation, which followed as a result of mass conservation [53] [72].

The value for P' is derived in [56] from conservation principles. The conservation of mass derivation is repeated here for convenience in equation (4.35), and the time advance in discrete terms for continuity in equation (4.36). A velocity corrector is required for the projection algorithm and is shown in equation (4.37) [53]. The intermediate pressure is defined notationally by P' and assures that the mass conservation over the split is enforced, and equation (4.38). Equation (4.37) is the velocity corrector which is derived from equations (4.35) and (4.36).

$$(4.35) \quad \frac{\partial \rho}{\partial t} = -\frac{\partial \rho u_i}{\partial x_i} = \frac{\partial U_i}{\partial x_i}$$

$$(4.36) \quad \frac{\rho^{n+1} - \rho^n}{\Delta t} = \frac{\partial U_i'}{\partial x_i}$$

$$(4.37) \quad U^{n+1} - U^* = \Delta t \frac{\partial P'}{\partial x_i}$$

$$(4.38) \quad U' = \theta_1 \left(\Delta t \frac{\partial P'}{\partial x_i} + U^* \right) + (1 - \theta_1) U_i^n$$

Equation (4.39) describes the change in pressure and or density that satisfies mass conservation, where $\Delta P = P^{n+1} - P^n$.

$$(4.39) \quad \frac{1}{c^2} \Delta P = \Delta \rho = -\Delta t \frac{\partial U'_i}{\partial x_i} = \Delta t^2 \theta_1 \left(\theta_2 \frac{\partial^2 P^{n+1}}{\partial x_i^2} + (1 - \theta_2) \frac{\partial^2 P^n}{\partial x_i^2} \right) - \Delta t \left(\theta_1 \frac{\partial \Delta U_i^*}{\partial x_i} + \frac{\partial U_i^n}{\partial x_i} \right)$$

$$(4.40) \quad \frac{1}{c^2} \Delta P - \theta_1 \theta_2 \frac{\partial^2 \Delta P}{\partial x_i^2} = \Delta t^2 \theta_1 \frac{\partial^2 P^n}{\partial x_i^2} - \Delta t \left(\theta_1 \frac{\partial \Delta U_i^*}{\partial x_i} + \frac{\partial U_i^n}{\partial x_i} \right)$$

Green's theorem is applied to weaken the second order partial derivatives in equation (4.40) to a first order ordinary differential equation, and employing the thermodynamic relation $(\Delta P/c^2) = \Delta \rho$ creates a weak form of ΔP and leads to equation (4.42). It is worthwhile to note $P^{n+1} = \Delta P + P^n$ which allows one to solve for $P' = \theta_2 P^{n+1} + (1 - \theta_2) P^n = \theta_2 \Delta P + P^n$ and gives gives the final expression for the specific momentum, equation (4.41).

$$(4.41) \quad \Delta U_i = U^{n+1} - U^n = \Delta U_* - U_n = \Delta U_* \Delta t \frac{\partial P'}{\partial x_i} = \Delta U_* - \Delta t \left(\theta_2 \frac{\partial \Delta P}{\partial x_i} + \frac{\partial P^n}{\partial x_i} \right)$$

The matrix form of equation (4.41) is given by equation (4.42).

$$(4.42) \quad \left(\left[\frac{\mathbf{M}_p}{c^2} \right] + \Delta t^2 \theta_1 \theta_2 \mathbf{H} \right) \{ \Delta P_i \} = \Delta t^2 \theta_1 \mathbf{H} \{ P_i^n \} - \Delta t (\theta_1 G \{ \Delta U_i^* \} + \mathbf{G} \{ U_i^n \} - \Delta t \{ \mathbf{F}_{p_i} \})$$

The tensor \mathbf{H} is a weakened second derivative term that gives the pressure boundary conditions. The values of θ_1 in equation (4.42) ranges from 0.5 to 1 and θ_2 from 0 to 1 depending on the amount of semi-implicitness is needed to solve the system [53]. The choice that $\theta_2 = 0$ allows the system to run an explicit time march and would work well for future pursuits towards exascale computing. If solving for $\Delta\rho$ in compressible flow problems, the velocity is updated as required by $\Delta P = c^2\Delta\rho$.

The FEM form of ΔU is given in equation (4.43), below. The final velocity is given using the corrector equation (4.44), which allows the mass velocity $\phi^{n+1}u_i^{n+1} = U_i^{n+1}$ to be solved with the extracted velocity as in equation (4.45).

$$(4.43) \quad \{\Delta\mathbf{U}_i\} = \Delta\mathbf{U}^* - \Delta t[\mathbf{M}_u^{-1}](\theta_2[\mathbf{G}]\{\Delta p_i\} + [\mathbf{G}]\{p_i^n\})$$

$$(4.44) \quad \{\Delta\mathbf{U}_i^{n+1}\} = \{\Delta\mathbf{U}_i\} + \{\Delta\mathbf{U}_i^n\}$$

$$(4.45) \quad \mathbf{u}^{n+1} = \frac{U^{n+1}}{\rho^{n+1}}$$

The updated density ϕ^{n+1} is recovered from the equation of state and is used in equation (4.45) to extract the updated velocity for the specific momentum. The speed of sound is determined by $c = \sqrt{\gamma RT}$, where R is the specific gas constant. Specifically for air, $R = 287\text{J/kg} \cdot \text{K}$, if the process involves multi-species transport then R can be determined by a mass-averaging process [53] [72].

4.4.3 Energy Conservation for Multi-Species and Turbulent Energy Transport

The equation for total energy with additional terms added to model species is given by equation (4.46). A full derivation of equation (4.46) can be found in [73].

$$(4.46) \quad \frac{\partial E}{\partial t} = -\frac{\partial}{\partial x_i}(Eu_i + pu_i) + \frac{\partial}{\partial x_i}\left(\kappa + \frac{\mu_t}{Pr_t}\right)\frac{\partial T}{\partial x_i} + \frac{\partial}{\partial x_i}(t_{ij} + \tau_{ij})u_j + \frac{\partial}{\partial x_i}\left[\rho \sum_{k=1}^{N_{species}} \bar{H}_k \left(D_k + \frac{\mu_t}{Sc_t}\right) \frac{\partial \gamma_k}{\partial x_i}\right] + \rho \sum_{j=1}^{N_{species}} \gamma_j f_j(x_i) \cdot U_i - \sum_{k=1}^{N_{species}} H_{ok} \omega_k$$

The shear stress is a function of the the molecular viscosity and the term t_{ij} is given by equation (4.47).

$$(4.47) \quad \mathbf{t}_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right)$$

The rate of flux for heat is given by (4.48), where the Dufour and Soret effects are neglected and reflect the assumption that radiation has some minimal effect for engine simulation combustion processes [72] that do not involve much soot formation, or a long latent fuel combustion in the combustion arena in an engine.

$$(4.48) \quad \dot{q} = \dot{q}_d + \dot{q}_{Dufour} + \dot{q}_{Soret} + \dot{q}_{radiation}$$

The heat flux rate from species diffusion is given (4.49).

$$(4.49) \quad \dot{q}_d = \rho \sum_{j=1}^{NumSpecies} H_j \gamma_j V_j$$

The enthalpy on the right hand side of equation (4.49) is given by equation (4.50).

$$(4.50) \quad H_j = H_{o,j} + \bar{H}_j = H_{o,j} + \int_{T_o}^T c_{p,j} dT$$

The application of Fick's Law of mass diffusion, to the mass species velocity $\gamma_j V_j$ gives equation (4.51), where D is assumed to be equal for all species.

$$(4.51) \quad \gamma_j V_j = -D \nabla \gamma_j$$

The simplification or assumption can be made that all but one of the species exists in small quantities so that the diffusion velocity of the j^{th} species interacts primarily with the most abundant species, N. This assumption leads to the simplified form of equation (4.51) as equation (4.52).

$$(4.52) \quad \gamma_j V_j = -D_{j,N} \nabla \gamma_j$$

The flux rate for heat can finally be shown as equation (4.53).

$$(4.53) \quad \dot{q}_d = - \sum_{j=1}^{NumSpecies} H_{o,j} \omega_j - \rho \sum_{j=1}^{NumSpecies} H_j D_{j,N} \nabla \gamma_j$$

The effect of weakening the second order PDE to a first-order differential equation gives the flux on the boundary $\{\mathbf{F}_{ei}\}$. For a thermally perfect gas, enthalpy and internal energy are functions of temperature only, so by solving for the energy equation, by default the temperature is evaluated. The final total energy is calculated by summing all in-source terms explicitly. Because equation (4.46) is already shown as the incompressible change in thermal energy for an incompressible fluid, we can see after separating the instantaneous parts into mean and fluctuating components we get the internal energy transport equation. Again, same equation (4.46). In equa-

tion (4.46) we are choosing to neglect species body forces and drop the notation used for averaging because our equation is now shown in closed form which accounts for Reynolds stress terms and species fluctuations.

4.4.4 Species Transport

The conservation of the species terms and the fluxes and sources of these species into and out of the domain results in a species transport mass fraction given by equation (4.54), in conservative form.

$$(4.54) \quad \gamma^j = \frac{\rho^j}{\rho}$$

$$(4.55) \quad \frac{\partial \rho \gamma_j}{\partial t} = -\frac{\partial \rho u_i \gamma_j}{\partial x_i} + \frac{\partial \rho D}{\partial x_i} \frac{\partial \gamma_j}{\partial x_i} + \gamma_j f_j(x_i) + \dot{w}_{chem}^i + \dot{w}_{spray}^j$$

The terms in the species transport equation (4.55), $\dot{w}_{chem}^i + \dot{w}_{spray}^j$ are for chemical reactions and spray evaporation which are not investigated herein. For breadth, the term $\gamma_j f_j(x_i)$ would be the j^{th} body force term.

4.4.5 Turbulent Species Transport

If and when the species concentration is not affected by the momentum they can be properly advected by use of the mean fluid flow. In these particular situations a scalar transport equation is used similar to that of the thermal energy transport equation. It full derivations can be found in [73] and other references [53] [72], but the weak form of the turbulent species transport equation in matrix form is given in equation (4.56):

$$(4.56) \quad \rho \Delta \gamma_j^j = -\Delta t [\mathbf{M}_\gamma^{-1}] \left[[\mathbf{A}_\gamma] \{\gamma_i^j\} + [\mathbf{K}_\gamma] \{\gamma_i^j\} + \{\mathbf{F}_{\gamma s_i^j}\} + \{\mathbf{Q}_{\gamma s_i^j}\} \right]^n$$

The matrix terms in the turbulent species transport equation (4.56) are for the j^{th} species, i.e., advection $[\mathbf{A}_\gamma] \{\gamma_i^j\}$, species diffusion $[\mathbf{K}_\gamma] \{\gamma_i^j\}$ and via the process to create a weakened equation one can see that the flux on the boundary elements in the FEM is the species body force term $\{\mathbf{F}_{\gamma s_i^j}\}$. There is also great care needed to account for the aggregate material properties, commonly referred to as composite material, which are determined from the molar fractions for the j^{th} species at each computational node i , this derivation is done properly in [73].

4.4.6 Petrov-Galerkin Stabilization Scheme

The advection terms are stabilized by use of the up-winded Petrov-Galerkin (PG) weighting method, which is used in addition to the characteristic terms. The weighting method is developed in [74] which uses the weighting as shown in equation (4.57).

$$(4.57) \quad W_i = N_i + \frac{\alpha h_e}{2V} (\mathbf{V} \cdot \nabla N_i)$$

In Equation (4.57) h_e is the element size and direction of local velocity given by: $\alpha = \coth(Pe/2 - 2/Pe)$. The cell Peclet number is given by $Pe = \rho h_e |V| / 2K_e$, where K_e is considered an effective diffusion within the cell Peclet number itself [53].

Adjustments are made to correct for inviscid flow by removing the viscosity, which can be done by setting $\alpha = 1$. By using this weighting method, one introduces a selective approach to artificial diffusion upon the numerical scheme that propagates along the local stream-wise direction of flow and effectively removes numerical dispersion effects [74]. The error that is introduced from dispersion from the advection

modeling is measured before advancing the is then appropriately removed during the integration.

4.4.7 Finite Element Matrix Terms for the Mass, Momentum, Energy and Species Equations

When integrating over the elements and summing the contributions the surface flux cancels everywhere except at the boundaries [53]. A clear distinction between other numerical methods and the FEM can be seen by noticing this. This leads to the understanding that FEM is precisely conservative unlike the FVM which induces a truncation error when evaluating the surface flux contributions at all points within the domain. The momentum predictor matrix equations for advection and diffusion are given by equation (4.58), and equation (4.59). The matrix equation for body forces and boundaries is provided by equation (4.60). The pressure or density corrector equation matrices are shown for the Laplacian by (4.61), gradient (4.62) and boundary (4.63).

$$(4.58) \quad \mathbf{A}_u = \int_{\Omega} \{N_i\} u_j \left(\left[\frac{N_k}{x_j} \right] \right) \{U_i\} d\Omega$$

$$(4.59) \quad \mathbf{K}_{\tau u} = - \left(\int_{\Omega} \{N_i\} \left[\frac{N_j}{\partial x_i} \right] \{\mu_t\} \left(\left[\frac{\partial N_j}{\partial x_i} \right] + \frac{1}{3} \left[\frac{N_j}{\partial x_i} \right] \right) \{u_j\} d\Omega + \int_{d\Omega} \left(\left[\mu + \mu_t \right] \left(\left\{ \frac{N_j}{\partial x_i} \right\} \left[\frac{N_j}{\partial x_i} \right] \{u_i\} + \frac{1}{3} \frac{\partial N_j}{\partial x_i} \{u_i\} \right) - \frac{2}{3} \delta_{ij} \rho k \right) d\Omega \right)$$

$$(4.60) \quad \mathbf{F}_u = - \int_{\Omega} \{N_i\} \rho \sum_{k=1}^{NumSpecies} [N_j] \{\gamma_k f_k(x_i)\} d\Omega + \int_{\Omega} \{N_i\} [N_j] \{f_{drop_i}\} d\Omega + \int_{d\Gamma} \{N_i\} [\mu + \mu_t] \left[\frac{\partial N_j}{\partial x_j} \right] \{U_i\} d\Gamma$$

$$(4.61) \quad \mathbf{H} = \int_{\Omega} \left\{ \frac{\partial N_i}{\partial x_j} \right\} \left[\frac{\partial N_j}{\partial x_i} \right] d\Omega$$

$$(4.62) \quad \mathbf{G} = \int_{\Omega} \left\{ \frac{\partial N_i}{\partial x_j} \right\} \left[\partial N_j \right] d\Omega$$

$$(4.63) \quad \mathbf{F}_{P_i} = \left(\int_{\Gamma} \{N_i\} \left[\frac{N_j}{\partial x_i} \right] \{\Delta P_i\} \right) + \left(\{N_i\} \left[\frac{N_j}{\partial x_k} \right] \{P_i^n\} \right) d\Gamma$$

The momentum phase interaction corrector matrices and energy related vectors and matrices are defined for the advection of energy in equation (4.64) and the pressure work term by equation (4.65)

$$(4.64) \quad \mathbf{A}_e = \int_{\Omega} \{N_i\} u_j \left(\left[\frac{\partial N_k}{\partial x_j} \right] \right) \{E_i\} d\Omega$$

$$(4.65) \quad \mathbf{C}_p = \int_{\Omega} \{N_m\} [N_l] \{p_i\} \left[\frac{\partial N_l}{\partial x_k} \right] \{u_k\} d\Omega$$

The equation for thermal diffusion is shown in equation (4.66) where weakening the equation produces the boundary flux integral [53] given by equation (4.71), while

the viscous dissipation is given by equation (4.67).

$$(4.66) \quad \mathbf{K}_T = - \left(\int_{\Omega} \{N_i\} \left[\frac{\partial N_j}{\partial x_i} \right] \left\{ \frac{\mu_t}{Pr_t} \right\} \left[\frac{N_j}{\partial x_i} \right] d\Omega \right) + \int_{\Omega} \left(\left[\kappa + \frac{\mu_t}{Pr_t} \right] \frac{\partial N_i}{\partial x_j} \frac{\partial N_i}{\partial x_j} \right) d\Omega$$

$$(4.67) \quad \mathbf{K}_{\tau} = \int_{\Omega} \left(\mu \left\{ \frac{\partial N_i}{\partial x_j} \right\} \left[\frac{\partial N_j}{\partial x_i} \right] - \frac{2}{3} \delta_{ij} \left[\frac{\partial N_j}{\partial x_i} \right] \right) \{u_i\} - \frac{2}{3} \delta_{ij} \rho k \left[N_j \right] \{u_k\} d\Omega$$

The body forces applied to the boundaries require the divergence free heat flux, equation (4.68), which includes the diffusion of species with individual chemical properties that require the diffusion velocity to be converted from a diffusive flux [53]. The full derivation is not shown here, but provided by equation (4.69), carefully note that the process of weakening the residual gives rise to a flux boundary of heat for the species that is absorbed by the term F_{es} via the boundary integral term. The volume forces acting on the j^{th} species are given by equation (4.70).

$$(4.68) \quad q_{div} = \nabla \cdot q = \frac{\partial}{\partial x_i} \left(\rho \sum_{k=1}^{NumSpecies} \overline{H}_k D_k \frac{\partial \gamma_k}{\partial x_i} \right)$$

$$(4.69) \quad q_{div} = \int_{\Omega} \left(\rho \sum_{l=1}^{NumSpecies} H_l D_{l,N} \left\{ \frac{\partial N_i}{\partial x_j} \right\} \left[\frac{\partial N_i}{\partial x_j} \right] \{ \gamma_{li} \} \right) d\Omega = \int_{\Omega} \rho \left(\sum_{l=1}^{NumSpecies} c_{pl} D_{l,N} \left\{ \frac{\partial N_i}{\partial x_j} \right\} \left[\frac{\partial N_i}{\partial x_j} \right] \{ \gamma_{li} \} T \right) d\Omega$$

$$(4.70) \quad F_{eb} = \int_{\Omega} \{N_i\} \left(\rho \sum_{l=1}^{NumSpecies} [N_j] \{U_k\} [N_j] \{ \gamma_{li} f_l(x_k)_i \} \right) d\Omega$$

$$(4.71) \quad F_{es} = \int_{\Gamma} \{N_i\} \hat{n} \cdot (\{N_j\} \{q_i\}) d\Gamma + \int_{\Gamma} \{N_i\} \hat{n} \cdot \left(\sum_{k=1}^{NumSpecies} \left[\frac{\partial N_j}{\partial x_i} \right] \{D_{kn} \gamma_k\}_i \right) d\Gamma$$

Source and sink terms for energy injections are applicable to spark injection direction injection combustion engines and other applications, but not relevant to the SST development herein. The overall heat associated with destruction or birth of a source and sink can be modeled by a term \mathbf{Q}_{ss} , and another volumetric heat source \mathbf{Q}_{vs} . The species flux matrix equations are not shown, but follow similarly and are both conservative and weakened to a first order differential and can be found in [53]. In order to evaluate the species flux one need only consider the boundaries of the domain in question [53].

4.4.8 Turbulent Model Matrices

The boundary term on the domain for turbulent flux is usually zero, thus not often needing to be defined, and is easily identified as the first term in equation (4.71). The advection for the k - ω turbulence model is shown in equations (4.72) and (4.73) and dissipation equations (4.74) and (4.75).

$$(4.72) \quad \mathbf{A}_k = \int_{\Omega} \{N_i\} u_j \left(\left[\frac{\partial N_k}{\partial x_j} \right] \right) \{k_i\} d\Omega$$

$$(4.73) \quad \mathbf{A}_{\bar{\omega}} = \int_{\Omega} \{N_i\} u_j \left(\left[\frac{\partial N_k}{\partial x_j} \right] \right) \{\omega_i\} d\Omega$$

$$(4.74) \quad \mathbf{K}_k = - \int_{\Omega} \{N_i\} \left[\frac{\partial N_j}{\partial x_i} \right] \left\{ \sigma^* \mu_t \right\} \left[\frac{N_j}{\partial x_i} \right] d\Omega + \int_{\Omega} \left[\mu + \sigma^* \mu_t \right] \frac{\partial N_i}{\partial x_j} \frac{\partial N_i}{\partial x_j} d\Omega$$

$$(4.75) \quad \mathbf{K}_{\bar{\omega}} = - \int_{\Omega} \{N_i\} \left[\frac{\partial N_j}{\partial x_i} \right] \left\{ \sigma \mu_t \right\} \left[\frac{N_j}{\partial x_i} \right] d\Omega + \int_{\Omega} \left[\mu + \sigma \mu_t \right] \frac{\partial N_i}{\partial x_j} \frac{\partial N_i}{\partial x_j} d\Omega$$

The production term shown in equation (4.76) is in its weakened matrix form and is modeled by using a switch that applies the weakened version of the SST equation given in equations (4.29) and (4.30) that were previously derived in section (4.3.5).

$$(4.76) \quad \mathbf{P}_k = \int_{\Omega} \left[\{N_j\} \{ \mu_{t_i} \} \left(\left[\frac{\partial N_j}{\partial x_i} \right] \{u_i\} + \left[\frac{\partial N_i}{\partial x_i} \right] \{u_i\} - \frac{2}{3} \frac{\partial N_k}{\partial x_k} \{u_k\} \delta_{ij} \right) - \delta_{ij} \frac{2}{3} \rho [N_j] \{k_i\} \right] d\Omega$$

4.5 Boundary Conditions

4.5.1 Boundary Conditions for Velocity

The boundary conditions for the velocity of an incompressible flow only require Dirichlet boundary conditions for an average velocity in viscous flows and are modeled as either a non-slip condition for boundaries located on solid objects and as fixed velocity at inlets as specified. The outlet boundary conditions require more thought and are produced by assuming a zero gradient for velocity, which is referred to as the common Neumann condition. The Neumann condition can be stated without regard for the traction terms [53]. The zero gradient assumption is needed on velocities at outflow boundaries and requires the computational a grid to be generated that matches the pressure outflow condition, in GridPro this is set as Pressure Outflow from the properties tab after the grid is generated. By not allowing traction at the outlet boundaries one allows for the vorticity leaving the domain to be unchanged by the zero gradient constraint associated with the Neumann boundary condition [53].

This statement or requirement can be relaxed if one chooses to use a viscous boundary condition such as that found in [64].

4.5.2 Viscous and Inviscid Boundary Conditions for Velocity

A second type of boundary condition is produced when the weak statement is derived. This boundary condition results in the viscous term shown in equation (4.77). Equation (4.77) is appropriately zero for walls where non-slip conditions are prescribed and also for inlet and or outlet velocities that are normal to the boundaries [53]. If these requirements are not met then these components are effectively found from the boundary integral and evaluated to allow for a relaxing of the zero gradient velocities at the appropriate pressure outflow boundaries [53].

For those non-penetrating or tangential velocity components the normal component is found and specified to be zero. This would be for the case of inviscid flow at velocity wall boundaries.

$$(4.77) \quad \int_{\Gamma} \{N_i\} \{\mu + \mu_t\} \left[\frac{\partial N_j}{\partial x_i} \right] \{U_i\} n_i d\Gamma$$

4.5.3 Boundary Conditions for Turbulent Kinetic Energy and Specific Dissipation Rate

For boundary layer flow in the presence of turbulence a defect layer and inner layer region [2]. The so-called defect region can be made to include a buffer zone between the defect layer and the viscous sublayer located next to the wall nodes. Where these two layers meet is considered the buffer zone if one is considering a three equation inner region, otherwise the viscous sublayer and defect region are blended by using single logarithmic equation and only the sublayer and the defect layer exist.

A logarithmic sub-layer meets to join the defect layer to that of the buffer zone with the outer region or it models the defect layer with the outer region, depending on whether or not the wall in question is modeled as a 2 or 3 equation wall. The outer portion of the defect region is considered a fully turbulent region, and the idea of requiring a 3rd buffer region to distinguish the fully turbulent portion of the defect layer is presumed.

