CREATING REDUCED ORDER METHANE-AIR COMBUSTION MECHANISMS THAT SATISFY THE DIFFERENTIAL ENTROPY INEQUALITY

A Thesis

by

ALLEN REAM

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Paul G. A. Cizmas
Committee Members,	Adonios N. Karpetis
	John C. Slattery
	Eric L. Petersen
Head of Department,	Rodney D. W. Bowersox

December 2015

Major Subject: Aerospace Engineering

ABSTRACT

To simulate the combustion of a flame, or any reacting fluid flow, it is necessary to incorporate a reaction mechanism which describes the incremental steps and associated rates leading from reactant species to products. Detailed mechanisms include all possible species and elementary reactions so as to provide accurate solutions in a wide range of simulation conditions. Unfortunately, there is a large computational cost associated with the complexity of detailed mechanisms. Reduced mechanisms are created for specific conditions where simplifying assumptions can be made to decrease the complexity of detailed mechanisms. However, it has been shown that common reduced mechanisms produce violations of the differential entropy inequality (DEI), a local form of the second law of thermodynamics.

In this thesis, a new method is developed to determine rate parameters for reduced mechanisms which satisfy the DEI. Two mechanisms were created using this method. They are labeled the optimized two- and three-step mechanisms. In one-dimensional simulations, the optimized two- and three-step mechanisms predicted flame characteristics better than previous methods. The sharp transition from creation to destruction of minor species was better resolved and the error in flame speed prediction was reduced by up to 47.5%. A numerical model of the Sandia flame A was used as a test case to investigate violations of the DEI. Unphysical flammability limits in all reduced mechanisms allowed pre-ignition of the flame, causing errors when compared to experimental data. Both the optimized two- and three-step mechanisms produced no violations of the DEI. A two-dimensional flame simulation using the detailed GRI 3.0 mechanism was run for the first time and violations of the DEI were found.

ACKNOWLEDGMENTS

This thesis would never have come together without the support and encouragement of many people. First and foremost, I would like to thank my adviser Dr. Paul Cizmas for all the help and guidance he has given me over the past few years. At every step along the way Dr. Cizmas has challenged me, and in so doing, made me a better researcher and engineer. I am truly grateful for all his mentoring and will never forget his dedication to his students. I would like to extend my gratitude to each of the members of my research committee: Dr. Adonios Karpetis, Dr. John Slattery, and Dr. Eric Petersen. Discussions with each of them were instrumental in directing this research to where it is today. I would also like to thank Dr. Waruna Kulatilaka for so graciously standing in at my thesis defense.

Throughout my time here at Texas A&M I have made many amazing friends who have helped remind me that there is life outside of research. I would especially like to thank Kristin Nichols, Stoian Borissov, Bharat Mahajan, and Doug Famularo for spending time playing games, watching movies and sports, discussing books, and bringing laughter to my life. I would also like to thank my colleagues Forrest Carpenter, Raymond Fontenot, and Robert Brown for their company and conversation during our group lunch outings. A special thanks must also be made to my office mate Neil Matula. His interest and constant willingness for a discussion has gone a long way toward making this research what it is.

Last but not least, I would like to thank my family. Without my Mom and Dad, Grandmom and Poppop, Galen, Kerry, and Aunt Cathy I would have never made it here. Their guidance, encouragement, enthusiasm, and love has made all of this possible.

TABLE OF CONTENTS

Page

Page

LIST OF FIGURES

FIGURE Page

FIGURE Page

LIST OF TABLES

NOMENCLATURE

Latin Letters

$a_{(n)}$	Activity of species n
$a_{i,(n)}$	NASA polynomial coefficients of species n
A_f	Area of cell face
$A_{(r)}$	Arrhenius pre-exponential term for reaction r
c	Total molar density
$c_{(n)}$	Molar concentration of species n
$c_{p,(n)}$	Specific heat at constant pressure of species n
$\vec{d}$	Displacement vector
$\vec{d}_{(n)}$	Mass transfer driving force vector of species n
$\overline{\overline{D}}$	Diagonal scaling matrix used in trust-region-reflective algorithm
$D_{(n),mix}$	Mass diffusion coefficient of species n into a mixture
$D_{(n)(m)}$	Generalized diffusion coefficient of species n into species m
$D_{T,(n)}$	Soret thermal diffusion coefficient of species n
$\mathcal{D}_{(n)(m)}$	Binary diffusion coefficient of species n into species m
e	Total mass specific energy
$E_{a,(r)}$	Activation energy of reaction r
f	Function to fit to data for least squares curve fitting
f	Function to minimize in trust-region-reflective algorithm
$f_{(n)}$	Fugacity of species n
$f^0_{(n)}$	Fugacity of species n at reference condition
$\vec{f}_b$	Body forces
$\vec{g}$	Gradient vector

$g_{(n)}$	Gibbs free energy of species n
$g^0_{f,(n)}$	Gibbs free energy of formation of species n
h	Total mass specific enthalpy
$h_{(n)}$	Mass specific enthalpy of species n
$h^0_{f,(n)}$	Mass specific enthalpy of formation of species n at reference condition
$\overline{\overline{H}}$	Hessian tensor
i	Index for data points
$\overline{\overline{I}}$	Identity tensor
j	Index for parameters
$\overline{\overline{J}}$	Jacobian tensor
$\vec{J}_{(n)}$	Diffusive mass flux vector for species n
k	Index for iterations
k	Thermal conductivity
$k_{(n)}$	Thermal conductivity of species n
$k_{(r)}$	Arrhenius rate of reaction r
$k_{b,(r)}$	Backward Arrhenius rate of reaction r
$k_{f,(r)}$	Forward Arrhenius rate of reaction r
k_B	Boltzmann constant
k_{eff}	Effective thermal conductivity
$K_{c,(r)}$	Equilibrium constant for concentrations of reaction r
m	Mass
m	Second species index
M_{mix}	Molecular weight of the mixture
$M_{(n)}$	Molecular weight of species n
$\mathcal{M}_{(n)}$	Chemical symbol of species n

n	Index for species
$n_{(m)}$	Number of moles of species m
N	Local neighborhood used in trust-region-reflective algorithm
N_{faces}	Number of faces on a cell
N_{nodes}	Number of nodes on a face of a cell
N_p	Number of points
N_r	Number of reactions
N_s	Number of species
N_{SS}	Number of steady-state species
N_η	Number of parameters
P	Pressure
q	Local approximate function in trust-region-reflective algorithm
q_i	NASA polynomial temperature exponents
$q'_{(n)(r)}$	Reactant rate exponent of species n in reaction r
$q''_{(n)(r)}$	Product rate exponent of species n in reaction r
r	Index for reactions
$\vec{r}$	Vector of residuals for least squares curve fitting
$\vec{R}$	Vector of species net rates of production of length N_s
$R_{(n)}$	Molar net rate of production of species n
$R^*_{(n)}$	Approximate molar net rate of production of species n
$R_{(n)(r)}$	Molar rate of production of species n in reaction r
$\hat{R}$	Universal Gas Constant
$\vec{s}$	Step in solution calculated using the trust-region-reflective algorithm
$\vec{s}_2$	Second subspace direction in trust-region-reflective algorithm
$s_{(n)}$	Mass specific entropy of species n

S	Sum of squares of residuals
S_h	Source term in the energy equation
S_u	Flame speed
t	Time
T	Temperature
T_{ref}	Reference temperature
$T^*_{(n)}$	Dimensionless temperature of species n
$T^*_{(n)(m)}$	Dimensionless temperature of species n into species m
v	Magnitude of velocity vector
$\hat{v}$	Molar specific volume
$\vec{v}$	Velocity vector
$\mathcal{V}$	Cell volume
$\vec{x}$	Vector of independent variable data
$X_{(n)}$	Mole fraction of species n
$\vec{y}$	Vector of dependent variable data
$Y_{(n)}$	Mass fraction of species n
$\overline{Y}_{(n)}$	Mass fraction of species n scaled by molecular weight
Z	Compressibility factor

Greek Letters

$\beta_{(r)}$	Arrhenius temperature exponent of reaction r
γ	Relaxation factor
$\gamma_{(n)(r)}$	Third body efficiency of species n on reaction r
$\Gamma_{(r)}$	Net effect of third bodies on reaction r
Δ	Positive scalar used in trust-region-reflective algorithm

$\vec{\epsilon}$	Energy flux vector
$\varepsilon_{(n)}$	Lennard-Jones characteristic energy of species n
$\varepsilon_{(n)(m)}$	Mixed Lennard-Jones characteristic energy of species n and m
$\vec{\eta}$	Solution vector for least squares curve fitting of length N_η
$\Delta\vec{\eta}$	Shift vector
λ	Bulk viscosity
λ^k	Variable parameter at iteration k for Levenberg-Marquardt algorithm
μ	Average dynamic viscosity
$\mu_{(n)}$	Dynamic viscosity of species n
$\mu_{(n)}$	Chemical potential of species n
$\mu^0_{(n)}$	Chemical potential of species n at reference condition
$\overline{\overline{\nu}}$	Matrix of difference in product and reactant stoichiometric coefficients
$\nu'_{(n)(r)}$	Reactant stoichiometric coefficient of species n in reaction r
$\nu''_{(n)(r)}$	Product stoichiometric coefficient of species n in reaction r
$\overline{\overline{\Pi}}$	Stress tensor
ρ	Density
$\rho_{(n)}$	Density of species n
$\sigma_{(n)}$	Lennard-Jones characteristic length of species n
$\sigma_{(n)(m)}$	Mixed Lennard-Jones characteristic length of species n and m
τ	Appropriate time step used in Stiff Chemistry Solver
$\overline{\overline{\tau}}$	Viscous stress tensor
ϕ	Equivalence ratio
ϕ	Cell-centered scalar flow variable used in FLUENT
ϕ_f	Face value of scalar flow variable
$\overline{\phi}_f$	Average face value of scalar flow variable

$\overline{\phi}_n$	Average node value of scalar flow variable
$\phi_{(n)(m)}$	Convenience parameter for calculating ideal gas mixture properties
$\vec{\omega}$	Vector of progresses of reactions of length N_r
$\vec{\omega}_{(r)}$	Vector of progress of reaction data for reaction r of length N_p
$\omega_{(r)}$	Progress of reaction r
$\omega'_{(r)}$	Net forward progress of reaction r
$\Omega_{V,(n)}$	Viscosity collision integral of species n
$\Omega_{D,(n)(m)}$	Diffusion collision integral between species n and m

Acronyms

CFD	Computational Fluid Dynamics
DEI	Differential Entropy Inequality
DNS	Direct Numerical simulation
GRI	Gas Research Institute
ISAT	In-Situ Adaptive Tabulation
PEA	Partial Equilibrium Approximation
QSSA	Quasi Steady State Approximation
UDF	User Defined Function

1. INTRODUCTION

1.1 Motivation and Background

With the ever increasing capabilities of computers, the field of computational fluid dynamics (CFD) has grown to into an extremely useful tool to model and understand the physics of fluid flows. CFD enables analysts to model flows over exceedingly large and complex geometries. The results of these simulations allow for rapid design iterations which can result in an optimized geometry quickly and cheaply. The importance of CFD cannot be overstated given that the time and cost to fabricate and test parts for aerospace applications, especially turbomachinery, can be prohibitively high.

Despite recent advances in computer hardware and CFD software, there are still some classes of problems that require so much time and computational resources that they become intractable. Current state-of-the-art models involve the direct numerical simulation (DNS) of frozen or fixed chemistry flows. DNS requires that all scales of the flow be resolved which leads to simulations with billions of grid points for seemingly simple flow domains. To run such a simulation, even when operating in parallel on a modern supercomputer, is a substantial investment of time and resources. If chemical non-equilibrium is desired to be added to the simulation then the run time can easily become insurmountable.

To put the extra effort required for chemical non-equilibrium into perspective, a typical frozen chemistry simulation requires the solution of five equations for five unknowns at every grid point. The current benchmark model for the combustion of methane fuel, which is the simplest hydrocarbon, involves 53 chemical species. Considering that each species adds another unknown, the amount of extra equations to

solve is more than 10 times greater. This additional time required is also increased due to the disparate time scales between convection and chemical kinetics. More complex fuels can involve hundreds of species and thousands of reactions. Obviously the solution times can become insurmountable very quickly. Unfortunately, some of the most interesting and necessary problems to solve involve the flow of high temperature reacting gases. Notable examples are the combustion within turbomachinery, scramjets, and planetary reentry. Obviously, to solve these problems, simplifying assumptions must be made to the models.

One such simplification is to use a much coarser grid than is required for DNS and to incorporate models for the sub-grid scale flow structures. This is a technique used heavily in the study of turbulence. Simulations involving chemical non-equilibrium stand to gain substantial speedup through the use of simplifying assumptions to the reaction models. Reduced reaction mechanisms are chemical models where the number of species and reactions have been decreased, in some cases to one global reaction. The methods for creating these reduced reaction mechanisms traditionally involve the elimination of some chemical species and reactions from a detailed mechanism. Approximations are made based on experimental data and simple simulations using the detailed mechanism. Quasi steady state approximations (QSSA) and partial equilibrium approximations (PEA) are the two most commonly used tools to reduce a detailed chemical mechanism to a desired final form. The assumptions made in the reduction process typically stem from observations about the relative importance of species and reactions for specific operating conditions. Therefore, the resultant mechanism is essentially tailored to a very narrow range of operating conditions. Outside this range unphysical results are produced. It bears mentioning that there are many possible methods to create reduced mechanisms with varying levels of sophistication. A review of currently used methods is presented in section 1.2.

The dilemma that is faced in the creation of reduced mechanisms, and the focus of the research in this thesis, is that the processes used to develop the reduced mechanisms rarely consider the underlying thermodynamics. Recent research [3, 4] has shown that many commonly used reduced mechanisms violate the differential entropy inequality (DEI), which is the local form of the second law of thermodynamics. If the second law of thermodynamics is being broken by these models then the conclusions that have been drawn from simulation results are put into serious doubt. Therefore, the work that will be detailed in this thesis seeks to tailor reduced mechanisms so that violations do not occur. Previous methods to modify existing reduced mechanisms are discussed, and a new method which builds on these is presented.

1.2 Literature Review

Methane, being the simplest of the hydrocarbons, has been studied extensively both for its simplicity and exceptionally high energy density. The current benchmark model for methane combustion, known as the GRI 3.0 mechanism, was developed by the Gas Research Institute (GRI) throughout the 1990s [2]. This model, while complete, is burdensome for multi-dimensional numerical studies due to the vast computational resources it requires. The GRI team even acknowledges the fact that this mechanism may take a long time to run and that it contains many species and reactions not strictly necessary for pure methane combustion [2, Overview of GRI-Mech]. In order decrease solution run times, reduced mechanisms are used.

One of the first reduced mechanisms developed was the two-step mechanism of Westbrook and Dryer [1]. Their goal was to create a simple method for determining the rate of a global one-step reaction for hydrocarbon fuels. The proposed method varied the pre-exponential term, A, in the Arrhenius rate formulation until the best fit was found with four chosen tabulated values: the flame speed for a stoichiometric

mixture, the maximum flame speed, and the high and low flammability limits. The other terms of the Arrhenius rate were set based on assumed or published values. The activation energy was chosen from published data, the temperature exponent was set to zero, and the concentration exponents were set based upon pressure sensitivity results from Adamczyk and Lavioe [5]. While this worked for some complex hydrocarbons, the adiabatic flame temperature was over-predicted when the method was applied to methane fuel. The suggested solution was to use a two-step model which included the oxidation of carbon monoxide. The form of the second reaction was based on the work of Dryer and Glassman [6]. This mechanism has been very successful and is still in wide use today in commercial software packages [7].

The next phase of reduced mechanism development for methane combustion sought to recitfy the discrepancies between numerical reaction models and the simplistic models used in asymptotic studies. One of the first major milestones was the paper by Peters [8], in which a four-step mechanism for lean methane flames was developed. The process used was to systematically eliminate chemical species and reactions through the use of quasi steady state (QSSA) and partial equilibrium approximations (PEA). The process and mathematical implications of QSSA and PEA are described in great detail by Lam [9]. Unlike the Westbrook and Dryer mechanism, where the reaction rate parameters were found through curve fitting, the method employed here produced reaction rates which were algebraic functions of the elementary reactions from the complete mechanism.

One of the major downsides to the creation of Peters' four-step mechanism was that the reduction process was only made possible by assuming the flame to be in a very narrow band of possible operating conditions. Because of this, the resultant mechanism only performed well for certain flames. To illustrate this, at sufficiently high pressures Peters' four-step mechanism can be further reduced to a three-step

mechanism by employing a QSSA on the hydrogen ion. This mechanism was the basis of an extensive asymptotic analysis by Peters and Williams [10]. While this mechanism produced good results for lean to stoichiometric flames above pressures of one atmosphere, it did not predict moderately rich to rich flames well. Additional mechanisms were needed to model flames operating at higher equivalence ratios.

The work of Bilger [11] expanded upon the development of Peters to create a four-step mechanism for non-premixed methane air flames. The resulting mechanism gave good predictions of the structure of laminar flames and flame extinction for slightly rich flames. Significantly higher equivalence ratio flames were able to be modeled by mechanisms created by Seshadri, Bai, and Pitsch [12, 13]. By relaxing steady state approximations previously made to some radical species, five- and six-step mechanisms were created which allowed for the asymptotic analysis of richer flames. With the creation of these mechanisms, nearly the entire range of equivalence ratios was able to be accurately modeled for methane in air combustion.

All of the reduced mechanisms discussed so far have been created by hand, usually with the intent to perform an asymptotic analysis on flame structure. Recent research has been directed at developing methods to automatically determine reduced mechanisms for numerical studies. These methods include computational singular perturbation (CSP) [14], intrinsic low dimensional manifolds (ILDM) [15], functional equation truncation (FET) [16], and even automated QSSA methods [17]. While these methods take the effort out of creating reduced mechanisms by automating the process, they still do not escape the fact that they are specific to a reduced set of operating conditions.

The use of the differential entropy inequality (DEI) as a tool to analyze chemically reacting flows was first investigated in the thesis of Chambers [4]. Conditions for the automatic satisfaction of the DEI were later given as a theorem by Slattery et al. [18].

The thesis by Jones [3] introduces a method to determine reaction rate parameters for reduced mechanisms by attempting to curve fit simulation data from a full detailed reaction mechanism. This method was later formalized in a paper by Jones, Cizmas, and Slattery [19].

1.3 Objective and Scope

This thesis seeks to further the study of reduced reaction mechanisms and combustion modeling in general. The work presented is divided into two main parts. The first part was to investigate methods to create reduced mechanisms which automatically satisfy the DEI. A new method was created with the intention to reduce the need for user intuition, to improve the prediction of flame characteristics, and to reduce computational complexity. The new method sought to achieve these goals by estimating the reduced mechanism reaction rates from the species production rates of a detailed mechanism solution. The Arrhenius rate parameters for each reaction could then be found by curve fitting the estimated reaction rates. This method decoupled the reaction rates from the species production rates and, in so doing, allowed for the influence of desired species to be amplified through a weighting function. Two new reduced mechanisms were created using the proposed technique. The elementary steps for these mechanisms were based on the Westbrook and Dryer two-step mechanism [1] and the Peters and Williams three-step mechanism [10]. The created mechanisms showed significant improvements in minor species modeling, flame speed prediction, and produced no violations of the DEI.

The second objective of this thesis was to use the DEI as a tool to investigate current methane-air combustion mechanisms and assess whether the results were physical. An updated DEI user defined function (UDF) was created to post-process solution results. The UDF could accept any reaction mechanism without requiring

user modifications. This UDF was used on a simulation using the detailed GRI 3.0 mechanism as well as on many reduced mechanisms. Valuable insights are given as to the results of the simulations and violations are explained in detail. Recommendations for improvements to combustion models based on the satisfaction of the DEI are discussed.

1.4 Novel Aspects of this Thesis

This thesis presents a new methodology for determining reaction rates of reduced mechanisms. A new algorithm is developed which aids in solution convergence, minimizes computational time, and produces mechanisms with more physical flame characteristics.

Both a two-step and three-step mechanism were created using this process. Neither created mechanism produces any violations of the DEI.

A generalized UDF was created which could be applied to any combustion mechanism. This UDF was used on simulations of various reaction mechanisms which had never been analyzed for DEI violations before. Most interestingly, the UDF was used on a solution of a detailed GRI 3.0 simulation for the first time and violations of the DEI were found.

1.5 Outline

Chapter 2 outlines the governing equations of fluid flow and the basics of numerical solution techniques. Combustion mechanisms are discussed in chapter 3 with emphasis placed on reaction rates and the reduction process. Chapter 4 introduces the differential entropy inequality and the physical interpretation of each of its terms. The creation of UDFs and the use of the DEI as a post-processing tool are also presented. Numerical simulations of a physical experiment were carried out so that solutions could be compared to real data. The experimental setup and numerical

domain validation are presented in chapter 5. An improved method for determining reaction rate parameters is discussed in chapter 6 and two created mechanisms are examined. Chaper 7 presents the results of the numerical simulations, DEI violations, and provides interpretations of findings. Finally, conclusions and closing remarks are presented in chapter 8.

2. TRANSPORT EQUATIONS AND NUMERICAL SOLUTION METHOD

The majority of the work presented in this thesis was carried out using a computer to simulate the motion and interactions of various chemical species. Computational Fluid Dynamics (CFD) computer programs were used to numerically approximate the flow and combustion of fluids within a desired domain. Before diving into the results from the simulations, it is necessary to detail the theory used to develop these programs so as to fully understand the solution process and the assumptions involved. This chapter outlines the modeling process that is used by CFD solution tools. In section 2.1 the transport equations which govern the fluid flow are presented. Next, section 2.2 gives an introduction to numerical methods used to approximate the transport equations. The CFD program and solution algorithms used for the work herein are given additional emphasis.

2.1 Transport Equations

The transport equations are the equations which govern fluid flow. This set of equations is composed of the mass, momentum, energy and species conservation equations. The following development will give a short introduction to each of these equations including: how they were derived, limiting assumptions, how the equations are structured, and the calculation of the individual components.

2.1.1 General Derivation Procedure

It is common to see the mass, momentum, energy, and species equations referred to as transport equations since they govern the transport of these parameters through a domain of interest. However, they are also commonly called conservation equations

since the derivation of each starts from a logical statement of conservation of one of the above properties. An example of this would be that mass is conserved for a fixed mass system. The conservation statement would be that the time rate of change of mass within a fixed mass system is zero, $\frac{dm}{dt} = 0$. While this statement may seem obvious and perhaps trivial, it is the foundation of the extremely important continuity equation. Similar statements can be written for momentum, energy, and chemical species as well.

Since fluids do not flow as blocks of fixed mass, but as a constantly mixing continuum, the statements for a fixed mass system above must be modified to apply to a control volume. Control volumes are more practical for analyzing fluid flow through a domain. Reynolds Transport Theorem [20, pp. 84-5] provides the link between control mass and control volume equations. Applying Reynolds Transport Theorem to the conservation statements results in integral equations with mixed volume and surface integrals.

Up to this point the mathematical equations developed exactly model fluid flow. Unfortunately, the conservation equations cannot be solved without constitutive equations to relate the viscous stresses, heat fluxes, and species diffusion to the flow variables. This is where modeling error in the equations come from. Constitutive equations are usually linear approximations of a physical phenomena, or thermodynamic models with given bounds of applicability. While this introduces error into the model, highly accurate results can still be calculated as long as the solution is within the range of applicability of the constitutive equations.

To write the equations in their most commonly seen differential forms, it is necessary to use Gauss's Divergence Theorem [21, p. 241]. This allows all of the terms in the integral equations to be written in volume integral form. After applying Gauss's Theorem, it can be argued that the domain of integration is arbitrary so

that everything within the integral must be equal to zero. This directly results in the transport equations in their differential form. Depending on the type of numerical solution technique to be used, either the integral or differential form of the conservation equations may be beneficial. This will be discussed in more detail in section 2.2.

The preceding process was used to derive the conservation equations for mass, momentum, energy, and species. Each of the following subsections will present one of the equations in a commonly seen differential form, explain its terms, and how each is evaluated.

2.1.2 Mass Conservation

The conservation of mass equation, commonly referred to as the continuity equation, is given in its conservative form by

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0. \tag{2.1}$$

Here ρ is the fluid density and $\vec{v}$ is the velocity vector. This form of the equation assumes that the flow is unsteady, compressible, and has no mass generation sources. Further simplifications can be made if the flow is taken to be steady and incompressible, resulting in

$$\nabla \cdot \vec{v} = 0. \tag{2.2}$$

2.1.3 Momentum Conservation

There are three momentum conservation equations, one for each component of velocity. They are collectively referred to as the Navier-Stokes equations. They can

be written compactly in one equation using vector notation as follows:

$$\frac{\partial\left(\rho\vec{v}\right)}{\partial t}+\nabla\cdot\left(\rho\vec{v}\vec{v}\right)=\nabla\cdot\overline{\overline{\Pi}}+\rho\vec{f_b}. \tag{2.3}$$

New variables introduced here are the stress tensor $\overline{\overline{\Pi}}$, and externally applied body forces $\vec{f_b}$. Body forces are typically force potentials such as gravity or electromagnetic fields. Body forces can be applied to individual species if desired by multiplying by the respective mass fraction and summing over all the species. The stress tensor includes contributions from both the thermodynamic pressure and viscous stresses as can be seen below

$$\overline{\overline{\Pi}}=\overline{\overline{\tau}}-P\overline{\overline{I}}. \tag{2.4}$$

The viscous stress tensor is denoted as $\overline{\overline{\tau}}$, P is the thermodynamic pressure, and $\overline{\overline{I}}$ is the identity tensor. The viscous stress tensor is a symmetric tensor and is defined as

$$\overline{\overline{\tau}}=\mu\left(\nabla\vec{v}+\nabla\vec{v}^{T}\right)+\lambda\left(\nabla\cdot\vec{v}\right)\overline{\overline{I}}. \tag{2.5}$$

Here μ and λ are the dynamic and bulk viscosities respectively. Typically Stokes' Hypothesis is used to approximate the bulk viscosity as $\lambda=-\frac{2}{3}\mu$ [22, p. 67].

The dynamic viscosity is a strong function of both temperature and species composition. For many flows with small temperature changes it is perfectly acceptable to model the viscosity as a constant value. However, since the simulations undertaken herein are of combustion processes, in which there are large and rapid temperature and composition changes, it is essential to model the viscosity as a variable parameter. To do this, kinetic theory is used to get the most accurate results possible.

The dynamic viscosity for an ideal gas mixture is calculated from the individual

viscosities of the constituent species by the simple summation [23, p. 429]

$$\mu = \sum_{n=1}^{N_s} \frac{X_{(n)}\mu_{(n)}}{\sum\limits_{m=1}^{N_s} X_{(m)}\phi_{(n)(m)}}. \tag{2.6}$$

In this equation: $X_{(n)}$ is the local mole fraction of species n, $\mu_{(n)}$ is the local dynamic viscosity of species n, and N_s is the total number of chemical species. $\phi_{(n)(m)}$ is a parameter defined for convenience as

$$\phi_{(n)(m)} = \frac{\left[1 + \left(\frac{\mu_{(n)}}{\mu_{(m)}}\right)^{\frac{1}{2}} \left(\frac{M_{(m)}}{M_{(n)}}\right)^{\frac{1}{4}}\right]^2}{\left[8\left(1 + \frac{M_{(n)}}{M_{(m)}}\right)\right]^{\frac{1}{2}}}, \qquad \begin{matrix} n = 1, \ldots, N_s \\ m = 1, \ldots, N_s \end{matrix}. \tag{2.7}$$

The only new variable to be defined is $M_{(n)}$, the molecular weight of species n.

The equation to calculate the viscosity of the individual species is a result of solving the Boltzmann equation. The solution was independently derived by Chapman and Enskog and the following equation bears both their names [23, p. 428]

$$\mu_{(n)} = 2.67 \times 10^{-6} \frac{\sqrt{M_{(n)}T}}{\sigma_{(n)}^2 \Omega_{V,(n)}}, \quad n = 1, \ldots, N_s. \tag{2.8}$$

This is the Chapman-Enskog equation for species viscosity with units kg/m/s. It is the first of many equations named after these two researchers that will be used herein. In this equation: T is the temperature in Kelvin, $\sigma_{(n)}$ is a characteristic length in Angstroms, and $\Omega_{V,(n)}$ is called the viscosity collision integral. The collision integral is a dimensionless function of $k_B T/\varepsilon_{(n)}$, where k_B is the Boltzmann constant and $\varepsilon_{(n)}$ is a characteristic energy value. Due to this dependence, the collision integral is usually

given as an analytical approximation in terms of a non-dimensional temperature $T^*_{(n)}$

$$T^*_{(n)} = \frac{T}{\varepsilon_{(n)}/k_B}, \quad n = 1, \ldots, N_s. \tag{2.9}$$

A commonly used approximate form for the collision integral is given by [24, p. 9.5]

$$\Omega_{V,(n)} = \frac{A}{\left(T^*_{(n)}\right)^B} + \frac{C}{\exp\left(DT^*_{(n)}\right)} + \frac{E}{\exp\left(FT^*_{(n)}\right)}, \quad n = 1, \ldots, N_s, \tag{2.10}$$

where the constants are

$$A = 1.16145 \quad B = 0.14874 \quad C = 0.52487$$
$$D = 0.77320 \quad E = 2.16178 \quad F = 2.43787.$$

To wrap up with the calculation of the dynamic viscosity, it should be noted that (2.10) is only valid for $0.3 \leq T^*_{(n)} \leq 100$. Also, the characteristic length and energy variables, $\sigma_{(n)}$ and $\varepsilon_{(n)}$ respectively, are known as Lennard-Jones parameters and come out of the Lennard-Jones 12-6 theory of inter-molecular force potential. They can be found tabulated by many sources and will appear again later in chapter 4 when binary diffusion coefficients are discussed.

The layout of the Navier-Stokes equations (2.3) is such that the terms that govern the spatial and temporal changes in momentum are given on the left hand side of the equation with the forces that cause these changes given on the right. Focusing on the right hand side, the first term represents all those forces which act at the surface of the control volume. The second term then gives the contribution from all the forces acting within the control volume from remote sources.

2.1.4 Energy Conservation

The energy conservation equation is needed when temperature changes due to viscous stresses, heat transfer, or chemical reactions start to play a role in the model. The energy equation can be written in many different forms depending on which parameters are of particular interest. For the purposes of this thesis the energy equation will be presented below in the form used by FLUENT [25, pp. 133-4]. This form is preferred since FLUENT is the software package used for the simulations undertaken herein and because this form clearly shows the contributions from chemical reactions. The energy equation is given by

$$\frac{\partial(\rho e)}{\partial t} + \nabla \cdot (\vec{v}(\rho e + P)) = \nabla \cdot \left(k_{eff} \nabla T - \sum_{n=1}^{N_s} h_{(n)} \vec{J}_{(n)} + (\overline{\overline{\tau}}_{eff} \cdot \vec{v}) \right) + S_h. \quad (2.11)$$

Here e is the the total mass specific energy, $h_{(n)}$ is the mass specific enthalpy of species n, and S_h is a source term which will be described in detail below. Also introduced here are the effective thermal conductivity and effective viscous stress tensor, k_{eff} and $\overline{\overline{\tau}}_{eff}$ respectively. Both of these terms can include laminar and turbulent components when applicable. The final new term is $\vec{J}_{(n)}$, which is the diffusive mass flux vector for species n. The calculation of the diffusive mass flux vector is an involved process built on many results from kinetic theory. Many of the terms involved are very important when talking about the derivation of the differential entropy inequality, so it is much more natural to delay the calculation of the diffusive mass flux vector until chapter 4 where the whole theory can be given together.

The total mass specific energy is given by the simple expression

$$e = h - \frac{p}{\rho} + \frac{v^2}{2}, \quad (2.12)$$

where v is the magnitude of the velocity vector and h is the total mass specific enthalpy. The enthalpy is calculated as a summation of the individual species enthalpies multiplied by their respective mass fractions $Y_{(n)}$

$$h = \sum_{n=1}^{N_s} Y_{(n)} h_{(n)}. \tag{2.13}$$

The individual species enthalpies can be calculated from first principals using partition functions from kinetic theory. However, it is common practice to numerically approximate the species enthalpy from the caloric equation of state

$$h_{(n)} = \int_{T_{ref}}^{T} c_{p,(n)} dT, \quad n = 1, \ldots, N_s. \tag{2.14}$$

The bounds of integration are from a common reference temperature T_{ref}, typically taken as 298.15 K or 0 K, to the temperature where the enthalpy is to be computed. The specific heat at constant pressure, $c_{p,(n)}$, in the above equation is approximated by a curve fit called the NASA polynomial [26]

$$\frac{c_{p,(n)}\,(T)}{\hat{R}} = \sum_{i=1}^{5} a_{i,(n)} T^{q_i}, \quad n = 1, \ldots, N_s. \tag{2.15}$$

For each species n, the coefficients $a_{i,(n)}$ are found by curve fitting experimental data to a fourth order polynomial; q_i values of 0, 1, 2, 3, and 4. Typically multiple sets of coefficients are tabulated to fit different temperature ranges. The temperature ranges most commonly encountered are $300 \leq T \leq 1000$K and $1000 \leq T \leq 5000$K. The coefficients are published by many sources, however, for consistency, only the values tabulated by CHEMKIN [27] were used. There are similar NASA polynomials for enthalpy and entropy as well, though these are less commonly encountered in

literature.