Most codes and in general most turbulence models only use a specific point that is close to the wall boundaries and not fully through to the inner layer. This is done as the alternative to integrating directly to the wall which considerably increases computational time, to avoid this a wall function is used as an assumption of the physics in this region. The two primary constituents needed to facilitate a wall function are the non-dimensional terms u^+ and y^+ , which are defined in equations (4.78) and (4.79).

$$(4.78) \quad y^+ = \frac{y\rho u^*}{\mu}$$

$$(4.79) \quad u^+ = \sqrt{\frac{u}{u^*}}$$

The term u^* is given by equation (4.80) which is the friction velocity. The wall shear stress is given by (4.80), which is that shear stress found at the wall in the viscous shear sublayer [2].

$$(4.80) \quad u^+ = \sqrt{\frac{\tau_w}{\rho}}$$

$$(4.81) \quad \tau_w = \mu \left(\frac{du}{dy} \right)_{y=0}$$

In the viscous shear layer we find the condition held by equation (4.82).

$$(4.82) \quad u^+ = y^+ = y\rho\sqrt{\frac{\tau_w}{\rho}}\mu^{-1}$$

Equation (4.82) is valid for $y^+ \leq 2 \approx 8$. If equation (4.82) sits above the range for incompressible flow over a smooth surface in the absence of a pressure gradient, the buffer zone is located in the range $8 < y^+ \leq 50$ and the wall equation [2] is used as shown in equation (4.83) and at larger distances up to $50 < y^+ \leq 200 \approx 400$ by equation (4.84). Equations (4.83) and (4.84) are the definition of the boundary conditions at the points within the inner layer, depending on the location of that point.

$$(4.83) \quad u^+ = 5\ln(y^+) - 3.05$$

$$(4.84) \quad u^+ = 2.5\ln(y^+) + 5.0$$

The wall function is solved by progressively iterating the transcendental relation, as described in [2]. Wilcox [2] states the appropriate equation to solve at a distance y_p from the wall in the presence of an adverse pressure gradient by calculation of equation (4.85). In equation (4.85), u_p is the tangential velocity at that grid point, B is a surface roughness factor, and $k_{vk} = 0.41$, which is the von Karman constant. The equations for turbulent kinetic energy and turbulent dissipation in the inner layer are given by equation (4.86) and equation (4.87) with the turbulent energy production, P^+ , provided equation (4.88).

$$(4.85) \quad u_p = u^* \left[\frac{1}{k_{vk}} \ln \left(\frac{\rho u^* y_p}{\mu} \right) + B - 1.13 \frac{y_p}{\rho (u^*)^2} \frac{dP}{dx} \right]$$

$$(4.86) \quad k_p = \frac{(u^*)^2}{\sqrt{\beta^*}} \left[1 + 1.16 \frac{\rho u^* y_p}{\mu} P^+ \right]$$

$$(4.87) \quad \bar{\omega}_p = \frac{u^*}{k_{vk} y \sqrt{\beta^*}} \left[1 + 0.30 \frac{\rho u^* y_p}{\mu} P^+ \right]$$

$$(4.88) \quad P^+ = \nu \tau_{xy} \frac{\left(\frac{\partial u}{\partial y} \right)}{\mu_\tau^4}$$

4.5.4 Artificial Compressibility

The PCS-fractional step method requires the usage of an artificial compressibility contribution or factor for the flow regimes that are close to, or fully incompressible [53]. The previously shown thermodynamic relation given in matrix form by equation (4.42) is locationally proper to apply this artificial compressibility factor. By distributing $\{\Delta P_i\}$ on the left hand side of equation (4.42) to both bracketed terms as shown in equation (4.89), one isolates the 1st term, $[\mathbf{M}_p/c^2]\{\Delta P_i\}$. The Pressure Poisson equation is now solvable, but has the specific requirement that both θ_1 and θ_2 are not equal to zero, and lie in the respective ranges: $0.5 < \theta_1 \leq 1$ and $0 < \theta_2 \leq 1$. This caveat would require unreasonable time-stepping constraints for most problems.

$$(4.89) \quad \left[\frac{\mathbf{M}_p}{c^2} \right] \{\Delta P_i\} + \Delta t^2 \theta_1 \theta_2 \mathbf{H} \{\Delta P_i\} =$$

$$\Delta t^2 \theta_1 \mathbf{H} \{P_i^n\} - \Delta t (\theta_1 G \{\Delta \mathbf{U}_i^*\} + \mathbf{G} \{\mathbf{U}_i^n\} - \Delta t \{\mathbf{F}_{p_i}\})$$

In order to relax the unreasonable time-stepping constraints, an artificial compressibility method is deployed by rearranging the 1st term in equation (4.89) as shown in equation (4.90). The artificial compressibility parameter β is introduced in equa-

tion (4.90) which is given as a constant through out the domain or can be found by inspecting the diffusive time step constraint [53]. Zienkiewicz [45] suggests finding the artificial compressibility parameter by inspecting the diffusive the time step constraint as it helps to mitigate issues that arise in global time step sizes and produces reasonable local time step sizes. This suggestion by Zienkiewicz [45] is the combined approach of dual-time stepping so one can recover the transient solution. The full approach to dual-time stepping not shown, but can be found in [75], in entirety.

$$(4.90) \quad \left[\frac{\mathbf{M}_p}{c^2} \right] \{ \Delta P_i \} \approx \left[\frac{\mathbf{M}_p}{\beta^2} \right] \{ \Delta P_i \} = \\ \Delta t^2 (\theta_1 \mathbf{H} \{ P_i^n \} - \theta_1 \theta_2 \mathbf{H} \{ \Delta P_i \} - \Delta t (\theta_1 G \{ \Delta \mathbf{U}_i^* \} + \mathbf{G} \{ \mathbf{U}_i^n \}) - \Delta t \{ \mathbf{F}_{p_i} \})$$

The value for β is found by application of equation (4.91) so to allow the convective and viscous time stepping to compete. In equation (4.91) ϵ is considered typically to be small as to ensure β is not too close to zero and vel_{conv} , vel_{diff} are the convective and diffusive velocities, respectively. Both vel_{conv} , vel_{diff} are defined in equations (4.92) and (4.93) with h_e the given element size.

$$(4.91) \quad \beta = \max(\epsilon, vel_{conv}, vel_{diff})$$

$$(4.92) \quad vel_{conv} = \|u\|$$

$$(4.93) \quad vel_{diff} = \frac{\mu}{\rho h_e}$$

It should be noted that no exact coupling exists between the energy equations and other equations in this algorithm. Specifically, the artificial compressibility method algorithm is valid for steady flows, and aids in the process to recover the transient

flow solutions that lie inside the inner iteration loops on the momentum and pressure solves. This is required for time steps smaller than the outer loop value and are continued until true convergence is reached. Using the the artificial compressibility parameter β by finding a global value has been validated [53] in addition to the use of the maximum value found for the locally determined values.

The specifics of PCS fractional step method are shown in this section. The mathematical concepts and applications developed in this section are referenced from the main work of Carrington [53], [60], but also other investigators [58], [69], [74], [75].

The MSST coding research outline and step-by-step implementation of the new turbulence model into the existing $k-\omega$ code is located in Appendix (A). The purpose of showing the MSST coding research outline is to aid any future development and give a detailed look into the process to implement the SST into the existing $k-\omega$ FEARCE code.

Chapter 5

Method Validation

5.1 Method Validation

The algorithms and performance of the FEARCE code using the new SST turbulence model were verified and validated with experimental data. Verification involves the process of understanding the model equations and their implementation, and seeks to understand if they are modeled correctly. Validation is an important part of the process to determine whether or not the model equations are numerically represented correctly and will give a physically reasonable solution to modeled phenomena.

The 2D axisymmetric duct is compared to both well established numerical data from Wilcox [2] and direct numerical simulation (DNS) data from Mansour et al. [3] for turbulent channel flow at $Re = 13,750$. The 2-D axisymmetric backward-facing step is validated with the use of both experimental data from Vogel and Eaton [1] and previous benchmarks for the $k-\omega$ PCS code using h -adaptation [58] [4] [59]. The two problem geometries are modeled in 2D because of the time constraint and technical de-bugging required to build and test the SST turbulence model. However, the 3D versions have shown excellent results in comparison to their 2D counterpart, only the 2D versions are validated. The meaning of 2D is misleading in some ways for these

the two problems studied herein, the term axisymmetric is a better choice because the code is inherently 3D and the choice of grid design and prescription of symmetry boundary conditions acts to represent a 2D solution and reduces the computational effort for de-bugging.

Note: The use of GridPro grids for these two benchmark problems was only possible after extensive development and integration of GridPro for LANL KIVA and FEARCE codes prior to coding the SST turbulence model, see Appendix B. The technical capability from integrating GridPro into the LANL KIVA and FEARCE codes is what made grid production using GridPro a possibility for these two benchmark problems.

The grids for the two benchmark problems were built using Gridpro for the 2D axisymmetric duct and the 2-D axisymmetric backward-facing step. The process to build these grids for their corresponding geometries are outlined herein.

The geometric configuration for turbulent flow the 2D duct is shown in Fig. (5.1), where the length of the grid was set to be $L = 25$ cm long, allowing for the extraction of data in the developed flow region at $L_{hd} = 22$ cm. The height of the duct, H is set to be 1 cm.

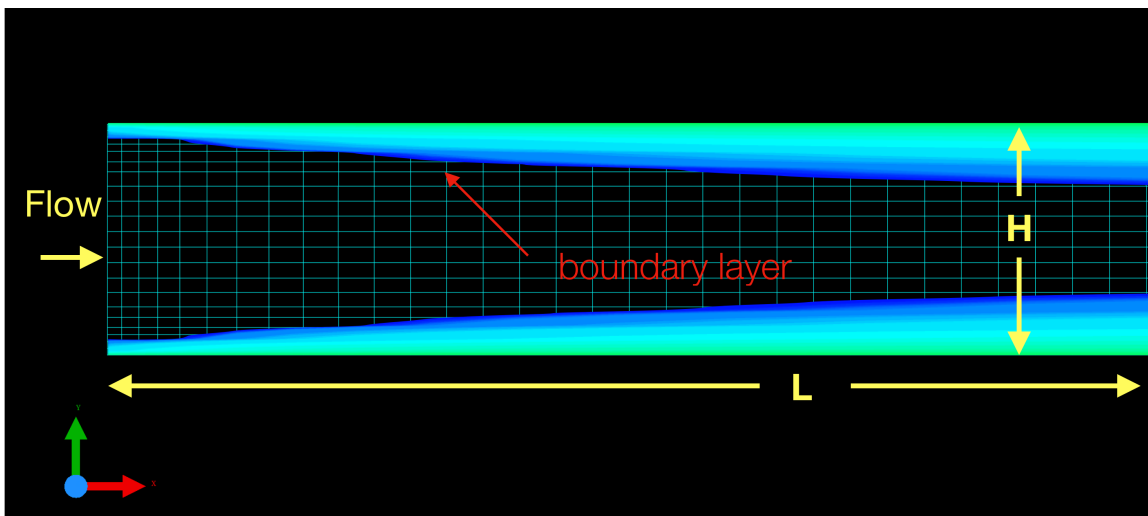


Figure 5.1: Schematic of the 2D duct, $H = 1$ cm and $L = 25$ cm.

The backward facing step geometric configuration for turbulent flow is shown in Fig. (5.2). The backward facing step with heat transfer is significantly more difficult benchmark problem. Specifically the choice for using the backward facing step with heat transfer is because it is an interesting problem to researchers in the areas of combustion, solidification, environmental flow and also compact heat exchanger design, Carrington et al. [59].

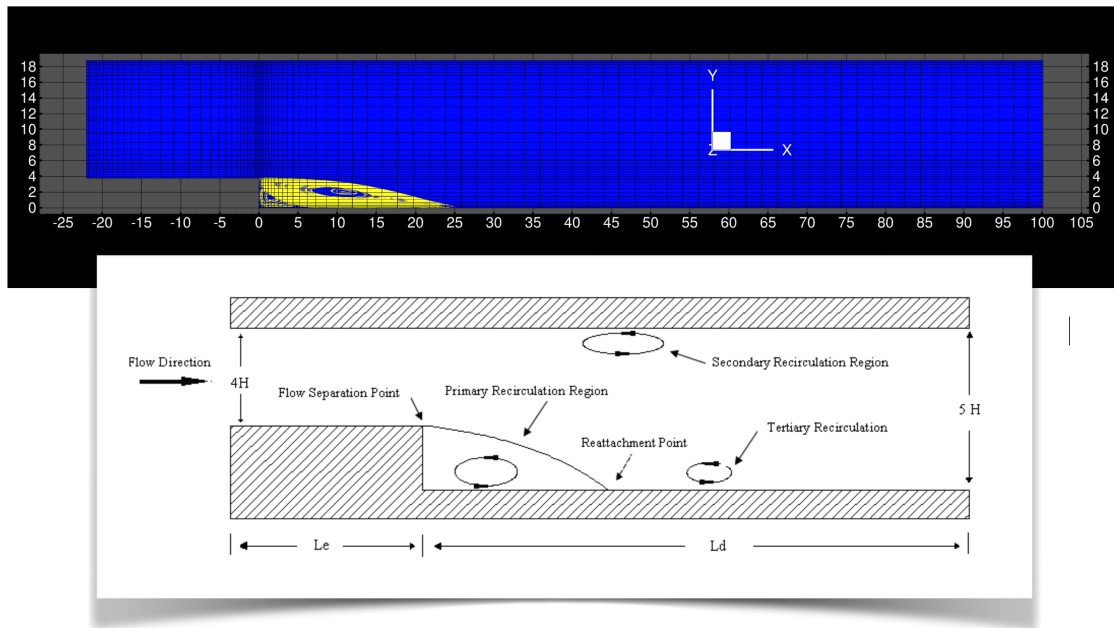


Figure 5.2: Schematic of the 2D backward facing step, the step height $H = 3.8 \text{ cm}$, $L_e = 22.0 \text{ cm}$ and $L_d = 100 \text{ cm}$.

The study of flow over the backward facing step with heat transfer is historically a good benchmark within the CFD community due to the simplicity of the geometry and plethora of rich flow physics that can be garnered from its analysis. The flow over a backward facing step includes physical phenomena interesting to the study of unsteady flow behavior, recirculation regions, flow separation and adverse pressure gradient specific problems. All of these are found in and useful to simulating the complexity of combustion engines where combustion reactions and radiation play important roles. The process to create the grid is done so by creating a 2D problem

by applying FEARCE `periodF` and `periodD` periodic boundary properties to the side boundaries of the backward facing step domain. The FEARCE convention and recent integration of the KIVA4 boundary properties allows for these to be set in the GridPro graphical user interface and exported to the FEARCE file format, a review of these surface properties can be found here [76]. The side boundaries are located to the left and right of the channel as viewed looking down the center of the grid over the step shown in Fig. (5.2), in the direction of flow, similarly for the duct as shown in Fig. (5.1). These nodes are given a zero normal velocity component to the side boundaries prescribed `periodD` and `periodF`. The use of `periodD` and `periodF` in GridPro for these side boundaries is simply to place these nodes in a list that the converter and FEM can see, and then later in the setup parameter files that drive FEARCE they can be dealt with. This, when coupled with a single element thick grid yields a 2D representation of the problem from the 3D code. This was done as a work around to de-bug and integrate the SST turbulence model.

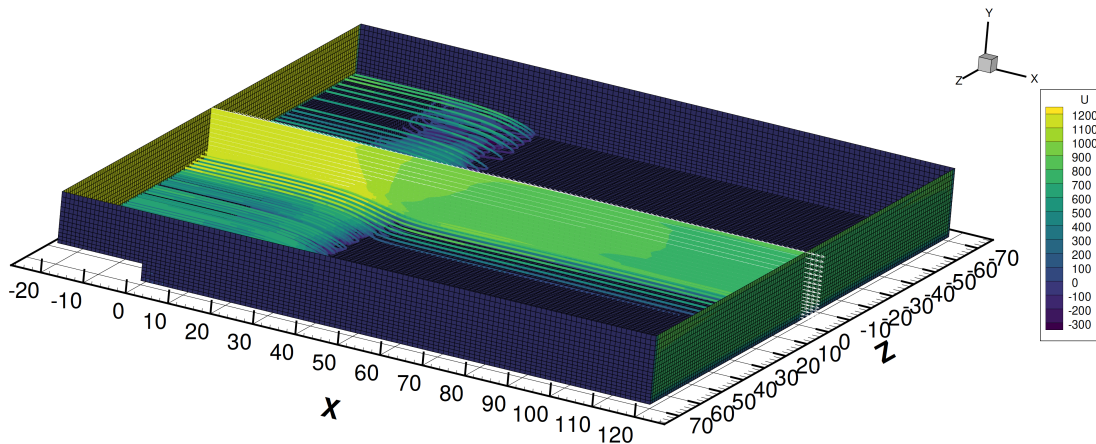


Figure 5.3: U -component of velocity [cm/s] streamline plot for the SST FEARCE turbulence model on a 375,000 element, wall bounded 3D GridPro grid at $t = 0.33s$. The solution looks very promising, further study will need to be continued with the addition of the SST model to the parallel FEARCE code.

The 2D backward facing step is somewhat likened to the solution one obtains from taking a center line plane cut of the 3D simulation shown in Fig. (5.3). However,

the current study and application of symmetry type boundary conditions is more representative of a 2D solution. In the future the code can be easily run in 3D when the SST subroutines are added to the primary FEARCE parallel LES code. Coarse grids have already been tested and working well, they are not shown in this manuscript due to the long computational time needed to allow the simulation to fully develop when the code is run in serial.

Chapter 6

Conclusions, Solutions and Results for 2D Duct

6.1 Solutions and Results for 2D Duct Flow

The solutions for the 2D duct channel flow in comparison to both DNS Mansour et al. [3] and Wilcox k - ω [2] are presented in this section. The 2D duct was used as a precursor to assist the 2D BFS study. It helped de-bug wall-function problems that arose and offered numerous insights into the codes behavior useful in predicting the more complicated 2D BFS flow in section (7.1). The 2D duct channel flow for the two-equation FEARCE SST was at Reynolds number 13,750 as determined by hydraulic inlet diameter of 1 cm . The inlet was prescribed a U velocity of 2060 [cm/s] and the density, dynamic viscosity and turbulence inlet settings and clips follow identically to that of the 2D BFS FEARCE SST parameters in the next section (7.1). The grid was generated using GridPro and consisted of 1684 nodes and 784 elements.

The turbulent energy production and dissipation, or creation and destruction as it is sometimes referred to, is shown in Fig. (6.3). One must take into account the coarse grid that was used for the 2D BFS FEARCE SST result. The dip in

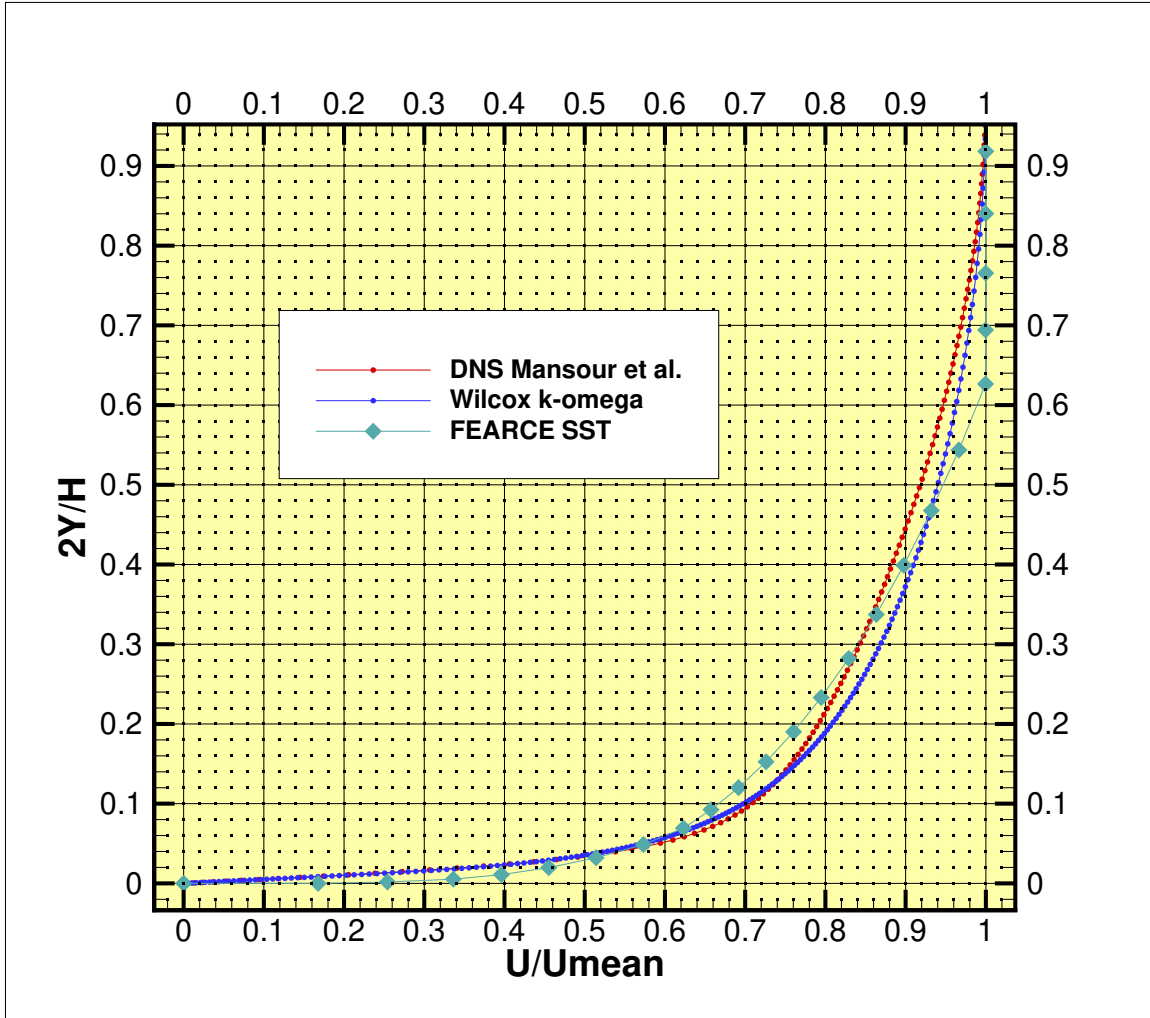


Figure 6.1: Duct channel flow $Re=13,750$ comparing the two-equation FEARCE *SST* code vs. Wilcox $k-\omega$ [2] vs. DNS Mansour et al. [3] at 22 hydraulic diameters from the inlet.

the production term is likely due to the span-wise gradient of the grid. Wilcox [2] defines P^+ and e^+ as the sublayer scaled value of turbulent energy production and dissipation, respectively. The terms plotted on the vertical axis of Fig. (6.3) are defined by equations (6.1) and (6.2). The predictions match reasonably well to both [2] vs. DNS Mansour et al. [3] considering this data is from a significantly finer grids. The reason to use a coarse grid for the turbulent energy production and dissipation data shown in Fig. (6.3) is because many changes to the turbulence parameters and clips were needed to be tested in order to match the DNS Mansour et al. [3] and Wilcox [2]

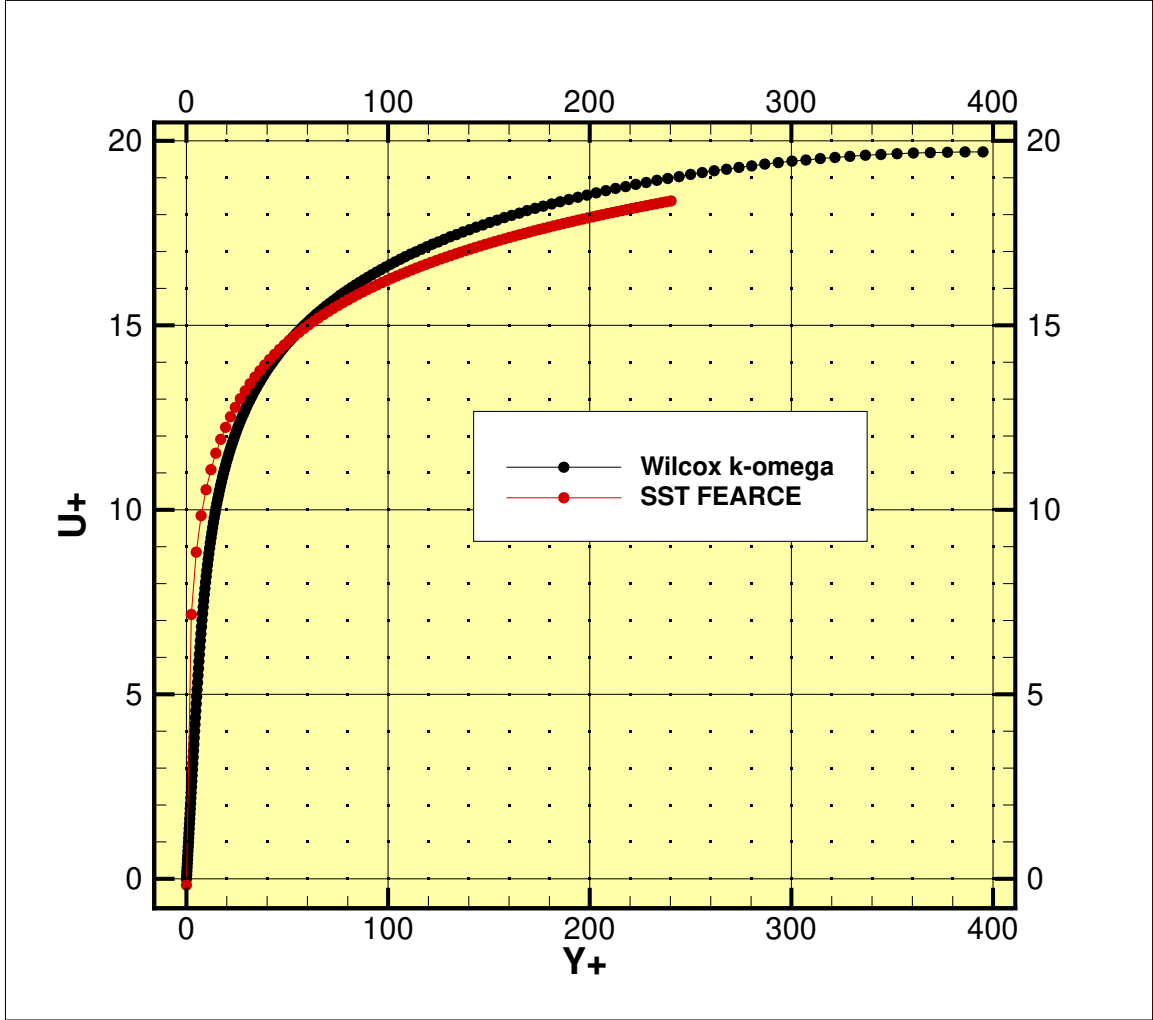


Figure 6.2: U^+ vs. Y^+ comparison of the two-equation FEARCE *SST* code vs. Wilcox $k-\omega$ [2].

data. A couple days per simulation on a serial code using a fine 2D channel duct GridPro grid was not accessible to tune the turbulence parameters and clips shown in (7.1).

$$(6.1) \quad P^+ = \nu \tau_{xy} \frac{\left(\frac{\partial u}{\partial y}\right)}{\mu_\tau^4}$$

$$(6.2) \quad \epsilon^+ = \frac{\nu \epsilon}{\mu_\tau^4}$$

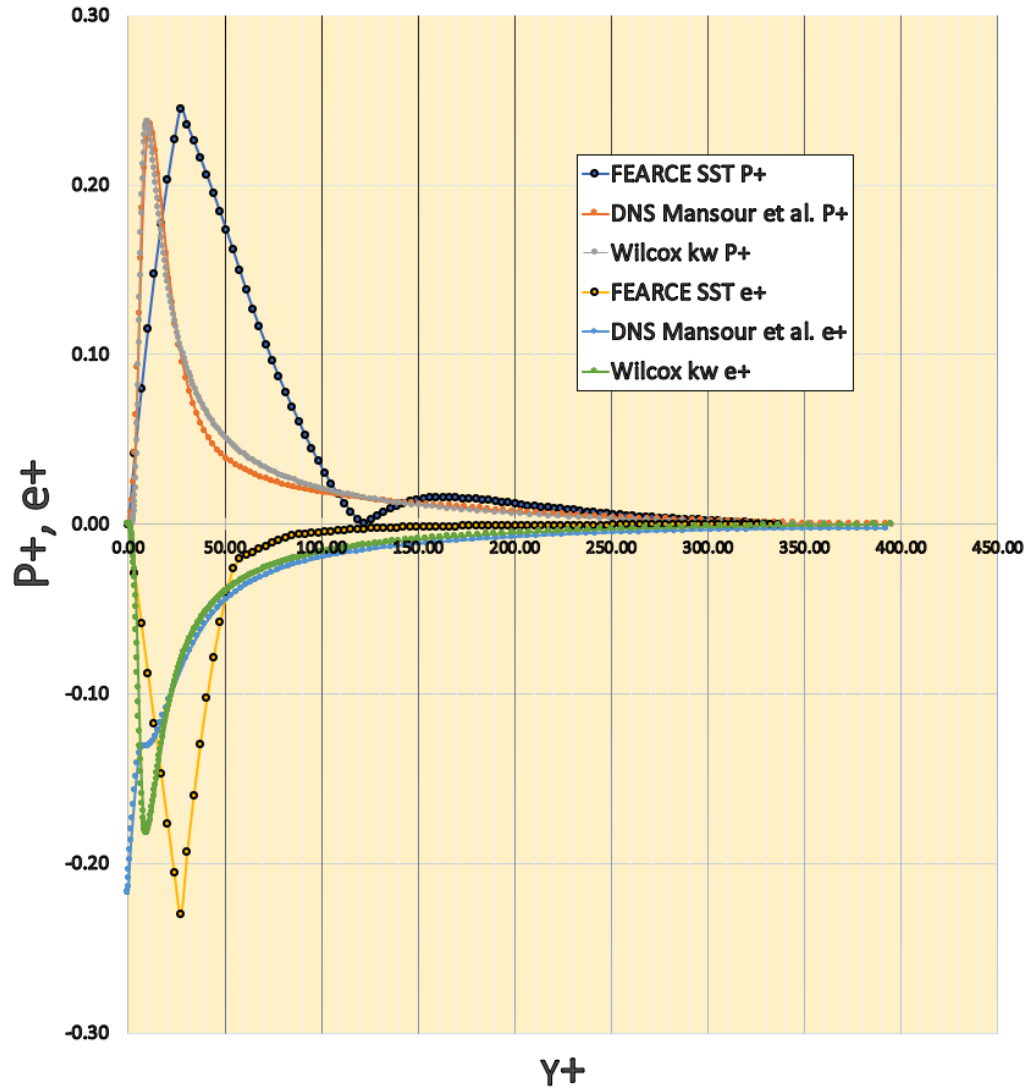


Figure 6.3: Turbulent energy production $P+$ vs. turbulent energy dissipation $e+$ for duct channel flow at $Re=13,750$ comparing the two-equation FEARCE *SST* code vs. Wilcox $k-\omega$ [2] vs. DNS Mansour et al. [3].

Chapter 7

Conclusions, Solutions and Results for 2D BFS

7.1 Solutions and Results for 2D BFS flow

In this section the solution for the 2D BFS for the incompressible flow regime with turbulence is shown. This result seeks to demonstrate the PCS algorithm with newly added SST turbulence model and show, to a reasonable level, validation of the SST turbulence model with the PCS method in the FEARCE code using semi-structured, automated GridPro grids.

In Fig. (7.1), the FEARCE PCS algorithm with the SST turbulence model is demonstrating 2D turbulent flow over a backward facing step as experimentally studied by Vogel and Eaton [1]. The Reynolds number is $Re = 28,000$ as determined by a step height $H = 3.8$ cm, with temperature entering the domain at 298 K, and $Pr = 0.71$. Downstream an applied constant heat flux surface behind the single-sided sudden expansion is prescribed 270 W/m². The insulating upper, and lower walls are given non-slip, Dirichlet, boundary conditions. The inflow face and corresponding nodes are applied the inlet U -velocity of 11.3 cm/s, while at the outflow boundary

located at $X = 26.31H$ behind the step, there is a zero gradient in the weak form sense for both velocity and heat transport. The outflow boundary face is assigned the pressure outflow 2D surface property in GridPro, and the inflow boundary given the velocity inflow 2D surface property. The symmetry walls are those that sit adjacent to the inflow, outflow, top, and bottom walls. They can also be described as sitting to the left and right of the inflow face, perpendicular to the Z -axis. The symmetry condition is set as periodD to the left wall and periodF to the right wall in the 2D surface property tab in GridPro. The nodes are saved in sets and converted using a GridPro to FEARCE converter after initially being exported as the FEARCE file type from GridPro. The non-slip walls are applied by default and the heat flux face is set as TFLX under the same GridPro property tab. In this process great care is taken to correctly form the adjacency node and face lists. The side walls then have their respective normal component of velocity, in this case the W -velocity component, set to zero. In the FEM and weakening process the zero W -velocity component forces a symmetry boundary condition on the side walls, mimicking a 2D solution within a 3D code.

The solutions that are presented for the 2D BFS FEARCE SST model at Reynolds number $Re = 28,000$, are determined for a density $\rho = 0.0012323 [g/cm^3]$, dynamic viscosity $1.846e-4 [g/(cm \cdot s)]$ that correspond to air. The inlet turbulent kinetic energy, k_{inlet} , is determined by a scale factor 0.00216 multiplied by the inlet velocity squared. The resulting free stream turbulent kinetic energy is then clipped to be above 0.0005 for this particular problem, and also the channel flow duct benchmark . The specific dissipation rate at the inlet was calculated based on $\mu_{scalein}$, where $\mu_{tinlet} = \mu_{scalein} \cdot \mu$, then $\omega_{inlet} = k_{inlet}/\mu_{tinlet}$. The reattachment could then be adjusted by ω_{clip} . By increasing or decreasing the ω_{clip} value the reattachment can be lengthened or shortened. The value for ω_{clip} was tested to be 50 for the 2D SST FEARCE code which yielded the corresponding predicted reattachment at $X_{re} = 6.655H$.

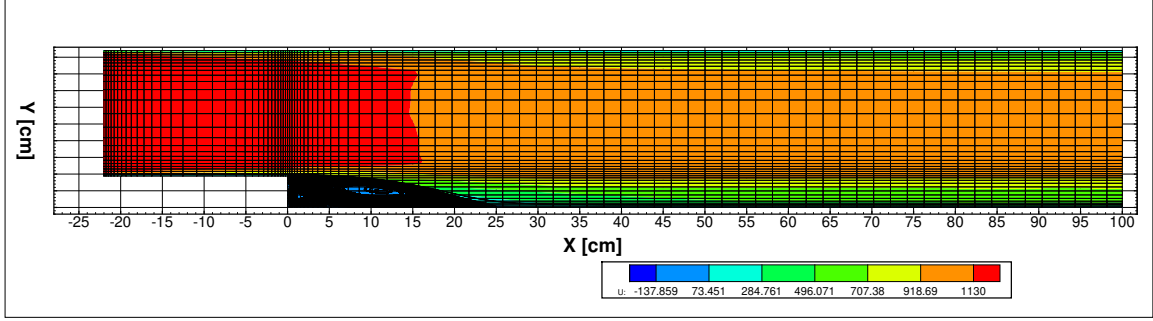


Figure 7.1: U -velocity [cm/s] BFS shown at $Re=28,000$ for the two-equation FEARCE SST code at solution time $1.35 [sec]$. The solution reaches state state appreciably quicker than $k-\omega h$ -adapted FEM [4] that reported stable recirculation at around $0.33 [sec]$. The two-equation FEARCE SST code finds stable recirculation at approximately $0.15 [sec]$. At solution time $1.35 [sec]$ the max L_2 norm was $4.32e - 12$, and the FEARCE SST code solution was considered converged. The reattachment is $25.29 cm$ corresponding to a $X_{re} = 6.655H$.

The chemical species composition for the fluid is that of air. The chemical species and corresponding mass fractions at $1.060e + 6 dynes/cm^2$ are shown in Table (7.1). The fluid is modeled with two species, and tests the species aggregation process [58].

Table 7.1: Species Composition for Air.

species 1	species 2	species 3
O_2	N_2	Ar
0.22	0.78	0.0

A wall function [4] with Wilcox [2] adverse pressure gradient formulation is used. The wall function requires we solve a transcendental stress equation for the wall friction velocity or wall shear stress τ_w . This is a Picard iterative procedure [77] that is formed at each time step.

The FEARCE SST code is shown in Fig. (7.1) for the BFS at $Re=28,000$ for the two-equation FEARCE SST code at solution time $1.35 [sec]$. The solution reaches state state appreciably quicker than $k-\omega h$ -adapted FEM [4] that reported stable recirculation at around $0.33 [sec]$. The two-equation FEARCE SST code reaches

a stable recirculation pattern at approximately 0.15 [sec]. At solution time 1.35 [sec] the max L_2 norm was $4.32e - 12$, and the FEARCE *SST* code solution was considered converged. The measured point of reattachment Fig. (7.2) was 25.29 cm corresponding to a $X_{re} = 6.655 H$ in comparison to the experimentally determined $X_{re_{vogel}} = 6.66667 H$ from Vogel and Eaton [1], it is in considerably good agreement. Typical simulation reattachments have been seen ranging from 6.8 to 7.2. This is a considerable improvement considering the data comparison to $k-\omega$ h -adapted FEM [4] result was performed with a final h -adapted grid approximately 153,000 nodes, and the two-equation FEARCE *SST* code used a semi-uniform, locally gradient GridPro grid with only 6014 nodes and 2880 elements.

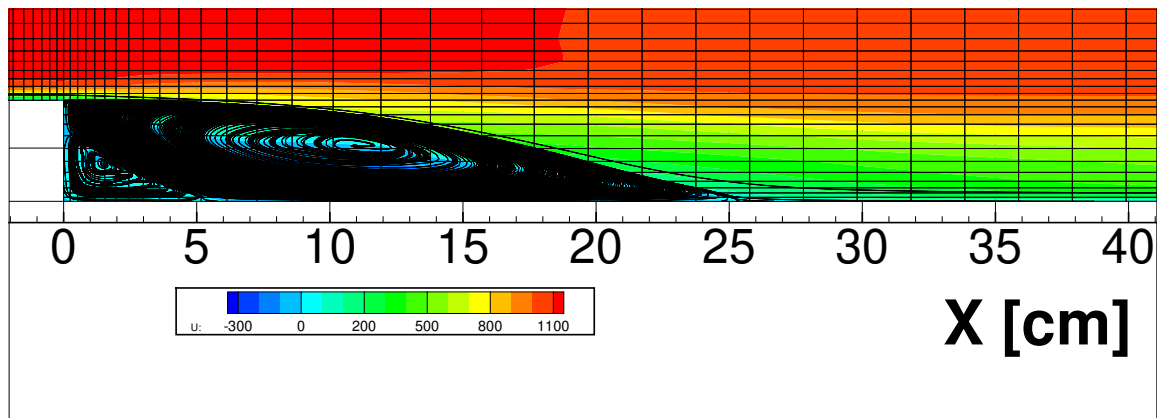


Figure 7.2: U -velocity [cm/s] BFS reattachment shown at $Re = 28,000$ for the two-equation FEARCE *SST* code at solution time 1.35 [sec]. At solution time 1.35 [sec] the max L_2 norm was $4.32e - 12$, and the FEARCE *SST* code solution was considered converged. The measured point of reattachment was 25.29 cm corresponding to a $X_{re} = 6.655 H$ in comparison to the experimentally determined $X_{re_{vogel}} = 6.66667 H$ from Vogel and Eaton [1], it is in considerably good agreement. Typical simulation reattachments for Vogel and Eaton [1] at $Re = 28,000$ have been seen ranging from $X_{re} = 6.8H$ to $7.2H$.

The conservation of mass states that the average mass flow rate is the same if at steady state or if incompressible, also the average speed is the same if the material is constant density. The verification analysis in Fig. (7.3) and in Fig. (7.4) was performed to ensure that the principle mass conservation is upheld for the incompressible

flow SST turbulence model. The principle mass conservation is clearly shown to be upheld for the SST turbulence model. The initial bump at the downstream location $X/H = 20$ is simply due to the fluid that is occupying that location in the domain at the initial start of the simulation. As the simulation proceeds the mass flow rate quickly approaches steady state at both sliced areas, $X/H = -3$ and $X/H = 20$.

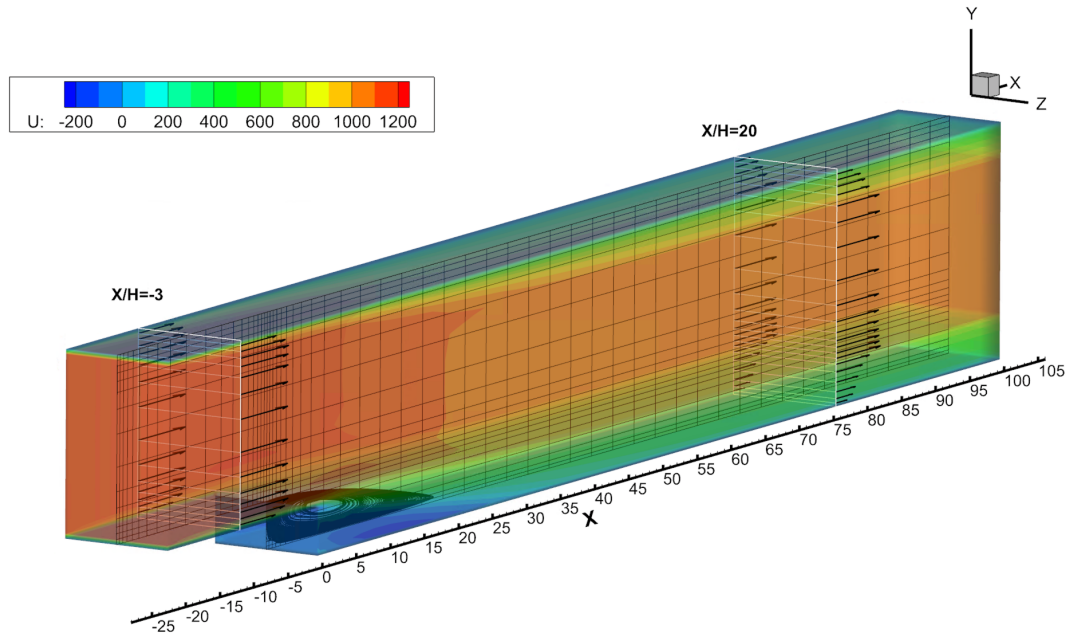


Figure 7.3: U -velocity [cm/s] BFS shown at $Re=28,000$ for the two-equation FEARCE SST code where the location of slices at $X/H = -3$ and $X/H = 20$ were integrated over to calculate the mass flow rate for Fig. (7.4). The location $X/H = -3$ was chosen for the reference velocity U_{ref} by Vogel and Eaton [1] as the mean velocity for the velocity profile analysis.

Fig. (7.5) shows the temperature and Fig. (7.6), and Fig. (7.7) show the thermal profiles for the backward facing step for $Re=28,000$ comparing the two-equation FEARCE SST code vs. $k-\omega$ h -adapted FEM [4], at $X/H = 0.33$ and $X/H = 4.3$, respectively.

Fig. (7.8) shows the velocity profiles for the backward facing step for $Re=28,000$ comparing the two-equation FEARCE SST code vs. $k-\omega$ h -adapted FEM [4], at $x = -0.67x_{re}$, $x = -0.33x_{re}$, $x = 0.0x_{re}$, and $x = 0.33x_{re}$. The velocity profiles are also compared to the experimental data from Vogel and Eaton [1] for $x = -0.67x_{re}$,

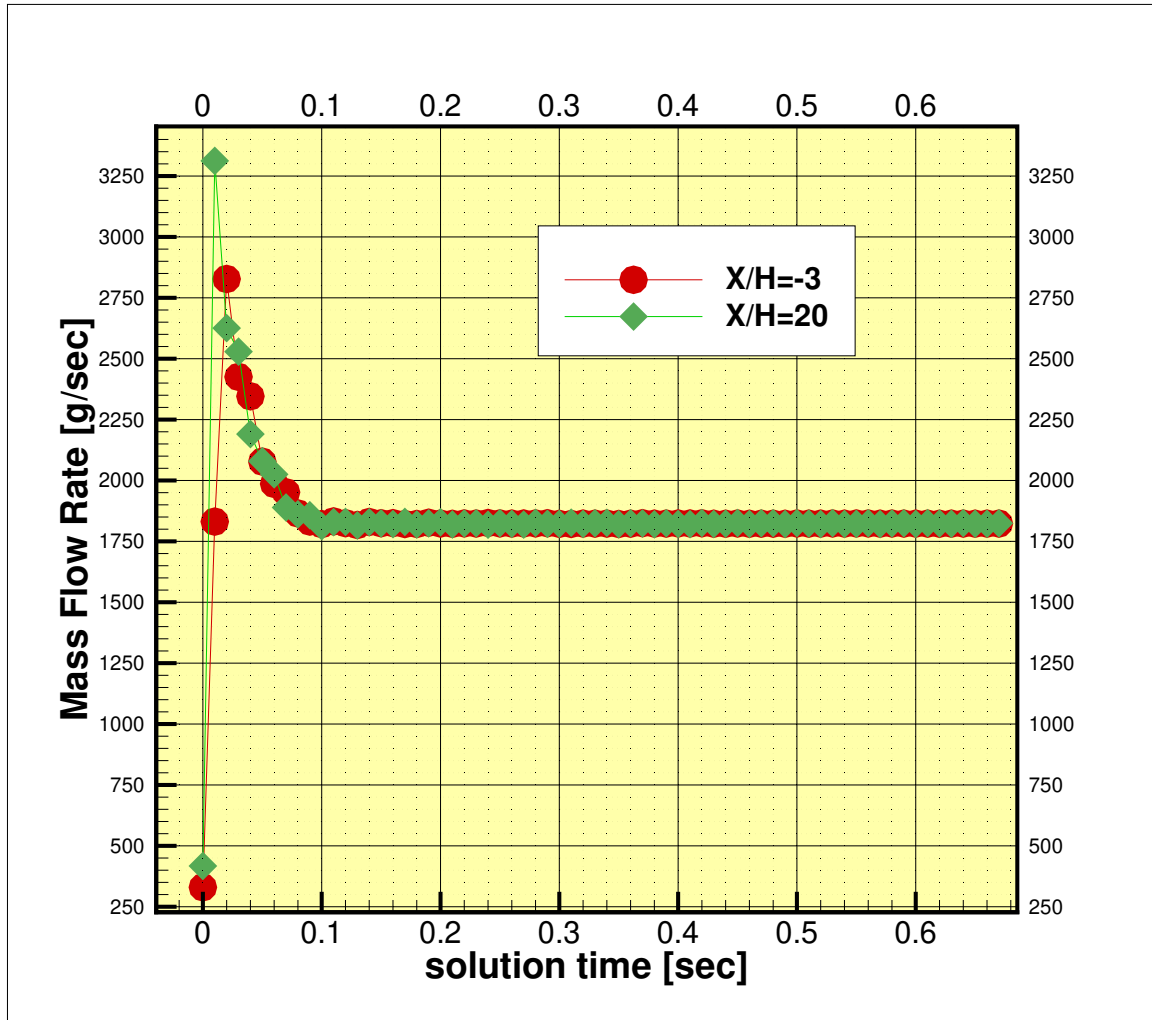


Figure 7.4: BFS mass flow rate analysis for comparing the mass flow rates at $X/H = -3$ and $X/H = 20$ at $Re=28,000$ for the two-equation FEARCE *SST* code.

$x = -0.33x_{re}$, $x = 0.0x_{re}$, and $x = 0.33x_{re}$. The reference velocity U_{ref} was taken at $X/H = -3$ as described in [1].

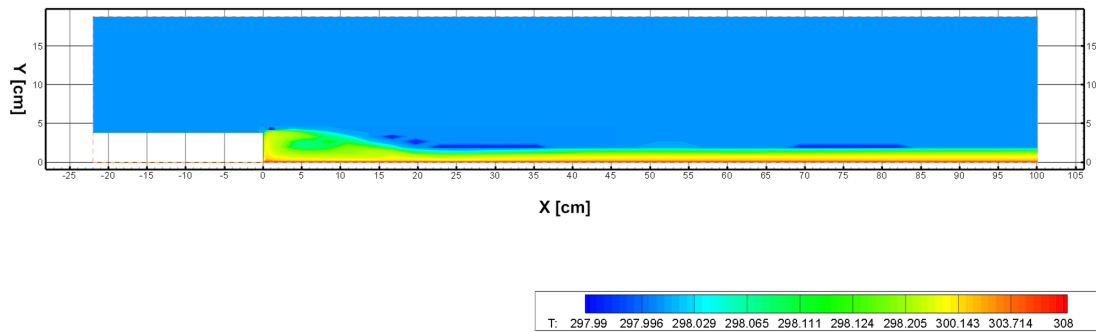


Figure 7.5: Temperature $[K]$ plot for the backward facing step at $Re=28,000$ for the two-equation FEARCE *SST*.