Much like the Navier-Stokes equations (2.3), the energy equation (2.11) is arranged such that the spatial and temporal changes in the energy are given on the left hand side while the forcing functions causing this change are given on the right. There are four terms on the right hand side of the energy equation. The first three are due to thermal conduction, mass diffusion, and viscous effects respectively. The final term S_h is a source term which accounts for any volumetric sources of energy addition to the fluid. Most notably this term encompasses the heat release due to chemical reactions, which is of great importance since the work presented herein focuses on combustion. The source term is given by the following equation [25, p. 136]

$$S_h = -\sum_{n=1}^{N_s} h^0_{f,(n)} R_{(n)}. \tag{2.16}$$

Here $h^0_{f,(n)}$ is the mole specific enthalpy of formation of species n and $R_{(n)}$ is the molar net rate of production of species n. This equation differs slightly from that given by FLUENT [25, p. 136] due to $R_{(n)}$ being defined herein as a molar rate of production and FLUENT defining it as a mass rate of production. The rate of production is a critical parameter when analyzing combustive systems. Therefore, it will be given special attention in chapter 3 where combustion mechanisms and their rates of reaction are developed together.

At this point the basic equations that govern the flow of viscous fluids have been discussed. Expanding this model to apply to a mixture material requires that additional equations be included. These equations are the species conservation equations and are detailed in the next subsection.

2.1.5 Species Conservation

Classic fluid dynamics considers the flowing medium to be homogeneous and non-reacting. However, newer modeling efforts seek to simulate reactive systems such as the mixing of fuels and oxidizers for combustion, or very high temperature flows where dissociation starts to occur. For these cases it is necessary to include extra equations to model the motion of each constituent species. The species equation shown below is the necessary inclusion for such models

$$\frac{\partial\left(\rho Y_{(n)}\right)}{\partial t}+\nabla\cdot\left(\rho Y_{(n)}\vec{v}\right)=-\nabla\cdot\vec{J}_{(n)}+R_{(n)}M_{(n)},\quad n=1,\ldots,N_s-1. \tag{2.17}$$

No new variables have been introduced in this equation, however a critical distinction must be noted. The density and velocity vector are both properties of the bulk flow while the mass fraction, diffusive mass flux vector, and net rate of production are all properties of individual species. This means that for each species to be included in the model an extra species equation needs to be included, up to a total of N_s-1 species equations. The reason for using one less equation than the number of species is due to the definition of the mass fraction. Since the sum of the mass fractions must equal one, $\sum_{n=1}^{N_s} Y_{(n)}=1$, once N_s-1 species equations have been solved, the mass fraction of the last species at any given point in the flow is known implicitly. The species production rate $R_{(n)}$ is multiplied by the molecular mass for consistent units.

The species equations are laid out in the same manner as all the other flow flow equations before, with the spatial and temporal changes on the left, and the forcing functions on the right. In this case, the right hand side has two terms which give changes due to mass diffusion and species creation and destruction respectively. As

mentioned previously these two terms are built on a substantial amount of theory and will be discussed in more detail later: the diffusive mass flux vector $\vec{J}_{(n)}$ in chapter 4, and the net rate of production $R_{(n)}$ in chapter 3.

2.1.6 Equation of State

A careful inspection of the transport equations will reveal that there are more unknowns that there are equations. Without considering the species conservation equations, for a three-dimensional flow, there are five equations but seven total unknowns: three components of velocity, density, pressure, temperature, and energy. To solve this problem two thermodynamic equations of state must be used to relate the unknown variables. These are the thermal and caloric equations of state.

One form of the caloric equation of state, for a process that occurs at constant pressure, was introduced in (2.14). This relates the enthalpy to the temperature through the specific heat. The enthalpy can then be related back to the internal energy through (2.12). In general, the caloric equation of state writes the internal energy or enthalpy in terms of two other thermodynamic state variables. Any two can be chosen, however by choosing the internal energy to be a function of density and temperature as in

$$e = e(\rho, T), \tag{2.18}$$

and the enthalpy to be a function of pressure and temperature as in

$$h = h(P, T), \tag{2.19}$$

the classic specific heats at constant volume and pressure can respectively be found.

Depending on the fluid and temperature range of the case to be simulated, different models can be used for the specific heats. For air at near atmospheric conditions

a constant specific heat is a commonly used model. For combustive systems where there are large variations in temperature a variable specific heat model is more appropriate. An example of variable specific heats are the NASA polynomials of (2.15).

Thermal equations of state relate any one thermodynamic state variable to any two other thermodynamic state variables. Commonly the pressure is written in terms of the density and temperature as in

$$P = P(\rho, T). \tag{2.20}$$

For a wide range of temperatures and pressures, including dense gases, the thermal equation of state has been experimentally shown to be

$$\frac{P\hat{v}}{\hat{R}T} = Z, \tag{2.21}$$

where $\hat{v}$ is the molar specific volume. The important variable introduced here is the compressibility factor, Z. The compressibility factor is a measure of how far the gas is from ideal behavior. For a gas at low pressure and high temperature, a dilute gas, the compressibility factor is unity and (2.21) becomes the famous thermally perfect equation of state

$$P\hat{v} = \hat{R}T. \tag{2.22}$$

Near a fluid's critical point, the point at which the specific volume is the same for a saturated liquid and a saturated vapor, the compressibility factor starts to increase. Much research has gone into accurately modeling the change in compressibility factor as the critical point is approached. A notable model is the virial equation of state [24, pp. 4.11-7]. Historically, real gas behavior was not modeled using the compressibility factor. Initial models sought to model real gases by adding additional terms to the

thermally perfect equation of state. The van der Waals equation of state [21, p. 22] used this method and was one of the first real gas models to be widely employed.

For air at atmospheric conditions the compressibility factor deviates from unity by less than 1% [21, p. 22]. When combustion occurs, as long as the pressure remains atmospheric, the temperature will increase making the fluid behave even more as a thermally perfect gas. Therefore, (2.22) can safely be used as the thermal equation of state.

At this point all the relevant equations have been given to solve fluid flow problems analytically. Unfortunately, besides a few simple cases, there are no analytical solutions to the transport equations. Therefore, numerical methods must be used to find an approximate solution. The next section will outline the basic methods of calculating these numeric solutions with particular emphasis on the methods most heavily used for the simulations herein.

2.2 Numerical Solution Method

To find solutions to the transport equations for complex flow problems, numerical methods must be used. Discretization of of the computational domain and the governing equations will be briefly discussed here. Emphasis will be placed on the numerical tools used herein and the solution algorithms implemented by these numerical tools.

Computational fluid dynamics (CFD) is the study of numerical solutions to the transport equations which govern fluid flow. For most practical applications it is desired to understand the flow field around complex bodies. These can range from entire vehicles, such as the space shuttle, to individual components, such as a single blade of a compressor. This incredible variety of complex geometries, coupled with the inherently difficult and coupled nature of the transport equations makes finding

analytical solutions impossible for all but the most simple of cases. CFD solves this problem by providing computational tools to numerically approximate solutions so that the flow around complex structures can still be understood and optimized.

The basic premise of CFD is to find approximate solutions to the governing transport equations by discretizing the domain of interest into finite pieces and evaluating the solution at these locations. The creation of the discretized domain is carried out using grid generation software. Derivatives in the transport equations can be locally approximated about one of the finite pieces as algebraic equations in terms of the values of its neighbors. Once the resulting set of algebraic equations has been formed, the solution can can be easily calculated.

The choice of which form of the transport equations to use influences how the grid is discretized. Using the differential form of the equations leads to finite differences which are based around points in the computational domain. The integral form of the equations leads to finite volumes which consider cells that the fluid flows through. While both methods can produce accurate results, finite volumes are more commonly used. There are many reasons for this, however, one of the most important is that finite volumes are able to model shocks more accurately. This is due to the differences in assumptions between the integral and differential forms of the governing equations. The integral equations impose no restrictions on the smoothness of the resulting solutions. To derive the differential equations Gauss's Divergence Theorem was employed which requires that solutions be smooth. While shocks are not necessarily discontinuities, they occur in such a small spatial distance that they appear discontinuous to numerical solutions at finite grid locations. Therefore, finite volumes are better able to resolve shocks and are used more commonly.

The order of accuracy of the solution is directly tied to the complexity of how the derivatives are algebraically approximated. First order solutions use linear models

for approximating the derivatives which only require values at the nearest neighbors. Higher order methods can approximate curvature in the solution but require values from more distant cells and take longer to calculate.

The algebraic approximation also influences how the solution must be solved. Depending on the discretization method, the solution will either be solved explicitly or implicitly. Explicit methods calculate the solution at each point from known quantities. Explicit solvers are sometimes called marching solutions since each point can be solved for consecutively. In implicit methods a coupled set of equations is used so that solutions must be calculated for all spatial points at the same time using matrix methods or iterative solution techniques. Explicit methods typically are easier to code and require less memory, but can be numerically unstable.

There are trade-offs to be made between grid complexity, solution time, order of accuracy, and numerical stability. The choice of solver is strongly driven by the flow to be solved. CFD is a rich and diverse field of study which is constantly advancing the state-of-the-art of fluids modeling. Therefore, only the aspects pertaining to the flow solver tools used will be addressed in greater detail.

All fluid flow simulations were carried out using the commercial finite volume flow solver FLUENT [7]. FLUENT is a sophisticated program capable of simulating a diverse variety of fluid flows. It contains far more functionality than was utilized herein so only the solvers used will be explained. Much of the following is described in greater detail in the FLUENT Theory Guide [25].

FLUENT contains both a pressure-based solver and a density-based solver option. The pressure-based solver was developed for low-speed incompressible flows, while the density-based solver was created for high-speed compressible flows. The flows investigated herein are all low speed so the pressure-based solver was chosen. The pressure-based solver is an iterative algorithm which solves for the velocity field and

mass fluxes using the momentum equations and a pressure correction equation. The pressure correction equation is derived from the continuity and momentum equations so that the pressure corrected velocity field satisfies continuity. This differs from the density-based solver which uses the momentum and continuity equations directly.

The pressure-based solver contains options for both a segregated and coupled algorithm. The segregated algorithm decouples the equations and solves for each flow parameter sequentially. First, it solves for the components of velocity field using the momentum equations. The pressure-correction equation is then solved and the velocity field is updated based on the calculated corrections. Depending on the flow simulated, any additional scalar equations are solved sequentially. These can include energy, species, turbulence, and user defined scalars. The coupled algorithm solves the same equations as the sequential algorithm. The difference is that it solves the momentum and pressure-correction equations simultaneously, much like a density based implicit solver. All other scalar equations are solved sequentially as in the segregated solver.

The difference between the two algorithms lies in rate of solution convergence rate and memory usage. The benefit of the segregated solver is that it is memory efficient since it only needs to store data for one equation at a time. The coupled algorithm uses $1.5 - 2$ times the memory of the segregated solver since it must store all data for the momentum and pressure-correction equations simultaneously. However, the trade off is that the coupled solver significantly improves solution convergence since the equations are solved together. In a test case performed by FLUENT the coupled algorithm arrived at the same converged solution as the segregated solved while using 88% fewer iterations and taking 82% less time [28]. For all cases run, memory requirements were never an issue so the coupled solver was utilized to improve solution convergence speed.

When the integral forms of the transport equations are applied to each finite control volume within the domain surface integrals are formed to account for the fluxes entering and leaving the faces of the cell. To calculate the surface integrals, values for the flow variables at the face of interest are needed. However, during the solution process FLUENT stores all flow variables at the cell centers and assumes them to be cell averaged values. To find the face values for the calculation of the fluxes FLUENT uses an upwinding algorithm. Upwinding means that the face values are derived from quantities of the cell upstream, or upwind, relative to the local velocity. Two schemes were used to achieve a converged solution. Initial convergence of the solution was carried out first-order to decrease solution time. The final solution would then be carried out second-order to get the most accurate results possible.

First-order upwinding simply assumes the value flow variable on a cell face, ϕ_f, to be equal to the cell-centered value, ϕ, of the cell upstream of the face. When second-order accuracy is required a more sophisticated algorithm is required. This is given by

$$\phi_{f,2^{nd}Ord} = \phi + \nabla\phi \cdot \vec{d}. \tag{2.23}$$

Here, all quantities are still taken from the upwind cell-center. However, now the gradient of the scalar across the cell, $\nabla\phi$, as well as the displacement of the face center from the cell centroid, $\vec{d}$, are also required.

The evaluation of the gradient in (2.23) requires an additional calculation. The gradient is calculated at the cell-center using the Green-Gauss theorem which can be written in discrete form as

$$\nabla\phi = \frac{1}{\mathcal{V}} \sum_{i=1}^{N_{faces}} \overline{\phi}_{f,i} A_{f,i}, \tag{2.24}$$

where $\mathcal{V}$ is the cell volume, $\overline{\phi}_{f,i}$ is an averaged face value for face i, and $A_{f,i}$ is the area of face i. The average face value can be calculated using either a cell-based or node-based method. FLUENT recommends using a node-based method to ensure second order accuracy for non-orthogonal grids [23, p. 707]. Using this method the average face value can be calculated as

$$\overline{\phi}_f = \frac{1}{N_{nodes}} \sum_{i=1}^{N_{nodes}} \overline{\phi}_{n,i}. \tag{2.25}$$

Here N_{nodes} is the number of nodes on a face of the cell and $\overline{\phi}_{n,i}$ is the average value at node i calculated as a weighted average of the surrounding cell-centered values.

This concludes the description of the basic FLUENT solver options used. More sophisticated options used to aid in solution convergence will be discussed in chapter 5. While this has barely even scratched the surface of the functionality of FLUENT, let alone the depths of the field of CFD, it serves to outline the specific solution processes used and highlight why these options were chosen over potential alternatives.

2.3 Summary

This chapter provided a brief introduction to the governing equations of fluid flow, their derivation, the calculation of each term, and numerical solutions methods which are commonly used to solve them. The set of equations is collectively known as the transport equations since they govern the transport of mass, momentum, energy, and chemical species in a fluid. They are also referred to as conservation equations since the derivation of each starts with a logical statement of conservation of one of the transported properties. The classic equations which govern the bulk flow of a fluid are the mass, momentum, and energy equations. For cases when the fluid cannot be considered homogeneous species equations are added to properly model chemical reactions and species diffusion. There are typically more unknowns than

there are conservation equations. Therefore, thermodynamic equations of state are needed to relate quantities so that the system can be solved. The transport equations are impossible to solve analytically for all but the most simple of cases. Numerical methods are needed to approximate solutions so that fluid flows around complex geometries can be studied. The CFD program FLUENT was used for all numerical flow studies herein.

3. COMBUSTION MECHANISMS

Simulating the fluid dynamics of a combustion process requires the inclusion of models for chemical reactions which occur at finite rates. To fully model all intermediate steps of a combustion process is a computationally intensive problem, even for the most simple fuels. Considerable effort has been expended to create reduced reaction mechanisms which can speed up combustion simulations. However, these mechanisms are typically valid for only a narrow range of operating conditions. In this chapter the creation of reduced reaction mechanisms will be covered, and examples for methane-air systems will be given. Section 3.1 outlines the phenomenological laws for the rates of chemical reactions. Detailed methane-air mechanisms, with particular emphasis on the GRI 3.0 mechanisms, are discussed in section 3.2. A look into various reduction methods is presented in section 3.3. Reduced reaction mechanisms which are used herein will be highlighted.

3.1 Finite Rate Chemistry

At the atomic scale reaction processes are driven by molecular collisions. While a fluid may appear to be stagnant at a macroscopic scale, the constituent molecules are all moving in random directions with large velocities. As they move around they stand the chance of colliding with other molecules. Given sufficient energy these collisions can be violent enough to break the molecular bonds between the atoms and create new molecules. It is desirable to construct reaction rates which accurately reflects this microscopic behavior. The following derivation is based in large part from the work of Williams [29, p. 554-94].

In a reactive system with N_s chemical species, any arbitrary reaction out of the

N_r possibilities can be written as

$$\sum_{n=1}^{N_s} \nu'_{(n)(r)} \mathcal{M}_{(n)} \rightarrow \sum_{n=1}^{N_s} \nu''_{(n)(r)} \mathcal{M}_{(n)}, \quad r = 1, \ldots, N_r, \tag{3.1}$$

where $\mathcal{M}_{(n)}$ is the chemical symbol for species n and $\nu'_{(n)(r)}$ and $\nu''_{(n)(r)}$ are the stoichiometric coefficients for species n in reaction r as a reactant and a product respectively. For this single reaction, it can be reasoned that the reaction rate is proportional to the rate of molecular collisions of reactant species and the energy of the collisions. In the study of chemical kinetics the reaction rate equation which encompasses these ideas is given by the phenomenological law of mass action

$$\omega_{(r)} = k_{(r)} \prod_{n=1}^{N_s} c_{(n)}^{\nu'_{(n)(r)}}, \quad r = 1, \ldots, N_r. \tag{3.2}$$

Here $\omega_{(r)}$ is the volumetric molar progress of reaction r, $c_{(n)}$ is the molar concentration of species n, and $k_{(r)}$ is termed a reaction-rate constant.

In the law of mass action, (3.2), the product of the species concentrations reflects the rate of molecular collisions. The product requires that all reactant species be present in high concentrations to maximize the reaction rate. The energy of the collisions is factored into the equation through the reaction-rate constant, $k_{(r)}$. Despite being called a constant, it is actually a strong function of temperature. Higher temperatures lead to faster moving, more energetic, molecules which increases the macroscopic reaction rate.

For (3.2) to be valid, it is assumed that the reaction in (3.1) is one that can actually take place at the microscopic level. This requires that (3.1) be an elementary reaction step where exactly $\nu'_{(n)(r)}$ molecules of each species collide. Thus, the law of mass action does not apply to overall reactions such as the global methane-oxygen

oxidation reaction, $CH_4 + 2O_2 \rightarrow CO_2 + 2H_2O$. While, in practice, it is still applied to global reactions, such as in the Westbrook and Dryer mechanism discussed below, the results are less meaningful.

The molar rate of production of species n in reaction r is related to the progress of reaction through the stoichiometric coefficients from (3.1) through

$$R_{(n)(r)} = \left(\nu''_{(n)(r)} - \nu'_{(n)(r)}\right)\omega_{(r)}, \qquad \begin{matrix} n = 1, \ldots, N_s \\ r = 1, \ldots, N_r \end{matrix}. \tag{3.3}$$

The net molar production rate of species n is found by simply summing the contributions from each reaction

$$R_{(n)} = \sum_{r=1}^{N_r} R_{(n)(r)} = \sum_{r=1}^{N_r} \left(\nu''_{(n)(r)} - \nu'_{(n)(r)}\right)\omega_{(r)}, \quad n = 1, \ldots, N_s. \tag{3.4}$$

To evaluate the progress of reaction an expression is needed for the reaction-rate constant $k_{(r)}$. As previously mentioned this proportionality constant is a strong function of temperature, and in most systems it is not dependent on any other thermodynamic state variables. The form of this reaction-rate constant is given by the empirical Arrhenius expression

$$k_{(r)} = A_{(r)} T^{\beta_{(r)}} \exp\left[\frac{-E_{a,(r)}}{\hat{R}T}\right], \quad r = 1, \ldots, N_r. \tag{3.5}$$

Three new terms are introduced here which are reaction specific constants. The first is the positive valued pre-exponential $A_{(r)}$ which has units such that the units of $\omega_{(r)}$ are kmol/m^3/s or mol/cm^3/s. Next is the temperature exponent $\beta_{(r)}$ which is typically in the range $-1 \leq \beta_{(r)} \leq 2$. Finally, $E_{a,(r)}$ is the activation energy which is a measure of the energy addition required to cause a reaction to spontaneously

transition from reactants to products. The form of (3.5) comes from assessing the probability that the energy of a collision exceeds the activation energy using Boltzmann statistics.

Values for the three Arrhenius parameters are tabulated for many elementary reactions. Typically the values come from curve fitting (3.5) to experimental data or through sensitivity analyses [2]. Unfortunately, for reduced mechanisms which do not include elementary reaction steps, values cannot be readily found. Chapter 6 presents a novel method to fit Arrhenius parameters for reduced mechanisms to experimental data.

Near chemical equilibrium it becomes important to consider that reactions do not necessarily proceed in one direction. Since elementary reactions are collision driven, at equilibrium it is just as likely for the product species from (3.1) to collide and react as it is for the reactant species. Instead of including a separate reaction for the reverse direction into a chemical mechanism, (3.1) can be modified to include the reverse reaction

$$\sum_{n=1}^{N_s} \nu'_{(n)(r)} \mathcal{M}_{(n)} \rightleftharpoons \sum_{n=1}^{N_s} \nu''_{(n)(r)} \mathcal{M}_{(n)}, \quad r = 1, \ldots, N_r. \tag{3.6}$$

Now that reactions are allowed to proceed in both the forward and reverse directions (3.2) can be rewritten as a net rate for the reaction pair

$$\omega'_{(r)} = k_{f,(r)} \prod_{n=1}^{N_s} c_{(n)}^{\nu'_{(n)(r)}} - k_{b,(r)} \prod_{n=1}^{N_s} c_{(n)}^{\nu''_{(n)(r)}}, \quad r = 1, \ldots, N_r. \tag{3.7}$$

Here $k_{f,(r)}$ and $k_{b,(r)}$ are the Arrhenius rates for the forward and backward reactions respectively and $\omega'_{(r)}$ is known as the net forward progress of reaction r. When the forward and backwards reactions are combined using (3.6) the net molar production

rate of species n can be written in terms of $\omega'_{(r)}$ with only slight modification

$$R_{(n)} = \sum_{r=1}^{N_r} \left(\nu''_{(n)(r)} - \nu'_{(n)(r)}\right) \omega'_{(r)}, \quad n = 1, \ldots, N_s. \tag{3.8}$$

At chemical equilibrium the net rate of production of each species is zero. When multiple reactions are considered, this could imply that either the sum of all reactions is zero, the forward and backward component of each reaction are balanced, or a combination of both. However, for a system with one reaction equilibrium only occurs when the forward and backward reaction rates are balanced such that $\omega'_{(r)} = 0$. By (3.7) the definition of the equilibrium constant for concentration $K_{c,(r)}$ is found

$$K_{c,(r)} = \frac{k_{f,(r)}}{k_{b,(r)}} = \prod_{n=1}^{N_s} c_{(n)}^{\nu''_{(n)(r)} - \nu'_{(n)(r)}}, \quad r = 1, \ldots, N_r. \tag{3.9}$$

This is the same equilibrium constant that is found from the classic derivation starting with the thermodynamic functions.

In practice it is not always desirable to restrict a mechanism to obey the law of mass action. Such is the case for the Westbrook and Dryer two-step mechanism which uses concentration exponents that are not the stoichiometric coefficients. This mechanism will be discussed in greater detail in section 3.3. FLUENT, the CFD flow solver used, allows for more generalized reaction mechanisms to be implemented. For a reversible reaction FLUENT calculates the molar production rate of species n in reaction r as follows.

$$R_{(n)(r)} = \Gamma_{(r)} \left(\nu''_{(n)(r)} - \nu'_{(n)(r)}\right) \left(k_{f,(r)} \prod_{m=1}^{N_s} c_{(m)}^{q'_{(m)(r)}} - k_{b,(r)} \prod_{m=1}^{N_s} c_{(m)}^{\nu''_{(m)(r)}}\right), \quad \begin{matrix} n = 1, \ldots, N_s \\ r = 1, \ldots, N_r \end{matrix} \tag{3.10}$$

Here the concentration exponents of the forward reaction have had the reactant

stoichiometric coefficients replaced with $q'_{(m)(r)}$ which can take on any user defined value. The reverse reaction rate exponents are still the product species stoichiometric coefficients. Third body reactions are captured by the term $\Gamma_{(r)}$ calculated as

$$\Gamma_{(r)} = \sum_{n=1}^{N_s} \gamma_{(n)(r)} c_{(n)}, \quad r = 1, \ldots, N_r, \tag{3.11}$$

where $\gamma_{(n)(r)}$ is the third body efficiency of species n in reaction r. Third body effects occur in reactions involving three molecules in either the reactants or products. In these reactions two of the three molecules will either combine or break apart. The third molecule is present to supply or absorb the additional energy necessary for the other two to react. The third body efficiencies are a measure of how well each species acts as a third body for these types of reactions.

FLUENT calculates the reverse Arrhenius rate, $k_{b,(r)}$, in one of two ways. Either the user can specify the Arrhenius rate coefficients for the reverse reaction, or the equilibrium constant and forward Arrhenius rate can be used to solve (3.9) for the reverse Arrhenius rate. Both of these options come with their own set of issues. If the reverse Arrhenius coefficients are manually input then the user is forced to use the product species stoichiometric coefficients as the concentration exponents. If the equilibrium constant is used, additional operations must be performed to calculate its value. Additionally, for many reduced mechanisms the reactions are not elementary steps so the equilibrium constant is not well defined. Luckily, FLUENT provides an option for non-reversible reactions which solves this issue

$$R_{(n)(r)} = \Gamma_{(r)} \left(\nu''_{(n)(r)} - \nu'_{(n)(r)} \right) \left(k_{f,(r)} \prod_{m=1}^{N_s} c_{(m)}^{\left(q'_{(m)(r)} + q''_{(m)(r)} \right)} \right), \quad \begin{matrix} n = 1, \ldots, N_s \\ r = 1, \ldots, N_r \end{matrix}. \tag{3.12}$$

Here only the forward reaction is considered and the concentration exponents are

the sum of the manually defined values $q'_{(m)(r)}$ and $q''_{(m)(r)}$ for reactants and products respectively. By splitting reversible reactions into two non-reversible steps they can be implemented with (3.12) while allowing user defined concentration exponents for both the forward and reverse reactions.

One final point worth noting is that FLUENT performs all of its solver calculations using mass instead of number of moles. Therefore, once all the molar species production rates are calculated, they are converted to mass species production rates by multiplying by the appropriate molecular weight.

At this point the calculation of finite chemical reaction rates has been laid out enough to start investigating chemical reaction mechanisms. In the following sections the creation of detailed reaction mechanisms and their reduction to more manageable levels will be presented.

3.2 Detailed Reaction Mechanisms

A chemical reaction mechanism is a collection of elementary reaction steps which, when combined, provide an unbroken route from reactant species to products. There are varying levels of complexity of reaction mechanisms which depend on the application it is to be used for. Computationally, the simulation of reacting flows can be quite expensive so there is a trade-off between the level of mechanism detail and computational time. In section 3.3 this trade-off is considered more closely. However, the focus of this section will be on mechanisms which pay no heed to computational efficiency and seek to represent the physics of the combustion process as accurately as possible.

Detailed reaction mechanisms are chemical mechanisms which seek to include the effects of all possible elementary reactions to an overall combustion process. An example of this is the initiation reaction for the oxidation of methane in air

at atmospheric pressure and temperatures near the adiabatic flame temperature. According to Williams [29, p. 582] the initiation step is believed to either be

$$\mathrm{CH_4} + X \rightarrow \mathrm{CH_3} + \mathrm{H} + X,$$

or

$$\mathrm{CH_4} + \mathrm{O_2} \rightarrow \mathrm{CH_3} + \mathrm{HO_2},$$

where X stands for any third body molecule. Of these two reactions, the second is more important at lower temperatures. An investigator interested in creating a mechanism to use in large CFD simulations would judiciously choose whichever initiation reaction is better suited to the conditions of the simulation so as to increase solution speed. A detailed mechanism would simply include both since they both play a role, even if one is more prevalent than the other.

3.2.1 Chain Reactions

A typical reaction mechanism includes a set of elementary reaction steps which, when carried out in order, trace a route from reactant species to products. Due to this sequential nature the term chain reaction has been adopted to describe the reaction process. Each step of a chain reaction fills a specific purpose. For a chain reaction to take off it must be started in some way. Elementary reactions which begin the chain reaction are known as initiation reactions. Two examples of initiation reactions can be seen for methane-air combustion above. Following initiation, the steps which progress the reaction forward are known as chain-carrying or propagation reactions. The formation of product species signals the end of the chain-reaction. Elementary reactions which form the final products are known as chain-breaking or terminations steps.

Propagation steps are where the interesting dynamics of a chemical reaction are found. Chain-carriers are the molecules which are seen in almost every reaction which drive a chain-reaction from one step to the next. Typically, chain-carriers are simple radicals such as H, O, and OH. In many propagation reactions one chain-carrier reacts with another molecule and is consumed only to create a new chain-carrier as one of the products. However, some propagation reactions consume one chain-carrier and create two. This is known as chain-branching and can lead to high concentrations of chain-carriers which can cause explosive reactions. Detailed reaction mechanisms seek to include all possible initiation, propagation, branching, and termination steps so that the model exhibits the actual physics of the combustion process over a large range of conditions.

Table 3.1: Hydrogen-Oxygen Explosion Mechanism

Reaction	Equation	Type
1	$H_2 + X \rightarrow 2H + X$	Initiation
2	$H + O_2 \rightarrow OH + O$	Branching
3	$O + H_2 \rightarrow OH + H$	Branching
4	$OH + H_2 \rightarrow H_2O + H$	Propagation
5	$H \rightarrow$ walls	Termination
6	$2H + X \rightarrow H_2 + X$	Termination
7	$H + OH + X \rightarrow H_2O + X$	Termination
8	$H + O_2 + X \rightarrow HO_2 + X$	Term./Prop.

A simple example of a reaction mechanism which includes each type of elementary step is the hydrogen-oxygen explosion reaction from Table 3.1. In this mechanism the

chain carriers are H, O, OH, and in some cases HO_2. Reaction 1 is a simple initiation reaction which develops the radical pool. Reactions 2 and 3 are chain-branching while reaction 4 is a simple propagation reaction. Notice that in 2 and 3 the number of radicals is increased. Termination can occur in a number of ways. In reaction 5 the hydrogen radical interacts with the walls of the container, removing it as a chain-carrier. Reactions 6 and 7 are three body recombination reactions which result in product species. Reaction 8 can either be a termination or propagation reaction depending on the pressure. At low pressures HO_2 is considered a stable radical and the reaction terminates. At high pressures it can react through additional steps to form H_2O_2.

3.2.2 GRI 3.0

For the combustion of methane in air a widely used detailed mechanism is the GRI 3.0 mechanism [2]. It was developed through computational and experimental research funded by the Gas Research Institute and carried out at universities across the United States. The GRI 3.0 mechanism is comprised of 325 elementary reaction steps involving 53 chemical species. It was computationally optimized for use with methane and natural gas fuels based on experimental data. Due to limited data sources the resultant mechanism was optimized for premixed systems with temperatures of approximately 1000 to 2500 K, pressures of 10 Torr to 10 atm and equivalence ratios of 0.1 to 5. These ranges fall right in line with simulation conditions that will be used later in this thesis so the GRI 3.0 mechanism was chosen as a benchmark to which other chemical models will be compared.

To properly use the GRI 3.0 mechanism it is important to understand how it was created and its limitations. The GRI 3.0 mechanism was optimized to include accurate NO mechanisms, however it completely ignores soot formation. As previ-

ously mentioned, it was optimized for the combustion of methane and natural gas so it contains chemical pathways for more complex hydrocarbons, such as propane, which can be found in small amounts in natural gas. The GRI 3.0 creators warn that, while these pathways are included, the GRI 3.0 mechanism should not be used for the combustion of pure complex hydrocarbons.

Fundamental to the GRI 3.0 mechanism are the Arrhenius rate parameters and the determination of their values. The Arrhenius parameters are found for elementary reactions by fitting experimental data using an optimization routine. The basic process is to initially assign values for all of the parameters either from published sources or by estimation. Experimental data is gathered which includes combustion properties the mechanism is to predict. Examples of combustion properties are flame speed, flammability limits, and ignition delay. A computational sensitivity analysis is then carried out to determine which rate constants have the greatest affect on the combustion properties of interest. A simultaneous parameter optimization process is then undertaken, starting with the most sensitive parameters previously determined. During the optimization, the parameters are incrementally adjusted within strict constraints to see if better performance is found. Once an optimal set of parameters is found it is checked against an exhaustive set of experiments as a final validation.