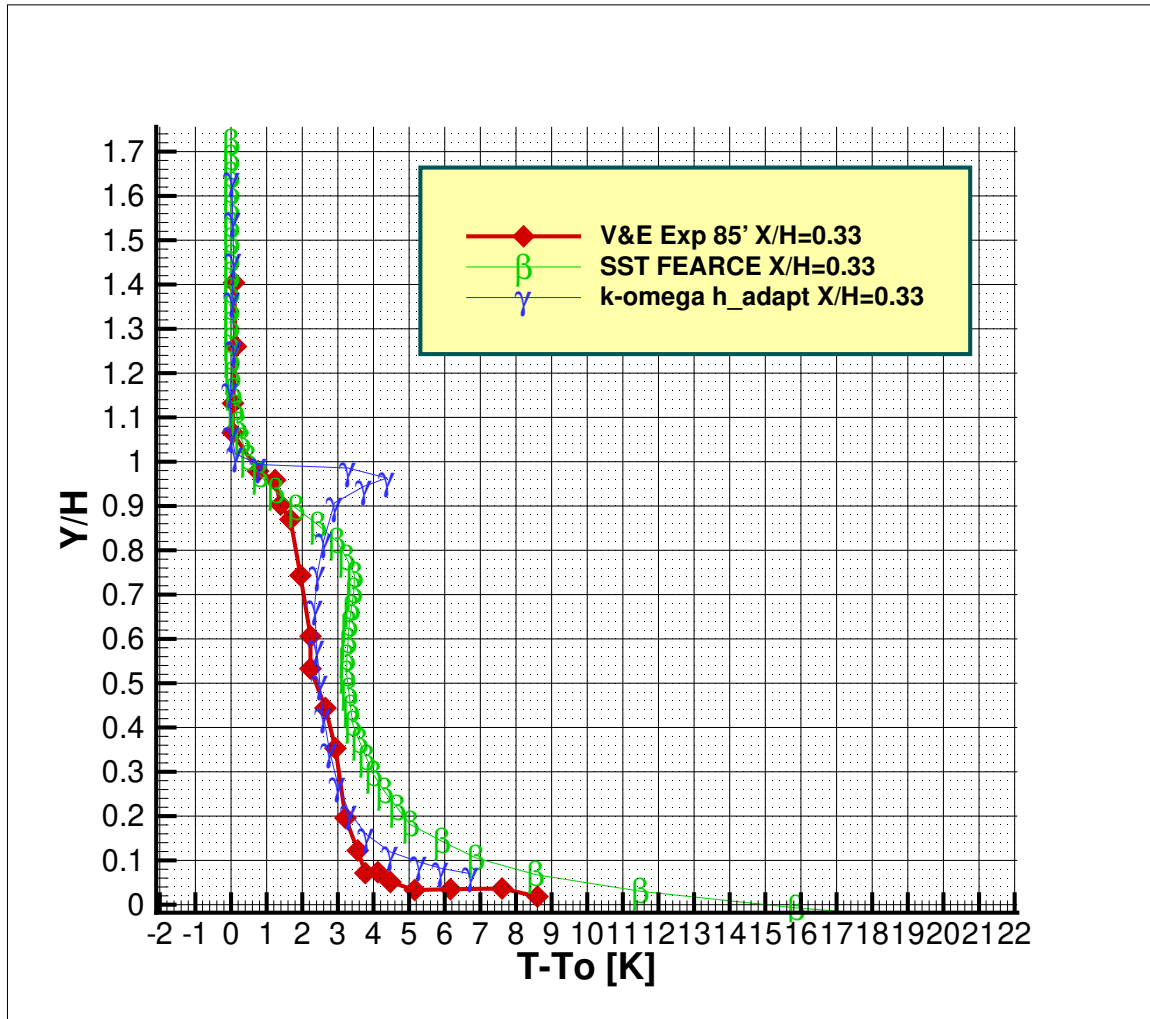


Figure 7.6: Thermal profiles for the backward facing step for $Re=28,000$ comparing the two-equation FEARCE *SST* code vs. $k-\omega$ h -adapted FEM [4]. The thermal profiles are also compared to the experimental data from Vogel and Eaton [1] at $X/H = 0.33$.

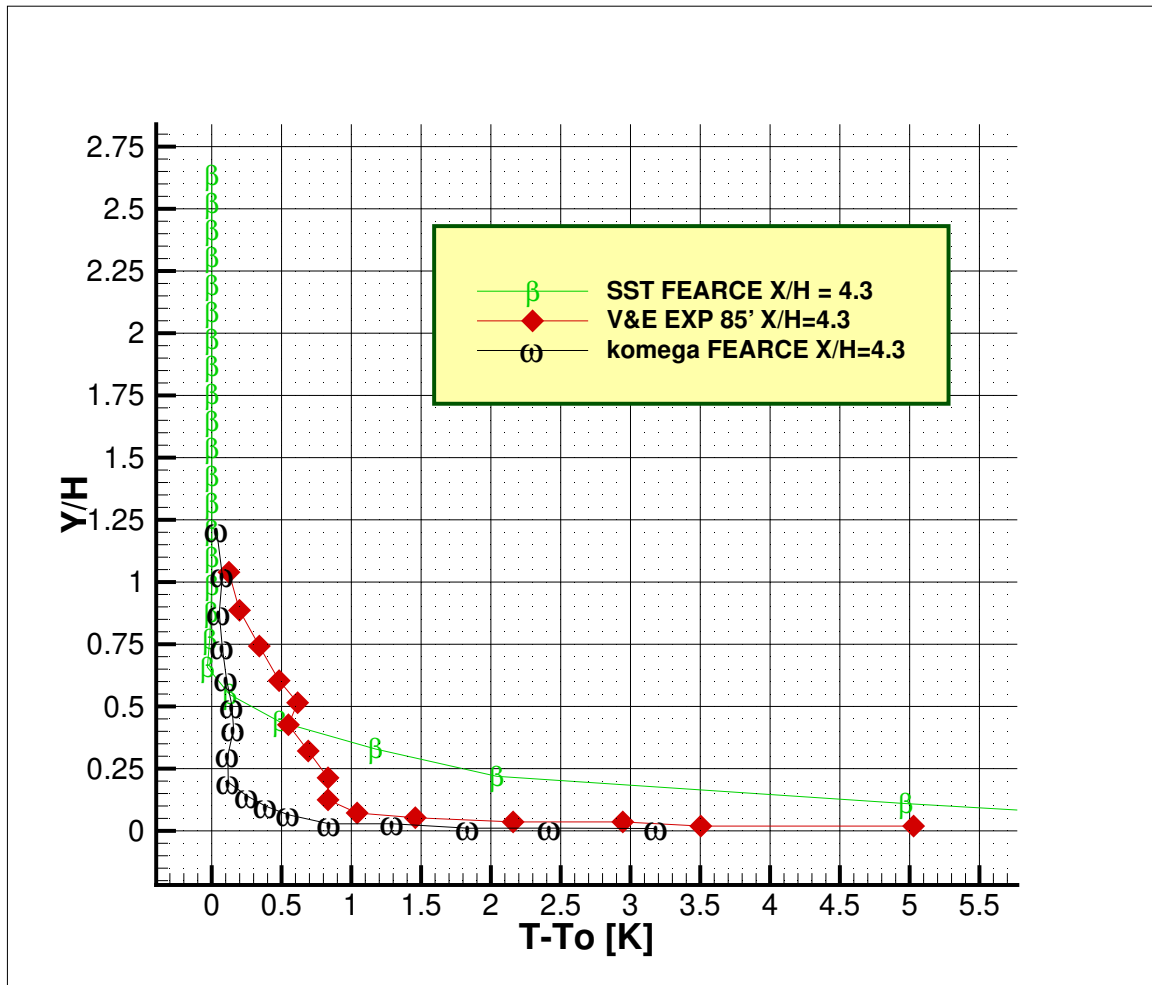


Figure 7.7: Thermal profiles for the backward facing step for $Re=28,000$ comparing the two-equation FEARCE *SST* code vs. FEARCE *k- ω* non-*h*-adapted FEM [4]. The thermal profiles are also compared to the experimental data from Vogel and Eaton [1] at $X/H = 4.3$.

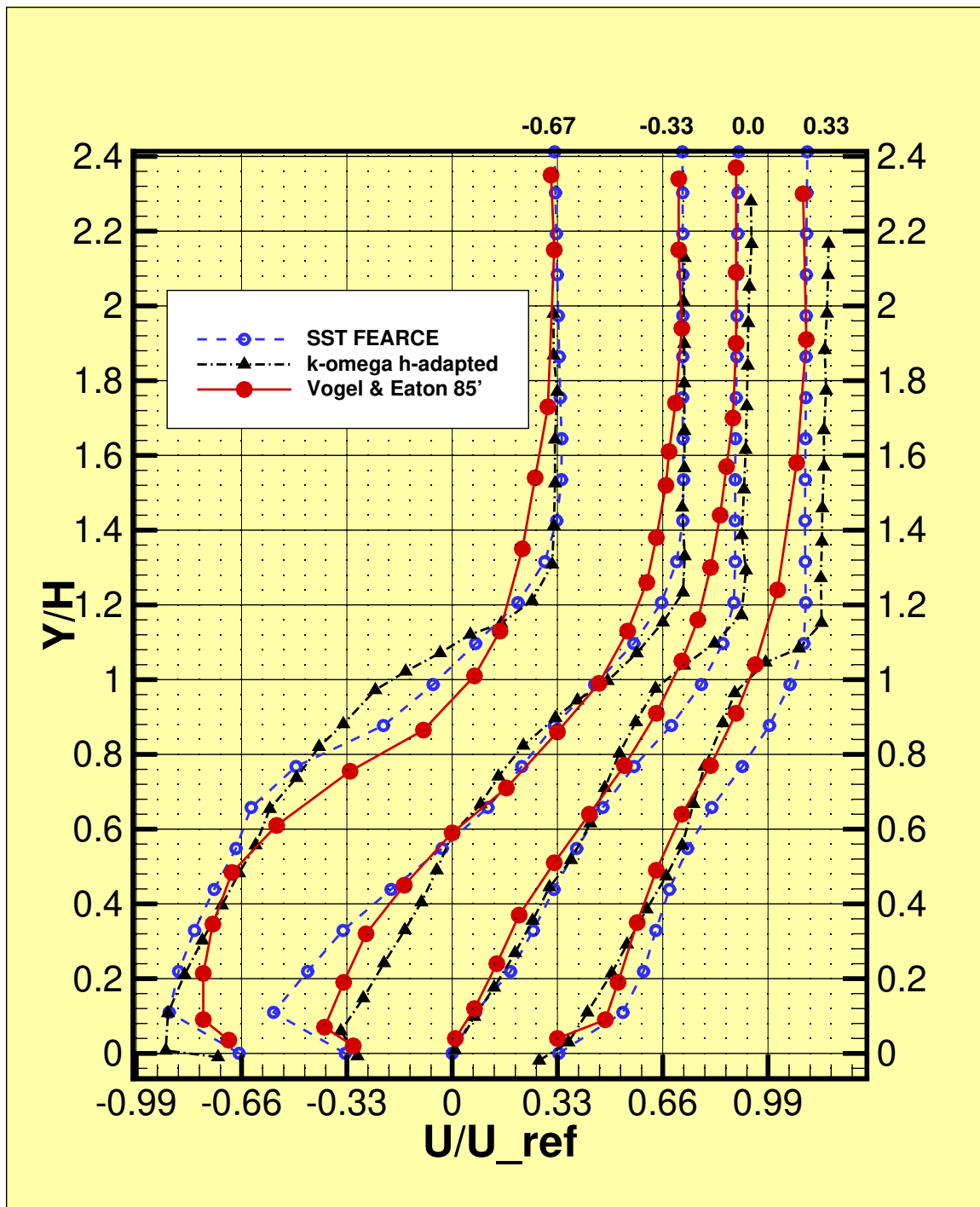


Figure 7.8: U -component velocity over U_{ref} velocity profiles for the backward facing step at $Re=28,000$ comparing the two-equation FEARCE SST code on a GridPro grid with 6014 nodes vs. h -adapted $k-\omega$ FEM [4] with a final adapted node count of 153,000 nodes. The velocity profiles are also compared to the experimental data from Vogel and Eaton [1] for $x = -0.67x_{re}$, $x = -0.33x_{re}$, $x = 0.0x_{re}$, and $x = 0.33x_{re}$.

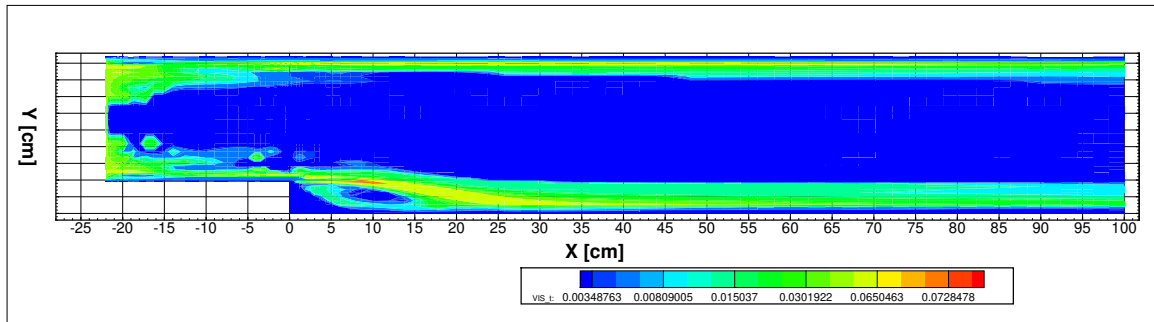


Figure 7.9: Turbulent viscosity plot for the backward facing step at $Re=28,000$ for the two-equation FEARCE *SST*.

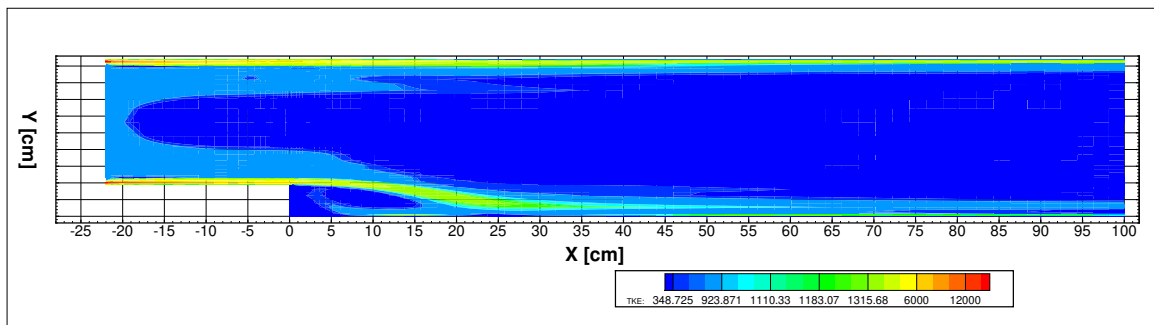


Figure 7.10: Turbulent kinetic energy plot for the backward facing step at $Re=28,000$ for the two-equation FEARCE *SST*.

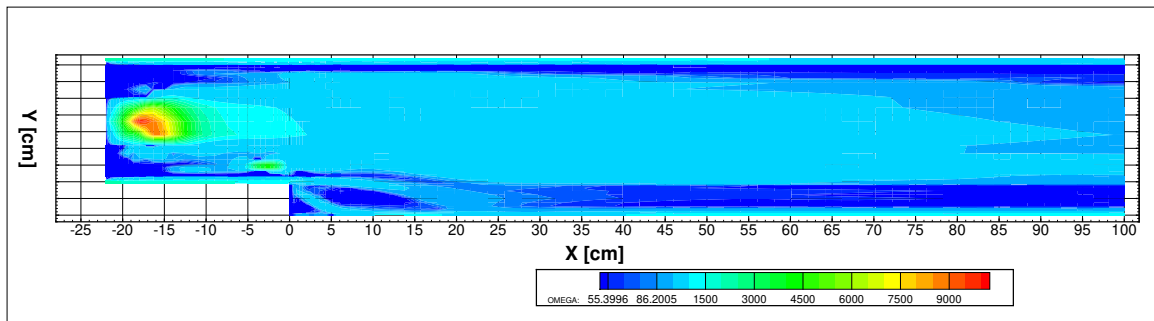


Figure 7.11: Turbulent dissipation plot for the backward facing step at $Re=28,000$ for the two-equation FEARCE *SST*.

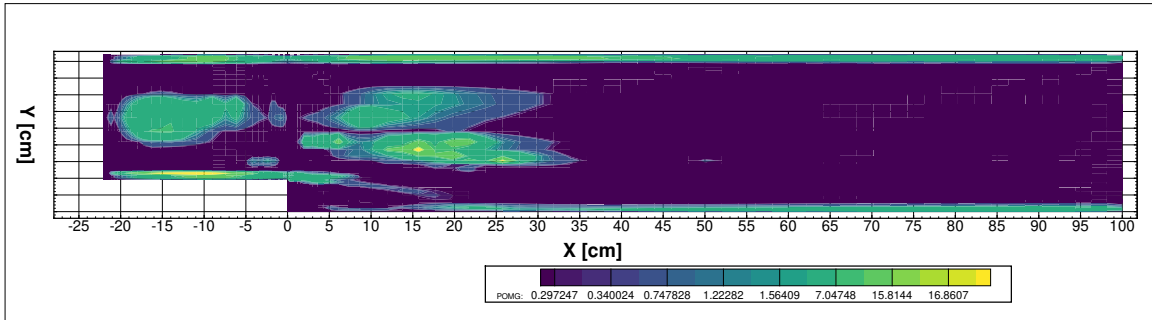


Figure 7.12: Turbulent energy dissipation plot for the backward facing step at $Re=28,000$ for the two-equation FEARCE *SST*.

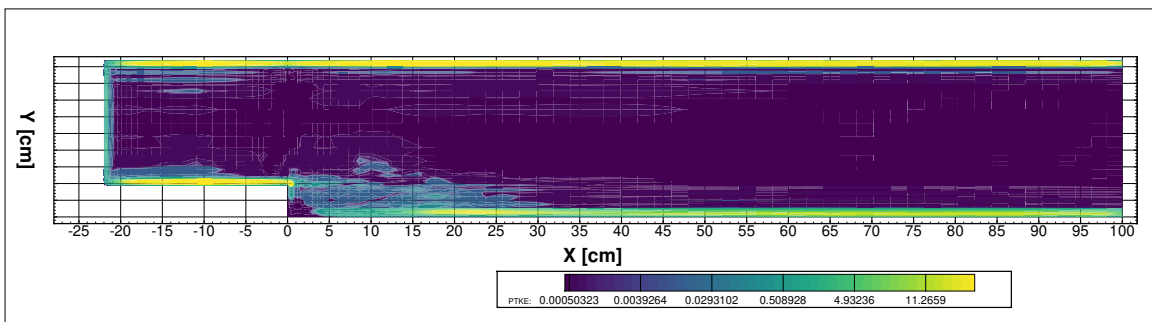


Figure 7.13: Turbulent energy production plot for the backward facing step at $Re=28,000$ for the two-equation FEARCE *SST*.

Appendix A

Appendix: Coding the Improved MSST

A.1 MSST Research Outline

The purpose of this appendix is to aid any future development and give a detailed look into the process, bug fixes and code changes to implement the SST into the existing $k-\omega$ FEARCE code.

A.1.1 Minor Bug Fixes

Looking at PRODOMG in file:

TwoEq_Turbulence_Wall_ALE_Sub_092117SUPG.f90

and comparing it to either of...

<https://turbmodels.larc.nasa.gov/wilcox.html>

<http://highorder.berkeley.edu/proceedings/aiaa-annual-2007/paper0547.pdf>

the following line...

$$+ \text{WZ}*(\text{MUTT}*((4.0*\text{WZ}/3.0)-(2.0/3.0)*(\text{UX}*\text{VY})) \&$$

is incorrect. It should read...

$$+ WZ*(MUTT*((4.0*WZ/3.0)-(2.0/3.0)*(UX+VY)) \&$$

The line occurs on lines...

1565

1783

2002

2214

in the following subroutines...

SUBROUTINE ASSMEB_TURBULENCE_DISS_Hexsub

SUBROUTINE ASSMEB_TURBULENCE_DISS_cHexsub

SUBROUTINE ASSMEB_TURBULENCE_DISS_Prismsub

SUBROUTINE ASSMEB_TURBULENCE_DISS_Pyramidsub

PRODTKE is defined 4 times depending on the subroutine...

SUBROUTINE ASSMEB_TURBULENCE_TKE_Hexsub

SUBROUTINE ASSMEB_TURBULENCE_TKE_cHexsub

SUBROUTINE ASSMEB_TURBULENCE_TKE_Prismsub

SUBROUTINE ASSMEB_TURBULENCE_TKE_Pyramidsub

The first subroutine uses NODE, but the final 3 times use NODE1. Also, the following factors are different when defining PRODTKE...

- * NHex(KK,IQ)*detJac(K,IQ)
- * NHex(KK,IQ)*detHex(m,IQ)
- * NPri(KK,IQ)*PridetJac(m,IQ)
- * NPyr(KK)*volpyr(m)

Each of the 4 instances of the following line...

$$-(2.0/3.0)*(UX*WZ))-(2.0*RhoTilda*TKEN/3.0)) \&$$

should be...

$$-(2.0/3.0)*(UX+WZ))-(2.0*RhoTilda*TKEN/3.0)) \&$$

The line occurs on lines...

425

A.1.2 Coding MSST-Step1:

Summary: Add something to PRODOMG and pass TkeSub, OmgSub to many more subroutines. Then calculate KX,OX, KY,OY, KZ, and OZ in each of the 8 subroutines.

Looking in file... TwoEq_Turbulence_Wall_ALE_Sub_092117SUPG.f90

Production term of interest is PRODOMG.

PRODTKE is defined 4 times depending on the subroutine...

SUBROUTINE ASSMEB_TURBULENCE_TKE_Hexsub

SUBROUTINE ASSMEB_TURBULENCE_TKE_cHexsub

SUBROUTINE ASSMEB_TURBULENCE_TKE_Prismsub

SUBROUTINE ASSMEB_TURBULENCE_TKE_Pyramidsub

The first subroutine uses NODE, but the final 4 times use NODE1. Also, the following factors are different when defining PRODTKE...

- * NHex(KK,IQ)*detJac(K,IQ)
- * NHex(KK,IQ)*detHex(m,IQ)
- * NPri(KK,IQ)*PridetJac(m,IQ)
- * NPyr(KK)*volpyr(m)

PRODOMG is defined 4 times depending on the subroutine...

SUBROUTINE ASSMEB_TURBULENCE_DISS_Hexsub

SUBROUTINE ASSMEB_TURBULENCE_DISS_cHexsub

SUBROUTINE ASSMEB_TURBULENCE_DISS_Prismsub

SUBROUTINE ASSMEB_TURBULENCE_DISS_Pyramidsub

The first subroutine uses NODE and $\text{TKEN} > 1.0\text{e-}06$, but the final 4 times use NODE1 and $\text{TKEN} > 1.0\text{e-}03$. Also, the following factors are different when defining PRODTKE...

- * NHex(KK,IQ)*detJac(K,IQ)
- * NHex(KK,IQ)*detHex(m,IQ)
- * NPri(KK,IQ)*PridetJac(m,IQ)
- * NPyR(KK)*volpyr(m)

The term we have to add to PRODOMG is...

$$2 (1 - F1) \rho \sigma_{\omega 2} / \omega (dk/dx d\omega/dx + dk/dy d\omega/dy + dk/dz d\omega/dz)$$

Where the following parameters are not yet in the ReactCFD code...

- $\sigma_{\omega 2} = 0.856$, and let's call this SigOmega2
- F1 is the blending parameter (which will have to be dynamically determined)

The following are in ReactCFD code...