A few final points of caution must be noted before the GRI 3.0 mechanism can be safely implemented. First is that the mechanism was optimized as a whole and any substitutions or omissions can drastically affect the results. Second, uncertainties still exist in the rates of the elementary steps even though the full mechanism may produce accurate results. These uncertainties can even exceed an order of magnitude for some elementary reactions [29, p. 581]. Therefore, if a reduced mechanism is desired to be constructed from the GRI 3.0 mechanism, the elementary rate parameters will need to re-optimized for the application.

3.3 Reduced Reaction Mechanisms

Reduced reaction mechanisms are used in situations where the complexity of a detailed mechanism is unneeded or overwhelming. In large computational simulations the use of a detailed mechanism might be too resource and time intensive so a simpler model is necessary. When discussing the structure of a flame, as in asymptotic studies, the great number of elementary reactions can muddle the underlying structure. In many cases it is sufficient to speak of a reaction as a single global step, as in the methane-oxygen oxidation reaction $CH_4 + 2O_2 \rightarrow CO_2 + 2H_2O$.

Reduced combustion mechanisms allow large complex combustion processes to be condensed to a much simpler set of reactions. These are more easily understood by an investigator, simple to implement and quick to calculate while still maintaining enough information about the underlying mechanisms to give accurate predictions about the structure of the flame. The measure of a good reduced reaction is its simplicity and its accuracy.

In this section two reduced reaction mechanisms are derived for the combustion of methane in oxygen. Both mechanisms have seen widespread use and represent different methods of creating reduced mechanisms. The first is the Westbrook and Dryer two-step mechanism [1] which fits rate equations to experimental data. The second is the Peters and Williams three-step mechanism [10] which makes steady state and equilibrium assumptions for certain operating conditions to reduce detailed mechanism complexity.

3.3.1 Westbrook and Dryer Two-Step Mechanism

In their 1985 paper [1], Westbrook and Dryer sought to develop a method to model hydrocarbon combustion using a single global reaction step

$$\text{Fuel} + \nu_1 \text{O}_2 \rightarrow \nu_2 \text{CO}_2 + \nu_3 \text{H}_2\text{O}, \tag{3.13}$$

where the ν_i are determined by the fuel used. The rate equation for this single step is a modified version of the forward-only progress of reaction from (3.2)

$$\omega = AT^{\beta} \exp\left(\frac{-E_a}{\hat{R}T}\right) c_{(Fuel)}^{q_1'} c_{(Oxidizer)}^{q_2'}. \tag{3.14}$$

Here q_1' and q_2' are variable parameters instead of the stoichiometric coefficients that are used in the law of mass action.

The process developed was to adjust the parameters of the rate equation until good agreement was found with the flame speed at stoichiometric conditions, the maximum flame speed, and the flammability limits. All optimization targets were found either experimentally or numerically using simulations of a detailed mechanism. To simplify the optimization the activation energy was taken from experimental results and the temperature exponent was set to zero. This left the pre-exponential and the exponents q_1' and q_2' to be determined.

To compute the rate of reaction for a given fuel, exponents q_1' and q_2' were chosen based on some given criterion, then the pre-exponential A was varied until the flame speed at stoichiometric conditions was matched. Initial testing used exponents $q_1' = q_2' = 1$ since a common assumption at the time was that the rate should be first order in both fuel and oxidizer. However, the best fit achievable with these exponents over-predicted the maximum flame speed and the rich flammability limit considerably. To

improve the results an approximation for the pressure dependence of the flame speed was used

$$S_u \propto P^{(q_1'+q_2'-2)/2}, \tag{3.15}$$

where S_u is the flame speed. By varying the value of the pressure exponent optimized rate constants were found. The best performing sets are seen in Table 3.2.

Table 3.2: Single-Step Rate Parameters From Westbrook and Dryer [1]

W&D Set No.	A	β	E	q_1'	q_2'
Set 2	1.3×10^8	0	48.8	-0.3	1.3
Set 4	1.0×10^{13}	0	48.8	0.7	0.8

Units are cm, mol, kcal, s, and K.

It is interesting to note that Set 2 gives the best results and has a negative fuel exponent meaning that the fuel acts as an inhibitor. While this may give the best agreement, it is numerically undesirable for complex combustion systems since it will result in infinite reaction rates where there is no fuel. For this reason Set 4 is used much more extensively in practice.

While the one step mechanism does a reliable job at matching the flame speed, it highly over-predicts the temperature. By neglecting additional product species, such as CO, the heat of reaction is falsely inflated. Therefore, Westbrook and Dryer implement the two-step mechanism seen in Table 3.3.

Table 3.3: Two-Step Mechanism Reactions

Reaction	Equation
I	$CH_4 + 1.5O_2 \rightarrow CO + 2H_2O$
II	$CO + 0.5O_2 \rightleftharpoons CO_2$

The reaction rate for the first equation can be any choice from Table 3.2 where only the pre-exponential needs to be modified slightly. The forward and reverse rates reaction II were determined similarly to those for the single-step reaction. The final rates for reaction II are

$$\omega_{(IIf)} = 10^{14.6} \exp\left(\frac{-40}{\hat{R}T}\right) c^1_{(\mathrm{CO})} c^{0.5}_{(\mathrm{H_2O})} c^{0.25}_{(\mathrm{O_2})}, \tag{3.16a}$$

$$\omega_{(IIb)} = 5 \times 10^8 \exp\left(\frac{-40}{\hat{R}T}\right) c^1_{(CO_2)}, \tag{3.16b}$$

and use the same units as in Table 3.2.

The Westbrook and Dryer two-step mechanism gives good predictions of flame speed and flammability limits and is simple to use in simulations. However, in some instances this simplistic nature is a shortcoming. Since the reactions and rates are both empirically determined there is little that can be learned about the underlying structure of the flame. Therefore, a more rigorous reduction method is desired that starts from a detailed mechanism and systematically makes simplifications until a reduced mechanism is found.

3.3.2 Peters and Williams Three-Step Mechanism

The method used to create the Peters and Williams three-step mechanism [10] has seen a great amount of success and is the foundation around which many asymptotic studies are built. The three-step mechanism presented here is a special case of a

four-step mechanism created by Peters [8]. Therefore, the following derivation will present the entire process used to create both mechanisms.

Table 3.4: Short Mechanism in the C1 Chain

Reaction	Equation	Influences
1	$CH_4 + H \rightarrow CH_3 + H_2$	Fuel Breakup
2	$CH_4 + OH \rightarrow CH_3 + H_2O$	
3	$CH_3 + O \rightarrow CH_2O + H$	Creation of Formaldehyde
4	$CH_2O + H \rightarrow CHO + H_2$	Decomposition of
5	$CH_2O + OH \rightarrow CHO + H_2O$	Formaldehyde
6	$CHO + H \rightarrow CO + H_2$	CO Creation
7	$CHO + X \rightarrow CO + H + X$	
8	$CHO + O_2 \rightarrow CO + HO_2$	
9	$CO + OH \rightleftharpoons CO_2 + H$	CO Oxidation
10	$H + O_2 \rightleftharpoons OH + O$	Oxygen Consumption,
11	$O + H_2 \rightleftharpoons OH + H$	Formation of Radicals
12	$OH + H_2 \rightleftharpoons H_2O + H$	
13	$OH + OH \rightleftharpoons H_2O + O$	
14	$H + O_2 + X \rightarrow HO_2 + X$	Three-Body Recombination
15	$H + OH + X \rightarrow H_2O + X$	
16	$H + HO_2 \rightarrow OH + OH$	Termination
17	$H + HO_2 \rightarrow H_2 + O_2$	
18	$OH + HO_2 \rightarrow H_2O + O_2$	

The first step was to create a "short" mechanism which consisted of the 40 known

elementary reaction in the C1 chain. Using the "short" mechanism a 1-D code was used to simulate the propagation of a stoichiometric methane-air flame. One of the outputs from this code were the influences of each of the reactions on the overall creation/destruction rates for each species. Using this it was possible to eliminate all reactions which did not significantly contribute while still maintaining a complete path from fuel and oxidizer to carbon dioxide and water. This resulted in the 18 reactions shown in Table 3.4 for which the Arrhenius parameters are all known. Note that this result is only valid for flames sufficiently close to stoichiometric.

For each species included in a reaction mechanism an additional species transport equation is needed. Therefore, the time required to calculate a solution scales much more heavily with the number of species than the number of reactions. To reduce the number of species some minor species can be assumed to be in steady state. To determine which species this is valid for requires the previously defined species equation (2.17).

Dividing (2.17) through by the molecular mass, rearranging, and using the definition of the net molar production rate from (3.4) the species equation can be written in terms of $\overline{Y}_{(n)} = Y_{(n)}/M_{(n)}$ as

$$\frac{\partial\left(\overline{Y}_{(n)}\right)}{\partial t} + \nabla\cdot\left(\rho\overline{Y}_{(n)}\vec{v}\right) + \nabla\cdot\frac{\vec{J}_{(n)}}{M_{(n)}} = \sum_{r=1}^{N_r}\left(\nu''_{(n)(r)} - \nu'_{(n)(r)}\right)\omega_{(r)}, \quad n = 1,\ldots,N_s. \tag{3.17}$$

The operator $\mathcal{L}$ can be defined as everything on the left hand side of (3.17)

$$\mathcal{L}(\overline{Y}_{(n)}) \equiv \frac{\partial\left(\rho\overline{Y}_{(n)}\right)}{\partial t} + \nabla\cdot\left(\rho\overline{Y}_{(n)}\vec{v}\right) + \nabla\cdot\frac{\vec{J}_{(n)}}{M_{(n)}}, \quad n = 1,\ldots,N_s, \tag{3.18}$$

such that

$$\mathcal{L}(\overline{Y}_{(n)}) = \sum_{r=1}^{N_r}\left(\nu''_{(n)(r)} - \nu'_{(n)(r)}\right)\omega_{(r)}, \quad n = 1,\ldots,N_s. \tag{3.19}$$

For intermediate species with small mass fractions all the terms in the operator $\mathcal{L}(\overline{Y}_{(n)})$ are small compared to the reaction rate terms and can be ignored. The quasi steady-state assumption (QSSA), therefore, is that

$$\mathcal{L}(\overline{Y}_{(n)}) = \sum_{r=1}^{N_r} \left(\nu''_{(n)(r)} - \nu'_{(n)(r)}\right) \omega_{(r)} = 0, \quad n = 1, \ldots, N_{SS}, \tag{3.20}$$

where N_{SS} is the number of steady-state species.

To apply the QSSA a metric is needed to determine if a given species has a low enough concentration. Peters [8] chose to apply the QSSA to all species with maximum weighted mole fractions of 1% or less. By running a simulation at stoichiometric conditions with the mechanism from Table 3.4 the intermediate species OH, O, HO_2, CH_3, CH_2O, and CHO were all found to be of low enough concentration to apply the QSSA.

For each species that the QSSA is applied to at least one rate equation can be eliminated as well. Since the linear operators $\mathcal{L}(\overline{Y}_{(n)})$ for the steady-state species are negligible, linear combinations of the steady-state species can be added to the non steady-state species to remove additional reaction rates. It is important to make sure to remove the fast reactions since the slow reactions are rate limiting for the overall mechanism. Through a careful analysis the following combinations can be found.

$$
\begin{aligned}
&\mathcal{L}(\overline{Y}_{(\mathrm{H})}) + \left[\mathcal{L}(\overline{Y}_{(\mathrm{OH})}) + 2\mathcal{L}(\overline{Y}_{(\mathrm{O})}) - \mathcal{L}(\overline{Y}_{(\mathrm{CH_3})}) + \mathcal{L}(\overline{Y}_{(\mathrm{CHO})}) - \mathcal{L}(\overline{Y}_{(\mathrm{HO_2})})\right] \\
&\qquad = -2(\omega_{(1)} + \omega_{(2)}) - 2(\omega_{(6)} + \omega_{(8)} + \omega_{(14)} + \omega_{(15)}) + 2(\omega_{(10)} + \omega_{(16)}) \\
&\mathcal{L}(\overline{Y}_{(\mathrm{H_2})}) + \left[-\mathcal{L}(\overline{Y}_{(\mathrm{OH})}) - 2\mathcal{L}(\overline{Y}_{(\mathrm{O})}) + 3\mathcal{L}(\overline{Y}_{(\mathrm{CH_3})}) + \mathcal{L}(\overline{Y}_{(\mathrm{CH_2O})}) + \mathcal{L}(\overline{Y}_{(\mathrm{HO_2})})\right] \\
&\qquad = 4(\omega_{(1)} + \omega_{(2)}) + (\omega_{(9)}) + (\omega_{(6)} + \omega_{(8)} + \omega_{(14)} + \omega_{(15)}) - 3(\omega_{(10)} + \omega_{(16)}) \\
&\mathcal{L}(\overline{Y}_{(\mathrm{O_2})}) + \left[\mathcal{L}(\overline{Y}_{(\mathrm{HO_2})})\right] = -(\omega_{(10)} + \omega_{(16)}) \qquad (3.21) \\
&\mathcal{L}(\overline{Y}_{(\mathrm{H_2O})}) + \left[\mathcal{L}(\overline{Y}_{(\mathrm{OH})}) + \mathcal{L}(\overline{Y}_{(\mathrm{O})}) - \mathcal{L}(\overline{Y}_{(\mathrm{CH_3})})\right] \\
&\qquad = -(\omega_{(1)} + \omega_{(2)}) - (\omega_{(9)}) + 2(\omega_{(10)} + \omega_{(16)}) \\
&\mathcal{L}(\overline{Y}_{(\mathrm{CO})}) + \left[\mathcal{L}(\overline{Y}_{(\mathrm{CH_3})}) + \mathcal{L}(\overline{Y}_{(\mathrm{CH_2O})}) - \mathcal{L}(\overline{Y}_{(\mathrm{CHO})})\right] = (\omega_{(1)} + \omega_{(2)}) - (\omega_{(9)}) \\
&\mathcal{L}(\overline{Y}_{(\mathrm{CO_2})}) = (\omega_{(9)}) \\
&\mathcal{L}(\overline{Y}_{(\mathrm{CH_4})}) = -(\omega_{(1)} + \omega_{(2)})
\end{aligned}
$$

The terms in the square brackets are the steady-state species and can be neglected. The subscripts of the reaction rates correspond to the reactions from Table 3.4. The reaction rates have been grouped for each species to suggest that certain combinations can be considered overall reaction rates for a reduced reaction mechanism. Using the leading coefficients for each of these groupings as stoichiometric coefficients, the reduced reaction mechanism in Table 3.5 is found.

Table 3.5: Four-Step Reaction Mechanism

Reaction	Equation	Rate
I	$CH_4 + 2H + H_2O \rightarrow CO + 4H_2$	$\omega_{(I)} = \omega_{(1)} + \omega_{(2)}$
II	$CO + H_2O \rightleftharpoons CO_2 + H_2$	$\omega_{(II)} = \omega_{(9)}$
III	$2H + X \rightarrow H_2 + X$	$\omega_{(III)} = \omega_{(6)} + \omega_{(8)} + \omega_{(14)} + \omega_{(15)}$
IV	$O_2 + 3H_2 \rightleftharpoons 2H_2O + 2H$	$\omega_{(IV)} = \omega_{(10)} + \omega_{(16)}$

Before reducing this four-step mechanism to the final three-step mechanism some additional work needs to be done with the rate equations. Expanding the reaction rates in Table 3.5 results in algebraic equations in terms of the Arrhenius parameters of the elementary reactions and the concentrations of the steady and non steady-state species. Since the steady-state species are not explicitly solved for during the solution process, additional relations must be found to relate their concentrations to the non steady-state species.

One method is to solve the large non-linear algebraic equations from the QSSA relations. However, this complicates the numerical solution considerably. A simpler method is to investigate the relative forward and reverse rates of individual elementary reactions to see if any are balanced. This is known as the partial equilibrium assumption (PEA). If any reactions are found to be in partial equilibrium, then simple algebraic equations can be constructed.

From simulations using the short mechanism from Table 3.4 at stoichiometric conditions it was found that reactions 11-13 were close to partial equilibrium in the high temperature region of the flame. Considering reaction 12, a simple algebraic equation can be found for the concentration of OH in terms of non steady-state

species and the equilibrium constant

$$c_{(\mathrm{OH})} = \frac{C_{(\mathrm{H})} C_{(\mathrm{H_2O})}}{C_{(\mathrm{H_2})} K_{c,(12)}}. \tag{3.22}$$

Similar relations can be found for the other steady-state species. It should be noted that PEA and QSSA produce results which are specific to a narrow range of operating conditions. If different operating conditions are desired these assumptions must be reevaluated and the reaction mechanism, and the associated rates, may need to be changed. Reaction mechanisms for flames operating at non-stoichiometric conditions have been created using this same method [11, 12, 13].

At pressures above approximately 1 atm an additional steady-state assumption can be made for the hydrogen ion [10]. This additional QSSA allows the four-step mechanism from Table 3.5 to be simplified to the desired three-step mechanism of Peters and Williams. This mechanism is shown in Table 3.6.

Table 3.6: Three-Step Reaction Mechanism

Reaction	Equation
I	$CH_4 + O_2 \rightarrow CO + H_2 + H_2O$
II	$CO + H_2O \rightleftharpoons CO_2 + H_2$
III	$O_2 + 2H_2 \rightarrow 2H_2O$

Much like the four-step mechanism, the reaction rates for the three-step mechanism are complex algebraic equations in terms of the elementary reaction Arrhenius parameters and species concentrations. These reaction rates can be cumbersome to deal with and many CFD flow solvers, such as FLUENT, do not accept them.

Chapter 6 introduces a novel method to create Arrhenius rate equations for reduced mechanisms so that they can be more easily implemented.

3.4 Summary

Chemical reactions are processes which are driven by the collision of molecules. At high enough energy levels the molecules can break apart and form into new molecules. The rate at which a chemical reaction progresses is mathematically described by the phenomenological law of mass action. This rate is proportional to number of collisions of the proper species and the energy of the collisions. A single global reaction shows reactants turning into products. In reality this process occurs through a series of elementary steps, each with its own rate. Each elementary step builds from the last in a process called a chain reaction. When all possible elementary reactions are considered, the result is a detailed chemical reaction mechanism. For the combustion of methane in air the GRI 3.0 mechanism is considered one of the best detailed mechanisms available. Detailed mechanisms are computationally expensive so reduced mechanisms are commonly used. Reduced mechanisms have diminished ranges of applicability since they are created using assumptions valid for only a narrow band of operating conditions. Westbrook and Dryer [1] created a two-step methane-air combustion mechanism based on empirical curve fitting experimental data. Peters and Williams [10] took a different approach and created a three-step mechanism by systematically reducing a detailed mechanism. The resultant reaction rates are complex algebraic equations. Later chapters will present a method for fitting Arrhenius rate equations to reduced reaction mechanisms.

4. THE DIFFERENTIAL ENTROPY INEQUALITY

The differential entropy inequality, hereafter referred to as the DEI, represents the local form of the second law of thermodynamics. It is derived by combining the energy balance equation with the entropy inequality. There are multiple forms of this equation, however the one most useful for the purposes of this thesis is given as [30, p. 451]

$$-\operatorname{tr}\left[\left(\overline{\overline{\Pi}}+P\overline{\overline{I}}\right)\cdot\nabla\vec{v}\right]+c\hat{R}T\sum_{n=1}^{N_s}\vec{J}_{(n)}\cdot\frac{\vec{d}_{(n)}}{\rho_{(n)}}+\sum_{r=1}^{N_r}\sum_{n=1}^{N_s}\mu_{(n)}R_{(n)(r)}+\frac{1}{T}\vec{\epsilon}\cdot\nabla T\leq 0. \quad (4.1)$$

In numerical simulations of fluid flow the conservation equations from chapter 2 are solved to give an approximation of how the fluid will behave in reality. A necessary, but not sufficient, condition to ensure solution accuracy would be to enforce that the solution obey the DEI at all points within the flow field. However, this is never done in current numerical simulations [18].

Part of the work herein is to investigate whether standard methane combustion models produce a solution that adheres to the DEI without it explicitly being enforced during the solution process. A user defined function (UDF) was created in FLUENT which post processes the solution data to see if there are violations of the DEI and which of the terms the violations are due to.

There are four primary terms to the DEI which represent the contributions to total entropy increase from viscous stresses, mass diffusion, chemical reactions, and heat transfer. Each of these terms will by discussed in detail in the following sections. Their terms will be fully explained and the necessary conditions for satisfaction of the overall DEI will be laid out.

4.1 First Term

The first term on the left hand side of the DEI represents entropy increase due to viscosity or shear stress. The full equation shows the first term as a function of three quantities, the stress tensor $\overline{\overline{\Pi}}$, the gradient of the velocity vector $\nabla\vec{v}$, and the thermodynamic pressure P, in the form

$$-\operatorname{tr}\left[\left(\overline{\overline{\Pi}}+P\overline{\overline{I}}\right)\cdot\nabla\vec{v}\right]. \tag{4.2}$$

As previously noted in equation 2.4 the stress tensor given by

$$\overline{\overline{\Pi}}=\overline{\overline{\tau}}-P\overline{\overline{I}},$$

which allows the first term of the DEI to be written as

$$-\operatorname{tr}\left[\overline{\overline{\tau}}\cdot\nabla\vec{v}\right]. \tag{4.3}$$

It is now clear that the entropy increase given by this term is due to shear stress only since pressure has been eliminated from the equation. By substituting equation 2.5 for the viscous stress tensor the most easily calculable form can be reached

$$-\operatorname{tr}\left[\mu\left(\nabla\vec{v}+\nabla\vec{v}^{T}-\frac{2}{3}\left(\nabla\cdot\vec{v}\right)\overline{\overline{I}}\right)\cdot\nabla\vec{v}\right]. \tag{4.4}$$

It should also be noted that Stokes' Hypothesis has been used to approximate the bulk viscosity as $\lambda=-\frac{2}{3}\mu$.

All of the above development for the first term is based on the idea that the material behaves as a compressible Newtonian fluid. Therefore, the first term automatically satisfies the DEI, that is $-\operatorname{tr}\left[\left(\overline{\overline{\Pi}}+P\overline{\overline{I}}\right)\cdot\nabla\vec{v}\right]\leq 0$, as long as the material

behaves as a compressible Newtonian fluid [18].

4.2 Second Term

The second term on the left hand side of the DEI is the contribution to entropy production from species diffusion. Rewriting this term below for analysis, it is apparent that it is a function of the total molar density c, the universal gas constant $\hat{R}$, temperature T, species density $\rho_{(n)}$, the diffusive mass flux vector $\vec{J}_{(n)}$ for species n relative to the mass averaged velocity, and the driving force for mass transfer corrected for temperature and pressure gradients $\vec{d}_{(n)}$ [19]

$$c\hat{R}T \sum_{n=1}^{N_s} \vec{J}_{(n)} \cdot \frac{\vec{d}_{(n)}}{\rho_{(n)}}. \tag{4.5}$$

There are four parameters in the second term that require additional information to calculate. Each of these parameters will be shown in terms of the dependent variables solved for in the numerical solution scheme. The first of these is the density of species n, $\rho_{(n)}$. It is calculated by the simple equation

$$\rho_{(n)} = \rho Y_{(n)}, \quad n = 1, \ldots, N_s, \tag{4.6}$$

where ρ is the density of the bulk flow and $Y_{(n)}$ is the local mass fraction of species n. Next, the total molar density c is given by

$$c = \sum_{n=1}^{N_s} c_{(n)}, \tag{4.7}$$

where

$$c_{(n)} = \frac{\rho_{(n)}}{M_{(n)}}, \quad n = 1, \ldots, N_s. \tag{4.8}$$

Recall that $M_{(n)}$ is the molecular weight of species n.

The driving force for mass transfer variable is given by Slattery as [30, p. 458]

$$\vec{d}_{(n)} = \nabla X_{(n)} + \frac{X_{(n)} M_{(n)}}{\hat{R}T}\left[\left(\overline{V}_{(n)} - \frac{1}{\rho}\right)\nabla P - \vec{f}_{b,(n)} + \sum_{m=1}^{N_s} Y_{(m)}\vec{f}_{b,(m)}\right],\ n = 1,\ldots,N_s. \tag{4.9}$$

This equation can be vastly simplified with two basic assumptions. The first is that the system is isobaric, meaning $\nabla P = 0$; and the second is that the external body forces, $\vec{f}_{b,(n)}$, act on all species evenly and are negligible. With these two assumptions the driving force for mass transfer vector can be written as [4, p. 103]

$$\vec{d}_{(n)} = \nabla X_{(n)}, \quad n = 1,\ldots,N_s. \tag{4.10}$$

The mole fraction is not directly calculated in the conservation equations of chapter 2. However, the mole fraction is related to the mass fraction, which is directly solved for, through the simple relationship [27, p. 21]

$$X_{(n)} = \frac{Y_{(n)}}{M_{(n)}\sum\limits_{m=1}^{N_s}\frac{Y_{(m)}}{M_{(m)}}}, \quad n = 1,\ldots,N_s. \tag{4.11}$$

To calculate $\vec{d}_{(n)}$ the gradients of the mole fractions are also needed. It is useful to express the gradients of the mole fractions in terms of the mass fractions and their gradients since the mass fractions are what are directly solved for. By taking the gradient of (4.11) the following relationship can be found

$$\nabla X_{(n)} = \frac{M_{(n)}}{\left(M_{(n)}\sum\limits_{m=1}^{N_s}\frac{Y_{(m)}}{M_{(m)}}\right)^2}\left[\nabla Y_{(n)}\sum_{m=1}^{N_s}\frac{Y_{(m)}}{M_{(m)}} - Y_{(n)}\sum_{m=1}^{N_s}\frac{\nabla Y_{(m)}}{M_{(m)}}\right], \quad n = 1,\ldots,N_s. \tag{4.12}$$

This can be further simplified by noting that the mean molecular weight of the

mixture, M_{mix}, is defined as [27, p. 21]

$$M_{mix} = \frac{1}{\sum\limits_{n=1}^{N_s} \frac{Y_{(n)}}{M_{(n)}}}. \tag{4.13}$$

This leads to the most easy to use form

$$\nabla X_{(n)} = \frac{M_{mix}^2}{M_{(n)}} \left[\nabla Y_{(n)} \sum_{m=1}^{N_s} \frac{Y_{(m)}}{M_{(m)}} - Y_{(n)} \sum_{m=1}^{N_s} \frac{\nabla Y_{(m)}}{M_{(m)}} \right], \quad n = 1, \ldots, N_s. \tag{4.14}$$

The final parameter to be developed is the diffusive mass flux vector for species n, $\vec{J}_{(n)}$. The diffusive mass flux vector has been encountered before in chapter 2 when exploring the energy and species conservation equations; (2.11) and (2.17) respectively. The calculation of the diffusive mass flux vector depends on the diffusion model used. Two different models are presented below.

A dilute solution is one where the mass fractions of all chemical species are small compared to a single carrier species. For dilute solutions Fickian diffusion is applicable and the diffusive mass flux vector is given by [23, p. 455]

$$\vec{J}_{(n)} = -\rho D_{(n),mix} \nabla Y_{(n)}, \quad n = 1, \ldots, N_s - 1. \tag{4.15}$$

Here $D_{(n),mix}$ is the mass diffusion coefficient for species n into the mixture. For solutions which cannot be assumed to be dilute, a multicomponent and thermal diffusion model must be used. The calculation of the diffusive mass flux vector for species n then becomes [23, p. 456]

$$\vec{J}_{(n)} = -\sum_{m=1}^{N_s-1} \rho D_{(n)(m)} \nabla Y_{(m)} - D_{T,(n)} \frac{\nabla T}{T}, \quad n = 1, \ldots, N_s - 1. \tag{4.16}$$

The new coefficients are: $D_{(n)(m)}$, the generalized diffusion coefficients of species n

into species m; and $D_{T,(n)}$, the Soret thermal diffusion coefficient of species n.

Equations (4.15) and (4.16) are only valid for the first N_s-1 species. The diffusive mass flux vector for the final species, $\vec{J}_{(N_s)}$, is found by noting that the sum of the diffusive mass flux vectors over all the constituent species is zero [30, p. 451]

$$\sum_{n=1}^{N_s} \vec{J}_{(n)} = 0. \tag{4.17}$$

Rearranging to explicitly solve for the diffusive mass flux vector of the final species gives

$$\vec{J}_{(N_s)} = -\sum_{n=1}^{N_s-1} \vec{J}_{(n)}. \tag{4.18}$$

In (4.15) the mass diffusion coefficient for species n into the mixture is calculated using [23, p. 455]

$$D_{(n),mix} = \frac{1 - X_{(n)}}{\sum\limits_{\substack{m=1 \\ m\neq n}}^{N_s} \left(X_{(m)} / \mathcal{D}_{(n)(m)} \right)}. \tag{4.19}$$

In (4.16) the generalized diffusion coefficients are calculated by [23, p. 457]

$$D_{(n)(m)} = [D] = [A]^{-1}[B], \qquad \begin{matrix} n = 1, \ldots, N_s - 1 \\ m = 1, \ldots, N_s - 1 \end{matrix}. \tag{4.20}$$

The matrices $[D]$, $[A]$, and $[B]$ are of size $(N_s - 1) \times (N_s - 1)$. The diagonal terms of the $[A]$ and $[B]$ matrices are given by

$$A_{(n)(n)} = -\left(\frac{X_{(n)}}{\mathcal{D}_{(n)(N_s)}} \frac{M_{mix}}{M_{(N_s)}} + \sum_{\substack{m=1 \\ m\neq n}}^{N_s} \frac{X_{(m)}}{\mathcal{D}_{(n)(m)}} \frac{M_{mix}}{M_{(n)}} \right), \quad n = 1, \ldots, N_s - 1, \tag{4.21}$$

$$B_{(n)(n)} = -\left(X_{(n)} \frac{M_{mix}}{M_{(N_s)}} + \left(1 - X_{(n)}\right) \frac{M_{mix}}{M_{(n)}} \right), \quad n = 1, \ldots, N_s - 1, \tag{4.22}$$

while the off-diagonal terms are given by

$$A_{(n)(m)} = X_{(n)} \left(\frac{1}{\mathcal{D}_{(n)(m)}} \frac{M_{mix}}{M_{(m)}} - \frac{1}{\mathcal{D}_{(n)(N_s)}} \frac{M_{mix}}{M_{(N_s)}} \right), \quad \begin{matrix} n = 1, \ldots, N_s - 1 \\ m = 1, \ldots, N_s - 1 \end{matrix}, \tag{4.23}$$

$$B_{(n)(m)} = X_{(n)} \left(\frac{M_{mix}}{M_{(m)}} - \frac{M_{mix}}{M_{(N_s)}} \right), \quad \begin{matrix} n = 1, \ldots, N_s - 1 \\ m = 1, \ldots, N_s - 1 \end{matrix}. \tag{4.24}$$

The new term encountered in (4.19) and (4.21)-(4.24) is the binary diffusion coefficient of species n into species m, $\mathcal{D}_{(n)(m)}$. The equation to calculate the binary diffusion coefficients bears a remarkable similarity to the species viscosity equation (2.8). This is because they both come from solutions to the Boltzmann equation by Chapman and Enskog. The working equation for the binary diffusion coefficients is [23, p. 463]

$$\mathcal{D}_{(n)(m)} = 0.00188 \frac{\left[T^3 \left(\frac{1}{M_{(n)}} + \frac{1}{M_{(m)}} \right) \right]^{1/2}}{P \sigma^2_{(n)(m)} \Omega_{D,(n)(m)}}, \quad \begin{matrix} n = 1, \ldots, N_s \\ m = 1, \ldots, N_s \end{matrix}. \tag{4.25}$$

Here, $\sigma_{(n)(m)}$ is a characteristic length in Angstroms for the intermolecular force law being used, and $\Omega_{D,(n)(m)}$ is the dimensionless collision integral for diffusion between species n and m.

The diffusion collision integral is almost identical to that of the viscosity collision integral. It is a dimensionless function of $k_B T / \varepsilon_{(n)(m)}$, where k_B is the Boltzmann constant and $\varepsilon_{(n)(m)}$ is a characteristic energy value. The diffusion collision integral given as an analytical approximation in terms of the non-dimensional temperature $T^*_{(n)(m)}$

$$T^*_{(n)(m)} = \frac{T}{\varepsilon_{(n)(m)} / k_B}, \quad \begin{matrix} n = 1, \ldots, N_s \\ m = 1, \ldots, N_s \end{matrix}. \tag{4.26}$$

The approximate form of the collision integral is given by [24, p. 11.6]

$$\Omega_{D,(n)(m)} = \frac{A}{\left(T^*_{(n)(m)}\right)^B} + \frac{C}{\exp\left(DT^*_{(n)(m)}\right)} + \frac{E}{\exp\left(FT^*_{(n)(m)}\right)} + \frac{G}{\exp\left(HT^*_{(n)(m)}\right)}, \quad \begin{matrix} n = 1, \ldots, N_s \\ m = 1, \ldots, N_s \end{matrix} \quad , \tag{4.27}$$

where the constants are

$$A = 1.06036 \quad B = 0.15610 \quad C = 0.19300 \quad D = 0.47635$$
$$E = 1.03587 \quad F = 1.52996 \quad G = 1.76474 \quad H = 3.89411.$$

The minor differences between the Chapman-Enskog equations for viscosity (2.8) and binary diffusion coefficients (4.25) are encapsulated in the fact that viscosity is only dependent on one chemical species while the binary diffusion coefficients are dependent on two. To use the same Lennard-Jones parameters, $\sigma_{(n)}$ and $\varepsilon_{(n)}$, in the binary diffusion equations, some mixing rules must be employed. Those usually chosen are [24, p. 11.6]

$$\sigma_{(n)(m)} = \frac{\sigma_{(n)} + \sigma_{(m)}}{2}, \quad \varepsilon_{(n)(m)} = \left(\varepsilon_{(n)}\varepsilon_{(m)}\right)^{1/2}, \quad \begin{matrix} n = 1, \ldots, N_s \\ m = 1, \ldots, N_s \end{matrix} \quad . \tag{4.28}$$

When using the thermal diffusion model in (4.16) the Soret thermal diffusion coefficients, $D_{T,(n)}$, must also be calculated. These are given by the empirical equa-

tion [23, p. 458]

$$D_{T,(n)} = -2.59 \times 10^{-7} T^{0.659} \times \left[\frac{M_{(n)}^{0.511} X_{(n)}}{\sum_{m=1}^{N_s} M_{(m)}^{0.511} X_{(m)}} - Y_{(n)} \right] \cdot \left[\frac{\sum_{m=1}^{N_s} M_{(m)}^{0.511} X_{(m)}}{\sum_{m=1}^{N_s} M_{(m)}^{0.489} X_{(m)}} \right], \quad n = 1, \ldots, N_s. \quad (4.29)$$

This form of the Soret thermal diffusion coefficient causes light molecules to diffuse more quickly towards hot surfaces than heavy molecules [3, 23]. The effect is rather small in many cases and is regularly ignored.

At this point all of the necessary terms to calculate the second term of the DEI have been detailed. The necessary conditions for this term to automatically satisfy the DEI are for the material to be a dilute gas, a dilute solution which obeys Fick's Law, or a well stirred chemical reactor [18]. Because of these restrictions the form of the diffuse mass flux vector given in (4.16) does not automatically satisfy the DEI. Simulations are carried out in chapter 7 to verify that positive values can be encountered when using multicomponent and thermal diffusion.

A common alternate form of the second term of the DEI is given by [30, p. 451]

$$c\hat{R}T \sum_{n=1}^{N_s - 1} \vec{J}_{(n)} \cdot \left(\frac{\vec{d}_{(n)}}{\rho_{(n)}} - \frac{\vec{d}_{(N_s)}}{\rho_{(N_s)}} \right). \quad (4.30)$$

A proof showing the equality of these two forms is given in appendix A.1. The form given in (4.5) is preferred here since it is easier to implement into code.

4.3 Third Term

The third term of the DEI represents the change in entropy due to chemical reactions. This term is of special significance for the work presented herein since combustion involves many competing and chain reacting chemical reactions. The

third term is given by the expression

$$\sum_{r=1}^{N_r} \sum_{n=1}^{N_s} \mu_{(n)} R_{(n)(r)}. \tag{4.31}$$

The two summations in this expression are over the total number of chemical species N_s and the total number of reactions being considered in the model N_r. The variables are $\mu_{(n)}$, the chemical potential of species n, and $R_{(n)(r)}$, the rate of production of species n in reaction r which was described in detail in chapter 3. A more compact form would be to evaluate the summation over the reactions implicitly by replacing the rate of production variable with $R_{(n)}$. This would be the total rate of production of species n with all reactions considered. While writing it like this would be shorter, the form given in (4.31) serves to reinforce that the number of reactions is an important parameter to consider.

The chemical potential for species n can be evaluated through the use the the relative activity, $a_{(n)}$, which is defined as a ratio of fugacities [31, p. 22]

$$a_{(n)} = \frac{f_{(n)}}{f^0_{(n)}}, \quad n = 1, \ldots, N_s. \tag{4.32}$$

Here $f_{(n)}$ is the fugacity of species n, and the superscript denotes the value at reference condition. The activity is related to the chemical potential through the following equation [31, p. 21]

$$\ln\left(a_{(n)}\right) = \frac{\mu_{(n)} - \mu^0_{(n)}}{\hat{R}T}, \quad n = 1, \ldots, N_s. \tag{4.33}$$

To write the third term in terms of known values, a couple assumptions need to be made. The first is that the fluid being considered is an ideal mixture, where the

volume change due to mixing zero. From this it can be shown that [31, pp. 36-7]

$$f_{(n)} = X_{(n)} f^0_{(n)}, \quad n = 1, \ldots, N_s. \tag{4.34}$$

Here $X_{(n)}$ is the mole fraction as defined previously. This equation resembles Dalton's law of partial pressures, but applies to non-ideal gases. Now the activity can be rewritten as

$$a_{(n)} = X_{(n)}, \quad n = 1, \ldots, N_s. \tag{4.35}$$

The second assumption has to do with the type of process being investigated. For a constant temperature, constant pressure process, the chemical potential at reference condition is equivalent to the Gibbs free energy of formation, $g^0_{f,(n)}$. Gibbs free energy of formation can be found in multiple ways. In some sources it is given as a tabulated value, or it can be calculated from other tabulated data by its definition

$$g_{(n)} = h_{(n)} - T s_{(n)}, \quad n = 1, \ldots, N_s, \tag{4.36}$$

where $h_{(n)}$ and $s_{(n)}$ are the enthalpy and entropy of species n respectively. Also, Gibbs free energy can be calculated from first principles using kinetic theory.

With both of these assumptions (4.33) can be rearranged and rewritten as

$$\mu_{(n)} = g^0_{f,(n)} + \hat{R} T \ln\left(X_{(n)}\right), \quad n = 1, \ldots, N_s. \tag{4.37}$$

Finally, the third term can now be given in its most easily calculable form

$$\sum_{r=1}^{N_r} \sum_{n=1}^{N_s} \left[g^0_{f,(n)} + \hat{R} T \ln\left(X_{(n)}\right)\right] R_{(n)(r)}. \tag{4.38}$$

The necessary conditions for the third term to automatically satisfy the DEI

are for all the chemical species to be dilute gases, and for all reactions to be both reversible and conform to the law of mass action. This is stated as a theorem by Slattery et al. [18]. This theorem is the basis of much of the work herein. Reduced reaction mechanisms can be created to follow the law of mass action, however, by their very nature are not complete and do not typically include reverse reactions. While reduced reactions do not strictly conform to these conditions for DEI satisfaction, efforts are taken to create mechanisms which automatically satisfy the DEI. These mechanism and their creation are discussed in chapter 6. Violations of the DEI of complete mechanisms, which adhere to this theorem exactly, are discussed in chapter 7.

4.4 Fourth Term

The fourth and final term of the DEI

$$\frac{1}{T}\vec{\epsilon}\cdot\nabla T, \tag{4.39}$$

gives the entropy generation due to heat transfer. There is only one term here that needs to be defined, $\vec{\epsilon}$, the energy flux vector. It is given by the following equation [30, p. 449]

$$\vec{\epsilon} = -k\nabla T - c\hat{R}T\sum_{n=1}^{N_s} D_{T,(n)}\frac{\vec{d}_{(n)}}{\rho_{(n)}}. \tag{4.40}$$

Fortunately, the only new term introduced here is the thermal conductivity k, which for an ideal gas mixture is calculated by the a simple summation [23, p. 437]

$$k = \sum_{n=1}^{N_s}\frac{X_{(n)}k_{(n)}}{\sum\limits_{m=1}^{N_s} X_{(m)}\phi_{(n)(m)}}. \tag{4.41}$$

Here $\phi_{(n)(m)}$ is the previously defined convenience term from (2.7). The individual species thermal conductivities, $k_{(n)}$, are calculated from kinetic theory by the equation [23, p. 436]

$$k_{(n)} = \frac{15}{4}\frac{\hat{R}}{M_{(n)}}\mu_{(n)}\left[\frac{4}{15}\frac{c_{p,(n)}M_{(n)}}{\hat{R}} + \frac{1}{3}\right], \quad n = 1, \ldots, N_s. \tag{4.42}$$

In this equation $\hat{R}$ is the universal gas constant, $M_{(n)}$ is the molecular weight of species n, $\mu_{(n)}$ is the dynamic viscosity of species n calculated from kinetic theory as in (2.8), and $c_{p,(n)}$ is the specific heat of species n calculated from a NASA polynomial curve fit by (2.15).

The only necessary condition for this term to automatically satisfy the DEI is that the material obey Fourier's law of heat conduction [18].

4.5 User Defined Function to Calculate Violations of the DEI

A user defined function (UDF) was created in FLUENT as a post processing tool to calculate the value of the DEI at each point in the domain of a reacting flow simulation. The UDF calculates the value of each term of the DEI and outputs data showing the location and magnitude of violations of each term individually and the DEI as a whole.

The structure of the UDF was originally created by Chambers [4, pp. 98,145-62]. However, it was specifically hard coded to only work for the Westbrook and Dryer two-step mechanism. Additionally, many values that were computed by FLUENT during the solution process were recomputed manually in the original UDF. This created two main problems. The first stems from FLUENT not publishing all values and equations used in their solution process. When manually calculating various terms, errors could be introduced by using different equations and parameters than those used by FLUENT. This has the potential to produce incorrect results when

calculating the DEI. The second issue is that the manual calculations are specific to only one set of options in FLUENT. If the user changed the solution options the hard coded calculations would be incorrect and need to be manually changed.

A significant portion of the work for this thesis was to generalize the UDF so that it could be applied to any reaction mechanism with no user input required. As many manual calculations as possible were removed and replaced with FLUENT macros which extract the necessary data from the FLUENT solver. This removed the issues of using different equations than FLUENT and the need to manually update the UDF if different solution options were chosen. Unfortunately, macros do not exist to extract all the diffusion coefficient data needed. Therefore, the UDF is limited to using full multicomponent and thermal diffusion calculated by kinetic theory only. In the following sections the use of FLUENT UDFs will be briefly explained and the specific options used to improve the DEI UDF will given.

4.5.1 FLUENT UDF Basics

A UDF is a tool used with FLUENT to enhance the standard features of the program. UDFs can be used for a wide variety of tasks such as: customizing reaction rates, calculating diffusivities, altering grids or boundary conditions, adjusting solution data, initializing solutions, or as a postprocessing tool. This is just a small sample of the possible uses of UDFs, but it illustrates how they can be used for a variety of different purposes. Depending on how it is to be implemented, a UDF can be called on demand to perform a certain task or can be hooked to the FLUENT solver and be automatically executed on every solution iteration. Not only can UDFs be used to provide additional information to the solver, they can be used in place of standard FLUENT function calls.

UDFs provide additional functionality to FLUENT and are therefore left to the

user to create. They are programmed as functions in the C programming language and hooked onto the FLUENT solver to be run. FLUENT provides an interface for hooking the UDF to the solver and a manual outlining their creation [32]. The coding, debugging, and validation of results is left to the user.

The basics of UDF implementation starts with creating the code file. Since UDFs are written in C, the file name must contain the `.c` extension. The inner workings of UDFs are based around what FLUENT calls macros. A macro is a pre-built piece of code which is used to extract or modify existing FLUENT solution data. These macros come packaged with the FLUENT installation in header files with the `.h` extension. Therefore, the first few lines of the UDF code are used to identify where the macros are stored using `include` statements. Every UDF must include the `udf.h` file which contains the bulk of common macros. If additional macros are needed extra header files can be included.

What each UDF does is characterized by a `DEFINE` macro. After the header files are included the proper `DEFINE` macro is called which helps FLUENT properly implement the UDF into the solution process. The body of the code is then written as desired implementing whichever FLUENT macros are needed. The FLUENT UDF manual [32] contains information about some commonly used macros, but the list is far from complete. If a certain piece of information is needed from the FLUENT solver, there is more than likely a macro for it. Unfortunately, there is very little documentation about many of the possible macros. The best place to search is through the header files. This will be discussed in more detail in the the next section.

To apply the UDF to the FLUENT solver the code must either be compiled or interpreted. Depending on the type of UDF and how it will be used, either one could be a viable option. Compiling typically allows for greater functionality and speed at

the cost of additional steps before the UDF can be used. Compiling will be discussed here since it was performed for all UDFs used herein. This is only a general outline to understand the basic steps. The detailed compilation process can be found in the UDF manual [32, pp. 287-309].

The compilation process is very similar to how any executable is created. A script called a `Makefile` is used to call a C compiler which creates the object file libraries necessary to run the UDF. These libraries are specific to the system architecture so will need to be recompiled if the UDF is to be used on a different computer. For a Unix based operating system there are two `Makefiles` and a file called `users.udf` which are required to perform the compilation process. These are all found in the FLUENT installation file structure. The user must then create a UDF library file structure by hand [32, pp. 296-301] and put the source code, `Makefiles` and `user.udf` in the correct places based on system architecture and if FLUENT is being run in serial, parallel, 2D, or 3D. The compiler is then called to create the object libraries.

The final step is to load the UDF into the FLUENT case file that it will be applied to. This can be done through the `Define` menu of the user interface by simply directing FLUENT to the location of the compiled library. Once the UDF is loaded into FLUENT it can either be executed on demand or hooked to the solver for execution at each solution iteration. Both options are found in the `Define` menu. FLUENT has specific instruction on how to hook each different type of UDF in the user manual.

4.5.2 Application to DEI

The UDF created to calculate the violations of the DEI was implemented as a post processing tool. For any given simulation the solution was run until convergence

was achieved. The UDF was then run to look for violations of the DEI in each cell of the flow domain. The macro used to create this UDF was the general purpose `DEFINE_ON_DEMAND` macro. UDFs built using the `DEFINE_ON_DEMAND` macro can be made to perform a wide variety of tasks which do not need to be executed during solution iterations. Due to the general nature of this macro the only required input argument is a name for the UDF. All other information needed for the specific UDF being created can be gathered through additional FLUENT macros and utilities.

To calculate violations of the DEI, the following UDF structure was implemented. First, variables describing the flow domain, cells, material, and reactions were set up. Loops were then taken over all species and reactions to extract material parameters which are constant over the entire domain. All necessary variable declarations were made and output files were opened. The next step was to loop over all the zones and cells of the domain to calculate the value of the DEI in each cell. Within this loop all necessary solution data was extracted from FLUENT using the appropriate macros. Each term of the DEI was then calculated and the results were written into user-defined memory locations for plotting using the `C_UDMI` macro. Finally, statistics about the number of violating cells were written to the screen and all output files were saved and closed.

Generalizing the UDF so that it could be applied to any reaction mechanism required that FLUENT macros be used to extract information about the constituent species and reactions. Unfortunately, as mentioned earlier, the FLUENT UDF manual does not contain a complete list of all available macros. To find the appropriate macros, the header files where the macros are stored were manually searched by brute force. Since there is no documentation for many of the macros used a discussion of the ones implemented is presented below.

Material and reaction constants were found by using FLUENT loop macros to pull

relevant information from the material data structure. Macros to do this were found in the `materials.h` file. To find material constants for each species the following macro was used to loop over the species.

```
mixture_species_loop(m,sp,i){...}
```

Here `m` is a pointer to the material, `sp` is a pointer to the species, and `i` is an integer which increments with each loop. Within this loop species properties can be extracted using the following macro.

```
MATERIAL_PROP(sp,prop)
```

The same pointer for species, `sp`, is used here as in the mixture loop macro. The input argument `prop` is a place holder for the actual property that is desired. The material properties and their function calls used in the DEI UDF are given in Table 4.1 below.

Table 4.1: Material Species Property Macros

Property	Macro	Units
Molecular Weight	`PROP_mwi`	kg/kmol
Lennard-Jones Characteristic Length	`PROP_charSigma`	Angstrom
Lennard-Jones Characteristic Energy	`PROP_charEpsK`	K
Enthalpy of Formation	`PROP_hform`	J/kg
Entropy of Formation	`PROP_sform`	J/kg/K
Reference Temperature	`PROP_reference_temp`	K

This table only contains the properties used in the DEI UDF. Additional properties can be found in the `material.h` file. When applicable the macro will output

values in mass specific form. However, in many cases they are input into the program as mole specific values. Therefore, units are given to alleviate confusion when using these values.

Reaction constants were acquired by using the `loop` macro.

```
loop(r,r_list){...}
```

The argument `r_list` is a pointer to a list of all the reactions while `r` points to a structure containing all the data for a single reaction. Table 4.2 outlines the available data within the structured array and its data type. Within the loop data can be extracted from the structure using the C arrow operator. For example, the activation energy can be found with the following syntax, `r->E`.

Table 4.2: Reaction Parameter Macros

Property	Macro	Data Type[Length]
Arrhenius Pre-exponential	`A`	`real`
Activation Energy	`E`	`real`
Temperature Exponent	`b`	`real`
Number of Reactants	`n_reactants`	`integer`
Reactant Species IDs	`reactant`	`integer[n_reactants]`
Reactant Stoichiometric Coefficient	`stoich_reactant`	`real[n_reactants]`
Reactant Concentration Exponent	`exp_reactant`	`real[n_reactants]`
Number of Products	`n_products`	`integer`
Product Species IDs	`product`	`integer[n_products]`
Product Stoichiometric Coefficient	`stoich_product`	`real[n_products]`
Product Concentration Exponent	`exp_product`	`real[n_products]`

Cell progresses of reaction were the last piece of solution data from FLUENT that did not have a documented macro. These were needed for the calculation of the third term of the DEI. The proper macro was found in the `species.h` file and it had the following syntax.

```
cell_reaction_rates(c,t,i,&rr,&rr_l,&rr_t)
```

The first two arguments are for additional built in macros, the cell thread, defined by `cell_t c`, and the thread pointer, defined by `Thread *t`, respectively. The third argument, `i`, is an integer to specify the reaction from which to get the desired information. The final three arguments return addresses for the reaction rate, a purely laminar reaction rate, and a purely turbulent reaction rate. To make this UDF as general as possible only the first output argument was used. The cell reaction rate macro is very versatile and allows progresses of reaction to be calculated in all possible ways. Reaction parameters can be input by hand into FLUENT; they can be loaded using the Chemkin mechanism importer; or they can be defined using a macro.

Despite all the work done to make the UDF as robust as possible, there are still a few limitations that must be discussed. First is that the UDF currently only accepts 2D simulations. If a 3D solution is desired, then additional work must be done to include gradients in the z-direction. The second limitation is that the UDF is hard coded to calculate the binary and thermal diffusion coefficients from kinetic theory. The reason for this that no macros could be found to correctly extract the diffusion coefficients from the solver.

There are two diffusion coefficients necessary for the calculation of the DEI, binary mass diffusion coefficients and thermal diffusion coefficients. No macro was ever found for the binary coefficients, however the macro `C_THERMAL_DIFF` returns the

thermal diffusion coefficients. This issue with this macro is that it does not return a valid value for the final species in the list. As discussed in chapter 2, the transport equation for the final species does not need to be evaluated since the sum of all the species mass fractions is unity. Therefore, FLUENT does not calculate the diffusion coefficients for the last species since it is not needed for their solver calculations. Unfortunately, it is required when calculating the DEI.

To acquire values for the diffusion coefficients it was necessary for them to be calculated manually within the UDF. All equations used were taken from FLUENT literature when possible. Unfortunately, the form of the collision integral from (4.27) was not published by FLUENT and had to be filled in from other sources [24, p. 11.6]. Therefore, slight variations in the value of the binary diffusion coefficients may be present between the FLUENT solver and the DEI UDF.

A second ramification of the diffusion coefficients being hard-coded in the UDF is that the UDF is specific to only one set of diffusion options. The options that must be used are as follows. In the **Species Model** dialog box both **Full Multicomponent Diffusion** and **Thermal Diffusion** must be selected. Then in the **Create/Edit Materials** dialog box **kinetic-theory** must be selected as the model for both **Mass Diffusivity** and **Thermal Diffusion Coefficient**. These options were chosen since the theory is built from the most basic principles and contains the least amount of assumptions. Solutions run using these options may run slightly slower, however they should produce more accurate results.

4.6 Summary

There are four main terms in the DEI which govern the production of entropy through viscosity, species diffusion, chemical reaction, and heat transfer. The third term is of particular significance since the conditions for its automatic satisfaction,

presented as a theorem by Slattery et al. [18], are not met by reduced reactions. Later chapters will discuss efforts undertaken to create reduced mechanisms which produce no violation of the DEI. The UDF created to calculate the value of the DEI at each point in the flow domain was explained. This UDF was originally hard coded for one specific reduced reaction and set of solver options. Great effort was expended modifying the UDF to accept any reaction and require no user modifications or interaction when solver options are changed. Unfortunately, the UDF could not be fully generalized since macros do not exist to extract all diffusion coefficient data from the solver. Because of this, the UDF is hard coded for binary mass diffusion coefficients and thermal diffusion coefficients which are calculated by kinetic theory only.

5. NUMERICAL MODEL OF PHYSICAL EXPERIMENT

To apply the DEI UDF to actual flow simulations a test case needed to be constructed. Experimental data from a laminar methane-air flame, named flame A, performed at the Sandia National Laboratories proved to be an ideal case to simulate. This chapter outlines the development of the numerical flow domain created to model the Sandia flame A. In section 5.1 the flame A experiment is briefly introduced. Section 5.2 outlines the reduction of the physical experiment to a numerical model. The three-dimensional experiment is reduced to a two-dimensional simulation through axisymmetric arguments. Physical and numerical boundary conditions are then compared. The design of the computational grid and a grid convergence study are presented to show that the results are grid independent. Finally, section 5.3 presents the numerical options used and makes comparisons to possible alternatives.

5.1 Sandia Flame A

Flame A is the first in a set of flame experiments conducted at the Combustion Research Facility of Sandia National Laboratories. These experiments were performed to study the effects of molecular diffusion and turbulence on species mass fractions [33] and scalar structure [34] for non-premixed and partially premixed methane-air flames. A variety of different flame conditions were tested, from laminar to turbulent, with varying fuel mixtures and burner configurations. Each combination of flame conditions was designated by a letter. Archives of data from these experiments are available [35]. These data are used for improving chemical models and validating numerical simulations.

Flame A was a laminar, premixed, simple jet, methane-air flame and was used as a daily calibration target. In the flame A experiment a simple tube releases a

premixed fuel mixture which stabilizes an attached flame at its exit. While there is always a small gap between the end of the tube and the flame, the flame can be considered attached when compared to a lifted flame. The tube was designed to be of a sufficient length to ensure that the velocity profile is fully developed at the exit plane. Additionally, while the fuel is premixed, it is at an equivalence ratio considerably above the fuel's flammability limit. The fuel tube protrudes slightly from the end of a square wind tunnel which provides a low speed entraining flow of air referred to as the co-flow. The entire experiment is open to ambient conditions and the fuel and co-flow are at standard atmospheric temperature at pressure before combusting. Further detailed information about the physical setup can be found in section 5.2.

Simultaneous measurements of CO, OH, H_2, NO, major species, temperature, and mixture fraction data were taken at three axial locations within the flame. These locations were 25, 50, and 100 mm downstream of the fuel tube exit. The imaging system used a combination of Raman scattering, Rayleight scattering, and laser-induced fluorescence. The imaging system is described in detail by Karpetis et al. [36]. The flame was scanned radially at the three axial locations and data was collected in single-point fashion. The radial resolution of the data was 500 microns. The data was presented as a single profile for each axial location spanning radially out from the axis of the flame.

5.2 Discretization and Modeling of the Physical Experiments

To use the Sandia flame A data as a comparison for created mechanisms a numerical model of the flame must first be created. This section outlines the efforts to accurately model the Sandia flame A. First, an idealized three-dimensional representation of the experimental setup will be given. Simplifications will then be incorpo-

rated to reduce the model to an axisymmetric two-dimensional domain. Boundary conditions will be detailed and the resultant grid will be discussed to highlight all important features. Finally, a numerical grid convergence study will be undertaken to show that the modeling domain used is grid independent.

5.2.1 Experimental Set-up and Axisymmetric Numerical Domain

The physical experimental setup for the Sandia Flame A, as presented above, consisted of a tube which released a premixed fuel mixture surrounded by a wind tunnel to provide an entraining co-flow of air around the flame. The flame was attached to the exit of the fuel tube. The wind tunnel was a square duct with an area of one square foot. The tube had dimensions of 7.72 mm inner diameter, 9.525 mm outer diameter, and extended 200 mm beyond the end of the tunnel. An illustration of the setup can be seen in Fig. 5.1. The entire setup was oriented vertically so that asymmetric effects on the flame due to gravity would be negligible. These experiments were open to ambient conditions so the flame can be considered isobaric.

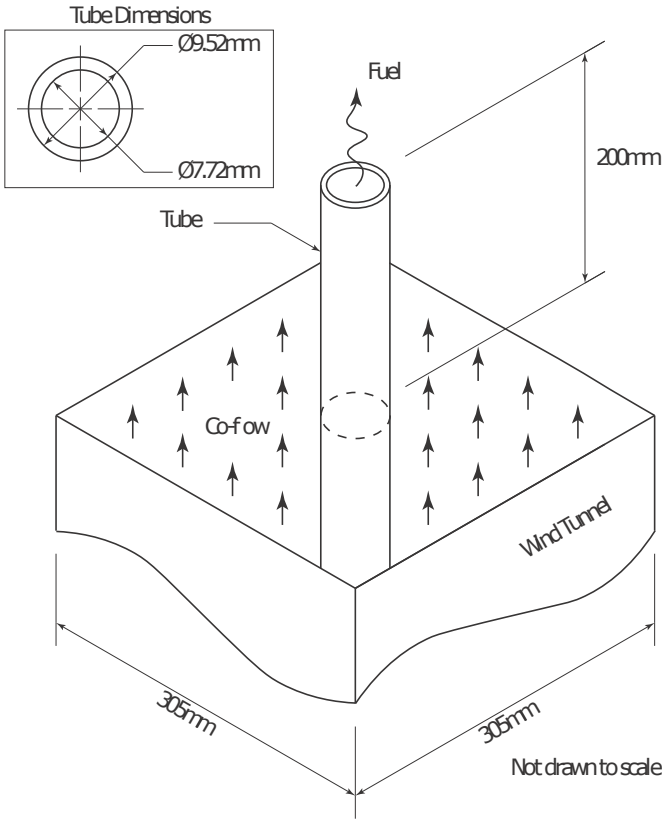

Figure 5.1: Isometric view of experimental setup for Sandia Flame A. All units are given in millimeters. Note that the drawing is not to scale.

The vertical orientation of the experiment is advantageous since all gravitational forces act parallel to the axis of the flame. By approximating the duct to be circular instead of square, the model of the flame can be simplified to a two-dimensional problem. This assumption is discussed further in section 5.2.3. Since there is no asymmetry in the radial or tangential directions only a two-dimensional slice of the flame needs to be modeled. This slice can then be revolved around the central axis of the flame to reconstruct the full three-dimensional profile. FLUENT is capable of performing two-dimensional axisymmetric simulations by simply choosing the option during the general problem setup and specifying the central axis of rotation [23, pp.

520-5]. The two dimensional modeling domain is shown in Fig. 5.2.

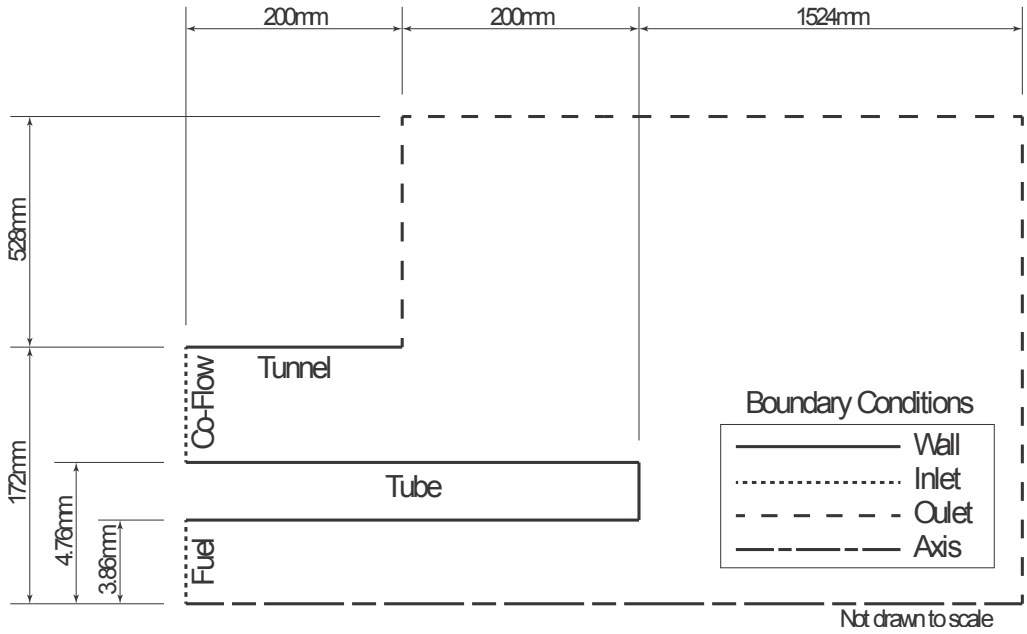

Figure 5.2: The two dimensional modeling domain simplified from the experimental setup. Dimensions and boundary conditions are supplied. Drawing is not to scale to clearly show inlet and wall boundary conditions.

5.2.2 Boundary Conditions

The boundary conditions that were utilized in this simulation were chosen to mimic the conditions from the Sandia experiments as closely as possible while allowing for the simulation to be simplified to a two-dimensional domain as discussed in section 5.2.1.

5.2.2.1 Wall Boundary Conditions

The centerline of the domain, around which the solution could be revolved to produce the three-dimensional flame, was modeled as an axis boundary condition. Hard surfaces, such as the wind tunnel and the fuel inlet tube, were modeled as

wall boundaries. All wall surfaces enforced no slip and no penetration of the flow adjacent to the wall. Thermally, the walls were modeled to allow no heat flux. Setting the temperature on the wall surfaces could negatively affect the solution. If the wall is hotter than the flame, premature ignition is possible. Alternately, if the wall temperature is too low the flame to become detached and dissipate. Therefore, choosing the heat flux through the wall to be zero, while not strictly physical, is a much safer assumption since it allows the wall temperature to equilibrate with the flame temperature.

5.2.2.2 Inlet Boundary Conditions

There are two fluid inlet boundaries to consider: the fuel inlet tube, and the co-flow. These were both modeled as velocity inlets where the flow velocity is the governing specified parameter. Based on mass flow rate data from the Sandia experiments, the average inlet velocity for the fuel and co-flow are 2.90 m/s and 0.40 m/s respectively. It should be noted that these velocities are taken to be constant across the width of the inlet. This would generally be a poor approximation of the velocity profile within a tube, however, as will be discussed in section 5.2.3, it is justified since the inlet boundary is at the beginning of a long tube through which the flow has ample time to fully develop boundary layers before exiting into the flame zone. This was verified by comparing the maximum velocity at the tube exit to that predicted by laminar pipe flow theory. The maximum velocity was within 3.1% of the predicted value which was considered sufficient for these studies. Other parameters specified at the velocity entrances are the static temperature, pressure, and composition. Values for all of these parameters can be seen in Table 5.1.

Table 5.1: Inlet and Outlet Boundary Conditions

Zone	Premixed Fuel	Co-Flow	Far Field
B.C. Type	Velocity Inlet	Velocity Inlet	Pressure Outlet
Velocity Magnitude [m/s]	2.90	0.40	N/A
Temperature [K]	300	300	300
Gauge Pressure [Pa]	0	0	0
Species Mass Fractions			
CH_4	0.1528	0.0000	0.0000
O_2	0.1944	0.2295	0.2295
CO_2	0.0004	0.0005	0.0005
CO	0.0000	0.0000	0.0000
H_2O	0.0066	0.0078	0.0078
N_2	0.6458	0.7622	0.7622

A couple points should be made regarding this information. First, the values for the mass fractions were taken from the Sandia experiments. The fuel is a premixed methane-oxygen mixture while the co-flow is ambient air. In simulations where additional species were incorporated, these were given zero values along the boundaries. Second, for this boundary condition type, the static gauge pressure is only an important parameter when dealing with supersonic entrances. For subsonic simulations, as are all the cases presented herein, the pressure is a calculated value and the value input is ignored by the program [23, p. 274].