- ω is called OMEG
- ρ is called RhoTilda ???
- dk/dx_i are called KX, KY, and KZ
- $d\omega/dx_i$ are called OX, OY, and OZ

So we get that the term to add to PRODOMG is (would be added 4 times)...

$$2 *(1 - F1_MSST) *RhoTilda*Sig2 / OMEG*(KX*OX + KY*OY + KZ*OZ)$$

For example the following code:

```
DO KK = 1, NUMN
  L=NODE(K, KK)
  IF ( TKEN > 1.0e-06 ) THEN
    PRODOMG(L) = PRODOMG(L) + ( ( Alp*OMEG/TKEN )      &
```

```

* ( UX*( MUTT*( (4.0*UX/3.0)-(2.0/3.0)*(VY+WZ) ) &
      -(2.0*RhoTilda*TKEN/3.0) ) &
+ VY*( MUTT*( (4.0*VY/3.0)-(2.0/3.0)*(UX+WZ) ) &
      -(2.0*RhoTilda*TKEN/3.0) ) &
+ WZ*( MUTT*( (4.0*WZ/3.0)-(2.0/3.0)*(UX+VY) ) &
      -(2.0*RhoTilda*TKEN/3.0) ) &
+ MUTT*(UY*UY+2.0*UY*VX+VX*VX) & ! tau 1 2 + tau 2 1
+ MUTT*(VZ*VZ+2.0*VZ*WY+WY*WY) & ! tau 2 3 + tau 3 2
+ MUTT*(UZ*UZ+2.0*UZ*WX+WX*WX) ) & ! tau 1 3 + tau 3 1
* NHex(KK,IQ)*detJac(K,IQ)
PRODOMG(L) = ABS(PRODOMG(L))
END IF
END DO

```

The code is changed into the following:

```

DO KK = 1, NUMN
L=NODE(K, KK)
IF ( TKEN > 1.0e-06 ) THEN
PRODOMG(L) = PRODOMG(L) + ( ( Alp*OMEG/TKEN ) &
* ( UX*( MUTT*( (4.0*UX/3.0)-(2.0/3.0)*(VY+WZ) ) &
      -(2.0*RhoTilda*TKEN/3.0) ) &
+ VY*( MUTT*( (4.0*VY/3.0)-(2.0/3.0)*(UX+WZ) ) &
      -(2.0*RhoTilda*TKEN/3.0) ) &
+ WZ*( MUTT*( (4.0*WZ/3.0)-(2.0/3.0)*(UX+VY) ) &
      -(2.0*RhoTilda*TKEN/3.0) ) &
+ MUTT*(UY*UY+2.0*UY*VX+VX*VX) & ! tau 1 2 + tau 2 1
+ MUTT*(VZ*VZ+2.0*VZ*WY+WY*WY) & ! tau 2 3 + tau 3 2
+ MUTT*(UZ*UZ+2.0*UZ*WX+WX*WX) ) +

```

```

2 *(1 - F1_MSST) *RhoTilda*Sig2 / OMEG*(KX*OX + KY*OY + KZ*OZ) ) & !
tau 1 3 + tau 3 1
    * NHex(KK,IQ)*detJac(K,IQ)
    PRODOMG(L) = ABS(PRODOMG(L))
    END IF
END DO

```

Notice that there is a KX and an OX in the new terms. Because of this and because other MSST calculations will require both KX and OX, make the appropriate lines from the KIVAhpFE3d_main_071216.f90 be the following:

```

allocate( TkeSub(NNODE) )
allocate( OmgSub(NNODE) )
Do I = 1, NUMSUBCYCLERANS
    TkeSub(1:NNODE) = TKE(1:NNODE)
    OmgSub(1:NNODE) = OMG(1:NNODE)
    Call ASSMEB_TURBULENCE_TKE_Sub(TkeSub, OmgSub)
End Do

Do I = 1, NUMSUBCYCLERANS
    TkeSub(1:NNODE) = TKE(1:NNODE)
    OmgSub(1:NNODE) = OMG(1:NNODE)
    Call ASSMEB_TURBULENCE DISS_Sub(OmgSub, TkeSub)
End Do

deallocate( TkeSub )
deallocate( OmgSub )

```

In the TwoEq_Turbulence_Wall_ALE_Sub_092117SUPG.f90 TKEsub, do the following replacements.

1 time:

Call VIRTUAL_NODES_CALCULATION(OMGsub)

Call VIRTUAL_NODES_CALCULATION(OMGsub)

9 times:

TkeSub)

TkeSub, OmgSub)

9 times:

OmgSub)

OmgSub, TkeSub)

5 times:

DOUBLEPRECISION, dimension(NNODE), Intent(inout) :: TkeSub

DOUBLEPRECISION, dimension(NNODE), Intent(inout) :: TkeSub,OmgSub

5 times:

DOUBLEPRECISION, dimension(NNODE), Intent(inout) :: OmgSub

DOUBLEPRECISION, dimension(NNODE), Intent(inout) :: OmgSub, TkeSub

4 times:

DOUBLEPRECISION :: KY,KX,KZ

DOUBLEPRECISION :: KY,KX,KZ,OX,OY,OZ

4 times:

DOUBLEPRECISION :: OX,OY,OZ

DOUBLEPRECISION :: OX,OY,OZ,KX,KY,KZ

Search the file for the phrase “FIRST DERIVATIVE TERMS”,

1st and 5th should have all of these

$$OX=OX+HexX(K, KK, IQ)*OmgSub(L)$$

$$OY=OY+HexY(K, KK, IQ)*OmgSub(L)$$

$$OZ=OZ+HexZ(K, KK, IQ)*OmgSub(L)$$

$$KX=KX+HexX(K, KK, IQ)*TkeSub(L)$$

$$KY=KY+HexY(K, KK, IQ)*TkeSub(L)$$

$$KZ=KZ+HexZ(K, KK, IQ)*TkeSub(L)$$

2nd and 6th should have all of these

$$OX=OX+dqHexX(m, KK, IQ)*OmgSub(L)$$

$$OY=OY+dqHexY(m, KK, IQ)*OmgSub(L)$$

$$OZ=OZ+dqHexZ(m, KK, IQ)*OmgSub(L)$$

$$KX=KX+dqHexX(m, KK, IQ)*TkeSub(L)$$

$$KY=KY+dqHexY(m, KK, IQ)*TkeSub(L)$$

$$KZ=KZ+dqHexZ(m, KK, IQ)*TkeSub(L)$$

3rd and 7th should have all of these

$$OX=OX+PriX(m, KK, IQ)*OmgSub(L)$$

$$OY=OY+PriY(m, KK, IQ)*OmgSub(L)$$

$$OZ=OZ+PriZ(m, KK, IQ)*OmgSub(L)$$

$$KX=KX+PriX(m, KK, IQ)*TkeSub(L)$$

$$KY = KY + \text{PriY}(m, KK, IQ) * \text{TkeSub}(L)$$

$$KZ = KZ + \text{PriZ}(m, KK, IQ) * \text{TkeSub}(L)$$

4th and 8th should have all of these

$$OX = OX + d\text{PyrX}(m, KK) * \text{OmgSub}(L)$$

$$OY = OY + d\text{PyrY}(m, KK) * \text{OmgSub}(L)$$

$$OZ = OZ + d\text{PyrZ}(m, KK) * \text{OmgSub}(L)$$

$$KX = KX + d\text{PyrX}(m, KK) * \text{TkeSub}(L)$$

$$KY = KY + d\text{PyrY}(m, KK) * \text{TkeSub}(L)$$

$$KZ = KZ + d\text{PyrZ}(m, KK) * \text{TkeSub}(L)$$

A.1.3 Coding MSST-Step 2:

Edit the following module in `KivaVarAllocateModuleMPI_091917.f90`

```

MODULE WILCOX_DATA ! For k-w and MSST turbulence model
IMPLICIT NONE
PUBLIC
DOUBLEPRECISION :: Alp, Beta, Srf, Sig, Sigstr, Clim
DOUBLEPRECISION :: Betastr, Karman
PARAMETER (Alp=5.0d0/9.0d0) ! approximately 5/9
PARAMETER (Beta=3.0d0/40.0d0)
PARAMETER (Betastr=9.0d0/100.0d0)
PARAMETER (Srf=9.0d0)
PARAMETER (Karman=0.41d0)
PARAMETER (Sig=0.5d0)

```

```

PARAMETER (Sigstr=0.5d0)
PARAMETER (Clim=0.875d0)
!
!
!
!      !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!      MSST MODULE !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!      !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

DOUBLEPRECISION :: Beta1,Beta2, BetaBlend
DOUBLEPRECISION :: Sigstr1, Sigstr2, SigstrBlend
DOUBLEPRECISION :: Sig1, Sig2, SigBlend
DOUBLEPRECISION :: Alp1, Alp2, AlpBlend
DOUBLEPRECISION :: a1_MSST, F1_MSST, F2_MSST
DOUBLEPRECISION :: ARG2,Omega_MSST
DOUBLEPRECISION :: distTEMP,ARG1,CD_komg
DOUBLEPRECISION, ALLOCATABLE, DIMENSION(:) :: distSmallestSST,
distSmallestElem
!      DOUBLEPRECISION, ALLOCATABLE, DIMENSION(:,:) :: distAllWalln-
odes

PARAMETER (a1_MSST=0.31d0)
PARAMETER (Beta1=0.075d0)
PARAMETER (Beta2=0.0828d0)
PARAMETER (Sig1=0.5d0)      ! Sig is sigma_omega
PARAMETER (Sig2=0.856d0)
PARAMETER (Sigstr1=0.85d0) ! Sigstr is sigma_k
PARAMETER (Sigstr2=1.0d0)
PARAMETER (Alp1=Beta1/Betastr-Sig1*Karman**2.0d0/(Betastr)**0.5d0) !Alp

```

is gamma

```

    PARAMETER (Alp2=Beta2/Betastr-Sig2*Karman**2.0d0/(Betastr)**0.5d0)
!
!   BSL Step 3
!   PARAMETER (Sigstr1=0.50d0) ! For BSL
!   PARAMETER (F1_MSST=0.0d0) ! For BSL
!   PARAMETER (BetaBlend=F1_MSST*Beta1 + (1.0d0-F1_MSST)*Beta2)
!   PARAMETER (SigstrBlend=F1_MSST*Sigstr1 + (1.0d0-F1_MSST)*Sigstr2)
!   PARAMETER (SigBlend=F1_MSST*Sig1 + (1.0d0-F1_MSST)*Sig2)
!   PARAMETER (AlpBlend=F1_MSST*Alp1 + (1.0d0-F1_MSST)*Alp2)
!
!
END MODULE WILCOX_DATA

```

A.1.4 Coding MSST-Step 3:

Summary: code baseline model setting F1 to be 0 or 1. Simplest case to begin. Annoyingly Sigstr1 needs to be 0.5d0 in the MODULE MSST. Most of this is eventually undone except for replacing some of the variables with their blended counterpart.

Also we will change the MODULE MSST by adding the following lines to determine F1 and the blended values:

```

PARAMETER (F1_MSST=0.0d0)
PARAMETER (BetaBlend=F1_MSST*Beta1 + (1.0d0-F1_MSST)*Beta2)
PARAMETER (SigstrBlend=F1_MSST*Sigstr1 + (1.0d0-F1_MSST)*Sigstr2)
PARAMETER (SigBlend=F1_MSST*Sig1 + (1.0d0-F1_MSST)*Sig2)
PARAMETER (AlpBlend=F1_MSST*Alp1 + (1.0d0-F1_MSST)*Alp2)

```

Making the following replacements to TwoEq_Turbulence_Wall_Ale_Sub_092117SUPZG.f90:

Beta -> BetaBlend (4 times)

Alp -> AlpBlend (4 times)

Sigstr -> SigstrBlend (to clarify 48 of the 52 times excluding the PeCell Sigstr) only term that appears in the 4 TKE subroutines, where as all others appear in the DISS subroutines

Sig -> SigBlend (to clarify 24 of the 28 times excluding the PeCell Sig)

Find and replace works for Beta if you search for Beta* (since Betastr exists).
Find and replace works for Sig if you search for Sig*

For the instances on the PeCell lines, search for PeCell and remove the blend terms, leave them unblended.

A.1.5 Coding MSST-Step 4:

Summary: Keeping the code as the BSL model for simplicity, but now letting F1_MSST vary. What we really want is Attempt 3's distance calculations that go into main, which will also be used for the actual MSST.

We have to get rid of the 5 parameter statements added in step 3.

Things needed to calculate F1_MSST:

RhoTilda (ρ)

TKEN (k)

OMEG (ω)

Betastr (β^*)

MuElem/RhoTilda ($\nu = \mu/\rho$)

sig2 ($\sigma_{\omega 2}$)

KX (dk/dx) <- not defined in 4 TKE subroutines

OX (d ω /dx) <- not defined in 4 DISS subroutine

Attempt 1:

Put the following code into the 8 subroutines just after the "FIRST DERIVATIVE TERMS", outside the loop:

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!! Calculate the "average" x,y,z of the cell
Xelem = 0.0d0 ! what unit is this if the SI switch is used?
Yelem = 0.0d0
Zelem = 0.0d0
DO KK=1,NUMN ! does this hit all nodes?
Xelem = Xelem + X(node(K,KK))
Yelem = Yelem + Y(node(K,KK))
Zelem = Zelem + Z(node(K,KK))
END DO
Xelem = Xelem / (NUMN*1.0d0)
Yelem = Yelem / (NUMN*1.0d0)
Zelem = Zelem / (NUMN*1.0d0)

!!!!!!! Calculate the distance of the closest wall node
distSmallest = 1.0d10
DO LL=1,NNODE
IF (novb(LL) < 0) THEN
distTEMP = sqrt((Xelem-X(LL))**2 +(Yelem-Y(LL))**2 +(Zelem-Z(LL))**2)
IF (distTEMP<distSmallest) THEN

```

```

    distSmallest = distTEMP
  END IF
END IF
END DO

```

!!!!!! Now calculate F1_MSST, <https://turbmodels.larc.nasa.gov/sst.html>

```

CD_komg = MAX(2*RhoTilda*Sig2*(KX*OX+KY*OY+KZ*OZ)/OMEG ,1.0d1**-

```

20)

```

ARG1 = MIN(MAX( sqrt(TKEN)/(Betastr*OMEG*distSmallest), &
  500*MuElem/(distSmallest**2*RhoTilda*OMEG) ), &
  4*RhoTilda*Sig2*TKEN/(CD_komg*distSmallest**2) )
F1_MSST = TANH(ARG1**4)

```

!!

```

BetaBlend=F1_MSST*Beta1 + (1.0d0-F1_MSST)*Beta2
SigstrBlend=F1_MSST*Sigstr1 + (1.0d0-F1_MSST)*Sigstr2
SigBlend=F1_MSST*Sig1 + (1.0d0-F1_MSST)*Sig2
AlpBlend=F1_MSST*Alp1 + (1.0d0-F1_MSST)*Alp2

```

Attempt 2:

May be better that we use the cntr variable in existing code instead of re-calculating every-time to get the node center. The suggestion has been made that we calculate the distance in the main function. May loose flexibility for moving parts overlay systems, h-adaptation etc.

!!!!!! Now calculate F1_MSST, <https://turbmodels.larc.nasa.gov/sst.html>

```

CD_komg = MAX(2*RhoTilda*Sig2*(KX*OX+KY*OY+KZ*OZ)/OMEG ,1.0d1**-

```

20)

```

ARG1 = MIN(MAX( sqrt(TKEN)/(Betastr*OMEG*distSmallest(K)), &
  500*MuElem/(distSmallest(K)**2*RhoTilda*OMEG) ), &
  4*RhoTilda*Sig2*TKEN/(CD_komg*distSmallest(K)**2) )
F1_MSST = TANH(ARG1**4)

```

```

BetaBlend=F1_MSST*Beta1 + (1.0d0-F1_MSST)*Beta2
SigstrBlend=F1_MSST*Sigstr1 + (1.0d0-F1_MSST)*Sigstr2
SigBlend=F1_MSST*Sig1 + (1.0d0-F1_MSST)*Sig2
AlpBlend=F1_MSST*Alp1 + (1.0d0-F1_MSST)*Alp2

```

Changing the 3 instances of distSmallest to distSmallest(K).

In the main file, do the following just before the loop for time stepping...

```
use WILCOX_DATA
```

In the main file, do the following just before the loop for time stepping...

!!!!!!! Calculate the distance of the closest wall node

```
allocate(distSmallest(NELEM))
```

```
DO L=1,NELEM
```

```
  distSmallest(L) = 1.0d10
```

```
  DO LL=1,NNODE
```

```
    IF (novb(LL) < 0) THEN
```

```

      distTEMP = sqrt((cntr(L,1)-X(LL))**2 +(cntr(L,2)-Y(LL))**2 +(cntr(L,3)-
Z(LL))**2)
      IF (distTEMP<distSmallest(L)) THEN
        distSmallest(L) = distTEMP
      END IF
    END IF
  END DO
END DO

```

In the main file, NO NEED to do the following just after the loop for time stepping...

```

      deallocate(distSmallest)

```

this will be used the whole time.

Attempt 3 - Better Yet:

To get more accurate results for hex cells when close to wall, put the following in the WILCOX_DATA module...

```

      DOUBLEPRECISION, ALLOCATABLE, DIMENSION(:) :: distSmallestSST,
distSmallestElem
      !    DOUBLEPRECISION, ALLOCATABLE, DIMENSION(:, :) :: distAllWalln-
odes

```

And this in KIVAhpFE3d_main_071216.f90 just before the time step loop:

```

      !!!!!!! Calculate the distance of the closest wall node
      allocate(distSmallestSST(NNODE)) !,distSmallestElem(NELEM)) !,distAllWalln-

```



```
odes(NNODE,NNODE))
```

```
! makes no sense to use distAllWallnodes in the above line, because not all the
values
```

```
! in the second dimension are wall nodes.
```

```
DO L=1,NNODE
```

```
! IF ( U(L) > zero ) CYCLE
```

```
distSmallestSST(L) = 1.0d10
```

```
DO LL=1,NNODE
```

```
IF (novb(LL) < zero .and. U(LL) /= 1130.0d0 ) THEN ! must match U inlet
velocity
```

```
distTEMP = dsqrt((X(L)-X(LL))**2 +(Y(L)-Y(LL))**2 +(Z(L)-Z(LL))**2)
```

```
IF (distTEMP < distSmallestSST(L)) THEN
```

```
distSmallestSST(L) = distTEMP
```

```
END IF
```

```
END IF
```

```
END DO
```

```
IF (distSmallestSST(L) == zero) THEN
```

```
distSmallestSST(L) = tiny
```

```
END IF
```

```
END DO
```

```
! DO L=1,NELEM ! take average of nodes in distSmallest for each element
```

```
! distSmallestElem(L) = 0.0d0
```

```
! DO LL=1,NUMN
```

```
! distSmallestElem(L) = distSmallestElem(L) + distSmallest(NODE(L,LL))
```

```
! END DO
```

```
! distSmallestElem(L) = distSmallestElem(L) / (1.0d0* NUMN)
```

```
! END DO
```

Also need to change `distSmallest(K)` to `distSmallestElem(K)` in the 8 subroutines of the the `TwoEq_Turbulence_Wall_ALE_Sub_092117SUPG.f90` (goes right after the “FIRST DERIVATIVE TERMS”) :

```

!!!!!! Now calculate F1_MSST, https://turbmodels.larc.nasa.gov/sst.html
CD_komg = MAX(2*RhoTilda*Sig2*(KX*OX+KY*OY+KZ*OZ)/OMEG ,1.0d1**-
20)
ARG1 = MIN(MAX( sqrt(TKEN)/(Betastr*OMEG*distSmallestElem(K)), &
500*MuElem/(distSmallestElem(K)**2*RhoTilda*OMEG) ),      &
4*RhoTilda*Sig2*TKEN/(CD_komg*distSmallestElem(K)**2) )
F1_MSST = TANH(ARG1**4)

BetaBlend=F1_MSST*Beta1 + (1.0d0-F1_MSST)*Beta2
SigstrBlend=F1_MSST*Sigstr1 + (1.0d0-F1_MSST)*Sigstr2
SigBlend=F1_MSST*Sig1 + (1.0d0-F1_MSST)*Sig2
AlpBlend=F1_MSST*Alp1 + (1.0d0-F1_MSST)*Alp2

```

Coding MSST-Step 5:

The Goal is to go from BSL to MSST. To do this μ_t will have to be calculated more carefully and change to `Sigstr1=0.85` from 0.5 as used in `KivaVarAllocateModuleMPI_091917.f90` WILCOX => MSST module. Step 6 will be worried about clipping and boundaries,

which should be the final step.

Expanding the vorticity term to simplify it:

$$W = 1/2 (du_i / dx_j - du_j / dx_i)$$

$$\Omega = \text{sqrt}(2 W_{ij} W_{ij})$$

Note that W is antisymmetric: $W_{ij} = -W_{ji}$

$$\text{So, } W = \begin{pmatrix} 0 & W_{12} & W_{13} \\ -W_{12} & 0 & W_{23} \\ -W_{13} & -W_{23} & 0 \end{pmatrix}$$

and $W_{ij}^2 = W_{ji}^2$

$$\begin{aligned} \Omega &= \text{sqrt}(2 (W_{11}^2 + W_{22}^2 + W_{33}^2) + 2 (W_{12}^2 + W_{13}^2 + W_{23}^2 + W_{21}^2 \\ &+ W_{31}^2 + W_{32}^2)) \\ &= \text{sqrt}(2 (0^2 + 0^2 + 0^2) + 4(W_{12}^2 + W_{13}^2 + W_{23}^2)) \\ &= \text{sqrt}((du_1 / dx_2 - du_2 / dx_1)^2 + (du_1 / dx_3 - du_3 / dx_1)^2 + (du_2 \\ &/ dx_3 - du_3 / dx_2)^2) \\ &= \text{sqrt}((UY - VX)^2 + (UZ - WX)^2 + (VZ - WY)^2) \end{aligned}$$

This can be coded and added to the eight subroutines

of the TwoEq_Turbulence_Wall_ALE_Sub_092117SUPG.f90 as:

! Now calculate MSST stuff, <https://turbmodels.larc.nasa.gov/sst.html>

Omega_MSST = dsqrt((UY-VX)**2 + (UZ-WX)**2 + (VZ-WY)**2) ! simplifies due to W being antisymmetric

CD_komg = MAX(2*RhoTilda*Sig2*(KX*OX+KY*OY+KZ*OZ)/OMEG ,1.0d1**-

20)

```

ARG1 = MIN(MAX( dsqrt(TKEN)/(Betastr*OMEG*distSmallestElem(K)),
&
500*MuElem/(distSmallestElem(K)**2*RhoTilda*OMEG) ),      &
4*RhoTilda*Sig2*TKEN/(CD_komg*distSmallestElem(K)**2) )
ARG2 = MAX( 2*dsqrt(TKEN)/(Betastr*OMEG*distSmallestElem(K)), &
500*MuElem/(distSmallestElem(K)**2*RhoTilda*OMEG) )
IF ( USE_MSST .eq. 1 ) then
F1_MSST = DTANH(ARG1**4)
F2_MSST = DTANH(ARG2**2)
MUTT = a1_MSST*TKE(L)*Rhotilda /
max(a1_MSST*OMG(L), Omega_MSST*F2_MSST) ! Nodal value, just as
above
SigstrBlend=F1_MSST*Sigstr1 + (1.0d0-F1_MSST)*Sigstr2
ELSE
F1_MSST = 1.0d0
SigstrBlend=Sigstr
END IF
BetaBlend=F1_MSST*Beta1 + (1.0d0-F1_MSST)*Beta2
SigBlend=F1_MSST*Sig1 + (1.0d0-F1_MSST)*Sig2
AlpBlend=F1_MSST*Alp1 + (1.0d0-F1_MSST)*Alp2

```

Need to change the above code to be nodal. Because the PG does not need blending, the visT is being advected. This will go just before the PROD terms (PRODTKE or PRODOMG) in each of the 8 sub-routines.

! ideally the next block should be made into a sub-routine at some point