5.2.2.3 Outlet Boundary Conditions

The final boundary conditions to discuss are the far field outlet boundaries. These boundaries represent the far extents of the numerical domain and are attempted to be modeled at such a distance from the flame as to not influence the solution. The far field boundaries of the domain are modeled using a pressure outlet boundary condition. This was chosen to enforce the static pressure along the boundary as ambient since the experiments were considered isobaric. Other conditions specified at these boundaries were the stagnation temperature and chemical composition. Values for these parameters can be seen in Table 5.1. It is important to note that this boundary condition only enforces the static pressure for normal outflow. All other parameters are only used in the case of backflow. In the case of backflow at some point along the boundary, the temperature and composition were set to most closely model the conditions of ambient air.

5.2.3 The Computational Grid

The mesh used for the studies herein was created using the commercial grid generation program Pointwise. The grid used for this research was created from scratch, however it was based off of the grids previously used by both Chambers [4] and Jones [3]. The first modification made to the previous grids was to re-size the tunnel entrance. In previous meshes the radius was taken to be six inches, corresponding to half the edge length of the square wind tunnel used in the physical experiments. Unfortunately, since this is an axisymmetric model, using the six-inch length would create a circular opening with a smaller cross-sectional area than the square box from the experiments. Assuming all other inlet properties to be the same, the mass flow rate would be reduced which could potentially alter the results. Therefore, the tunnel radius was increased such that the circular area being used in

the axisymmetric model was the same as the area used in the experiments. This ensured that the mass flow-rates would be equal.

The mesh is designed to provide the most accurate flow conditions at the start of the flame as possible. To achieve this, great care was taken to model the wind tunnel and gas inlet tube correctly. Initial grid iterations by Chambers [4, p. 92] consisted of a rectangular grid with distinct areas along the left wall that acted as inlets for the premixed fuel and the wind tunnel co-flow. The structure of the flame that resulted from this grid did not match the experimental results. This was due to the inlet flow having a constant velocity profile across the tube instead of taking into consideration the radial velocity variation due to the boundary layers [4, pp. 92-4]. To remedy this issue the current grid has an extra region added to the left hand side of the mesh to model the flow through the inlet tube. This region was added such that the flow could fully develop boundary layers before exiting into the area where combustion would occur. A comparison of both domains can be seen in Fig. 5.3.

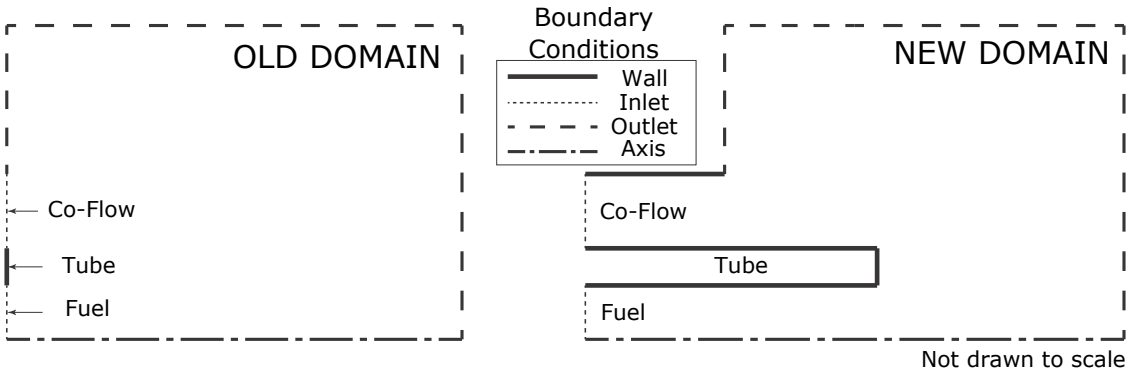

Figure 5.3: Comparison of domain structure iterations. The solution calculated on the old domain did not match experimental results. The new domain includes a segment of the tube and wind tunnel so as to better model the boundary layers along wall surfaces.

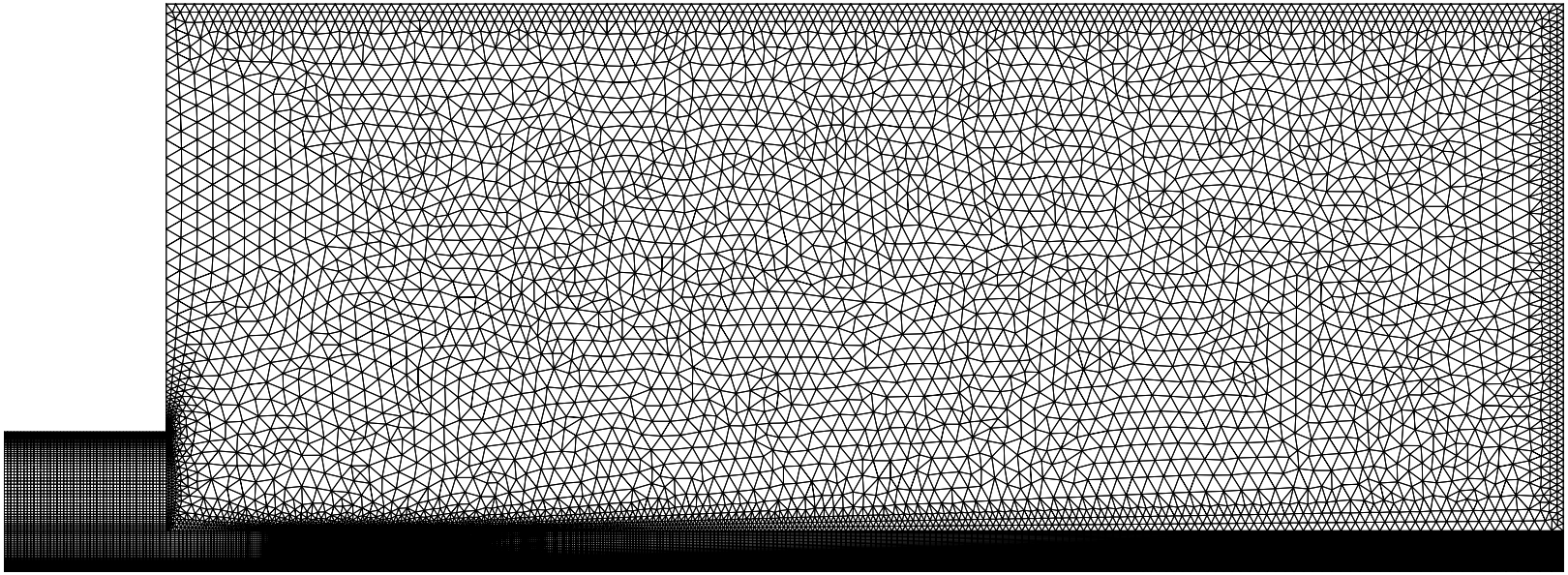

Figure 5.4: View of the mesh for the entire computational domain. A structured rectangular grid is used close to the center line to capture boundary layers and help with cell growth. An unstructured grid is used in the free stream to decrease the number of cells where resolution is less important.

An additional effort that was undertaken to make sure the boundary layers developed properly was the use of a structured grid all throughout the entrance tube and co-flow areas. Structured grids allow for the precise size of the cells along the walls to be specified. Having control of the size of the cells near boundaries is especially important since the grid influences the solution in two important ways. Firstly, the size of the cells must be appropriate for the flow structures that are desired to be resolved. Since boundary layers exist in such small regions of the flow next to wall boundaries, it is necessary to size the cells small enough such that proper resolution of the boundary layer is obtained. Secondly, the grid influences the solution through the cell growth rate. If the size of adjacent cells are radically different the solution can be negatively affected. A good way to make sure adjacent cells are similarly sized is to impose a growth rate in the direction normal to the wall.

The use of a structured grid allows for easy control of both the cell size and growth rate in critical areas where boundary layers are developing. Along the walls of the fuel entrance tube and the wind tunnel, the thickness of the first cell is determined such that the y^+ number is approximately 1 and the growth rate is set to be no greater than 1.3. In a turbulent flow the velocity profile is linear for $y^+ \leq 5$ [22, p. 415]. By placing the first cell within this viscous sublayer and growing slowly outwards, all structures in the boundary layer can be accurately captured. All the flows studied herein are laminar, so it would seem that these values are excessive. However, chemical reactions occur on a considerably faster time scale than laminar flows so a much finer grid is necessary to get proper resolution. The full grid is shown in Fig. 5.4. A close up view of the mesh around fuel tube exit is shown in Fig. 5.5 to emphasize the grid clustering.

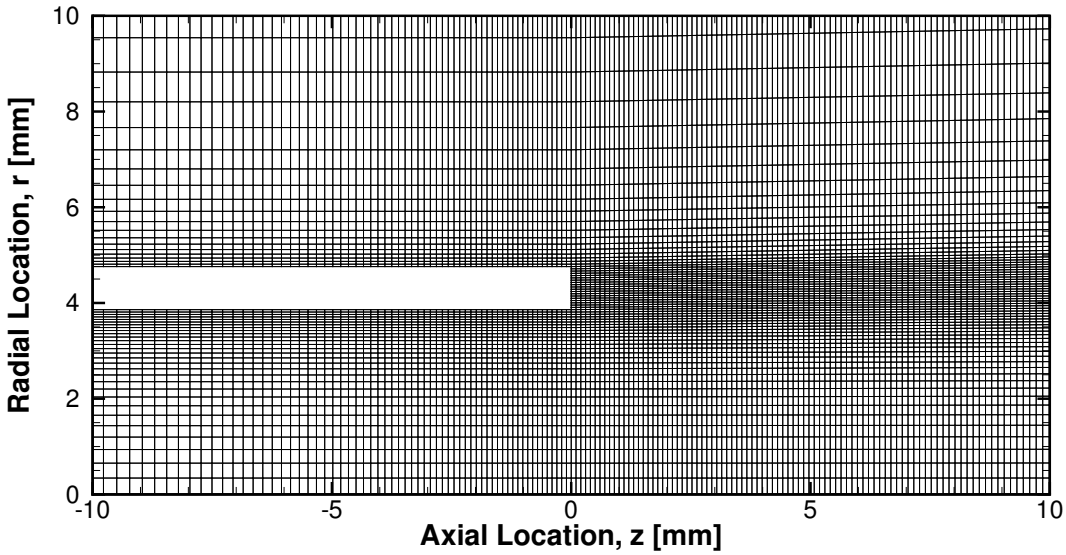

Figure 5.5: Zoomed in view of the mesh around the methane tube exit. The grid is clustered in the radial direction along both the top and bottom walls of the tube to capture the boundary layers of the bulk flow. Clustering is also used in the axial direction at the tube exit to help provide higher resolution where reactions are occurring.

Further mesh design characteristics were to cluster cells axially around the fuel tube exit as seen in Fig. 5.5. Cells were clustered axially near the tube exit to help provide additional resolution to the flame where the reactions are taking place most rapidly. In previously used meshes this clustering was achieved by using an unstructured grid. However, initial tests with the unstructured grid produced unphysical waves in the solution. The use of a structured grid all throughout the flame zone of the mesh helped to eliminate these waves and had the additional benefit of mildly speeding up the solution convergence time.

The final mesh characteristic to note is the use of an unstructured grid in the ambient air section of the grid which can be seen in Fig. 5.4. This was used to reduce computational time by using fewer, larger cells where solution gradients were small.

This concludes the notable design characteristics of the mesh used in this study.

To make sure that the grid chosen is producing accurate results a grid convergence study was undertaken and will be discussed next.

5.2.4 Grid Convergence Study

A grid convergence study was carried out to ensure that the solution was grid independent while using as few nodes as possible to speed up solution convergence time. Four different grids were constructed for this study. The number of cells in each grid are shown in Table 5.2 and a characteristic view of each grid is shown in Fig. 5.6. A common scheme for grid convergence studies uses three nested meshes. For a two-dimensional structured mesh the medium grid contains four times the number of cells as the coarse grid and the fine grid has 16 times as many. By looking at the number of cells in Table 5.2 this trend is loosely followed when considering the extra-coarse, coarse, and fine grids as the three grids for the study. The inclusion of the grid labeled as medium will be explained below.

Table 5.2: Number of Cells in Grids for Convergence Study

Grid	Number of Cells
Extra-Coarse	22999
Coarse	84780
Medium	153847
Fine	260158

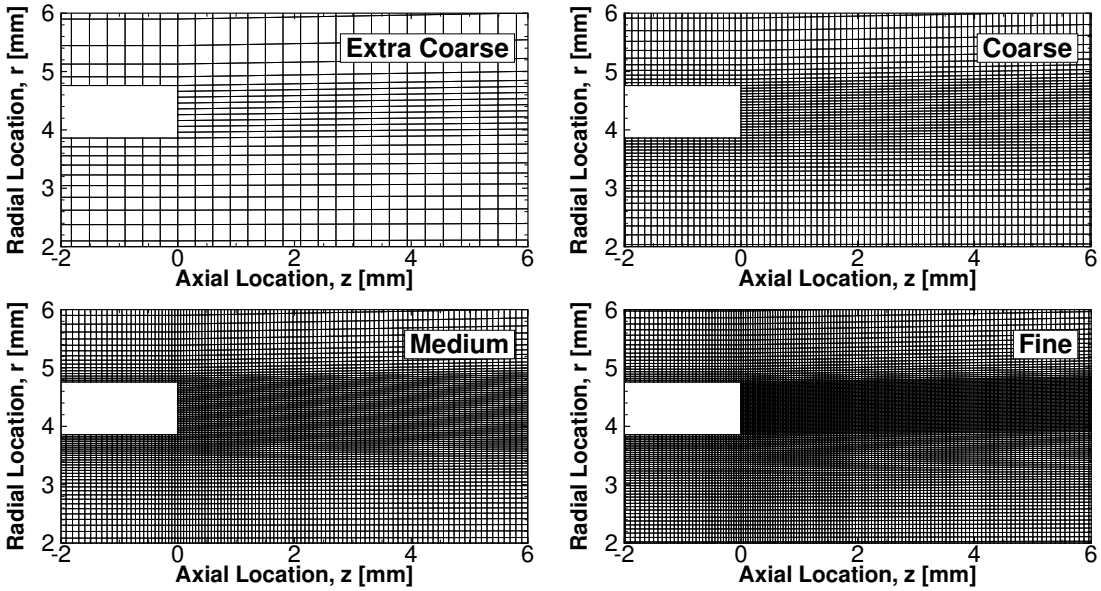

Figure 5.6: Comparison of a characteristic section near the fuel tube exit of all four meshes used in the grid convergence study. A close view of each grid is shown instead of the full mesh to make differences as clear as possible.

The reason that the number of cells does not exactly follow the factor of four rule between the extra-coarse, coarse, and fine grids is due to the unstructured region of the grid used. As mentioned in section 5.2.3, the unstructured grid was used to help reduce the number of cells, and thus speed up convergence time, in a section of the grid where gradients are small. Doubling the number of points along an edge of the unstructured domain does not necessarily double the number of cells. Therefore, the resultant meshes contain close to, but not quite, a factor of four times the number of cells as the next smaller one.

To prove that the mesh chosen for the numerical studies produces results that are grid independent, the same case was run on each mesh and then characteristic parameters were plotted to shown that the solutions align. For this study two parameters were chosen to show convergence: the temperature and the methane mass fraction. These were chosen since there is experimental data from the Sandia experi-

ments measuring these two quantities as a function of radius at three axial positions. The results of the grid convergence study are shown in figures 5.7 and 5.8.

As can be seen in the grid convergence results the solutions for each of the grids produces very similar results. In fact, even the temperature and methane mass fraction profiles from the extra-coarse grid sit very close to the other curves. However, due to the lack of resolution that this grid produces, as made apparent by the chunkiness of the solution results, and some slight differences in the temperature profiles at larger radial locations, the extra-coarse grid proves to be insufficient to produce grid independent results.

The results from the coarse, medium, and fine grids sit essentially right on top of each other. This shows that for the coarse grid and higher, the solution is grid independent. Adding additional grid points will only provide higher resolution at the cost of increased run time. Since the coarse grid gives the same solution profiles as the finer grids while using a fraction of the number of cells, and therefore runs in a fraction of the time, the coarse grid was chosen as the grid to run all simulations on.

The medium grid was created as a midway point between the coarse and fine grids in an attempt to get finer resolution of the solution profiles without the additional time penalty of running the fine mesh. However, in the cases it was used for, the increased resolution was not noticeable. It has been added here solely to add further evidence that the solution of the coarse grid, the one chosen to run all of the simulations on, is truly grid independent.

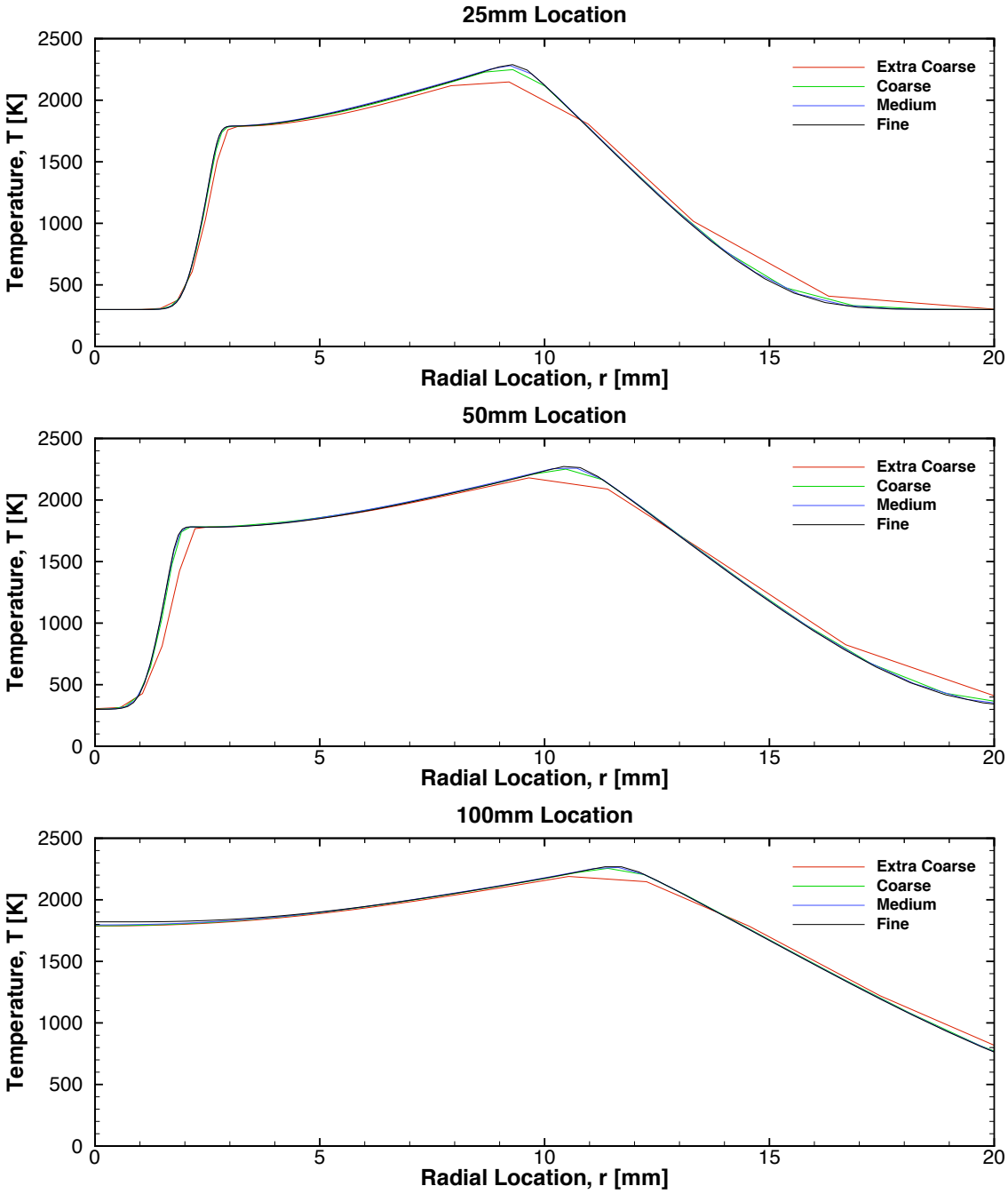

Figure 5.7: Temperature profiles showing grid convergence study results. Solutions computed using Westbrook and Dryer two-step mechanism [1]. Profiles shown, from top to bottom, at 25 mm, 50 mm, and 100 mm downstream of fuel tube.

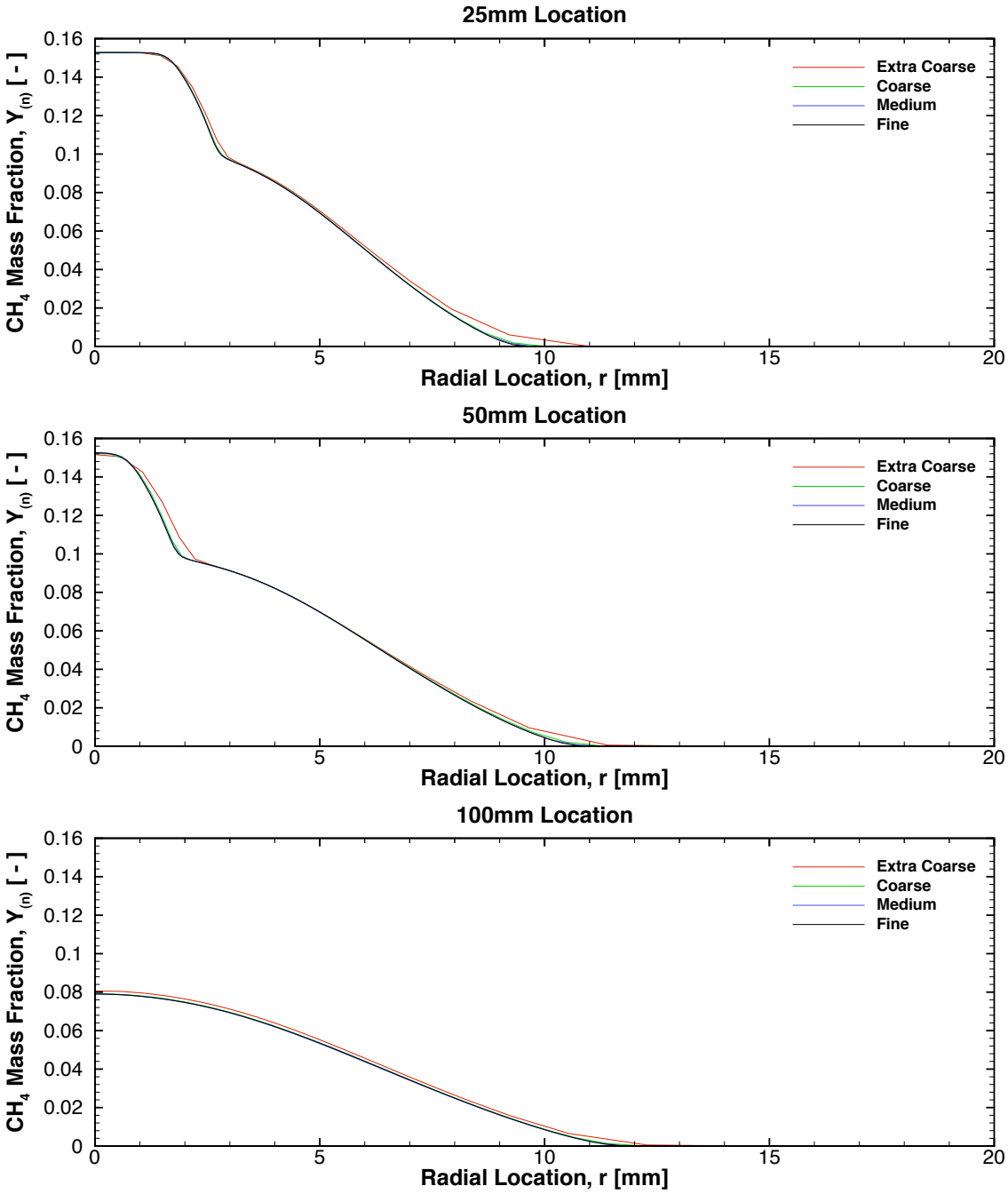

Figure 5.8: Methane mass fraction profiles showing grid convergence study results. Solutions computed using Westbrook and Dryer two-step mechanism [1]. Profiles shown, from top to bottom, at 25 mm, 50 mm, and 100 mm downstream of fuel tube.

5.3 Numerical Options Used

To obtain a converged solution with a reacting flow can be a difficult undertaking. Typically the large amount of heat released during combustion processes causes a strong coupling between the Navier-Stokes equations and the species transport equations. The excess heat drives rapid changes in density which in turn causes the flow to accelerate. A two-step solution process can greatly aid these coupling issues. The first step is to run the simulation without reactions to produce a cold-flow solution. This cold-flow solution can be thought of as a solution initialization process to provide a more realistic starting point for the combustion process. The cold-flow solution can be obtained by either manually disabling reactions in the solver or, if the solution requires a spark to start the combustion process, withholding the spark until the flow is well established. Once the cold-flow solution is obtained the reactions can be started and the simulation converged to its final result.

For the axisymmetric flame A simulation there was no natural source of energy to ignite the flame. Therefore, the cold-flow solution was obtained by simply letting the simulation run until the flow of fuel from the tube was steady. A numerical spark was then applied to ignite the combustion by patching a small area at the tube exit with a temperature of 2000 K. To apply the spark, the region to patch is defined through the **Region Adaptation** dialogue box by choosing **Select Points with Mouse** and highlighting a small square area just downstream of the tube. In the **Solution Initialization** menu **Patch** is selected and the temperature of the region can be increased to a specified value.

Another issue which can affect solution convergence of reacting flows is when there is a mismatch in the time scales between the reactions and the bulk flow. For laminar low speed flows, as are being studied here, the reaction time scales are

considerably faster than the time scales of the convection and diffusion terms. When this is the case the set differential equations are called stiff and the solution becomes numerically difficult to calculate. Stiff equations typically necessitate small time steps in order to avoid divergence.

For some simple reaction mechanisms the solver may be able to produce a solution without any additional numerical techniques to help convergence. This was the case for the work done by Chambers [4] and Jones [3]. Unfortunately, the solutions produced contained an instability which required a temperature limiter to keep from diverging. Additionally, when this simple solution technique was applied to larger mechanisms, such as the GRI 3.0 mechanism [2], the solution would diverge almost immediately.

To aid with convergence issues in reacting systems, FLUENT utilizes a Stiff Chemistry Solver option which can be used with the pressure-based solver [23, pp. 912-3]. For steady simulations the Stiff Chemistry Solver allows the flow solver to take larger time steps by approximating the species reaction rates as

$$R^*_{(n)} = \frac{1}{\tau} \int_0^\tau R_{(n)} dt, \quad n = 1, \ldots, N_s. \tag{5.1}$$

Here $R^*_{(n)}$ is the approximate reaction rate of species n and τ is an appropriate time step. If τ is taken to be zero, the reaction rate becomes exact and as τ tends to infinity the reaction rate goes to zero. There is a trade-off here in that as τ is increased, the solution is easier to calculate and tends to diverge less, however, the reaction rate is decreased until the flow is essentially non-reactive. The default value for τ is one-tenth of the minimum convective or diffusive time scale in the cell.

Numerically calculating the integral from (5.1) can be quite costly. Luckily, since the chemical mechanisms are deterministic, once an integration has been performed

it can be stored in a table and used in the future to speed up solution times. This process is called in-situ adaptive tabulation (ISAT) [25, pp. 298-91]. FLUENT claims a speed up of two to three orders of magnitude is possible when using ISAT over direct integration.

ISAT tables are termed a solution acceleration technique since they aid in decreasing the time required to perform successive iteration of the solution. On the first iteration of the solution all integrations are directly calculated and stored in the ISAT tables. In the following iteration, initial values within a specified tolerance tabulated data can be interpolated from the ISAT tables. As the solution proceeds and the ISAT tables fill up the solution shifts from calculating the integrals to finding the solution in a table. As long as a table look-up is faster than computing the integral time is being saved.

The use of the Stiff Chemistry Solver allowed for all desired mechanisms to be run to convergence. Not only did the Stiff Chemistry Solver allow for the detailed GRI 3.0 mechanism to be simulated, but it aided in solving all mechanisms and producing more stable results. As mentioned earlier, solutions computed without using the Stiff Chemistry Solver contained a numerical instability which required the use of a temperature limiter to keep the solution from diverging. The effects of the temperature limiter on the laminar methane-air flame simulation without the Stiff Chemistry Solver was studied by Chambers [4, pp. 112-3]. As the temperature limiter was increased the maximum temperature seen in the domain increased to match the limiter. After a temperature limiter of approximately 2900 K the flame would enter the fuel tube, an unphysical result, and the solution would diverge. The use of the Stiff Chemistry Solver completely eliminated the need for the temperature limiter.

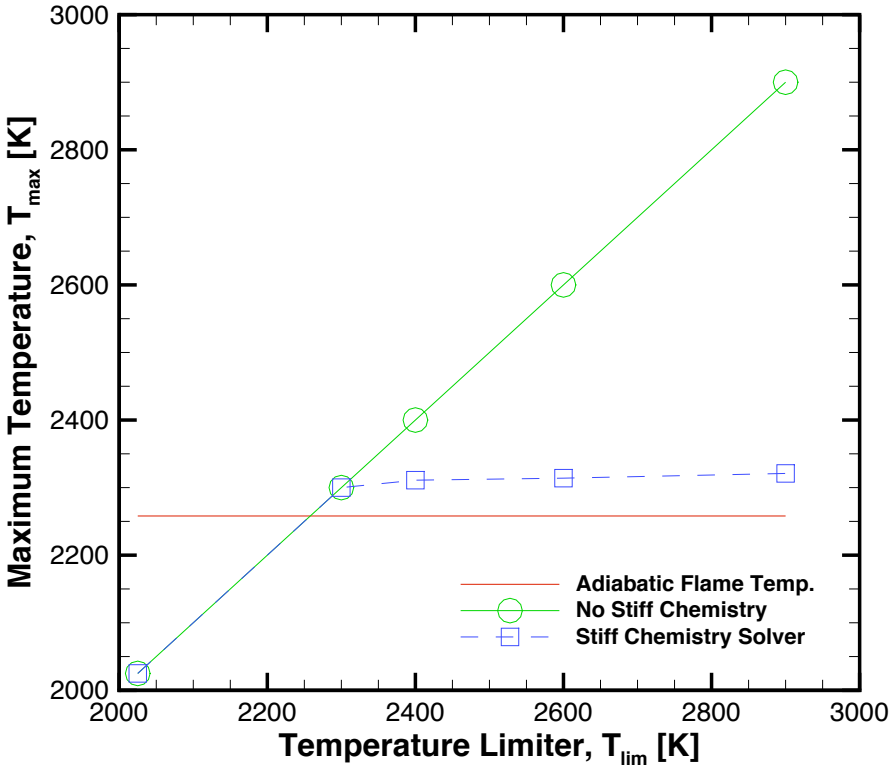

Figure 5.9: Maximum computed temperature in FLUENT axisymmetric 2D simulation as a function of temperature limiter. All simulations performed using the Westbrook and Dryer two-step mechanism [1]. The Stiff Chemistry Solver does not require a limiter and produces results slightly above the predicted adiabatic flame temperature.

As can be seen in Fig. 5.9, without the Stiff Chemisty Solver the maximum temperature rises linearly with the temperature limiter until the solution diverges. When the Stiff Chemistry solver is used, as the temperature limiter is increased the maximum temperature reaches a value that it does not rise beyond. The adiabatic flame temperature is plotted for comparison. The adiabatic flame temperature was predicted to be 2258 K at standard atmospheric initial conditions for the Westbrook and Dryer two-step mechanism which contains six species [1]. Good agreement is found between the maximum temperature when using the Stiff Chemistry Solver and

the adiabatic flame temperature, with the value being over predicted by 2.5%.

An addition point should be made regarding the results of the temperature limiter tests from Fig. 5.9. In the cases where the Stiff Chemistry Solver was not used, the only way to get a converged solution was to set the temperature limiter to a relatively low value, approximately 2100 K. Only when the flame had stabilized could the temperature limiter be increased to its final value without the solution diverging. However, when using the Stiff Chemistry Solver the temperature limiter was set to its final value when the solution was initialized and never changed. This shows how much more robust and user-friendly the Stiff Chemistry Solver is. The solution can be just be initialized and let run without needing to adjust the limiter for final convergence.

Despite not needing the temperature limiter for the solution to converge when using the Stiff Chemistry Solver, it is still beneficial to set the limiter to a reasonable value to aid in solution time. As the solution converges the temperature may temporarily increase to a value higher than the adiabatic flame temperature. If the temperature limiter is set too high the ISAT tables will fill up more and require additional time to interpolate values from. Simulations performed using the Stiff Chemistry Solver with the temperature limiter set to 2900 K took noticeably longer to converge than when it was set to 2400 K even though they were both given the same number of iterations and converged to the same value.

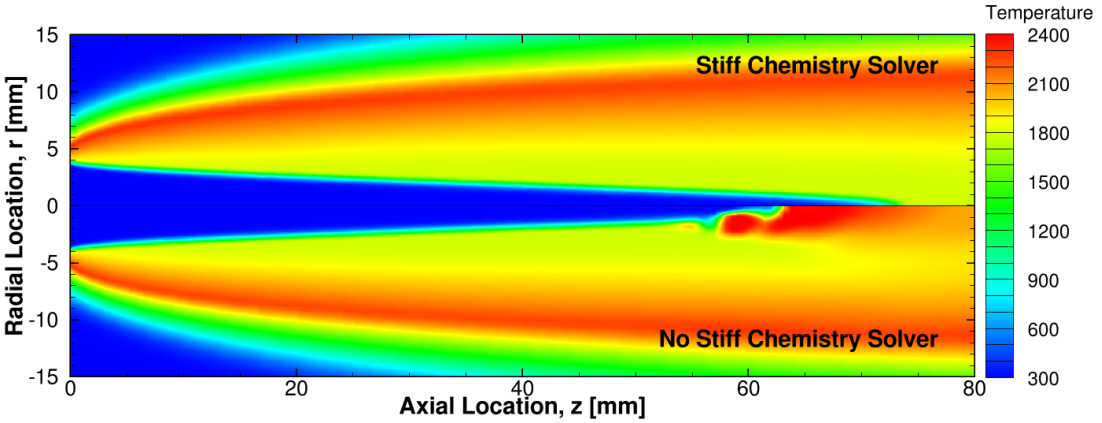

Figure 5.10: Results from simulations with and without the Stiff Chemistry Solver presented side by side for comparison. Both simulations were calculated using the Westbrook and Dryer two-step mechanism [1] with the temperature limiter set to 2400 K, slightly above the adiabatic flame temperature.

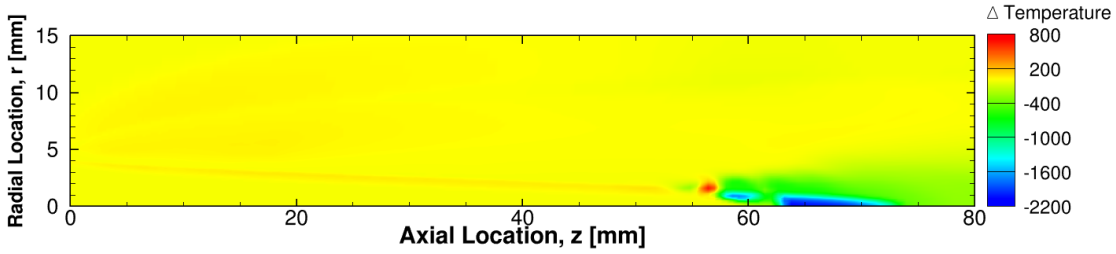

Figure 5.11: Difference in temperature profiles ($T_{SCS} - T_{noSCS}$) shown between simulations with and without the Stiff Chemistry Solver using the Westbrook and Dryer two-step mechanism [1]. Instabilities in the numerics when not using the Stiff Chemistry Solver cause unphysical waves to form near the center line producing temperature differences of nearly 2100 K.

Figures 5.10 and 5.11 demonstrate the numerical instability which causes solutions calculated without the Stiff Chemistry Solver to require a temperature limiter. In Fig. 5.10 a side by side comparison of the results from solutions calculated with and without the Stiff Chemistry Solver are shown, while in Fig. 5.11 the difference in temperature profiles is shown. All other solution options are the same between the models. The temperature limiter is set to 2400 K so as to be close to, but slightly

above, the adiabatic flame temperature so that it does not influence the results of the case using Stiff Chemistry Solver. In the top pane of Fig. 5.10 the results using the Stiff Chemistry solver are shown, while in the bottom pane the solution was calculated without using it.

Up to approximately 50 mm downstream of the fuel tube the solutions show close agreement. This best demonstrated in Fig. 5.11 where the difference in temperature up to this point is barely distinguishable on the contour plot. Beyond 50 mm a wave occurs in the solution calculated without the Stiff Chemistry Solver and the temperature difference, for this case when the temperature limiter is set to 2400 K, spikes to nearly 2100 K. This wave oscillates axially in time and locally increases the temperature due to an over-prediction of the reaction rates. The temperature is increased so much in this area that it will always reach the value of the temperature limiter as seen in Fig. 5.9. If the temperature limiter is set too high the heat released from this local instability will accelerate reaction rates nearby and cause the flame to spread throughout the domain. This is a purely numerical phenomenon which does not reflect any real world physics.

Many strategies were attempted prior to implementing the Stiff Chemistry Solver to removed this instability. These included under-relaxation, running the solution first-order, and using a Relax to Equilibrium option. Relax to Equilibrium solves the problem by assuming that the flow is at chemical equilibrium at each grid point. While this was able to remove the instability for a few simple cases, it was disregarded as a viable option since it did not work reliably for larger detailed mechanisms and would overly thin the flame by ignoring the chemical kinetics.

Jones [3, pp. 73-81] attempted to avoid the negative affects of the numerical instability by modifying the fuel inlet boundary condition composition. In an attempt to get more complicated mechanisms than the Westbrook and Dryer two-step mech-

anism [1] to run, the oxygen in the fuel was replaced by non-reacting nitrogen. When simulating a three-step mechanism based off that of the Peters and Williams [10], Jones had an issue in which the numerical instability caused the flame to propagate up the fuel tube even though the fuel mixture was too rich to combust naturally. The use of the Stiff Chemistry Solver completely resolved this issue. A similar three-step mechanism to the one used by Jones was successfully run without the need to modify any boundary conditions. The creation of this three-step mechanism is discussed in chapter 6 and simulation results using it are presented in chapter 7. The only reliable method to remove the instability was to use the Stiff Chemistry Solver. As can be seen in Fig. 5.10, the result is a smooth, steady flame.

Using the Stiff Chemistry Solver is beneficial for four main reasons. First, it allows detailed mechanisms to be run where they could not before. Second, regardless of the mechanism used, the temperature limiter does not need to be implemented to constrain the solution. The adiabatic flame temperature, which is well predicted, acts as a physical limiter which keeps the solution bounded. It is typically preferable to allow the solution to be limited by physical phenomena than to impose arbitrary limiters. Third, no modifications of boundary conditions are necessary for solutions to behave properly. Boundary conditions can be used directly from the experimental setup and good results can be expected. Fourth, and finally, the downstream behavior of the flame is much better modeled. If the simulation is to be used to predict pollutants or other combustion products, the numerical instability will incorrectly create higher concentrations along the central axis. Therefore, using the Stiff Chemistry Solver can greatly aid in the design of cleaner and more efficient combustion systems.

5.4 Summary

A reliable computational flow domain and solver was needed in order to compare various methane-air combustion mechanisms for violations of the DEI. The computational domain chosen was a two-dimensional axisymmetric representation of the Sandia flame A, for which comparative data are available. A great amount of effort was put into the design of the computational domain so that it mimicked the actual experimental conditions as closely as possible. Boundary conditions were chosen to ensure that the simulation was isobaric and that flow structures due to boundary layers were fully developed. A grid convergence study was carried out to ensure that the results were grid independent while using the least amount of computational resources possible. Finally, a Stiff Chemistry Solver was used with the flow solver to help solution convergence. Without using the Stiff Chemistry Solver, the solution contained a numerical instability which necessitated a temperature limiter to keep the solution bounded. The Stiff Chemistry Solver eliminated this instability and need for the temperature limiter. The maximum temperature of simulations calculated using the Stiff Chemistry Solver was found to show good agreement with the predicted adiabatic flame temperature.

6. FITTING ARRHENIUS PARAMETERS TO REDUCED MECHANISMS

Currently the DEI is never used in CFD reacting flow solvers as a solution limiter to ensure that the second law of thermodynamics is being obeyed at all points within the computations domain. Therefore, to ensure physical results, the reaction mechanisms used must automatically satisfy the DEI without the need for a limiter. In this chapter a method is developed to find Arrhenius parameters for reduced reaction mechanisms which satisfy the DEI. In section 6.1, methods for determining Arrhenius parameters for reduced mechanisms are discussed, including an overview of the method developed in this thesis. The derivation and solution method for least squares optimization are discussed in section 6.2. Least squares is then applied to reduced mechanisms in section 6.3. The process is applied to an example mechanism to illustrate the process. Section 6.4 outlines the algorithm created in MATLAB to solve for the Arrhenius parameters. Special attention will be given to the built-in MATLAB subroutines utilized in this code. Finally, in section 6.5, the resulting mechanisms created using this method are presented along with issues encountered, methods for avoiding these issues, and possible future improvements.

6.1 Introduction to Parameters Fitting

One of the first attempts to fit Arrhenius parameters to experimental data for a reduced mechanism was undertaken by Westbrook and Dryer [1]. Westbrook and Dryer found Arrhenius parameters for their two-step mechanism through a trail and error process. Their method involved repeatedly running a one-dimensional combustion code for a variety of inlet compositions. At each iteration the Arrhenius parameters were allowed to vary. This was continued until a set of parameters was

found that showed close agreement with published values for the flame speed, the maximum flame speed, and the high and low flammability limits.

The Westbrook and Dryer method attempted to find a mechanism that was applicable to a broad range of equivalence ratios by imposing relatively loose constraints on the solution. An alternate route is to search for a set of parameters which work for only a very limited range of operating conditions but are optimized for those conditions. Jones, following the latter mindset, used temperature and concentration data from a detailed flame simulation at a fixed inlet composition to determine the Arrhenius parameters [3, 19]. The resulting mechanism produces results with fewer DEI violations than the Westbrook and Dryer mechanism [3].

The method presented herein follows the tactic used by Jones, in that it attempts to create a mechanism optimized for a single operating condition. The basic idea of this method is to find the Arrhenius rate parameters for a reduced mechanism through a series of curve fittings. Starting with a data set of species production rates as a function of temperature and composition, the progresses of each reaction in the reduced mechanism are estimated using a linear least squares method. The progresses of reaction are then used in a series of non-linear least squares curve fittings to find the Arrhenius rate parameters for each reaction. The original data set can either come from experimentation or from a simulation using a detailed mechanism. For this work, a one-dimensional simulation was carried out using the GRI 3.0 mechanism. The method presented herein makes heavy use of least squares curve fitting techniques so the next section presents an in-depth derivation. Following that, in section 6.3, the method developed for this thesis is outlined.

6.2 Least Squares Curve Fitting Derivation

Least squares curve fitting is a process where a function can be optimally fit to a data set by minimizing the sum of the squares of the difference between the data set and the value of the function at a known point. Least squares is a very common process which can be used to match both linear and non-linear functions to a data set. A detailed derivation is included here to aid in understanding the implications of each term when they are applied to a reduced mechanism in section 6.3. The following derivation has been adapted from the book by Dennis [37, pp. 218-36] and will be presented in component form as well as vector form where applicable.

The first phase of the least squares technique is to acquire a data set composed of independent and dependent variables. The independent variable will be called x and the dependent variable will be called y. As this is written there is only one independent variable, however this process can easily be extended to multivariate problems by taking x to be a vector of the independent variables.

Least squares seeks to minimize the sum of the squares of the difference between the dependent variable data and the function to be fit. This difference is termed the residual and is defined as

$$r_i = y_i - f\left(x_i, \vec{\eta}\right), \quad i = 1, \ldots, N_p. \tag{6.1}$$

Here r_i is the residual at point i of a data set containing N_p points. The associated independent and dependent coordinates are x_i and y_i respectively. The equation to fit the data to is f which is a function of the independent variable and the solution vector $\vec{\eta}$; $\vec{\eta} \in \mathbb{R}^{N_\eta}$ where N_η is the number of parameters. Written in vector form the residual becomes

$$\vec{r} = \vec{y} - \vec{f}\left(\vec{x}, \vec{\eta}\right). \tag{6.2}$$

Summing the square of the residual over all the data points yields

$$S = \sum_{i=1}^{N_p} r_i^2, \tag{6.3}$$

where S is the sum. To minimize the residuals, S is differentiated with respect to the components of $\vec{\eta}$ and the result is set equal to zero

$$\frac{\partial S}{\partial \eta_j} = 2 \sum_{i=1}^{N_p} r_i \frac{\partial r_i}{\partial \eta_j} = 0, \quad j = 1, \ldots, N_\eta. \tag{6.4}$$

Here N_η is the number of components of the solution vector. Notice that (6.4) represents a set of N_η independent equations, one for each component η_j. The partial derivatives of the residual with respect to the components of the solution vector turn out to be important and are typically collected into a matrix called the Jacobian.

$$J_{ij} = -\frac{\partial r_i}{\partial \eta_j} = \frac{\partial f\left(x_i, \vec{\eta}\right)}{\partial \eta_j}, \qquad \begin{matrix} i = 1, \ldots, N_p \\ j = 1, \ldots, N_\eta \end{matrix} \tag{6.5}$$

This allows (6.4) to be simplified and rewritten as

$$\sum_{i=1}^{N_p} J_{ij} r_i = 0, \quad j = 1, \ldots, N_\eta, \tag{6.6}$$

or in matrix form as

$$\overline{\overline{J}}^T \vec{r} = 0. \tag{6.7}$$

When f is a linear function of the parameters η_j, the Jacobian is only a function of the independent variable. As proven in appendix A.2, when f is linear in the parameters

$$f\left(x_i, \vec{\eta}\right) = \sum_{j=1}^{N_\eta} J_{ij} \eta_j, \quad i = 1, \ldots, N_p, \tag{6.8}$$

or in matrix form

$$\vec{f}\,(\vec{x},\vec{\eta}) = \overline{\overline{J}}\vec{\eta}. \tag{6.9}$$

Substituting (6.9) into (6.2) and then into (6.7) yields

$$\overline{\overline{J}}^T\overline{\overline{J}}\vec{\eta} = \overline{\overline{J}}^T\vec{y}. \tag{6.10}$$

Consequently, the parameters η_j that minimize S of (6.3) are calculated by solving (6.10). Note that (6.9) and (6.10) are only valid for linear problems.

When f is a non-linear function of the parameters η_j, a solution can still be found, however it has to be approached iteratively. An initial guess for the parameters must be given and each successive iteration of the solution increments the parameters towards their final values in the following manner

$$\vec{\eta}^{k+1} = \vec{\eta}^{k} + \Delta\vec{\eta}^{k}. \tag{6.11}$$

Here k is an index to increment the iterations of the solution process. The bounds of k are determined by the convergence criterion imposed. The term $\Delta\vec{\eta}$ is known as the shift vector or the direction vector.

Expressing the value of f as a first-order Taylor series approximation with respect to the parameters gives

$$\begin{aligned} f\left(x_i,\vec{\eta}^{k+1}\right) &\approx f\left(x_i,\vec{\eta}^{k}\right) + \sum_{j=1}^{N_\eta}\frac{\partial f\left(x_i,\vec{\eta}^{k}\right)}{\partial\eta_j}\left(\eta_j^{k+1}-\eta_j^{k}\right) \\ &\approx f\left(x_i,\vec{\eta}^{k}\right) + \sum_{j=1}^{N_\eta} J_{ij}^{k}\Delta\eta_j^{k}, \quad i=1,\ldots,N_p. \end{aligned} \tag{6.12}$$

The residual at the new time step can be written in terms of the residual at the

current time step and the Jacobian by substituting (6.12) into (6.1).

$$\begin{aligned} r_i^{k+1} &= y_i - f\left(x_i, \vec{\eta}^{k+1}\right) \\ &\approx y_i - \left[f\left(x_i, \vec{\eta}^{k}\right) + \sum_{j=1}^{N_\eta} J_{ij}^k \Delta\eta_j^k \right] \\ &\approx r_i^k - \sum_{j=1}^{N_\eta} J_{ij}^k \Delta\eta_j^k, \quad i = 1, \ldots, N_p \end{aligned} \tag{6.13}$$

Inserting this result into (6.6), rearranging, and expressing in matrix form results in the final non-linear least squares method equation

$$\bar{\bar{J}}^{T,k} \bar{\bar{J}}^{k} \Delta\vec{\eta}^{k} = \bar{\bar{J}}^{T,k} \vec{r}^{k}. \tag{6.14}$$

Notice that this equation shares the same structure as the linear least squares method in (6.10).

The non-linear least squares method given in (6.14) is commonly known as the Gauss-Newton method [37, 38]. The solution technique typically proceeds as follows. First, the x and y data set is collected from an experimental or analytical source. Next, the form of the function f is determined from a physical model or by making assumptions. Partial derivatives of f with respect to the parameters are then taken. For the first iteration a guess of the parameters must be made. This step turns out to be quite important since this method tends to converge to local minima. If the initial guess is too far from the global minimum then incorrect or sub-optimal results will be found. This is discussed in greater detail in section 6.5. Once the initial guess is made the residual vector $\vec{r}^k$ and the Jacobian matrix are calculated for the current iteration. The shift vector $\Delta\vec{\eta}^k$ is found by solving (6.14). Finally, the parameters are updated for the next iteration based on (6.11) and the process starts over. This

is repeated until some desired convergence criterion is met.

In some instances, if the full value of the direction vector is used the solution will never converge to a final answer. It may bounce back and forth around the solution without ever reaching a stable value or it may diverge entirely. To aid with convergence a relaxation factor is commonly used. This modifies (6.11) as follows

$$\vec{\eta}^{k+1} = \vec{\eta}^{k} + \gamma \Delta \vec{\eta}^{k}. \tag{6.15}$$

Here γ is the relaxation factor and is in the range of $(0, 1]$. It is typically up to the analyst to find a good value for the under relaxation factor. Too large and the solution may diverge. Too small and the solution may take a very long time to converge.

6.3 Non-Linear Least Squares Method for Fitting Arrhenius Parameters

The technique developed herein to find Arrhenius parameters for a reduced mechanism uses the least squares optimization processes previously introduced. Linear least squares is used to approximate the progresses of reaction for the reduced mechanism from experimental species production rate data. Reversible reactions are split into two unidirectional reactions and treated as independent reactions. The non-linear least squares method is then used for each reaction to find the best fit of Arrhenius parameters. The entire process is detailed in this section.

To begin this development some useful rate equation terms will be drawn from the derivations in chapter 3 and modified for unidirectional reactions. The Arrhenius rate for reaction r of any reaction mechanism is

$$k_{(r)} = A_{(r)} T^{\beta_{(r)}} \exp\left[\frac{-E_{a,(r)}}{\hat{R}T}\right], \quad r = 1, \ldots, N_r. \tag{3.5}$$

The progress of reaction r, only considering the forward reaction and allowing for non-stoichiometric concentration exponents, is given by

$$\omega_{(r)} = k_{(r)} \prod_{n=1}^{N_s} c_{(n)}^{q'_{(n)(r)}}, \quad r = 1, \ldots, N_r. \tag{6.16}$$

Lastly, the volumetric molar net rate of production of species n is

$$R_{(n)} = \sum_{r=1}^{N_r} \left(\nu''_{(n)(r)} - \nu'_{(n)(r)} \right) \omega_{(r)}, \quad n = 1, \ldots, N_s. \tag{3.4}$$

The first step in the rate fitting method is to obtain a data set of the dependent and independent variables. A one-dimensional flame simulation is run using a detailed mechanism to obtain values for the temperature, species concentration, and time rate of change of concentration through a flame. The independent variable in this process is the temperature, T. The dependent variable when solving for the Arrhenius parameters is the progress of each reaction r, $\omega_{(r)}$. The data set from the detailed mechanism simulation does not contain usable progress of reaction data since the detailed mechanism does not use the same reaction steps as the reduced mechanism. Therefore, the progress of each reaction must be approximated from the species production rates through (3.4).

In most cases (3.4) is an over-constrained linear system which can be represented in matrix form as

$$\vec{R} = \overline{\overline{\nu}} \vec{\omega}. \tag{6.17}$$

The size of $\overline{\overline{\nu}}$ is $N_s \times N_r$, where N_s and N_r are the number of species and reactions in the reduced mechanism, respectively. Since $\overline{\overline{\nu}}$ is generally non-square, it must be solved as an optimization problem as in (6.10). Rewriting in the form of (6.10) and

solving for the reaction progress rates results in

$$\vec{\omega} = \left(\overline{\overline{\nu}}^T \overline{\overline{\nu}}\right)^{-1} \overline{\overline{\nu}}^T \vec{R}. \tag{6.18}$$

The solution to (6.18) gives the progress of each reaction r at every temperature T_i of the one-dimensional flow simulation.

The next step is to perform a set of non-linear least squares curve fits to find the Arrhenius parameters for each reaction. The temperature is the dependent variable and the progresses of each reaction just solved for are the dependent variable data. This leads to the residual from (6.1) being calculated as

$$r_{(r),i} = \omega_{(r),i} - f_{(r),i}\left(T_i, \vec{\eta}_{(r)}\right), \quad i = 1, \ldots, N_p. \tag{6.19}$$

Since each progress of reaction is independent, a separate non-linear least squares curve fit will have to be performed for each reaction. Each curve fit will produce the Arrhenius parameters for one reaction.

The form of the function in the residual is found by substituting the Arrhenius rate equation (3.5) into the progress of reaction equation (6.16)

$$f_{(r),i}\left(T_i, \vec{\eta}_{(r)}\right) = A_{(r)} T_i^{\beta_{(r)}} \exp\left[\frac{-E_{a,(r)}}{\hat{R} T_i}\right] \prod_{n=1}^{N_s} c_{(n),i}^{q'_{(n)(r)}}, \qquad \begin{matrix} i = 1, \ldots, N_p \\ r = 1, \ldots, N_r \end{matrix}. \tag{6.20}$$

To conform to the law of mass action the exponent $q'_{(n)(r)}$ is taken to be $\nu'_{(n)(r)}$, the reactant stoichiometric coefficient of species n in reaction r. The solution vector is composed of the Arrhenius parameters for reaction r

$$\vec{\eta}_{(r)} = \begin{bmatrix} A_{(r)} & \beta_{(r)} & E_{a,(r)} \end{bmatrix}^T, \quad r = 1, \ldots, N_r. \tag{6.21}$$

It should be noted that the process used here can be easily extended to cases where the species exponents do not follow the law of mass action. In these cases, the $q'_{(n)(r)}$ are left as undetermined parameters and are added to the solution vector. The length of the solution vector then becomes $3 + N_s$ for each reaction. This option is not explored herein.

The Jacobian for reaction r is then calculated by taking the partial derivatives of (6.20) with respect to the Arrhenius parameters at each point i. Recall that $i = 1, \ldots, N_p$ where N_p is the total number of data points.

$$\overline{\overline{J}}_{(r)} = \begin{bmatrix} \frac{\partial f_{(r),1}}{\partial A_{(r)}} & \frac{\partial f_{(r),1}}{\partial \beta_{(r)}} & \frac{\partial f_{(r),1}}{\partial E_{a,(r)}} \\ \vdots & \vdots & \vdots \\ \frac{\partial f_{(r),N_p}}{\partial A_{(r)}} & \frac{\partial f_{(r),N_p}}{\partial \beta_{(r)}} & \frac{\partial f_{(r),N_p}}{\partial E_{a,(r)}} \end{bmatrix}, \quad r = 1, \ldots, N_r \tag{6.22}$$

The partial derivatives that appear in the Jacobian can either be numerically approximated or pre-calculated. Manually taking the derivatives of (6.20) the following simple relations can be found

$$\frac{\partial f_{(r)}}{\partial A_{(r)}} = \frac{f_{(r)}}{A_{(r)}}, \quad r = 1, \ldots, N_r, \tag{6.23a}$$

$$\frac{\partial f_{(r)}}{\partial \beta_{(r)}} = f_{(r)} \ln(T), \quad r = 1, \ldots, N_r, \tag{6.23b}$$

$$\frac{\partial f_{(r)}}{\partial E_{a,(r)}} = \frac{-f_{(r)}}{\hat{R}T}, \quad r = 1, \ldots, N_r. \tag{6.23c}$$

The last step that must be taken before a solution can be calculated is to provide an initial guess of the parameters for the iterative non-linear least squares curve fitting process. Since each reaction is independent of the others, there are only three parameters for each curve fit that need to be found. This allows for the initial guess to be found by simply testing an assortment of combinations of the parameters to

see which has the lowest sum of squares of the residuals.

Finally, the solution can be calculated by solving (6.14) for the shift vector as in

$$\Delta\vec{\eta}^{k} = \left(\overline{\overline{J}}^{T,k}\overline{\overline{J}}^{k}\right)^{-1}\overline{\overline{J}}^{T,k}\vec{r}^{k}.$$

The solution vector is then updated using (6.15) and the process is repeated until some measure of convergence is reached.

This method is rather involved, so a simple outline is provided to aid in understanding and implementation.

1. A detailed mechanism simulation is run to generate a data set containing temperature, species concentration, and species production rate through a flame front.

2. Progresses of reaction are approximated at each data point from the species production rates using linear least squares as in (6.18).

3. The Arrhenius parameters are found for each reaction by performing a nonlinear least squares curve fit of the approximated progresses of reaction. This operation is composed of the following steps:

   (a) The form of function $f_{(r)}$ is found by applying (6.20) to each reaction r.

   (b) An initial guess of the solution vector for each reaction r is found by testing an assortment of combinations of the Arrhenius parameters.

   (c) The residual and Jacobian are calculated for each reaction r using (6.19) and (6.22).

   (d) The solution vector is updated for each reaction r solving (6.14).

(e) If a convergence criterion is met the solution has been found. If not, the guess of the solution vector is updated using (6.15) and the residual and Jacobian are recomputed for the new guess.

6.3.1 Example of Fitting Process

It is typically useful to see an application of any mathematical process to better understand how it works. Therefore, the method developed in this thesis will be applied to the Westbrook and Dryer model here for illustrative purposes. The Westbrook and Dryer mechanism is composed of two reactions with the second being reversible. It is useful to consider reversible reactions as two separate reactions so that the Arrhenius parameters for both the forward and reverse directions can be found independently. The resulting mechanism is shown in Table 6.1 below.

Table 6.1: Two-Step Mechanism Split into All Forward Reactions

Reaction	Equation
1	$CH_4 + 1.5O_2 \rightarrow CO + 2H_2O$
2	$CO + 0.5O_2 \rightarrow CO_2$
3	$CO_2 \rightarrow CO + 0.5O_2$

There are three reactions and five species in this model. A sixth species, N_2, is also present for combustion of methane in air. However, it does not contribute to the reactions and is not included in the model. The mole fraction of N_2 can be found by noting that the sum of all mole fractions is one, $\sum_{n=1}^{N_s} X_{(n)} = 1$. The first step is to construct the over-constrained system relating the known species production rates

and the desired progress of reactions at each data point i as in (6.17)

$$\begin{bmatrix} R_{(\mathrm{CH_4}),i} \\ R_{(\mathrm{H_2O}),i} \\ R_{(\mathrm{O_2}),i} \\ R_{(\mathrm{CO}),i} \\ R_{(\mathrm{CO_2}),i} \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 2 & 0 & 0 \\ -1.5 & -0.5 & 0.5 \\ 1 & -1 & 1 \\ 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} \omega_{(1),i} \\ \omega_{(2),i} \\ \omega_{(3),i} \end{bmatrix}, \quad i = 1, \ldots, N_p. \tag{6.24}$$

Each entry in the matrix of parameters is the difference between the product and reactant stoichiometric coefficients for species n in reaction r, $\nu''_{(n)(r)} - \nu'_{(n)(r)}$. This can be solved for the vector of reaction progress rates at each data point by using (6.18).

The calculated progress of reaction at each data point can be collected into individual vectors as in,

$$\begin{aligned} \vec{\omega}_{(1)} &= \begin{bmatrix} \omega_{(1),1} & \cdots & \omega_{(1),N_p} \end{bmatrix}^T, \\ \vec{\omega}_{(2)} &= \begin{bmatrix} \omega_{(2),1} & \cdots & \omega_{(2),N_p} \end{bmatrix}^T, \\ \vec{\omega}_{(3)} &= \begin{bmatrix} \omega_{(3),1} & \cdots & \omega_{(3),N_p} \end{bmatrix}^T. \end{aligned} \tag{6.25}$$

The functions that will be used to calculate the residual for each reaction are

$$\begin{aligned} f_{(1),i}\left(T_i, \begin{bmatrix} A_{(1)} & \beta_{(1)} & E_{a,(1)} \end{bmatrix}^T\right) &= A_{(1)} T_i^{\beta_{(1)}} \exp\left[\frac{-E_{a,(1)}}{\hat{R} T_i}\right] c_{(\mathrm{CH_4}),i} c_{(\mathrm{O_2}),i}^{1.5}, \\ f_{(2),i}\left(T_i, \begin{bmatrix} A_{(2)} & \beta_{(2)} & E_{a,(2)} \end{bmatrix}^T\right) &= A_{(2)} T_i^{\beta_{(2)}} \exp\left[\frac{-E_{a,(2)}}{\hat{R} T_i}\right] c_{(\mathrm{CO}),i} c_{(\mathrm{O_2}),i}^{0.5}, \\ f_{(3),i}\left(T_i, \begin{bmatrix} A_{(3)} & \beta_{(3)} & E_{a,(3)} \end{bmatrix}^T\right) &= A_{(3)} T_i^{\beta_{(3)}} \exp\left[\frac{-E_{a,(3)}}{\hat{R} T_i}\right] c_{(\mathrm{CO_2}),i}, \quad i = 1, \ldots, N_p. \end{aligned} \tag{6.26}$$

Recall that the independent variable is the temperature at each data point i, T_i. Also, the exponent $q'_{(n)(r)}$ is taken to be $\nu'_{(n)(r)}$, the reactant stoichiometric coefficient

of species n in reaction r, as dictated by the law of mass action.

At this point a non-linear least squares curve fit is performed for each reaction r. In this case there are three to be carried out. The steps are identical for each so only the first reaction will be shown here for brevity. The first step in the non-linear least squares process is to define the residual vector. For the first reaction each component of the residual is calculated as

$$r_{(1),i} = \omega_{(1),i} - f_{(1),i}\left(T_i, \begin{bmatrix} A_{(1)} & \beta_{(1)} & E_{a,(1)} \end{bmatrix}^T\right), \quad i = 1, \ldots, N_p. \tag{6.27}$$

For the first iteration of the non-linear least squares curve fitting an initial guess of the solution vector must be supplied. A very basic method was used, consisting of simply testing a random assortment of combinations of the parameters. The set which produced the minimum sum of squares of the residuals, as calculated in (6.3), was chosen as the starting point. Once this is determined it is simply a matter of filling in the Jacobian matrix as in (6.22) and (6.23). The shift vector $\Delta\vec{\eta}$ is then calculated by solving (6.14) and the guess of the parameters is updated through (6.15). This is repeated until some convergence criterion is met, typically when the change in the sum of squares of the residuals between two consecutive iterations is below a threshold value. This whole process is then repeated for each reaction in the mechanism.

The entire process developed in this thesis to find Arrhenius parameters for a reduced mechanism has now been explained in full detail. While the example above was specific to the reaction steps of the Westbrook and Dryer mechanism, it is very simple to apply it to any given reduced mechanism. In the next section the algorithm used to perform these calculations is presented.

6.4 MATLAB Non-Linear Curve Fitting Algorithm

The process of finding the Arrhenius parameters for a reduced mechanism described in the proceeding sections was implemented in a MATLAB algorithm and used to find optimized parameters for the reaction steps of both the Westbrook and Dryer two-step mechanism and the Peters and Williams three-step mechanism. In this section the MATLAB algorithm and options used will be explained.

The code written follows the algorithm presented in the previous section. First, concentration and rate of change data for each species are input and a linear least squares process is used to approximate the progress of each reaction from the species production rates. This procedure is performed exactly as presented above. The non-linear least squares step that comes next uses the built in MATLAB function `lsqcurvefit`. While this follows the same basic solution process, there is additional functionality which aids in solution convergence that needs to be discussed.

The `lsqcurvefit` subroutine bundles all of the non-linear least squares curve fitting steps into a single easy to use function. The calculation of the residual, the Jacobian, and even the iterative solution are all taken care of within the subroutine. From the MATLAB documentation [39] the function call uses the following syntax.

```
x = lsqcurvefit(fun,x0,xdata,ydata,lb,ub,options)
```

Here the output is the vector `x` of length N_η which contains the optimized parameters. The input variables `x0`, `lb`, and `ub` are also vectors of length N_η and are the initial guess of the parameters, lower bounds, and upper bounds for the parameters respectively. The independent and dependent variable data are `xdata` and `ydata` which are vectors of length N_p. If non-default solver options are desired the `options` input can be used which is set through the `optimoptions` subroutine. The final, and most important, input is the function handle `fun`. The function handle `fun` sets the

form of the equation which the `lsqcurvefit` optimization routine fits to the data. `fun` must be a function of both the parameters `x` and the independent variable data `xdata`. Additional information about the inputs and outputs can be found in the MATLAB documentation [39].