```

! Now calculate MSST stuff, https://turbmodels.larc.nasa.gov/sst.html
IF ( USE_MSST .eq. 1 ) then
  IF (OMG(L) == ZERO ) THEN
    OMG(L) = tiny
  END IF
  Omega_MSST = dsqrt((UY-VX)**2 + (UZ-WX)**2 + (VZ-WY)**2 ) ! sim-
plifies due to W being antisymmetric
  CD_komg = MAX( 2.0d0* Rhotilda * Sig2*(KX*OX + KY*OY + KZ*OZ
)/ OMG(L) , 1.0d1**20 ) ! Why? * Rhotilda should be here
  ARG1 = MIN(MAX( dsqrt(TKE(L))/(Betastr*OMG(L)*distSmallestSST(L)),
&
    500.0d0*MUFLUID(L) / ( Rhotilda * distSmallestSST(L)**2 *OMG(L)
) ), & ! Nu = Mufluid/Rhotilda
    4.0d0*Rhotilda*Sig2*TKE(L) / ( CD_komg*distSmallestSST(L)**2
) )
  ARG2 = MAX( 2.0d0*dsqrt(TKE(L))/(Betastr*OMG(L)*distSmallestSST(L)),
&
    500.0d0 * MUFLUID(L) / ( Rhotilda * distSmallestSST(L)**2 *OMG(L)
) )
  F1_MSST = DTANH(ARG1**4)
  F2_MSST = DTANH(ARG2**2)
  MUTT = a1_MSST*TKE(L)*Rhotilda /
max(a1_MSST*OMG(L), Omega_MSST*F2_MSST) ! Nodal value, just as
above
  SigstrBlend=F1_MSST*Sigstr1 + (1.0d0-F1_MSST)*Sigstr2
ELSE
  F1_MSST = 1.0d0

```

```

    SigstrBlend=Sigstr
END IF
BetaBlend=F1_MSST*Beta1 + (1.0d0-F1_MSST)*Beta2
SigBlend=F1_MSST*Sig1 + (1.0d0-F1_MSST)*Sig2
AlpBlend=F1_MSST*Alp1 + (1.0d0-F1_MSST)*Alp2

```

This now needs to be placed just before the TRAC terms into the each of the 8 sub-routines. LL is wanted instead of L:

```

    ! Now calculate MSST stuff, https://turbmodels.larc.nasa.gov/sst.html
    IF ( USE_MSST .eq. 1 ) then
        IF ( OMG(LL) == ZERO ) THEN
            OMG(LL) = tiny
        END IF
        Omega_MSST = dsqrt((UY-VX)**2 + (UZ-WX)**2 + (VZ-WY)**2 ) ! sim-
        plifies due to W being antisymmetric
        CD_komg = MAX(2.0d0 * Rhotilda * Sig2*(KX*OX + KY*OY + KZ*OZ
        )/ OMG(LL) , 1.0d1**-20 ) ! Why? * Rhotilda should be here
        ARG1 = MIN(MAX( dsqrt(TKE(LL))/(Betastr*OMG(LL)*distSmallestSST(LL)),
        &
            500.0d0*MUFLUID(LL) / ( Rhotilda * distSmallestSST(LL)**2 *OMG(LL)
        ) ), & ! Nu = Mufluid/Rhotilda
            4.0d0 *Rhotilda*Sig2*TKE(LL) / ( CD_komg*distSmallestSST(LL)**2
        ) )
        ARG2 = MAX( 2.0d0 *dsqrt(TKE(LL))/(Betastr*OMG(LL)*distSmallestSST(LL)),
        &

```

```

500.0d0 * MUFLUID(LL) / ( Rhotilda * distSmallestSST(LL)**2
*OMG(LL) ) )

```

```

F1_MSST = DTANH(ARG1**4)

```

```

F2_MSST = DTANH(ARG2**2)

```

```

MUTT = a1_MSST*TKE(LL)*Rhotilda /

```

```

max(a1_MSST*OMG(LL), Omega_MSST*F2_MSST) ! Nodal value, just as

```

above

```

SigstrBlend=F1_MSST*Sigstr1 + (1.0d0-F1_MSST)*Sigstr2

```

```

ELSE

```

```

F1_MSST = 1.0d0

```

```

SigstrBlend=Sigstr

```

```

END IF

```

```

BetaBlend=F1_MSST*Beta1 + (1.0d0-F1_MSST)*Beta2

```

```

SigBlend=F1_MSST*Sig1 + (1.0d0-F1_MSST)*Sig2

```

```

AlpBlend=F1_MSST*Alp1 + (1.0d0-F1_MSST)*Alp2

```

```

Edits to the _KivaVarAllocate file_: add

```

```

INTEGER :: USE_MSST

```

to module runparam

put the following wherever you want...

```

READ(7,*)USE_MSST

```

(just be sure to add a line in the correct spot in the .par file)

also add 1

USE_MSST 0 = K-Omega, 1 = MSST” as the first parameter read in the .par main file.

A.1.6 Coding MSST-Step 6

Summary: Boundary conditions and wall function and clipping.

In the stability analysis routine, I made the following change:

```

!      TURBULENT VISCOSITY EXCEPT AT WALL ELEMENT
!
!      MUT(:) = zero
!      IF ( FLOW == 1 .AND. Wall_Func == 1 ) THEN
!      DO L=1,NNODE
!      LOOPEXIT=0
!      IF ( IFNODE(L) /= L ) Cycle
!      IF (TKE(L)<FREE_STREAM_TKE ) THEN
!      TKE(L)=FREE_STREAM_TKE
!      END IF
!      IF (OMG(L)<FREE_STREAM_OMG ) THEN
!      OMG(L)=FREE_STREAM_OMG
!      END IF
!      MUT(L)=RHO(L)*TKE(L)/OMG(L)
!!      IF ( YPLUS(L) < 10.0 ) MUT(L) = zero
!      IF ( MUTMAX < MUT(L) ) MUTMAX=MUT(L)
!      TKE(WALLFNODE(L))=zero
!      MUT(WALLFNODE(L))=zero
!      OMG(WALLFNODE(L))=OMG(IFNODE(L))
!      END DO

```

```

IF ( FLOW == 1 .AND. Wall_Func >= 0) THEN

```



```

DO L=1,NNODE
  IF ( (NSTEP == 1 .OR. NADAPT == 1) .AND. TKE(L)<FREE_STREAM_TKE)
THEN!
  TKE(L)=FREE_STREAM_TKE
END IF
  IF ( OMG(L)<FREE_STREAM_OMG ) THEN
    OMG(L)=FREE_STREAM_OMG
  END IF
  MUT(L) = zero

  MUT(L)=RHO(L)*TKE(L)/OMG(L)
!   IF ( YPLUS(L) < 10.0 ) MUT(L)=zero
  IF (MUTMAX<MUT(L)) MUTMAX=MUT(L)
END DO
END IF

```

The Y^+ ranges are modified in SUBROUTINE WALL_FUNCTION to better suit that of Wilcox suggestion as below:

```

IF (YPLUS(LL) >= 30.0d0 .and. YPLUS(LL) < 500.0d0 ) THEN
  UPLUS(LL) = LOG(YPLUS(LL))/KARMAN + 5.5d0
  IF ( mod(NSTEP,500) == 0 ) THEN
    WRITE(101,*)'YPLUS(LL) > 30.0',YPLUS(LL),UPLUS(LL),TAUw(LL),TKE(LL),LL
  END IF
ELSE IF (YPLUS(LL) < 30.0d0 .AND. YPLUS(LL) > 5.0d0 ) THEN
  UPLUS(LL) = 5.0d0*LOG(YPLUS(LL)) - 3.05
  IF ( mod(NSTEP,500) == 0 ) THEN
    WRITE(102,*)'YPLUS(LL) <= 30.0',YPLUS(LL),UPLUS(LL),TAUw(LL),TKE(LL),LL
  END IF

```

```
END IF
ELSE IF ( YPLUS(LL) <= 5.0d0 ) THEN ! .AND. YPLUS(LL) >= zero) THEN
  UPLUS(LL) = YPLUS(LL)
  IF ( mod(NSTEP,500) == 0 ) THEN
    WRITE(103,*)'YPLUS(LL) < 10.0',YPLUS(LL),UPLUS(LL),TAUw(LL),TKE(LL),LL
  END IF
END IF
IF (YPLUS(LL) < 10.0 ) THEN
!  WRITE(*,*)'Y+ < 10.0', YPLUS(LL),UPLUS(LL),TAUw(LL),DIST(I),I,LL
END IF