The `lsqcurvefit` optimization algorithms can be applied to reduced chemical mechanisms quite simply. The parameters which will be optimized and output in the `x` vector are the three Arrhenius parameters for the given reaction as in (6.21). Creating the function handle is the most involved operation. The form of the function to be created is that of the progress of reaction r, fully expanded as in (6.20). The function handle for the first reaction of the Westbrook and Dryer mechanism is given below as an example. Compare this to the first equation from (6.26).

```
fun =@(x,xdata)(x(1).*(xdata.^x(2)).*exp(-x(3)./(R.*xdata))...
.*ch4.*(o2.^1.5))
```

Here `x` is a vector with three indices corresponding to the three Arrhenius parameters to solve for. The other input, `xdata`, is a vector containing the temperature at each data point. Everything on the first line of the example function above is the same for any reaction in a mechanism to be curve fit. The species concentration vectors and exponents in the second line, here `ch4` and `o2`, are specific to a given reaction and will need to be modified based on the current reaction being curvefit. The `@` symbol is known as the function handle and it is required syntax. The final point to note is that all multiplication, division, and exponent operations should be prefaced by a dot. This indicates to MATLAB that the operation is to be carried out for each component individually and not as a typical linear algebra operation.

The optimization algorithms which MATLAB employs in the `lsqcurvefit` subroutine are the trust-region-reflective algorithm and the Levenberg-Marquardt algo-

rithm. Both algorithms offer different functionality which can limit their applicability. The trust-region-reflective algorithm does not accept underdetermined problems while the Levenberg-Marquardt algorithm does not allow for bounded solutions [39].

The Levenberg-Marquardt algorithm is a more sophisticated version of the Gauss-Newton method derived in (6.14). It seeks to help solution convergence by turning the shift vector towards the direction of steepest descent. Two forms of the Levenberg-Marquardt method which are available for use

$$\left(\overline{\overline{J}}^{T,k}\overline{\overline{J}}^{k} + \lambda^{k}\overline{\overline{I}}\right)\Delta\vec{\eta}^{k} = \overline{\overline{J}}^{T,k}\vec{r}^{k}, \tag{6.28a}$$

$$\left(\overline{\overline{J}}^{T,k}\overline{\overline{J}}^{k} + \lambda^{k}\mathrm{diag}\left(\overline{\overline{J}}^{T,k}\overline{\overline{J}}^{k}\right)\right)\Delta\vec{\eta}^{k} = \overline{\overline{J}}^{T,k}\vec{r}^{k}. \tag{6.28b}$$

The first form was developed by Levenberg [40] and the second was an improved method developed by Marquardt almost 20 years later [41]. The idea is for the variable parameter λ^k to adjust the magnitude and direction of the shift vector at each solution iteration. When λ^k is set to zero the solution boils down to the Gauss-Newton method from (6.14). As λ^k goes to infinity the solution begins to resemble a gradient descent method and the shift vector spins towards the steepest descent direction, however the magnitude tends to zero. By adjusting the value of λ^k at each iteration it is possible to ensure that the solution continually descends. MATLAB claims to use a shift vector which is a cross between the Gauss-Newton direction and the steepest descent direction [42].

An issue that the first form in (6.28) has is that when λ^k is large there tends to be slower convergence for the term with the smallest gradient. The second form helps avoid this by scaling the gradients such that there is more movement along directions with smaller gradients. This is typically beneficial when there is large curvature in the solution.

The other optimization algorithm that MATLAB uses with the `lsqcurvefit` subroutine is the trust-region-reflective algorithm. This algorithm can be used for many different types of minimization problems so the notation used to describe the method will be rather general. The idea is to take some function $f(\vec{\eta})$ which returns scalar values and find the input vector $\vec{\eta}$ which minimizes its output. In the case of non-linear curve fitting for reduced mechanisms, the sum of squares of the residual (6.3) is the function to minimize and the components of the input vector are the Arrhenius parameters.

The trust-region-reflective algorithm goes about finding the minimum of the function by iteratively taking small steps away from an initial guess of the solution until a minimum is found. This is the same basic procedure the Levenberg-Marquardt method uses. The difference lies in the way these steps are calculated. The trust-region-reflective algorithm calculates the step, labeled here as $\vec{s}$, by locally approximating the function $f(\vec{\eta})$ in a neighborhood, or trust-region, N, with a much simpler function $q(\vec{s})$. The step is then found by minimizing the simpler function within the neighborhood. This can be written mathematically as

$$\min_{\vec{s}} \left\{ q(\vec{s}),\ \vec{s} \in N \right\}. \tag{6.29}$$

Once the step is calculated the function is tested at $f(\vec{\eta}+\vec{s})$. If $f(\vec{\eta}+\vec{s}) < f(\vec{\eta})$ then the step is taken and the process is started over from the beginning. If not, the size of the trust-region N is reduced and a new step is calculated for the smaller region.

The main obstacle for the trust-region-reflective method is how to choose an appropriate approximation function and trust-region. This problem was discussed in the paper by Moré [43]. Typically the function is approximated by a quadratic which is found by taking the first two terms of a Taylor Series about the point $\vec{\eta}$.

The trust-region is usually taken to be a sphere or ellipsoid. With this in hand the step can be found by solving the problem

$$\min\left\{\frac{1}{2}\vec{s}^T\overline{\overline{H}}\vec{s}+\vec{s}^T\vec{g}\text{ such that }\left\|\overline{\overline{D}}\vec{s}\right\|\leq\Delta\right\}. \tag{6.30}$$

Here $\overline{\overline{H}}$ is the Hessian matrix, a symmetric matrix of second derivatives, $\vec{g}$ is the gradient vector of $f(\vec{\eta})$, $\overline{\overline{D}}$ is a diagonal scaling matrix, and Δ is a positive scalar.

An issue that arises is that the solution of (6.30) can be quite costly when optimizing over the entire solution space. Therefore, the MATLAB trust-region-reflective algorithm performs an additional step which is to limit the solution to a two-dimensional subspace [44]. The subspace is defined as the space spanned by the gradient vector $\vec{g}$ and $\vec{s}_2$ which can be either the Newton direction or the direction of negative curvature. These are defined respectively as the solution to

$$\overline{\overline{H}}\vec{s}_2=-\vec{g}, \tag{6.31a}$$

$$\vec{s}_2^T\overline{\overline{H}}\vec{s}_2<0. \tag{6.31b}$$

The idea is to limit the solution of the trust-region problem to a subspace which ensures global convergence and provides fast local convergence. This is achieved by solving (6.30) in the plane described by the gradient and the Newton step.

The MATLAB documentation [42] gives a quick outline of the solution solution process which nicely sums up the steps taken. First, the two-dimensional trust-region problem is formulated. Next, (6.30) is solved for the step $\vec{s}$. The step is then tested in the function to see if it produces a smaller value. If so, the process starts over; if not, a smaller Δ is chosen.

As a final note, the `lsqcurvefit` subroutine uses the exact same solution algo-

rithms as the `lsqnonlin` subroutine. The benefit of `lsqcurvefit` is that it has been specialized for curve fitting operations so the function input is very simple for these types of problems. The downside is that `lsqcurvefit` does not accept a weighting function as an input to help correct for varying uncertainty in the measurements of the dependent variable data. A weighting function was not needed so `lsqcurvefit` was the best choice of solvers. However, if a weighting function needs to be applied it is much easier to use the `lsqnonlin` subroutine instead.

6.5 Curve-Fitting Results

The results of fitting Arrhenius parameters to reduced mechanisms through non-linear least squares are presented here. In the first section all algorithmic choices are defended and alternatives are discussed. Next, necessary steps to pre-process the data are covered. Last, the created mechanisms are presented and possible future improvements are discussed.

6.5.1 Discussion of Algorithm Choices

The algorithm presented in section 6.3 for finding Arrhenius parameters through a series of least squares optimizations was developed to improve upon existing methods in three main areas: speed of solution, accuracy of matching species production rates, and the ability of the algorithm to operate with little need for user intuition. The main method used to address these goals solve for the progresses of the reduced reactions before performing the non-linear least squares curve fitting.

6.5.1.1 Solving for the Progress of Reaction

The method presented herein distinguishes itself from other methods by first solving for the progress of each reaction, $\omega_{(r)}$, in the reduced mechanism before curve fitting the Arrhenius parameters. This step is critical and improves over previous

methods in two ways: better representation of the underlying physics, and a reducion in solution time. These will both be explained in further detail below.

To help understand the benefits of solving for the progress of reaction before attempting the non-linear least squares curve fitting, the species production rates, $R_{(n)}$, for the Westbrook and Dryer two-step mechanism will be used as an example. By taking the reaction steps from Table 3.3 and expanding the species production rates using (3.4) the following equations are found.

$$
\begin{aligned}
R_{(\mathrm{CH_4})} &= -\omega_{(1)} \\
R_{(\mathrm{H_2O})} &= 2\omega_{(1)} \\
R_{(\mathrm{O_2})} &= -1.5\omega_{(1)} - 0.5\omega_{(2f)} + 0.5\omega_{(2b)} \\
R_{(\mathrm{CO})} &= \omega_{(1)} - \omega_{(2f)} + \omega_{(2b)} \\
R_{(\mathrm{CO_2})} &= \omega_{(2f)} - \omega_{(2b)}
\end{aligned}
$$

Cantera directly outputs species production rates, the left hand side of the above equations, from its one-dimensional free flame simulation. However, the Arrhenius parameters that are desired to be solved for are contained within the progresses of reaction on the right hand side of the above equations. If the species creation rates are used to solve for the Arrhenius parameters directly, the analyst is presented with a dilemma. The Arrhenius parameters of the first reaction can be independently solved for using the species production rates of CH_4 and H_2O independently. The analyst must choose one, and in so doing, the influence of the other species on the Arrhenius parameters is completely ignored. Ideally the choice of which species production equation would not influence the results, however Jones found that this was not the case [3, p. 99]. Therefore, the analyst must test both possibilities and

hope one produces a favorable result.

The method presented herein avoids this issue by solving the system for the progresses of the reactions, $\omega_{(r)}$. Since there are more species than reactions the system is overdetermined and no solution exists. Therefore, the problem switches from finding an exact solution to finding a solution that is the best fit. Linear least squares (6.10) is the method chosen where the measure of the best fit is the minimization of the sum of squares of the residuals (6.3). Each progress of reaction that is solved for using linear least squares incorporates the influence of all species which take in that reaction. Therefore, no species have to be neglected and the natural physics are better represented. An additional benefit is that the analyst has less work and does not have to make judgment calls about which species to neglect.

The second benefit mentioned of solving for the progresses of reaction was a decrease in solution time. The forward and backward rates of the second reaction in the Westbrook and Dryer mechanism are a good example of the potential solution speed-up. As can be seen in the rate equations above, the second reaction shows up in the species production rates of O_2, CO, and CO_2. Unfortunately, since the backward reaction is a linear combination of the forward reaction, no combination of these species exists to separate the rates so they can be solved for independently. Therefore, if the species production rates, $R_{(n)}$, are desired to be used, the Arrhenius parameters for both reactions must be solved for simultaneously. If the progresses of reaction, $\omega_{(r)}$, are first solved for using linear least squares then the Arrhenius parameters for each reaction can be solved for independently.

Where solution time comes into play is in the determination of the initial starting point for the non-linear least squares method. This is found by testing an assortment of combinations of the parameters to see which produces the lowest sum of squares of the residuals. The computational time required for this operation is related to

the number of parameters through, $t \propto C^{N_\eta}$. Here t is time, C is a constant greater than one, and N_η is the number of parameters. If the forward and backwards rate parameters must be solved for simultaneously the time to find an initial starting point is, $t \propto C^6$. However, by decoupling the reactions by first solving for the species progresses of reaction, the time can be reduced to $t \propto 2C^3$. As an example of the time savings, if only three values are tested for each parameter the time required is reduced by 92.6%. Considering that run times are typically on the order of an hour, this speed up is very noticeable and useful.

The example of the Westbrook and Dryer two step mechanism used above is about the most simple mechanism that could have been chosen. For more complicated mechanisms, the benefits of solving for the progresses of reaction, $\omega_{(r)}$, before fitting the Arrhenius parameters only become more pronounced.

6.5.1.2 MATLAB Algorithm

To perform the non-linear least squares optimization in MATLAB there were two choices for solver algorithms as previous discussed: the trust-region-reflective and the Levenberg-Marquardt algorithms. The trust-region-reflective algorithm was chosen over the Levenberg-Marquardt algorithm for two main reasons.

The first was that the trust-region-reflective algorithm better predicted the near zero rates of reaction at ambient temperature. The Levenberg-Marquardt algorithm was able to produce results with nearly identical residual norms as the trust-region-reflective algorithm implying that both methods reached equivalent local minimums. However, the resultant mechanisms of the Levenberg-Marquardt method would typically have a non-zero reaction rate at ambient temperature. This non-zero rate would cause the reaction to spontaneously ignite. Not only is this a non-physical phenomenon, but the predicted flame speed would also be at least an order of mag-

nitude too high. In the axisymmetric FLUENT simulation the flame would spread uncontrolled throughout the computational domain. Solutions using the trust-region-reflective algorithm much more accurately modeled the low temperature reaction rates and therefore produced more physical results.

The second reason why the trust-region-reflective algorithm was chosen was due to its ability to impose bounds on the solution. Tests using unbounded algorithms would produce results with parameters considerably outside of typical ranges seen in literature. For instance, very large negative activation energies were being predicted for unbounded cases. While negative activation energies are permissible, large values are typically not encountered since they tend to cause reactions to spontaneously activate at low temperatures. Since methane does not spontaneously combust at standard atmospheric conditions, these large negative activation energies would not be physical. Therefore, the ability to put reasonable bounds on the solution using the trust-region-reflective algorithm proved to be very useful.

Initial attempts at performing the non-linear least squares curve fitting resulted in the solution converging extremely slowly. This was due to the vastly different scaling of the parameters to be found. The pre-exponential term and the activation energy are of similar size, but both are orders of magnitude larger than the temperature exponent. Where the temperature exponent is on the order of 1, the pre-exponential, depending on the units, can be on the order of 10^{15} or larger. The imbalance in parameter scaling would cause the algorithm to essentially ignore the temperature exponent since any change in this variable would be negligible in comparison to the other two terms. Very small tolerances had to be imposed to get the solution to converge properly causing the solution time to increase substantially. To combat this problem a scaling factor on the order of 10^6 was applied to the temperature exponent so that all the terms would be of similar magnitude. This scaling factor

allowed the algorithm tolerances to be reset to their default values which greatly reduced the solution time.

It is natural to wonder if scaling the temperature exponent causes any negative impacts on the solution. Luckily, not only does it not negatively affect the solution; it is actually recommended to rescale the variables to achieve faster solution convergence [37, pp. 155-9]. Rescaling the variables changes the steepest descent direction but does not affect the Newton direction. This is because the Newton method uses a quadratic model of the system which is not affected by unit changes. The steepest descent direction is altered because the value of the gradient is dependent on the units of the problem. The MATLAB algorithms use the steepest descent direction for the initial global convergence then locally converge using the Newton direction. Therefore, rescaling the temperature exponent can drastically improve the solution time by speeding up the global convergence.

The method used herein only incorporates a scaling factor on the temperature exponent. However, it is possible that better performance could be achieved by non-dimensionalizing the variables. This was not done here simply because the temperature exponent was easy to incorporate and produced the desired speed up. For future research, non-dimensionalization of the equations is recommended. However, care should be taken since the units of the pre-exponential term, $A_{r)}$, are variable.

6.5.2 Necessary Preprocessing Steps

To use the algorithm described in the preceeding sections it is necessary to have a data set containing temperature and species production rates from a physical experiment or detailed simulation. Herein a one-dimensional simulation was carried out using the GRI 3.0 mechanism to produce the data set. Cantera, the free chemical kinetics, thermodynamics, and transport processes software was used to perform this

simulation [45]. A python script was written to call the Cantera free flame utility which models the one-dimensional combustion of a premixed laminar flat flame. The python script used can be found in appendix section B.1 and the outputs of the simulation can be seen in figures 6.1 and 6.2.

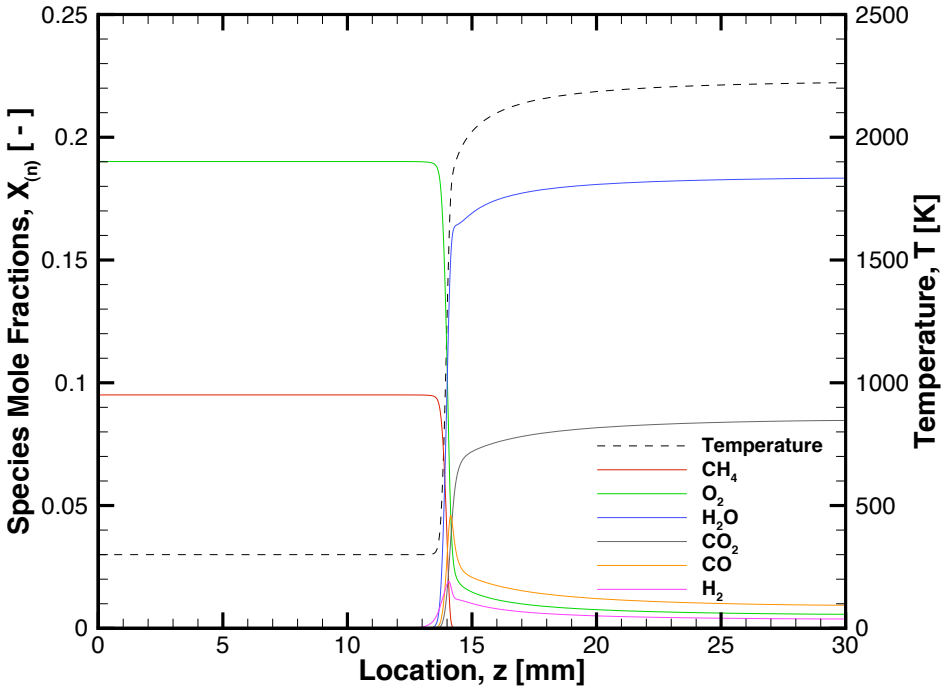

Figure 6.1: Temperature and mole fraction results from one-dimensional simulation of the GRI 3.0 combustion model [2] at atmospheric conditions with a stoichiometric mixture of methane and air. Data preprocessing is necessary to extract species production rates from this.

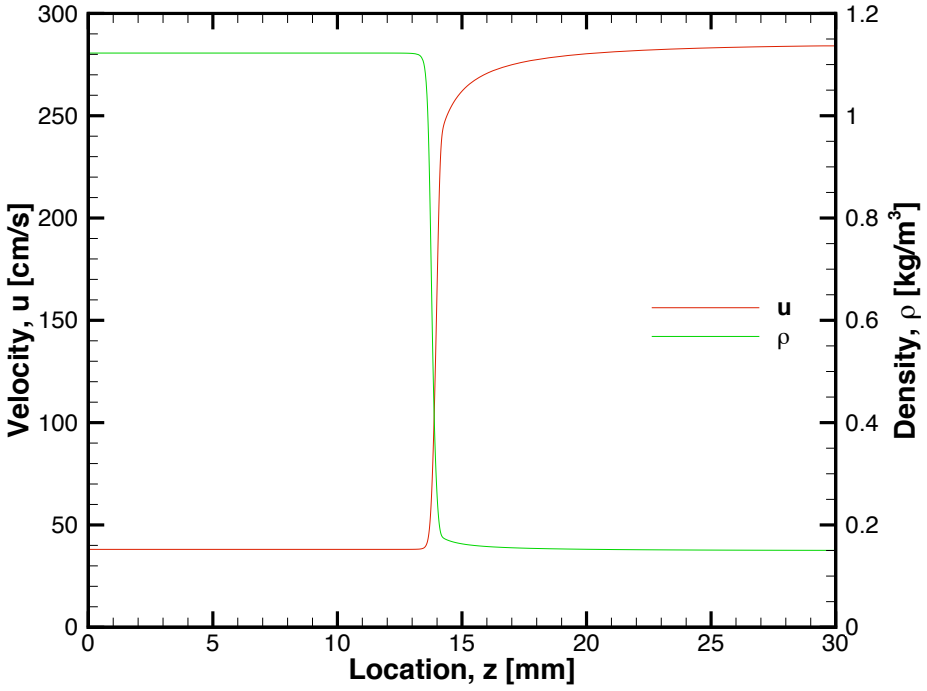

Figure 6.2: Flow velocity and density results from one-dimensional simulation of the GRI 3.0 combustion model [2] at atmospheric conditions with a stoichiometric mixture of methane and air.

The outputs of the detailed simulation are the species mole fractions, flow velocity, density, and temperature as a function of distance along the one-dimensional flow domain. Since the species production rates are the values of interest for the fitting algorithm, some preprocessing of the data must be performed. The first step is to convert the species mole fractions $X_{(n)}$ to molar concentrations $c_{(n)}$. The necessary equation is given in the Chemkin documentation [27, p. 22],

$$c_{(n)} = X_{(n)} \frac{\rho}{M_{mix}}, \quad n = 1 \ldots, N_s. \tag{6.32}$$

Recall that ρ is the density and M_{mix} is the average molecular weight of the mixture

calculated as

$$M_{mix} = \sum_{n=1}^{N_s} X_{(n)} M_{(n)}. \tag{6.33}$$

The next step is to approximate the species production rate based off of the flow speed. At each spatial grid point an approximation of the time it would take for a particle to reach that point can be calculated using the first-order backwards differencing method

$$t_i - t_{i-1} = \frac{z_i - z_{i-1}}{u_{i-1}}, \quad i = 2 \ldots, N_p. \tag{6.34}$$

Here t is time, z is the spatial location, and u is the velocity. With the time at each grid point calculated the species production rates can be approximated using a central differencing method,

$$R_{(n),i} = \frac{c_{(n),i+1} - c_{(n),i-1}}{t_{i+1} - t_{i-1}}, \qquad \begin{matrix} n = 1, \ldots, N_s \\ i = 2, \ldots, N_p - 1 \end{matrix}. \tag{6.35}$$

Special consideration needs to be paid to the first and last points when using central differencing since the stencil would include points outside the data set. This can easily be remedied by either using a one-sided differencing method, or by simply choosing the value at the boundary to be the same as the first point inside the data set. As can be seen in Fig. 6.1, the gradients at the boundary are nearly zero so either choice will produce reasonable results. One-sided differencing was used herein.

The species production rates are plotted as a function of temperature in Fig. 6.3. Positive values indicate species creation while negative values indicate destruction. Since this plot is from a one-dimensional flame simulation, all rates go to zero at 300 K and approximately 2000 K. These temperatures correspond to ambient conditions

and where the reactants have been depleted, respectively.

The final preprocessing step that could be undertaken is to add additional points in areas where greater resolution is desired. This can be accomplished by using cubic spline interpolation. Initial testing between a raw and interpolated data set produced no differences in the solution to the least squares curve fitting algorithm. However, it should be noted that all tests were run without using a weighting function. If a weighting function were used and additional points were included in areas of heavy weighting, it is possible that the solution could be influenced. No tests of this were carried out for the work herein.

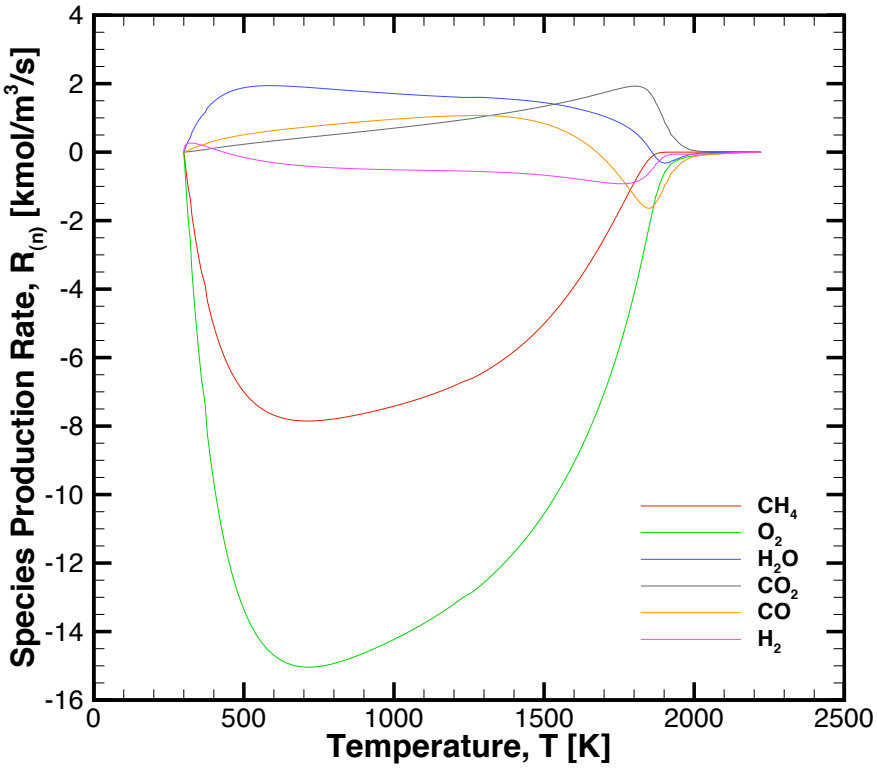

Figure 6.3: Species production rates as a function of temperature for the GRI 3.0 combustion model [2] at atmospheric conditions with a stoichiometric mixture of methane and air. This is the desired output from preprocessing the data for use in the Arrhenius parameter fitting algorithm.

6.5.3 Created Mechanisms

The Arrhenius parameter fitting algorithm presented in section 6.3 was used to find optimized parameters for the reaction steps of both the Westbrook and Dryer two-step mechanism [1] and the Peters and Williams three-step mechanism [10]. The newly created mechanisms will be called the optimized two-step and optimized three-step mechanisms. The values of the parameters are given in tables 6.2 and 6.3 for the optimized two-step and optimized three-step mechanisms, respectively.

Table 6.2: Optimized Two-Step Mechanism Arrhenius Parameters

Reaction	Equation	A	β	E
1	$CH_4 + 1.5O_2 \rightarrow CO + 2H_2O$	3.1623×10^{14}	0.8308	2.3855×10^4
2_f	$CO + 0.5O_2 \rightarrow CO_2$	4.2094×10^6	0.1251	7.3969×10^3
2_b	$CO_2 \rightarrow CO + 0.5O_2$	1.4286×10^9	0.2851	1.7072×10^5

Units are cm, mol, cal, s, and K.

Table 6.3: Optimized Three-Step Mechanism Arrhenius Parameters

Reaction	Equation	A	β	E
1	$CH_4 + O_2 \rightarrow CO + H_2 + H_2O$	4.8801×10^{12}	0.4452	2.3849×10^4
2_f	$CO + H_2O \rightarrow CO_2 + H_2$	2.3037×10^{11}	-1.0206	2.3901×10^3
2_b	$CO_2 + H_2 \rightarrow CO + H_2O$	1.4286×10^{12}	0.2851	1.7072×10^5
3	$O_2 + 2H_2 \rightarrow 2H_2O$	1.0000×10^9	2.5903	1.1360×10^2

Units are cm, mol, cal, s, and K.

Comparisons of the optimized two- and three-step mechanisms with the detailed

GRI 3.0 mechanism are shown in figures 6.4, 6.5, and 6.6 for predicted species mole fractions, species production rates, and temperature profiles, respectively. All simulations were calculated using the one-dimensional free flame utility in Cantera. As can be seen in figures 6.4 and 6.5 both created mechanisms accurately match the GRI 3.0 mechanism for the major species CH_4, H_2O, and O_2. Neither mechanism matches the profiles for CO_2 and the minor species CO and, in the case of the optimized three-step mechanism, H_2 very well. This is due to the fact that the minor species are typically transient, meaning that they have an initial burst of creation followed by a quick burst of destruction. This is most easily seen in Fig. 6.5. The major species are all characterized by a single spike of either creation or destruction while the minor species have a characteristic double spike profile.

This transient behavior is difficult to capture with a reduced mechanism for two reasons. First, the minor species are typically present only in small concentrations. They can easily become overshadowed by the major species in the curve fitting process. Second, to accurately capture the double spike profile at least two chemical reactions are necessary. When using reduced mechanisms, especially the optimized two-step mechanism, only a bare handful of reactions are used and these are typically targeted towards modeling the major species. This leaves too few reactions to properly model the minor species behavior, degrading the results.

The accuracy of the CO_2 profiles for the created mechanisms suffers due to the inability of the mechanisms to properly resolve the transient behavior of the minor species. In both mechanisms the only route from CH_4 to CO_2 is through CO. The inability to accurately model CO slows the transition of the carbon atoms to CO_2. The additional release of heat that would accompany the production of CO_2 is reduced which further exacerbates the issue.

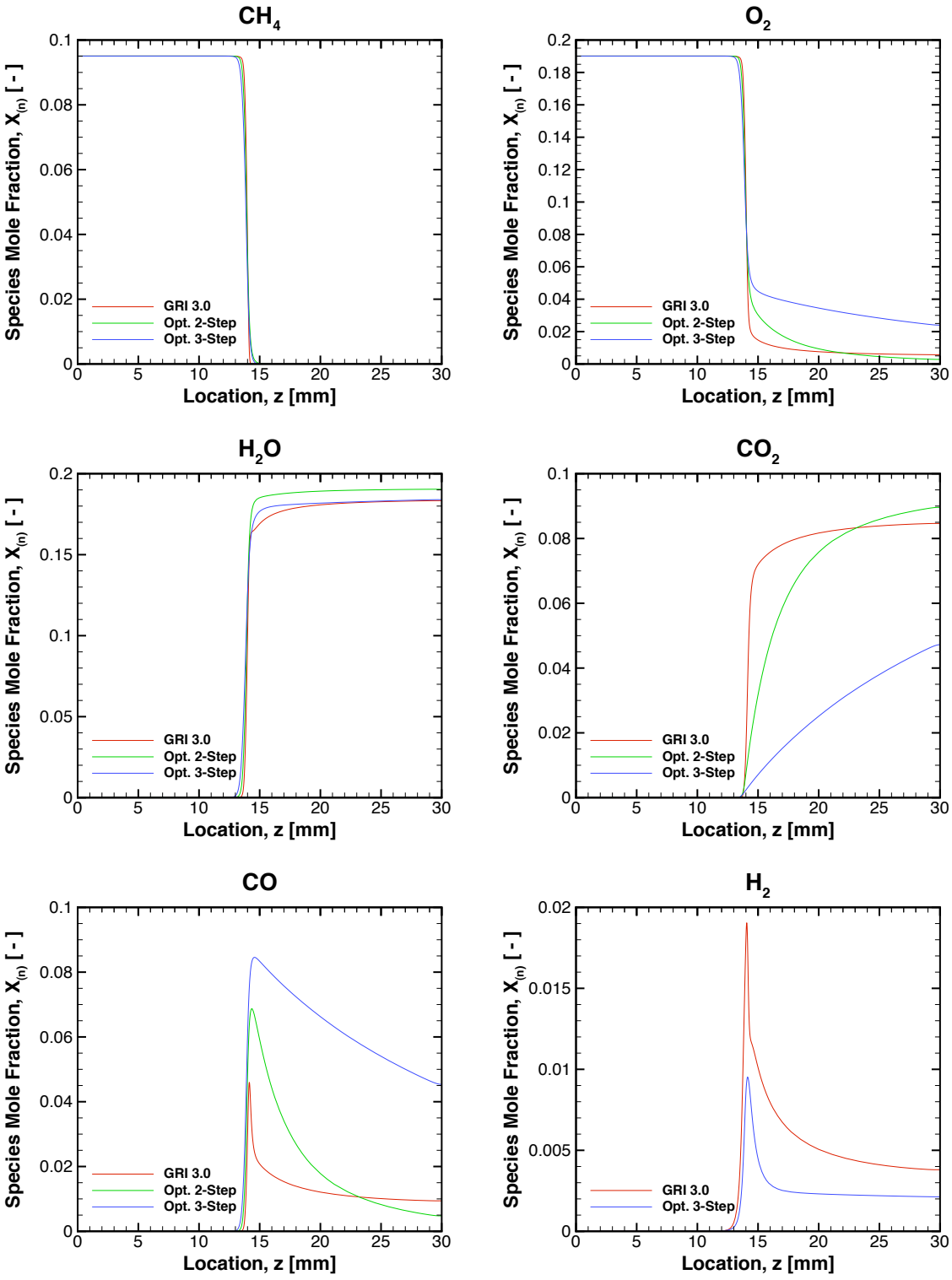

Figure 6.4: Comparison of species mole fractions as a function of distance for the GRI 3.0 [2] mechanism and the optimized two- and three-step mechanisms. Calculated using Cantera one-dimensional free flame package.