IF (YPLUS(LL) < 4.0 ) THEN
!  WRITE(*,*)'Y+ < 4.0', YPLUS(LL),UPLUS(LL),TAUw(LL),DIST(I),I,LL
END IF
```

Appendix B

Appendix: GridPro for KIVA and FEARCE

KIVA is a large legacy Fortran program developed and maintained by LANL's T3 solid mechanics and fluid mechanics group that solves turbulent reactive flows for internal combustion engines. KIVA-4mpi is an updated parallel version of KIVA4 that still involves the labor intensive task of producing intricate computational grids. In summary the legacy $k-\epsilon$ KIVA codes use traditional FV formulation while the parts movement system is based on local node snapping, the parts must be matched and the grid generation is painfully tedious. KIVA-4mpi is the current maintained available release from LANL, but will soon be overtaken with the release of FEARCE, formerly KIVA4-hpFE, in the near future. FEARCE employs the FEM with a recently invented moving marker parts overlay system that takes full advantage of the high quality automatic grid generation of GridPro. The grids are also referred to as semi-structured due to the way the hexahedral elements can be shaped vs. traditional structured grid generation softwares. The moving parts surfaces are easy to generate and are overlaid onto hexahedral elements which are automatically generated for complex convex domains. FEARCE and GridPro also give rise to even further au-

tomating the grid generation process for complex geometries with the introduction of modular based topology for spray injectors, spark systems and different port/head and cylinder configurations.

The process to develop and implement GridPro into FEARCE began with KIVA-4mpi grid generation and a full 2-valve combustion engine simulation example test as shown in Fig. (B.1) and Fig. (B.2).

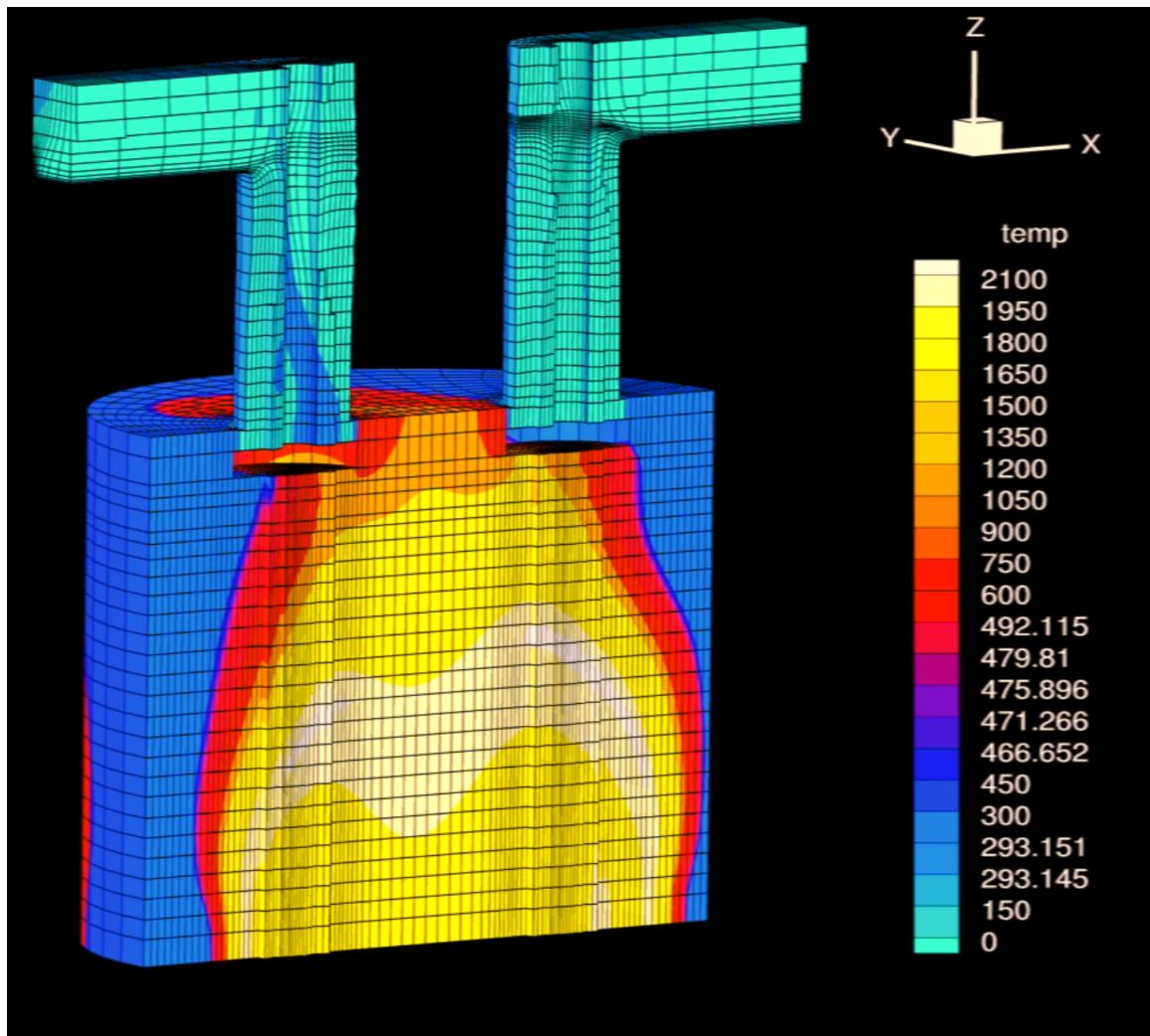


Figure B.1: Temperature distribution for a KIVA-4mpi turbulent combustion engine simulation with snapping moving parts on a GridPro grid 2 valve, 2 port test system.

The KIVA-4mpi traditional meshing is labor intensive on domains with imbedded bodies, requiring a node matched grid as shown in Fig. (B.3) for the snapping rou-

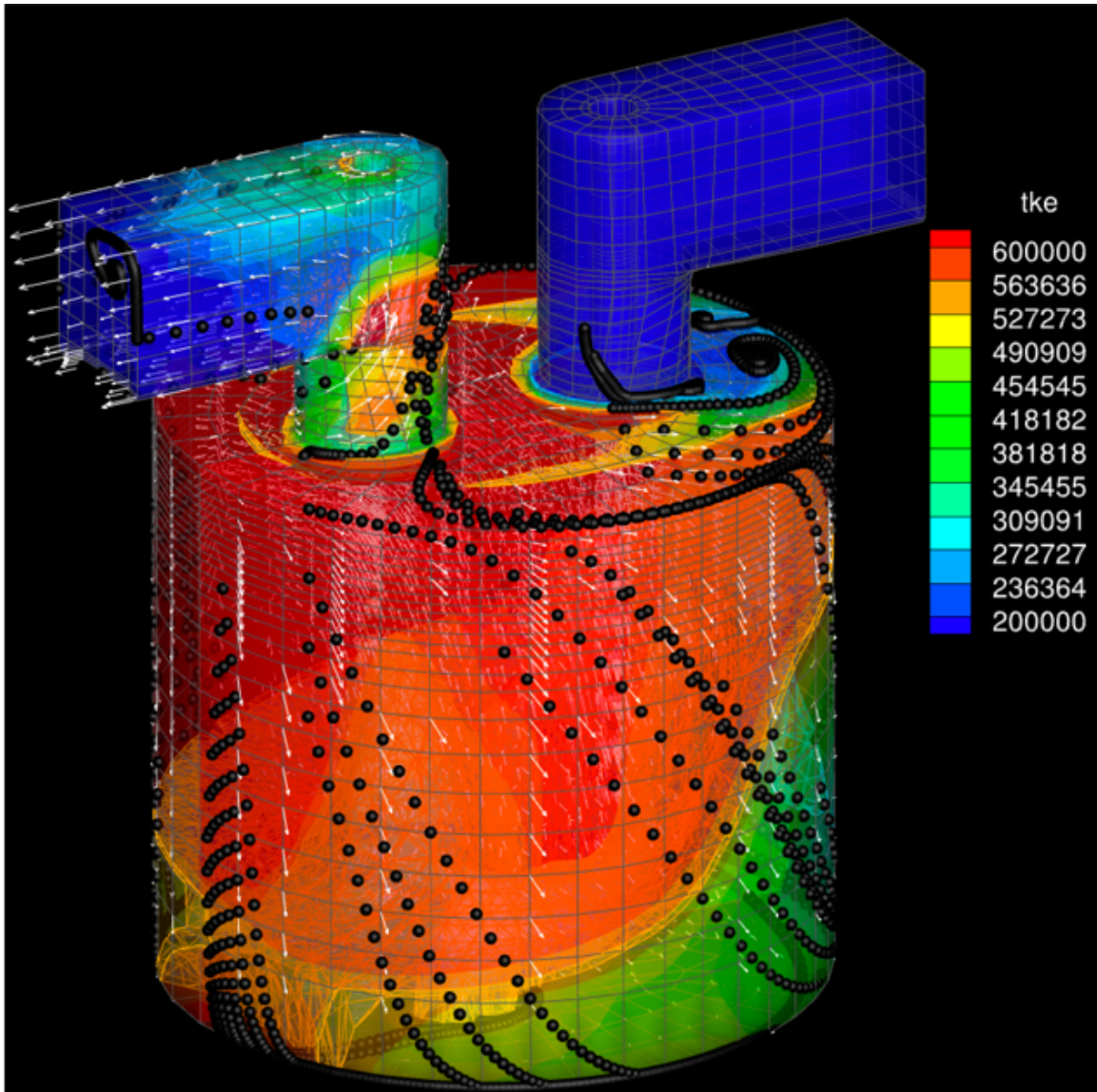


Figure B.2: Turbulent kinetic energy plot distribution with bead stream-traces of a KIVA-4mpi turbulent combustion engine simulation with snapping moving parts on a GridPro grid 2 valve, 2 port test system.

tine. The meshing for KIVA-4mpi was made significantly easier by using GridPro's semi-structured, automated grid generation to produce small aspect ratio hexahedral blocks that help facilitate stability in FV schemes. GridPro essentially reduces grid generation to topology generation, and allows the primary user to input a pattern of points referred to as the grid topology. Although more intuitive the embedded parts

snapping routines of KIVA-4mpi where used as a proof of concept exercise and even with GridPro still require a great deal of effort and work around for a quality mesh.

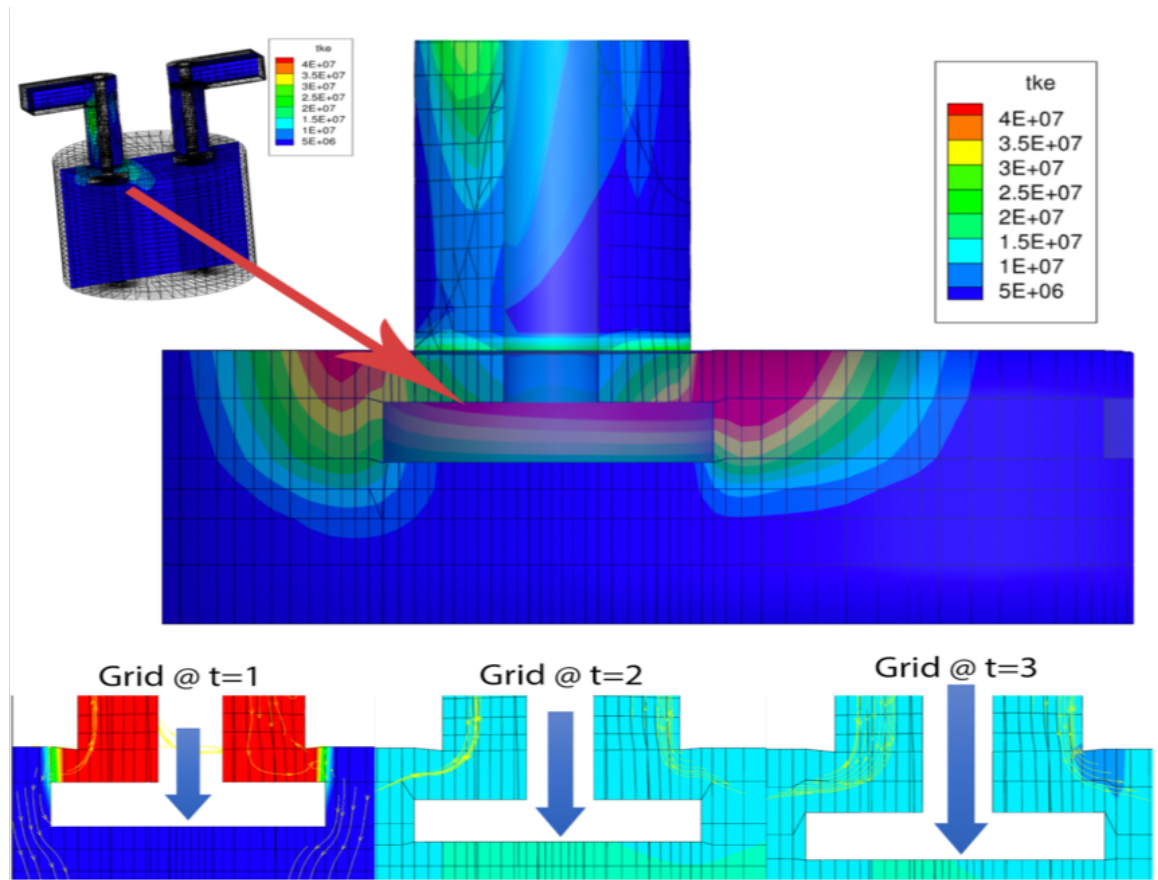


Figure B.3: KIVA-4mpi Snapping System for Moving Immersed Parts (GridPro 2 valve, 2 port idealized test system).

The newer FEM based FEARCE code can get by with less stringent requirements on aspect ratio and near wall spacing. In addition, the newly invented moving marker system for immersed bodies in the FEARCE code is presented here mainly for documentation reasons. The FEARCE overset, moving marker system takes full advantage of the automatic grid generators used in GridPro, which can provide a quality grid in a convex domain without consideration for internally attached grid nodes to moving parts as was previously done in prior KIVA codes. In FEARCE the body immersion is supplied by the FEM codes moving parts methods and uses triangulated surfaces overlaying Eulerian fluid hexahedral grid elements to allow for the fluid system to be

solved in the Eulerian frame. The development of the engine grids using GridPro for older versions of KIVA and FEARCE were a precursor to the grids used in this study.

GridPro grids are far superior when a low aspect grid can be created. The ability to modularize the topology and re-purpose its use is beneficial to the engine designer. The ability to create complex features and realize complex domains is of great benefit, however some of the control needed to enforce aspect ratio and similar element sizing can become tedious but still far easier to maintain when compared to other grid generators. This worry will be lessened with the introduction of module based topology.

The overlay system uses regional 3 dimensional block properties to turn the ports on and off, as shown in Fig. (B.4). Trying to use a sheet made from an internal surface first, then a slice 2-3 nodes about this internal surface exported and fed back into the grid (to better align the converged grid to the slice that edges the block).



Figure B.4: Schematic breakdown of the FEARCE/GridPro moving marker valve overlay 3D port blocking properties.

With this technique we have had mild success, most recently this approach has been abandoned in favor of sheets bisecting the domain and a small center cylinder running the length of the combustion arena to increase aspect ratio so the 3D region where the center of the head can have the lowest aspect as possible, shown in Fig. (B.4) 2a and 2b.

A low aspect ratio, highly uniform grid domain with smooth gradients improves the transfer of information from one cell to the next and would be considered preferred. The inherent need to immerse a triangular surface .stl file, such as the valves depicted in in Fig. (B.4) 2a, into the hexahedron of fluid grid must be scalable. For example, a finer engine grid would imply a finer tet, on a moving surface such as a valve or piston, is needed. 2-3 .stl tet surface elements are needed for one 3D element or isoparametric brick in the fluid domain upon which it is overlaid. This is done in order to identify an immersed boundary, more or less tets per fluid hex causes un-needed compute time, less may cause improperly recognized boundary.

In Fig. (B.7), the domain was extracted using x-ray tomography and the valves, bowled piston, head, intake and exhaust ports were gridded using GridPro. This extent of realism in a previous KIVA grid was not possible due to the organic shapes of the surfaces involved. The process of topology based nodes building a skeletal structure around these extracted x-ray tomography surfaces files, allows GridPro to project a control net surface onto the shape. This control net surface generation is something not seen in other grid generators, the technology more likened to that of the Hollywood Pixar animation industry. Using the FEARCE approach to moving marker overlay of moving components one further increases the quality of the fluid domain. In previous legacy KIVA grid generation the moving parts had to be coupled to the nodes in the domain, this perverting the fluid domain nodes as seen in the snapping routines of older KIVA grids. This is no longer needing to be done with FEARCE and significantly increases engine design from simulation to market.

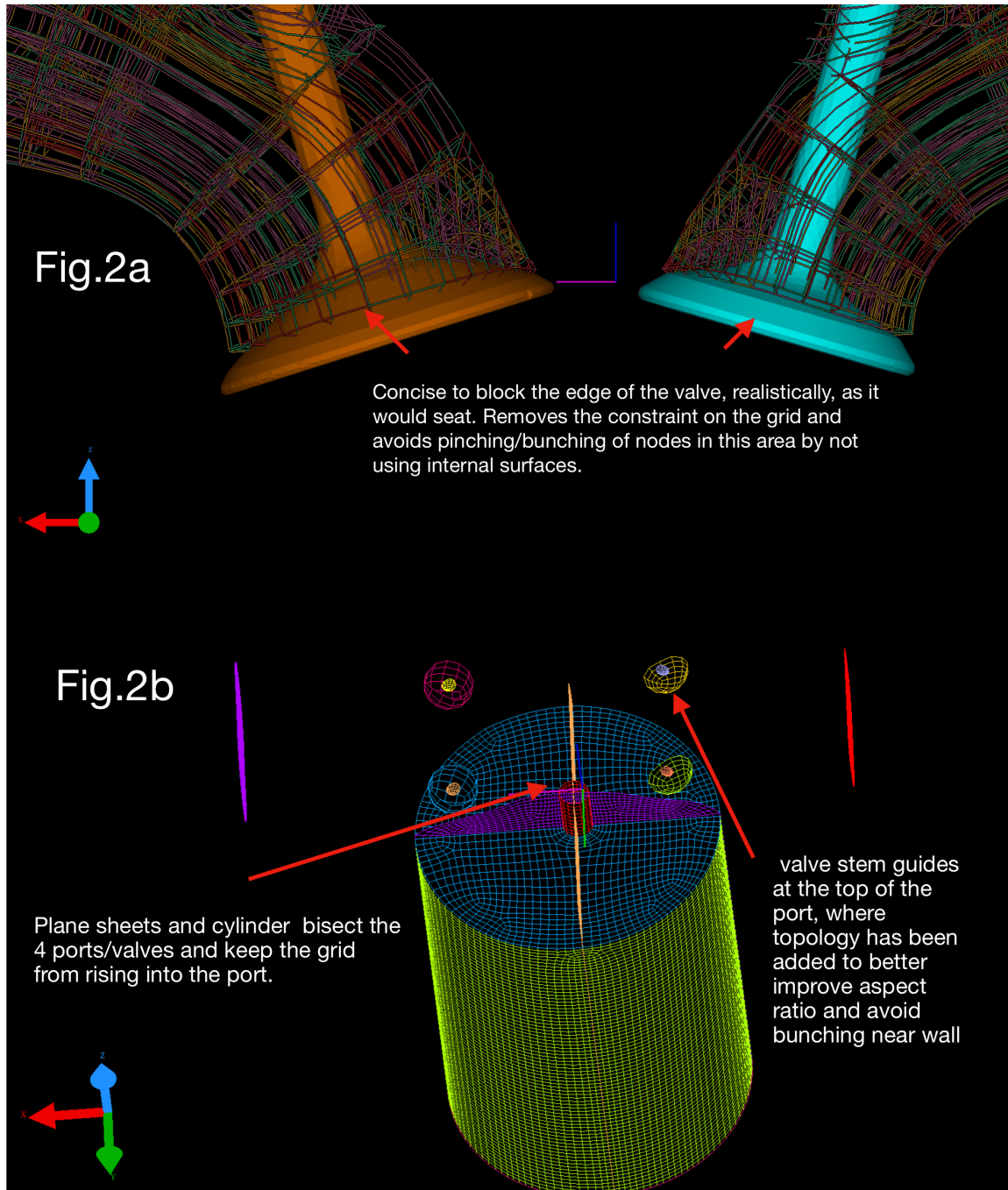


Figure B.5: Schematic breakdown of the FEARCE/GridPro moving marker overlay grid system.

Modular based topology from GridPro is outlined in the figures that follow. The parts overlaid are interchangeable and the topology modules are re-useable for different engine designs, i.e., the spark and Spray-G modules can easily be fitted to a 2V

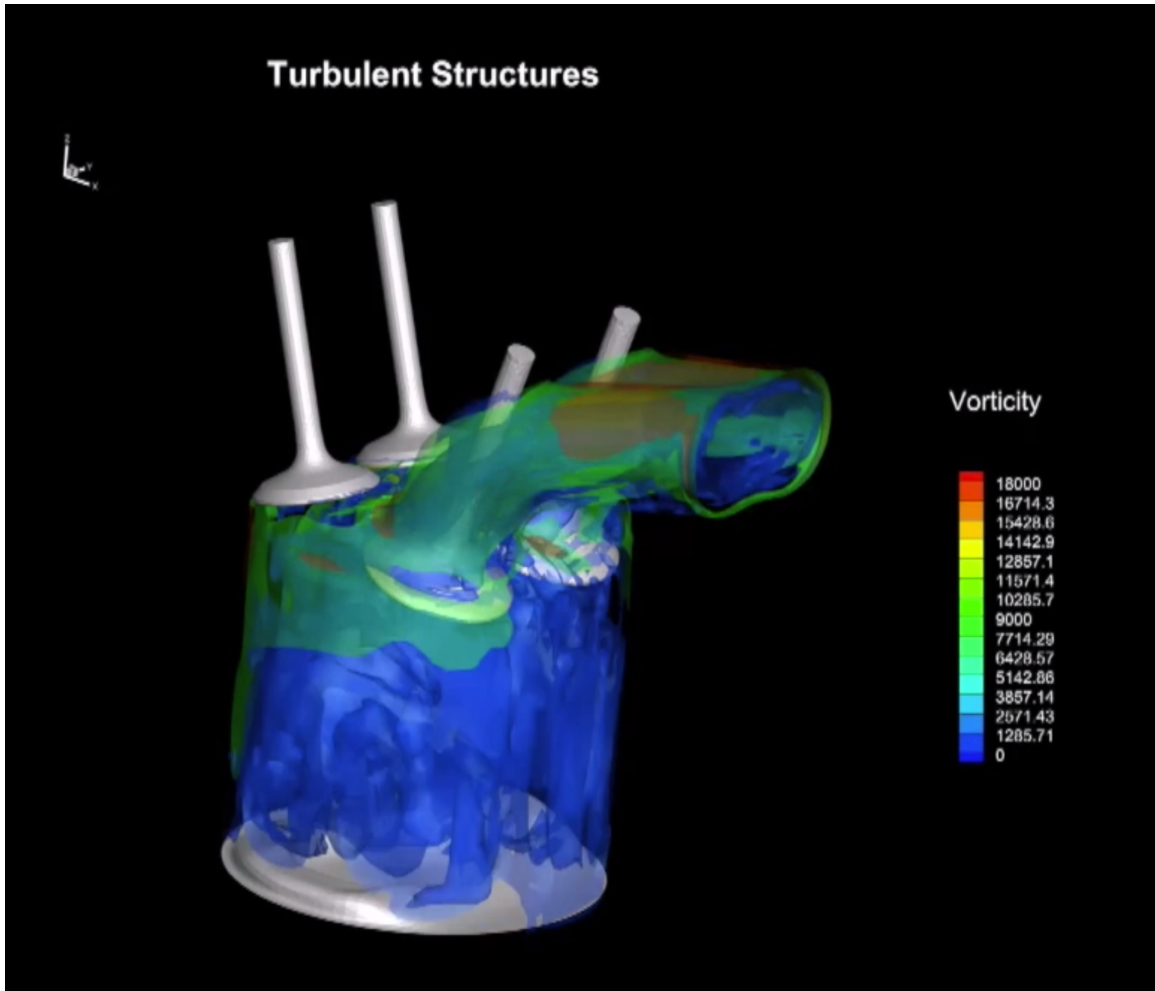


Figure B.6: A 4-valve DISI moving marker system FEARCE test simulation on GridPro grid. Bowled piston, valves, and manifold surfaces created in GridPro from x-ray tomography.

engine design or as shown a 4V valve manifold. The process to build the topology in GridPro is intuitive and gives the design engineer more freedom in the grid creation process, significantly decreases the time and effort to produce, and also increases grid quality.

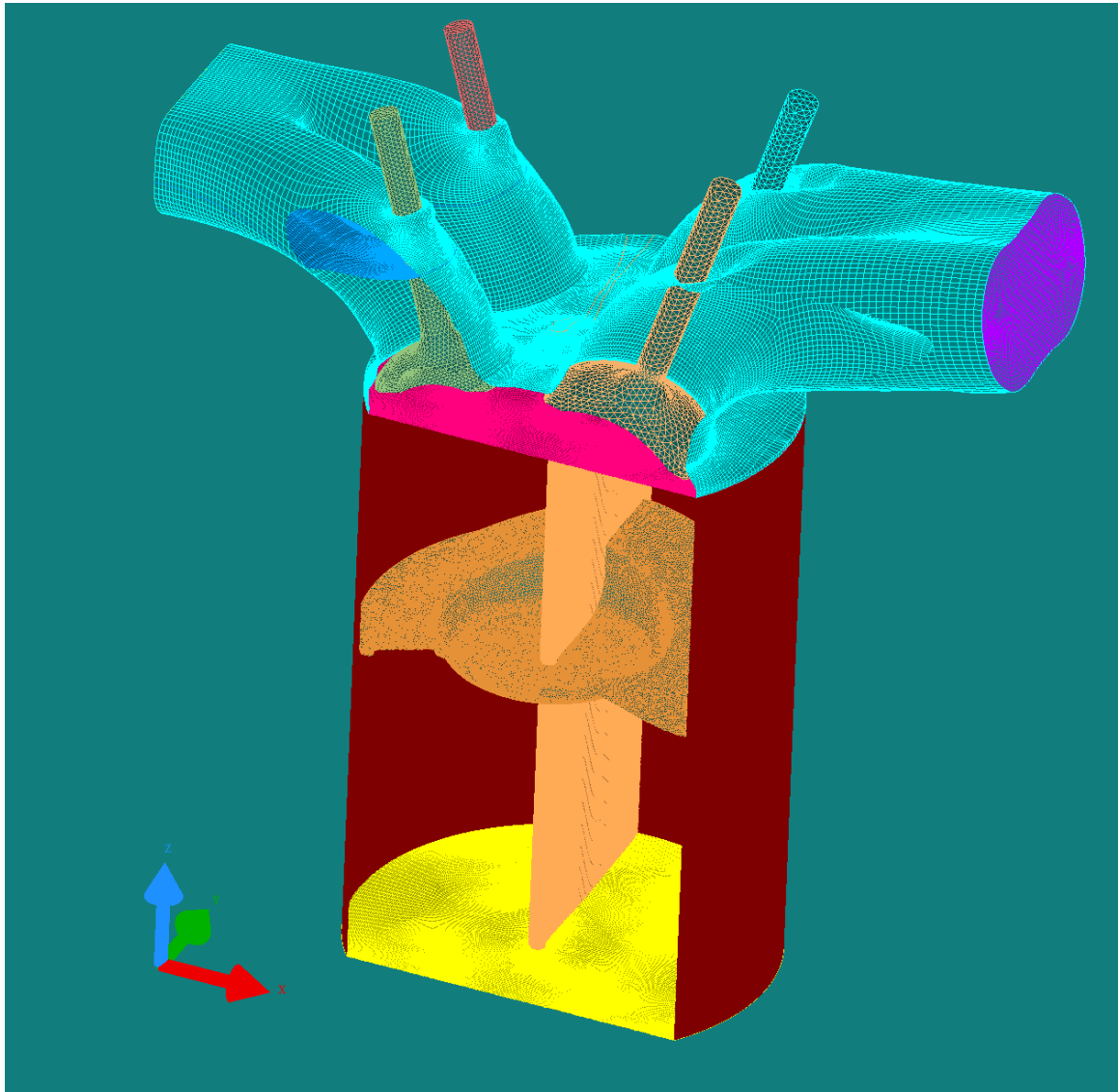


Figure B.7: A 1.2 million node 4 valve direct injection spark injection (DISI) engine from x-ray tomography, with .stl triangulated (tet) moving marker valves and bowled piston, grid generated by with GridPro. The grid has been sparsened down from 40 million.

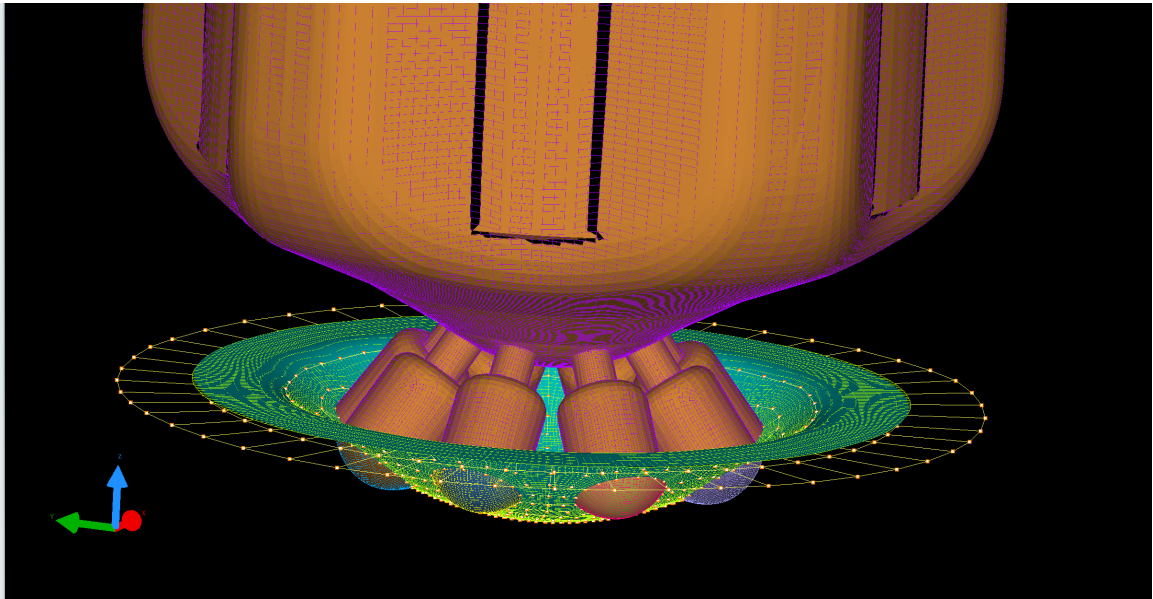


Figure B.8: Spray-G needle valve injector GridPro grid, the image shows an outer realm of topology nodes along with the surfaces used to create the grid overlaid onto the grid.

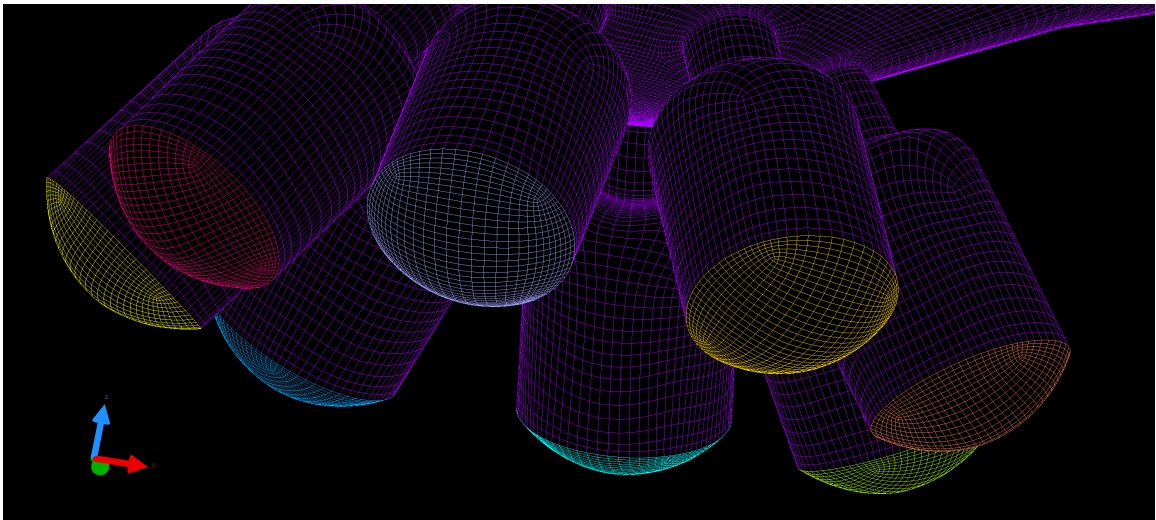


Figure B.9: Spray-G GridPro injector nozzle grid.

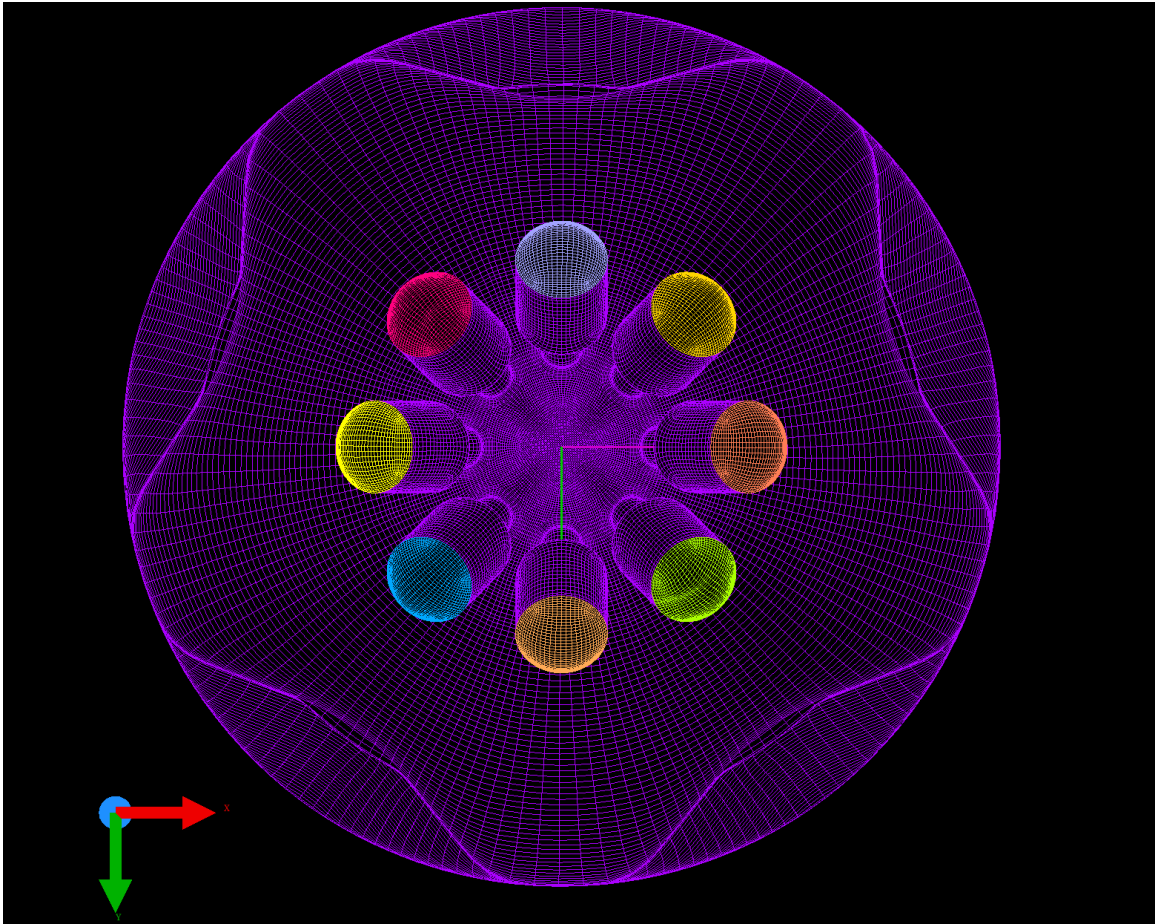


Figure B.10: Spray-G GridPro injector nozzle grid with needle valve sleeve.

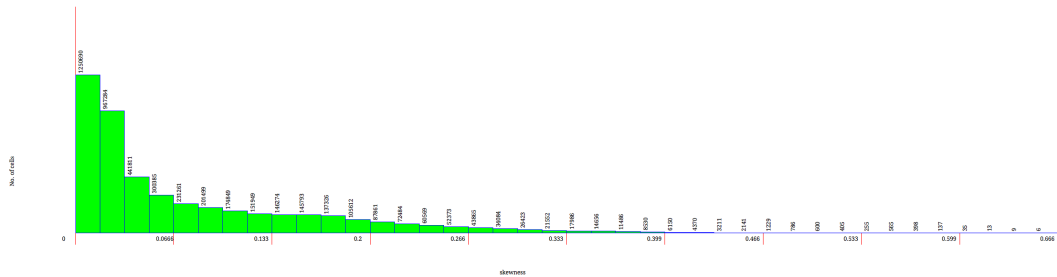


Figure B.11: Spray-G GridPro injector nozzle cell vs skew distribution, exhibiting the high quality found in GridPro’s capability to mesh complex components.

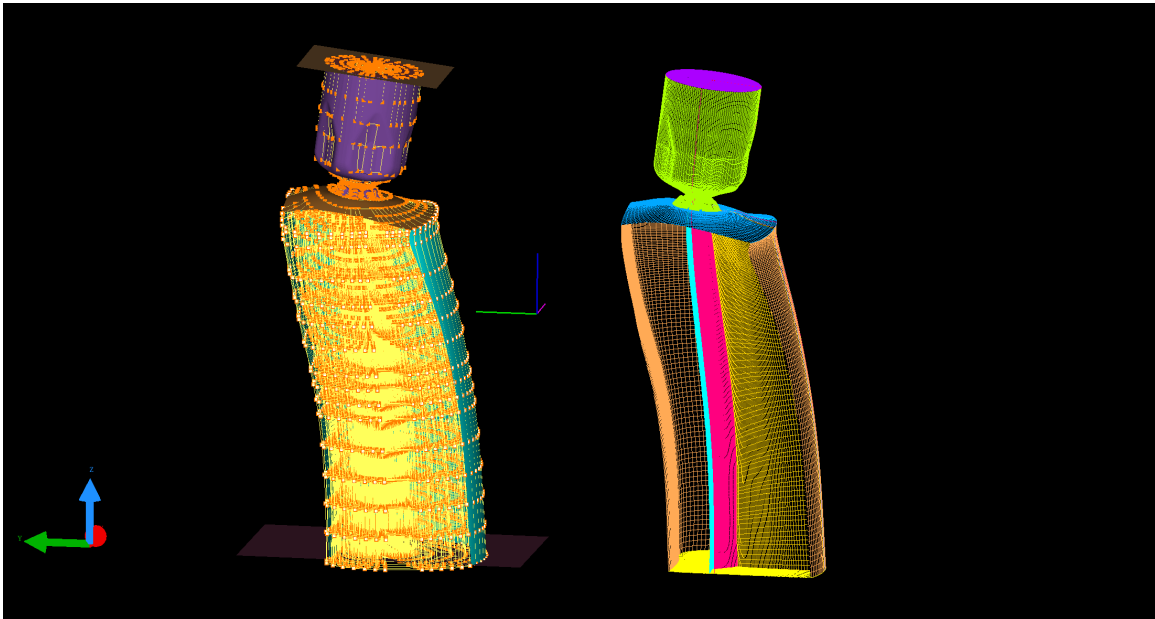


Figure B.12: The Spray-G GridPro injector was then "modularized" by creating a standalone topology, this module can now be effectively plugged into test engine designs. Surfaces and Topology(left), Grid(right)

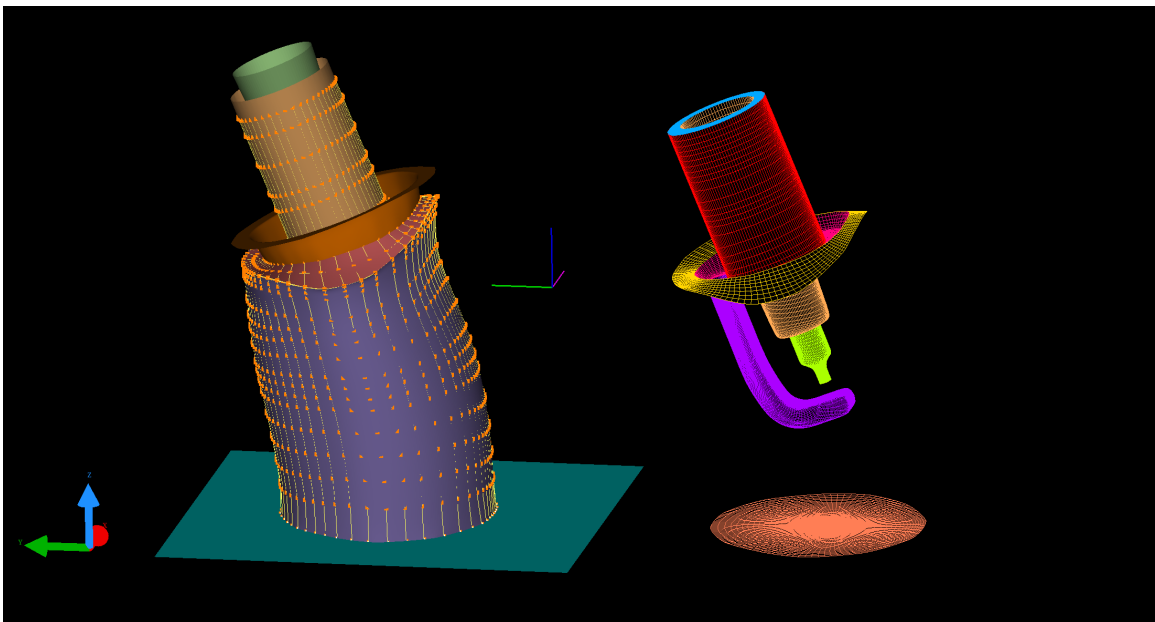


Figure B.13: A GridPro spark system was then "modularized", this module can also now be plugged into test engine designs. Surfaces and Topology(left), Grid(right)

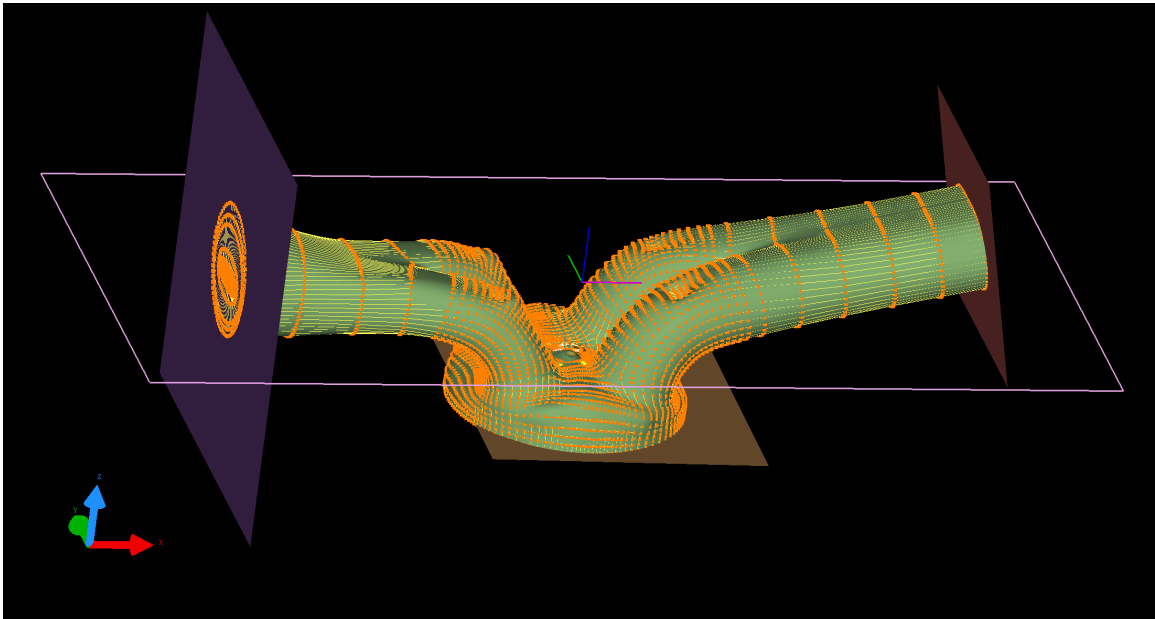


Figure B.14: A GridPro 4V manifold with insert topology for spark and Spray-G modules.

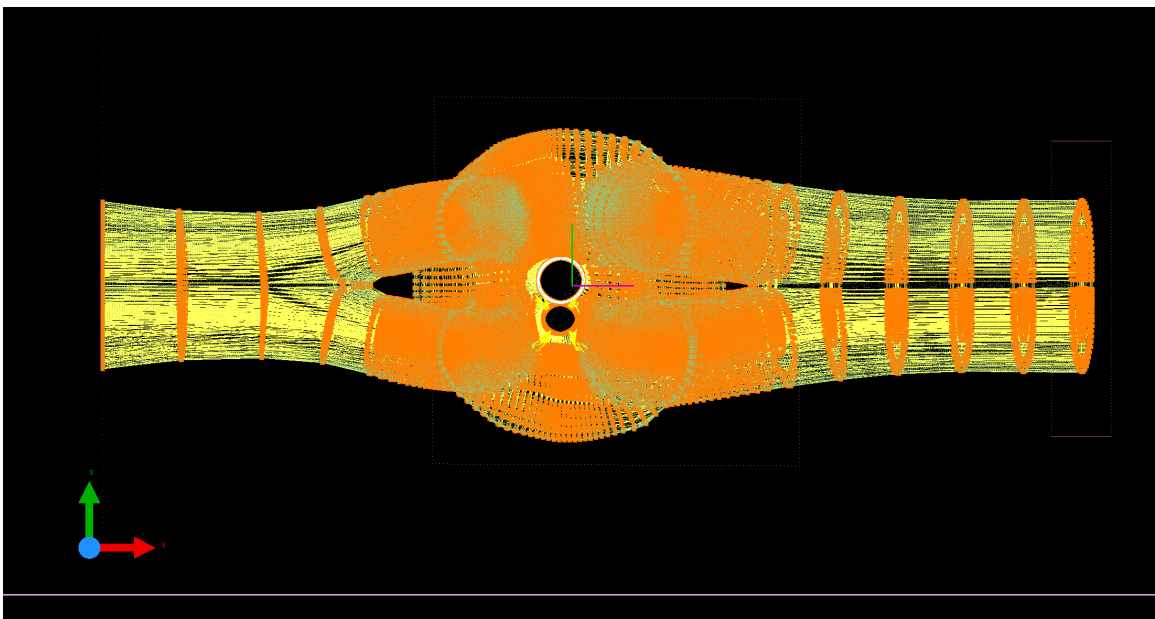


Figure B.15: A GridPro 4V manifold with insert topology for spark and Spray-G modules, top down view.

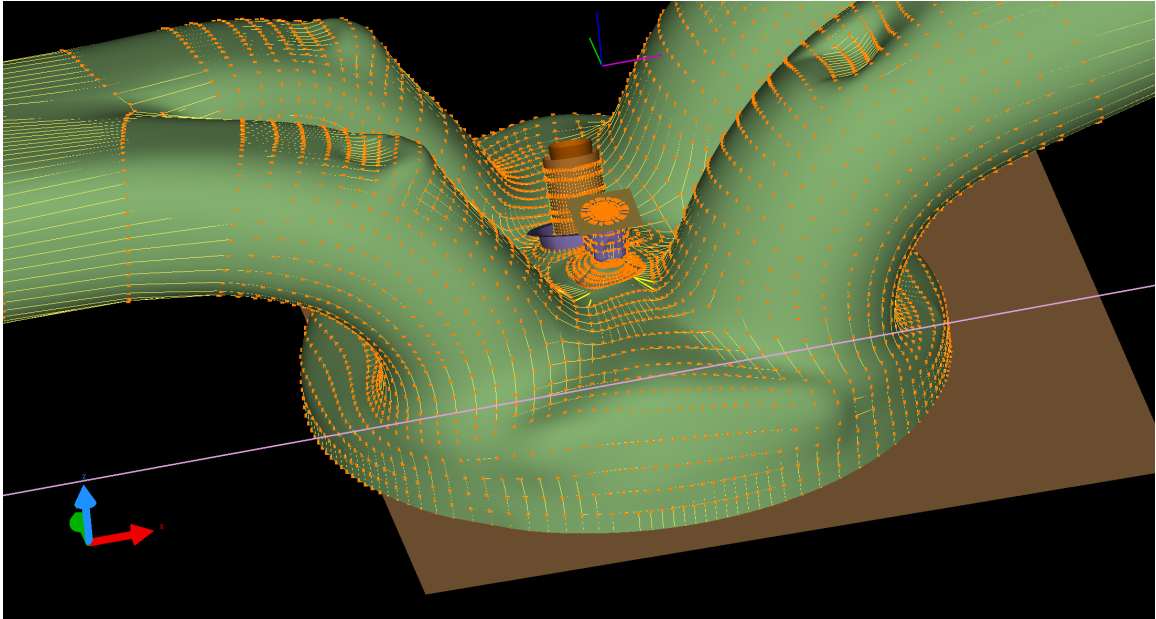


Figure B.16: A GridPro 4V manifold with inserted topology spark and Spray-G modules.

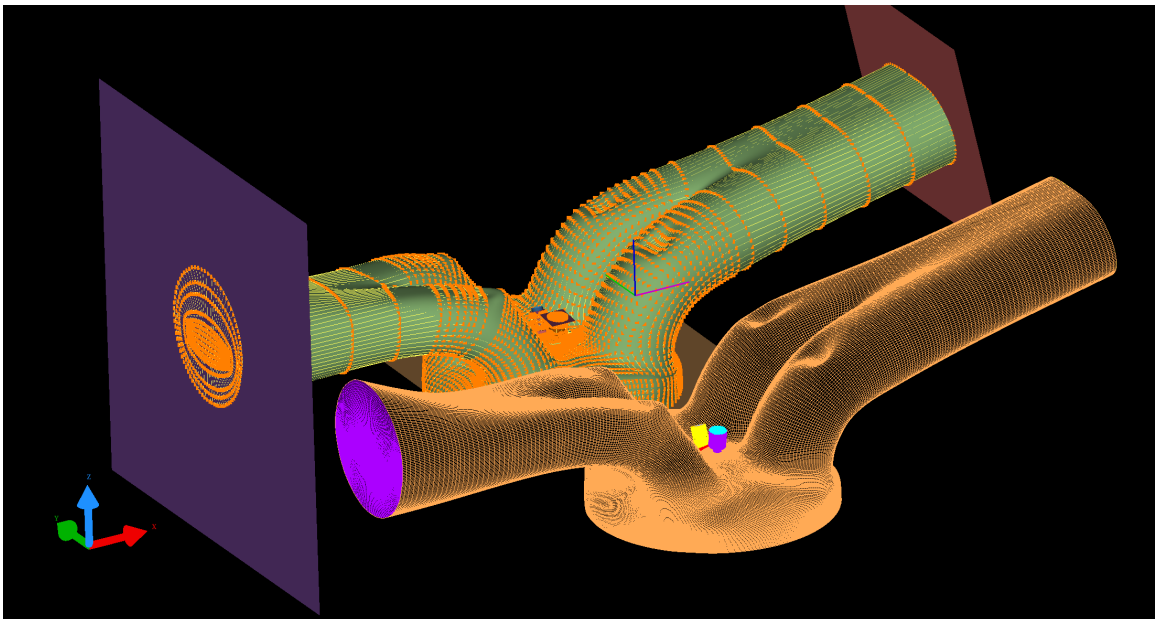


Figure B.17: A GridPro 4V manifold with inserted topology spark and Spray-G modules(left) and validated grid(right).

References

- [1] J. Vogel and J. Eaton, *Combined heat transfer and fluid dynamic measurements downstream of a backward-facing step*, *Journal of heat transfer* **107** (1985) 922.
- [2] D. C. Wilcox, *Turbulence Modeling for CFD*, vol. 3rd Edition. DWC Industries, 2004.
- [3] R. D. Moser, J. Kim and N. N. Mansour, *Direct numerical simulation of turbulent channel flow up to $Re \tau = 590$* , *Physics of fluids* **11** (1999) 943.
- [4] D. B. Carriington, *An h-adaptive finite element method for turbulent heat transfer*, tech. rep., Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2009.
- [5] F. R. Menter, *Improved two-equation k-omega turbulence models for aerodynamic flows*, *NASA STI/Recon Technical Report N 93* (1992) .
- [6] P. Krishnakumar Rajagopalan and R. Eiseman, *Automatic nested refinement—a technique for the generation of high quality multi-block structured grids for multi-scale problems using gridpro®*, .
- [7] J. Boussinesq, *Théorie de L'Écoulement Tourbillonnant Et Tumultueux Des Liquides Dans Les Lits Rectilignes a Grande Section*, no. v. 1. BiblioBazaar, 2010.

- [8] P. Bradshaw, *Possible origin of prandtl's mixing-length theory*, *Nature* **249** (1974) 135.
- [9] J. Alembert, *Essai d'une nouvelle theorie de la résistance des fluides*. David l'ainé, 1752.
- [10] F. R. Menter, *Two-equation eddy-viscosity turbulence models for engineering applications*, *AIAA journal* **32** (1994) 1598.
- [11] F. H. Harlow, *Hydrodynamic problems involving large fluid distortions*, *Journal of the ACM (JACM)* **4** (1957) 137.
- [12] M. W. Evans, F. H. Harlow and E. Bromberg, *The particle-in-cell method for hydrodynamic calculations*, tech. rep., LOS ALAMOS NATIONAL LAB NM, 1957.
- [13] R. Bridson, *Fluid simulation for computer graphics*. AK Peters/CRC Press, 2015.
- [14] F. H. Harlow and J. E. Welch, *Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface*, *The physics of fluids* **8** (1965) 2182.
- [15] F. H. Harlow and A. A. Amsden, *A numerical fluid dynamics calculation method for all flow speeds*, *Journal of Computational Physics* **8** (1971) 197.
- [16] L. Cloutman, *Sola-star: a one-dimensional iced-ale hydrodynamics program for spherically symmetric flows*, tech. rep., Los Alamos Scientific Lab., 1980.
- [17] J. Jiménez, *The contributions of an kolmogorov to the theory of turbulence*, *Arbor* **178** (2004) 589.
- [18] H. K. Versteeg and W. Malalasekera, *An introduction to computational fluid dynamics: the finite volume method*. Pearson Education, 2007.

- [19] X. Han, *Development and Application of Hybrid Wray-Agarwal Turbulence Model and Large-Eddy Simulation*, Ph.D. thesis, Washington University in St. Louis, 2018.
- [20] C. L. Rumsey and P. R. Spalart, *Turbulence model behavior in low reynolds number regions of aerodynamic flowfields*, *AIAA journal* **47** (2009) 982.
- [21] P. E. Smirnov and F. R. Menter, *Sensitization of the sst turbulence model to rotation and curvature by applying the spalart–shur correction term*, *Journal of Turbomachinery* **131** (2009) 041010.
- [22] A. Hellsten, *Some improvements in menter’s k-omega sst turbulence model*, in *29th AIAA, Fluid Dynamics Conference*, p. 2554.
- [23] B. F. Armaly, F. Durst, J. Pereira and B. Schönung, *Experimental and theoretical investigation of backward-facing step flow*, *Journal of fluid Mechanics* **127** (1983) 473.
- [24] W. Cherdron, F. Durst and J. H. Whitelaw, *Asymmetric flows and instabilities in symmetric ducts with sudden expansions*, *Journal of Fluid Mechanics* **84** (1978) 13.
- [25] A. Gosman and W. Pun, *Lecture Notes for Course Entitled” calculation of Recirculating Flows”*. Imperial College of Science and Technology, University of London, 1974.
- [26] A. Kumar and K. S. Yajnik, *Internal separated flows at large reynolds number*, *Journal of Fluid Mechanics* **97** (1980) 27.
- [27] J. D. Anderson, *Computational fluid dynamics*, vol. 206. Springer, 1995.

- [28] J. Lorenz and P. R. Zingano, *The navier-stokes equations for incompressible flows: solution properties at potential blow-up times*, *arXiv preprint arXiv:1503.01767* (2015) .
- [29] O. Reynolds, *On the dynamical theory of incompressible viscous fluids and the determination of the criterion*, *Philosophical Transactions of the Royal Society of London. A* **186** (1895) 123.
- [30] S. Childress, *Topological fluid dynamics for fluid dynamicists*, *preprint* (2004) .
- [31] D. Ruelle and F. Takens, *On the nature of turbulence*, *Communications in mathematical physics* **20** (1971) 167.
- [32] J. Guckenheimer and P. Holmes, *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields*, vol. 42. Springer Science & Business Media, 2013.
- [33] G. Boeing, *Visual analysis of nonlinear dynamical systems: chaos, fractals, self-similarity and the limits of prediction*, *Systems* **4** (2016) 37.
- [34] W. Layton, *The 1877 boussinesq conjecture: Turbulent fluctuations are dissipative on the mean flow*, *To appear* (2014) .
- [35] W. Jones and B. E. Launder, *The prediction of laminarization with a two-equation model of turbulence*, *International journal of heat and mass transfer* **15** (1972) 301.
- [36] S. B. Pope, *Turbulent flows*, 2001.
- [37] W. K. George, *Lectures in turbulence for the 21st century*, .
- [38] W. M. Kays, *Convective heat and mass transfer*. Tata McGraw-Hill Education, 2012.

- [39] R. Curl and K. Pitzer, *Volumetric and thermodynamic properties of fluids enthalpy, free energy, and entropy*, *Industrial & Engineering Chemistry* **50** (1958) 265.
- [40] B. J. Daly and F. H. Harlow, *Transport equations in turbulence*, *The Physics of Fluids* **13** (1970) 2634.
- [41] B. G. Galerkin, *Series solution of some problems of elastic equilibrium of rods and plates*, *Vestn. Inzh. Tekh* **19** (1915) 897.
- [42] B. D. Reddy, *Introductory functional analysis: with applications to boundary value problems and finite elements*, vol. 27. Springer Science & Business Media, 2013.
- [43] G. Green, *An essay on the application of mathematical analysis to the theories of electricity and magnetism*. Wezäta-Melins Aktiebolag, 1828.
- [44] A.-L. Cauchy et al., *Oeuvres complètes d'augustin cauchy. 1ère série, table générale*, .
- [45] O. C. Zienkiewicz, R. L. Taylor, O. C. Zienkiewicz and R. L. Taylor, *The finite element method*, vol. 36. McGraw-hill London, 1977.
- [46] R. A. Adams and J. J. Fournier, *Sobolev spaces*, vol. 140. Elsevier, 2003.
- [47] E. Kreyszig, *Introductory functional analysis with applications*, vol. 1. wiley New York, 1978.
- [48] K. Rektorys, *Variational methods in mathematics, science and engineering*. Springer Science & Business Media, 2012.
- [49] J. N. Reddy, *An introduction to the finite element method*, vol. 2. McGraw-hill New York, 1993.

- [50] D. L. Logan, *A first course in the finite element method*. Cengage Learning, 2011.
- [51] D. W. Pepper and J. C. Heinrich, *The Finite Element Method: Basic Concepts and Applications with MATLAB, MAPLE, and COMSOL*. CRC Press, 2017.
- [52] D. W. Pepper and J. C. Heinrich, *Intermediate finite element method: fluid flow and heat transfer applications, textbook edition 1* (1999) pg. 327.
- [53] D. Carrington, X. Wang and D. W. Pepper, *A predictor-corrector split projection method for turbulent reactive flow*, in *ICHMT DIGITAL LIBRARY ONLINE*, Begel House Inc., 2012.
- [54] O. C. Zienkiewicz and R. Codina, *A general algorithm for compressible and incompressible flowpart i. the split, characteristic-based scheme*, *International Journal for Numerical Methods in Fluids* **20** (1995) 869.
- [55] O. Zienkiewicz, K. Morgan, B. S. Sai, R. Codina and M. Vasquez, *A general algorithm for compressible and incompressible flowpart ii. tests on the explicit form*, *International Journal for Numerical Methods in Fluids* **20** (1995) 887.
- [56] D. Kelly, S. Nakazawa, O. Zienkiewicz and J. Heinrich, *A note on upwinding and anisotropic balancing dissipation in finite element approximations to convective diffusion problems*, *International journal for numerical methods in engineering* **15** (1980) 1705.
- [57] T. J. Hughes and M. Mallet, *A new finite element formulation for computational fluid dynamics: Iii. the generalized streamline operator for multidimensional advective-diffusive systems*, *Computer Methods in Applied Mechanics and Engineering* **58** (1986) 305.

- [58] D. W. Pepper and D. B. Carrington, *Application of h-adaptation for environmental fluid flow and species transport*, *International journal for numerical methods in fluids* **31** (1999) 275.
- [59] X. Wang, D. Carrington and D. W. Pepper, *Adaptive fem model for unsteady turbulent convective flow over a backward-facing step*, *Computational Thermal Sciences: An International Journal* **1** (2009) .
- [60] D. Carrington, *A fractional step hp-adaptive finite element method for turbulent reactive flow*, *Los Alamos Report LA-UR-11-00466*, *Los Alamos National Laboratory* (2011) .
- [61] J.-L. Guermond, P. Mineev and J. Shen, *An overview of projection methods for incompressible flows*, *Computer methods in applied mechanics and engineering* **195** (2006) 6011.
- [62] A. J. Chorin, *The numerical solution of the navier-stokes equations for an incompressible fluid*, *Bulletin of the American Mathematical Society* **73** (1967) 928.
- [63] R. Temam, *Remark on the pressure boundary condition for the projection method*, *Theoretical and Computational Fluid Dynamics* **3** (1991) 181.
- [64] P. M. Gresho, S. T. Chan, R. L. Lee and C. D. Upson, *A modified finite element method for solving the time-dependent, incompressible navier-stokes equations. part 1: Theory*, *International Journal for Numerical Methods in Fluids* **4** (1984) 557.
- [65] L. Gubin, B. Polyak and E. Raik, *The method of projections for finding the common point of convex sets*, *USSR Computational Mathematics and Mathematical Physics* **7** (1967) 1.

- [66] R. Löhner, K. Morgan, J. Peraire and M. Vahdati, *Finite element flux-corrected transport (fem-fct) for the euler and navier-stokes equations*, *International Journal for Numerical Methods in Fluids* **7** (1987) 1093.
- [67] E. Hylin and J. McDonough, *Derivation of projection methods from formal integration of the navier-stokes equations*, *Journal of computational and applied mathematics* **81** (1997) 349.
- [68] H. Bhatia, G. Norgard, V. Pascucci and P.-T. Bremer, *The helmholtz-hodge decompositiona survey*, *IEEE Transactions on visualization and computer graphics* **19** (2013) 1386.
- [69] A. J. Chorin, *Numerical solution of the navier-stokes equations*, *Mathematics of computation* **22** (1968) 745.
- [70] C. A. Fletcher, *Computational galerkin methods*, in *Computational Galerkin Methods*, pp. 72–85. Springer, 1984.
- [71] R. Rannacher, *On chorin’s projection method for the incompressible navier-stokes equations*, in *The Navier-Stokes equations IItheory and numerical methods*, pp. 167–183. Springer, 1992.
- [72] D. B. Carrington, *An h-adaptive finite element method for turbulent heat transfer*, tech. rep., Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2009.
- [73] C. K. Law, *Combustion physics*. Cambridge university press, 2010.
- [74] C.-C. Yu and J. C. Heinrich, *Petrovgalerkin method for multidimensional, time-dependent, convective-diffusion equations*, *International Journal for numerical methods in engineering* **24** (1987) 2201.

- [75] P. Nithiarasu and O. Zienkiewicz, *Analysis of an explicit and matrix free fractional step method for incompressible flows*, *Computer Methods in Applied Mechanics and Engineering* **195** (2006) 5537.
- [76] D. Torres, *Kiva-4 manual*, *Los Alamos National Laboratory* (2007) .
- [77] V. Berinde, *Picard iteration converges faster than mann iteration for a class of quasi-contractive operators*, *Fixed Point Theory and Applications* **2004** (2004) 716359.