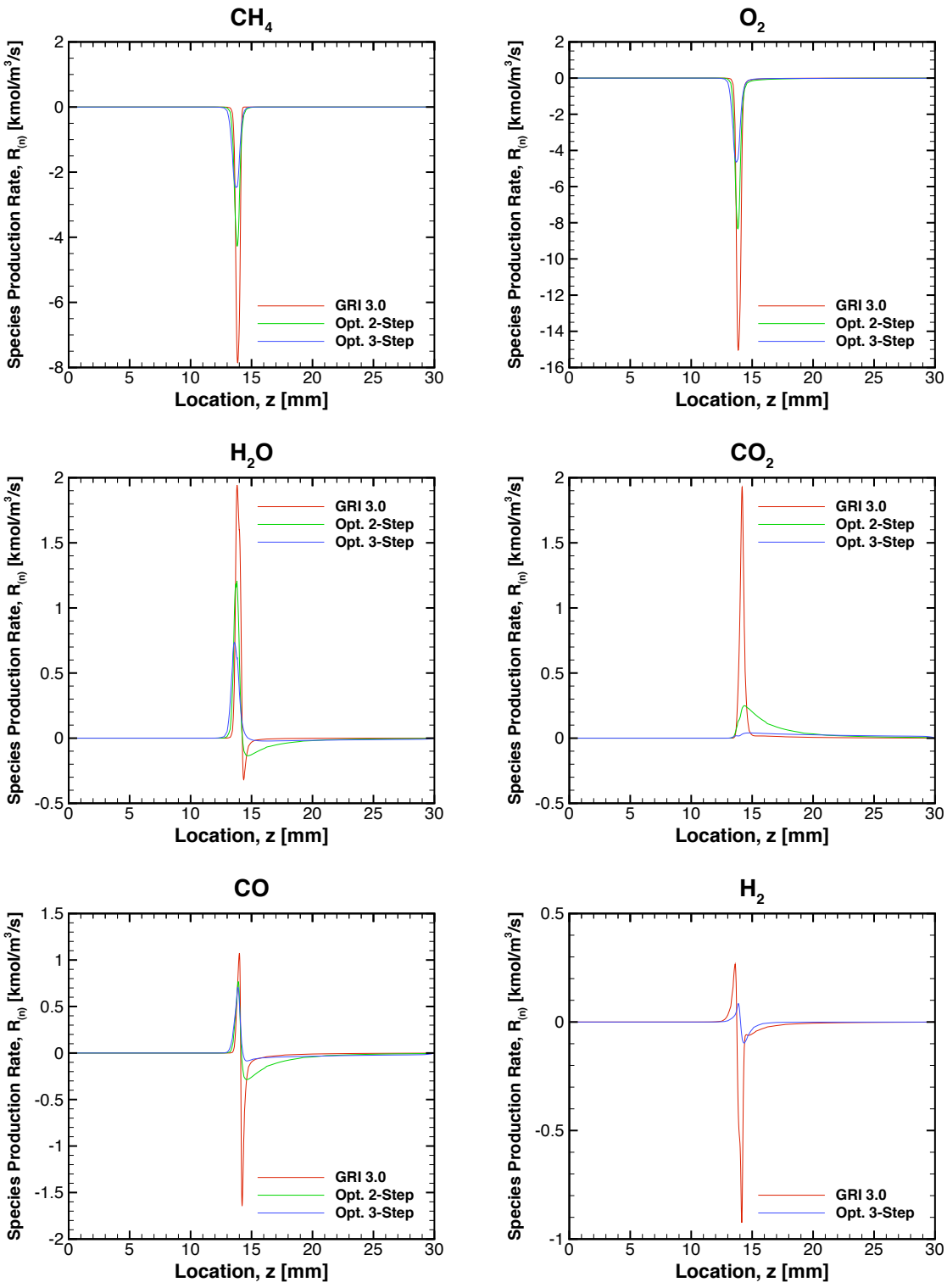

Figure 6.5: Comparison of species production rates as a function of distance for the GRI 3.0 mechanism [2] and the optimized two- and three-step mechanisms. Calculated using Cantera one-dimensional free flame package.

The optimized two-step mechanism matches the GRI 3.0 mechanism better than the optimized three-step mechanism. This can also be explained by the difficulty of modeling the minor species. The optimized three-step mechanism includes a second minor species, H_2, which exists in even smaller quantities than CO. Refer to Fig. 6.1 for a relative comparison of species mole fractions. This doubles the issue of accurately modeling CO_2 since it is now dependent on two minor species through the water-gas-shift reaction.

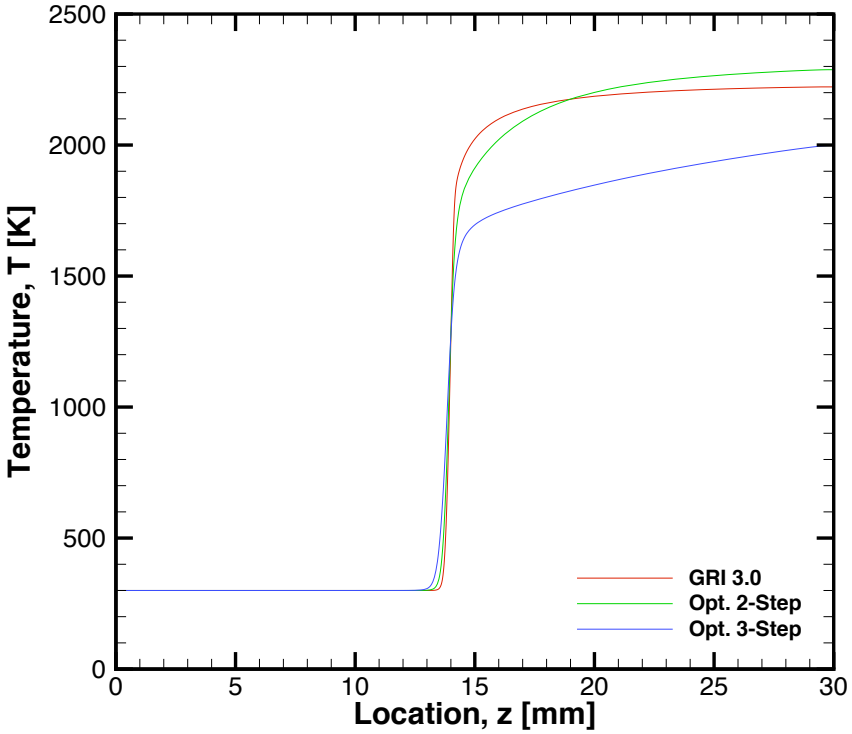

Figure 6.6: Temperature profiles as a function of distance for all created mechanisms compared to the GRI 3.0 mechanism [2]. Calculated using Cantera one-dimensional free flame package.

The first reaction of the optimized three-step mechanism oxidizes the methane fuel into water and both the minor species, CO and H_2. These then compete on

opposite sides of the water-gas-shift reaction to produce CO_2. Since the dynamics of both these species are poorly modeled by so few reactions, it slows down the production of CO_2 which delays the reaction. This effect is also seen by the temperature profiles in Fig. 6.6. The initial jump in temperature for the optimized three-step model is due to the fuel oxidation reaction. Once this reaction is completed the rise in temperature is much slower than the GRI 3.0 mechanism since the production of CO_2 is slowed by carbon getting stuck as CO.

Despite the difficult of modeling the minor species, both mechanisms created with the new algorithm produced very favorable results. With the exception of CO_2, the major species are predicted fairly accurately. Both mechanisms model the decomposition of the fuel so closely that the resulting profiles are nearly indistinguishable from the detailed mechanism.

As previously mentioned, the process for fitting Arrhenius rate parameters to reduced reactions developed in this thesis was inspired by the work of Jones [3, 19]. In his thesis, Jones created a two-step mechanism based on the Westbrook and Dryer two-step mechanism which he labeled Mech 2 [3, p. 103]. The Arrhenius parameters for this mechanism are shown in Table 6.4. The units have been updated to match those used in this thesis.

Table 6.4: Jones Mech 2 Arrhenius Parameters [3]

Reaction	Equation	A	β	E
1	$CH_4 + 1.5O_2 \rightarrow CO + 2H_2O$	1.5291×10^{13}	2.3077	6.0031×10^4
2_f	$CO + 0.5O_2 \rightarrow CO_2$	2.1685×10^8	-0.6022	-2.1560×10^1
2_b	$CO_2 \rightarrow CO + 0.5O_2$	1.4286×10^9	2.8571	1.7072×10^5

Units are cm, mol, cal, s, and K.

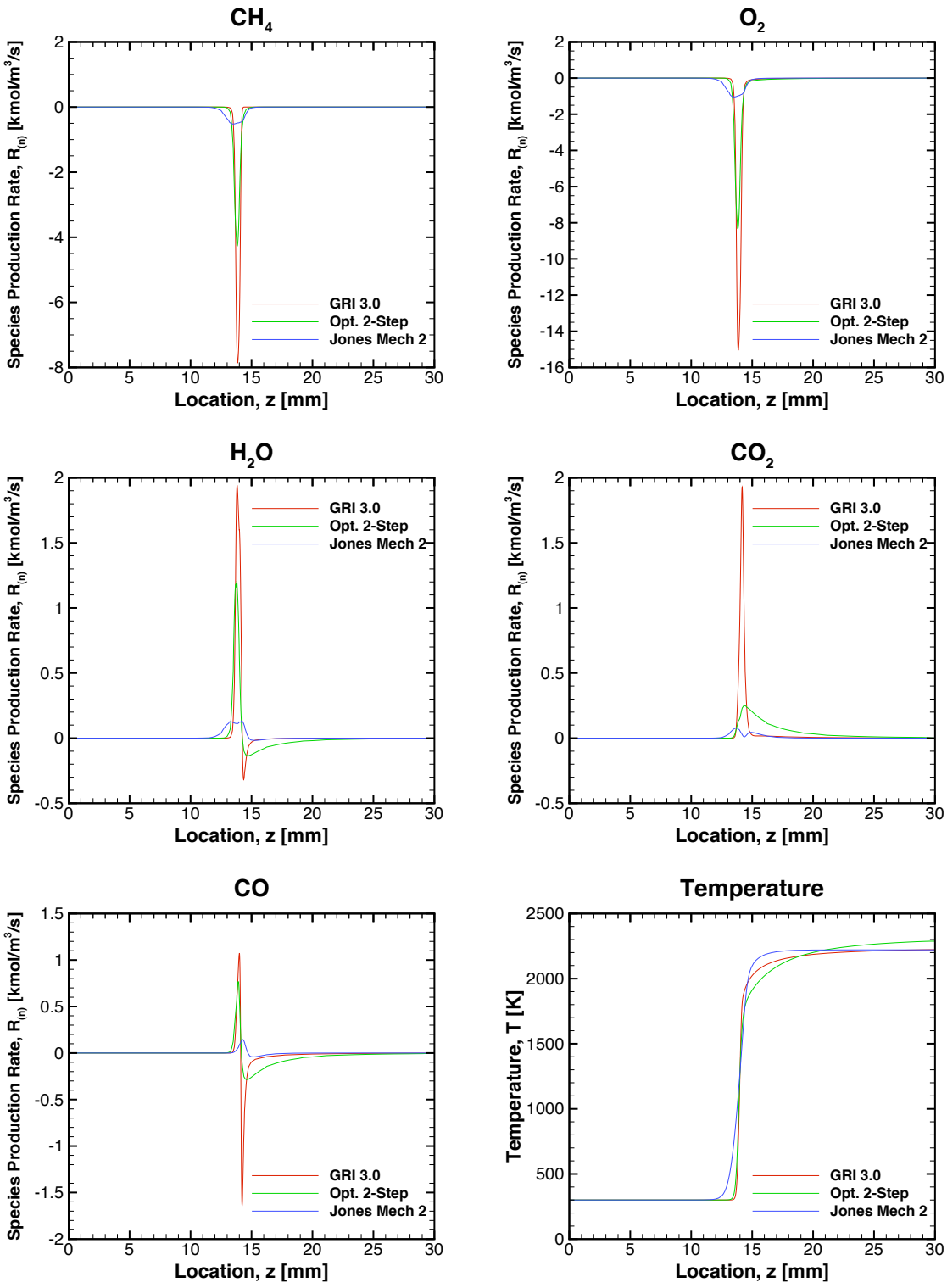

Figure 6.7: Comparison of species production rates temperature profiles as a function of distance for the optimized two-step mechanism, Jones Mech 2 [3] and the GRI 3.0 mechanism [2]. Calculations performed using Cantera free flame package.

Both the optimized two-step mechanism and Jones Mech 2 were created by finding Arrhenius rate parameters for the reaction steps of the Westbrook and Dryer two-step mechanism. Since they are based on the same reaction steps, and were fit to the same data set, these mechanisms are ideal for highlighting the improvements gained with the new fitting method. Figure 6.7 compares the species production rates and temperature profiles for one-dimensional simulations using both mechanisms to the GRI 3.0 mechanism.

For each of the species production rates shown in Fig. 6.7 the optimized two-step mechanism does a much better job of matching the GRI 3.0 mechanism. Jones Mech 2 has production rate profiles characterized by a small rounded bump while the optimized two-step mechanism profiles resemble a sharp spike, much closer to that of the detailed GRI 3.0 mechanism. This results in the maximum production rate being better predicted and the sharp transition at the flame front being more clearly defined for the optimized two-step mechanism. Additionally, for CO_2, Jones Mech 2 incorrectly shows two regions of species production where there should only be one. The optimized two-step mechanism consistently outperforms Jones Mech 2 for all species production rate predictions. Jones Mech 2 does do a slightly better job of predicting the maximum temperature. However, the optimized two-step mechanism shows good temperature prediction as well, with an error of less than 3%.

One of the most important features of the optimized two-step mechanism is its ability to predict the quick transition from creation to destruction of some chemical species. For H_2O and CO, which both exhibit the double spike of creation followed by destruction, Jones Mech 2 is only able to model the creation side of the profile. The optimized two-step mechanism, however, is able to realize both the creation and destruction rates of these two species. This is a critical advantage of mechanisms created using the new fitting method. Reduced mechanisms typically do not contain

enough reactions to properly model the creation and destruction of minor species. Finding that the new fitting method is able predict this behavior for such a small mechanism is very encouraging.

Table 6.5 gives a comparison of the flame speeds for the optimized two-step and three-step mechanisms, the Jones Mech 2, and the detailed GRI 3.0 mechanism. These were all calculated using Cantera for standard atmospheric conditions and a stoichiometric mixture of methane and air. Both the optimized two-step and three-step mechanisms substantially reduced the error in the predicted flame speed from the Jones Mech 2. The optimized two-step mechanism sees an improvement in the flame speed error of 47.5% over Jones Mech 2.

Table 6.5: Flame Speed Comparison of Various Mechanisms

Mechanism	Flame Speed [cm/s]	Error [%]
GRI 3.0	38.05	-
Optimized 2-step	28.43	25.28
Optimized 3-step	22.57	40.28
Jones Mech 2	10.35	72.80

The improvement in flame speed prediction is due to the difference in species production rates as seen in Fig. 6.7. The low, wide, rounded profiles for Jones Mech 2 show that the reactions are occurring much slower than for the optimized two-step mechanism. Since the reactions are occurring so much more slowly, the flame consumes the reactants slower and consequently moves slower.

Up to this point all of the discussion of the created mechanism has centered around the Cantera one-dimensional flame simulations. Comparisons of all mech-

anisms when applied to the axisymmetric FLUENT model of flame A, including violations of the DEI, will be discussed in chapter 7.

6.5.4 Future Improvements

The method for finding Arrhenius parameters for reduced mechanisms developed above produced promising results, however there are areas which stand to be improved upon. One that was previously noted was the difficulty of modeling the minor species properly due to their small concentrations relative to the major species. To increase the importance of the minor species in this method they can be weighted during the initial linear least squares step. This weighting can be implemented with minor modifications to (6.18)

$$\vec{\omega} = \left(\overline{\overline{\nu}}^T \overline{\overline{W}} \overline{\overline{\nu}}\right)^{-1} \overline{\overline{\nu}}^T \overline{\overline{W}} \vec{R}. \tag{6.36}$$

Here $\overline{\overline{W}}$ is a diagonal weighting matrix of size $N_s \times N_s$. Each entry along the diagonal weights the importance of a given species relative to the others. The result of this is that the progresses of the reactions, used in the non-linear least squares curve fitting, are preferentially weighted towards the species which are determined to be most important.

The use of a weighting matrix was briefly tested in an attempt to improve the accuracy of the species production rate for CO in the optimized two-step mechanism. A plot comparing the CO production rate using both the weighted and unweighted methods to the exact profile from the GRI 3.0 mechanism can be seen in Fig. 6.8. The weighting matrix shows initial promise by reducing the norm of the residual by nearly 11%. This greater accuracy has the potential to help improve the modeling of the transient behavior of the minor species which can lead to better reduced reaction parameter fitting.

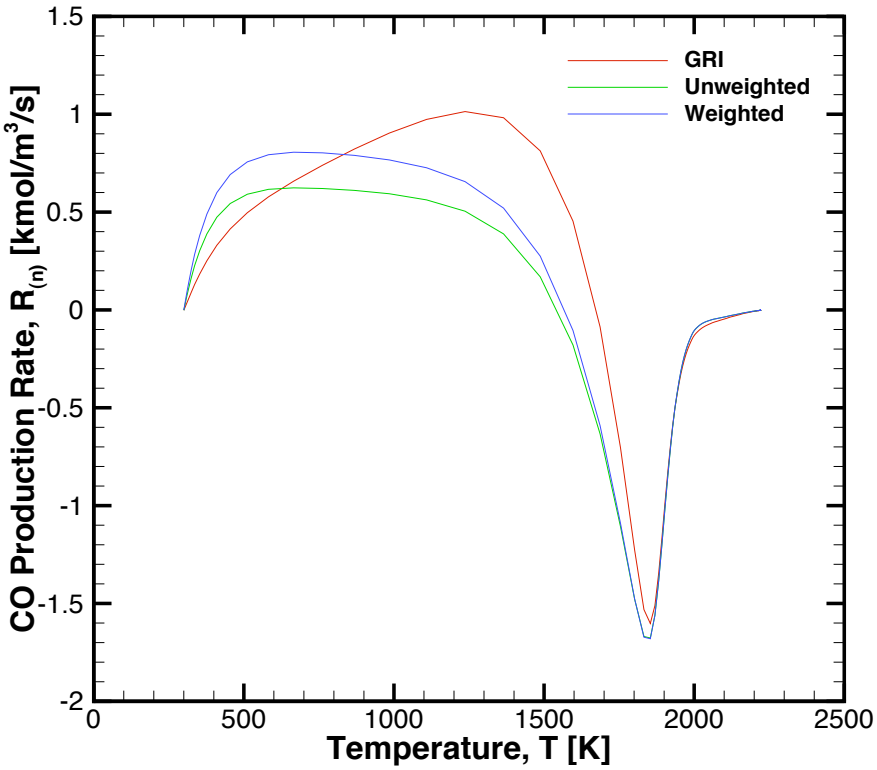

Figure 6.8: Initial results showing improvement of the CO production rate fitting by using weighted least squares.

Another possible improvement is to incorporate bounds into the linear least squares step of the algorithm. A close inspection of the definition of the progress of reaction in (6.16) will reveal that for a positive pre-exponential, $A_{(r)}$, the function is strictly positive. Therefore, better parameter fits are possible for the reduced mechanism if the progresses of reaction solved for during the linear least squares step of the algorithm are bounded to be greater than zero.

Additional improvements that could be made are more efficient methods of finding an initial guess for the solution. The linear least squares step aids this by drastically reducing the number of combinations of points to test. However, this is still the most

time consuming aspect of the fitting process. Line searches and gradient methods are just two possibilities to help find a reasonable guess of the solution.

6.6 Summary

A new method for determining the Arrhenius parameters of a reduced chemical mechanism was developed for this thesis. This method seeks to find an optimal set of parameters for a specific operating condition. The basic strategy is to approximate the progress of each reaction in a reduced mechanism from the species production rates of a detailed mechanism. A series of non-linear least squares curve fittings are then carried out to find the optimal Arrhenius parameters for each reaction. This process was implemented into a MATLAB code and was used to find parameters for the reaction steps of the Westbrook and Dryer two-step mechanism and the Peters and Williams three-step mechanism. Both optimized mechanisms showed significant improvement over previous mechanisms in minor species production rates and flame speed predictions. Both optimized mechanisms showed good agreement with the detailed mechanism for species mole fractions and production rates of most major species. Minor species and CO_2 were poorly predicted due to insufficient reactions to fully model the rapid creation and destruction of the minor species. Potential improvements include preferentially weighting minor species and using a positively bounded linear least squares approach.

7. COMPARISON OF MECHANISMS

Reduced mechanisms are created to decrease the complexity and computational cost of a detailed mechanism. The simplifying assumptions used to create a reduced mechanism decreases the range of applicability of the resultant mechanism. For instance, the Westbrook and Dryer two-step mechanism was made to predict flame speed and flammability limits well, but has been shown to produce violations of the DEI. The process presented in chapter 6 instead focuses on creating reduced mechanisms which automatically satisfy the DEI. In this chapter, the results of simulating the Sandia flame A with five combustion mechanisms, including the detailed GRI 3.0 mechanism, will be explored to compare the relative merits of each. Section 7.1 examines the two-dimensional structure of the simulated flames and shows comparisons to experimental data. Violations of the DEI are discussed in section 7.2. The automatic satisfaction of each term of the DEI is investigated to discover the the root causes of DEI violations.

7.1 Axisymmetric Flame Profiles

The Sandia flame A experiment was simulated using five different methane-air combustion mechanisms. These included the detailed GRI 3.0 mechanism, Westbrook and Dryer two-step mechanism, Jones Mech 2, and the optimized two- and three-step mechanisms from tables 6.2 and 6.3. Before delving into violations of the DEI, it is useful to understand the structure of the resulting flames.

7.1.1 Temperature and Composition Contours

In each simulation of the Sandia flame A the temperature limiter was set to 2400 K so that it would not influence the converged solution. Therefore, the maximum tem-

perature found in the results should be a numerical approximation of the adiabatic flame temperature. Table 7.1 gives a comparison of the maximum temperature for each simulation along with predicted values for the adiabatic flame temperature and the calculated error. Since the adiabatic flame temperature is a function of the chemical species present, only mechanisms which include the same species can be directly compared. For all others, only a comparison of error is possible.

Table 7.1: Maximum Temperature of Various Mechanisms

Mechanism	Num. Spec.	Adia. F.T. [K]	Max Temp. [K]	Error [%]
GRI 3.0	53	2225.5	2268.0	+1.91
Westbrook & Dryer	6	2258.2	2305.4	+2.09
Jones Mech 2	6	2258.2	2229.2	-1.28
Optimized 2-step	6	2258.2	2311.2	+2.35
Optimized 3-step	7	2246.4	2261.0	+0.65

The maximum flame temperature error is quite low, at less than 2.4% for each mechanism. For all mechanisms besides Jones Mech 2, the maximum temperature is higher than the predicted adiabatic flame temperature. This was previously seen in Fig. 5.9. Interestingly, of all the six species models, Jones Mech 2 had the lowest error in temperature. This result was also found in the one-dimensional flame simulations seen in Fig. 6.7.

Figures 7.1 and 7.2 show contour plots of temperature and methane mass fraction respectively for each mechanism. Recall that the temperature limiters were set to 2400 K for all simulations. This value was chosen to decrease ISAT table lookup time while not influencing the solution results, as discussed in chapter 5.

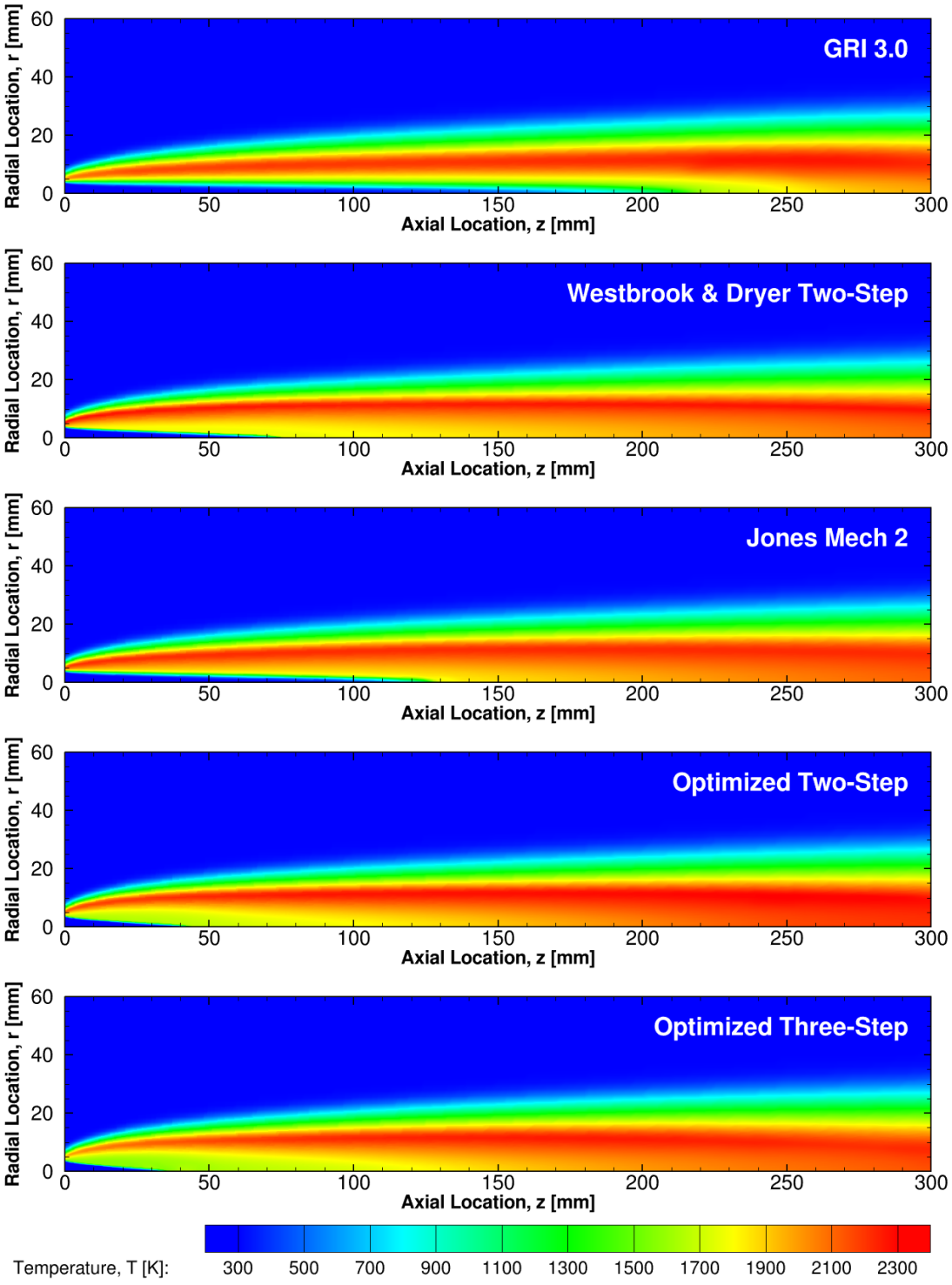

Figure 7.1: Temperature profiles of various mechanisms. From top to bottom: GRI 3.0 [2], Westbrook and Dryer two-step [1], Jones Mech 2 [3], optimized two-step, optimized three-step.

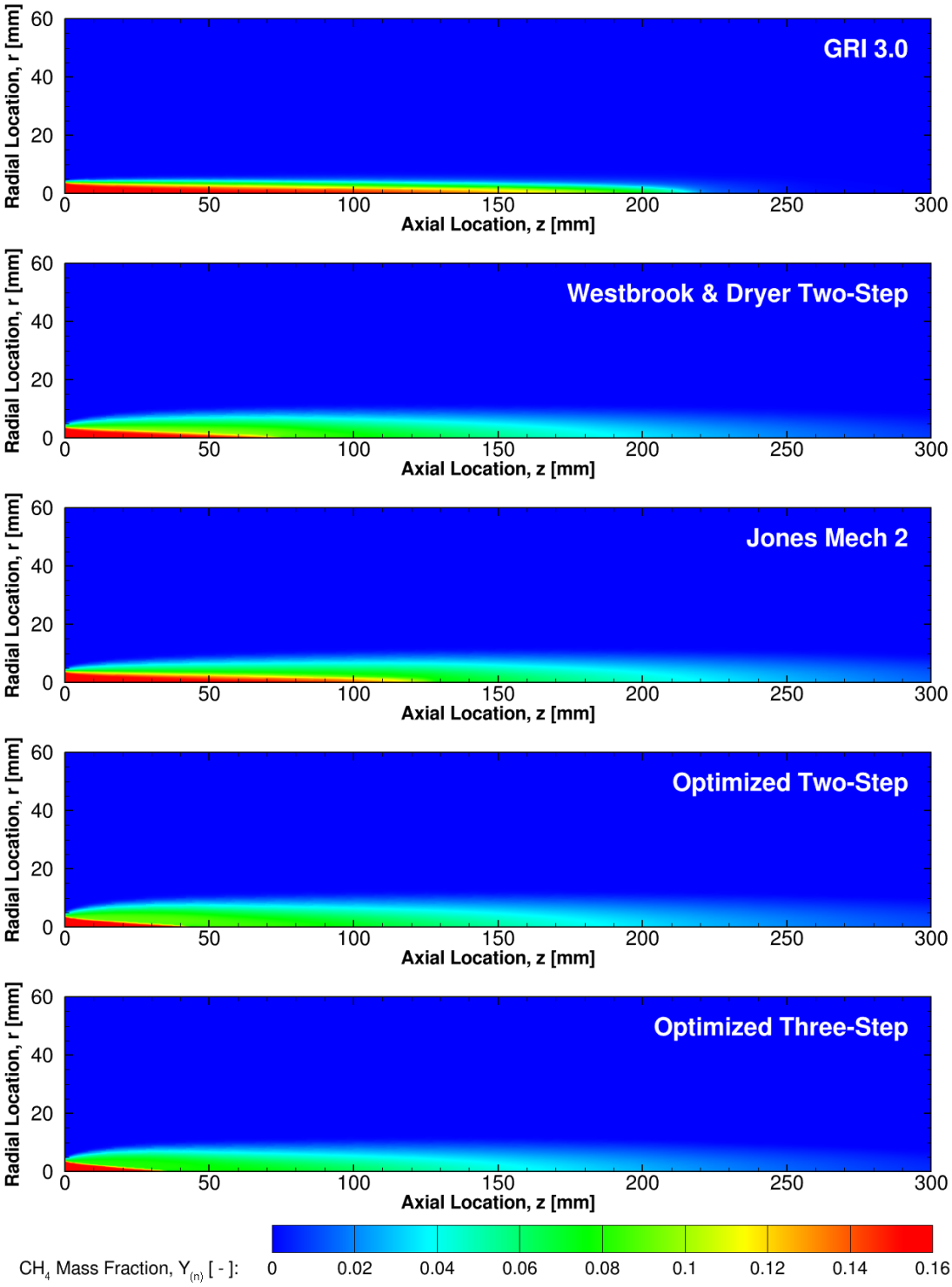

Figure 7.2: Methane (CH_4) mass fraction profiles of various mechanisms. From top to bottom: GRI 3.0 [2], Westbrook and Dryer two-step [1], Jones Mech 2 [3], optimized two-step, optimized three-step.

A quick inspection shows that the basic structure of the flame for each mechanism is the same. Each flame contains a cool conic interior section of unburnt fuel, and a zone of maximum temperature which is approximately centered radially in the flame. Temperature gradients exist on the inside and outside surfaces of the flame and are where reactions are occurring most prominently.

In both figures 7.1 and 7.2, it can be qualitatively seen that the cool interior section of the flame is a different length for each mechanism while the outer surfaces seem to occur in nearly the same location. As discussed in chapter 5, the experimental data was collected at 25, 50, and 100 mm downstream of the fuel tube exit. Clearly for some of the reduced mechanisms, the cool interior doesn't extend out 100 mm. Therefore, the greatest differences in the solutions are expected to occur near the centerline of the flame, and indeed this is the case, as will be displayed in section 7.1.2.

An interesting phenomena is found in the methane mass fraction profiles which may help to explain the differences in structure of the flames. In Fig. 7.2 it can be seen that the unburnt fuel (shown in red on the contour plots) extends a variable distance into the flame. However, the distance at which the fuel is completely depleted is approximately 200 mm downstream and is nearly the same for each mechanism. The variable internal length is due to premature ignition of the premixed fuel and differences in reaction rates between the mechanisms. The nearly constant distance at which fuel depletion occurs is due to the oxygen diffusion rate being approximately constant between each simulation.

For all the reduced reactions the premixed oxygen is quickly depleted since the mixture is fuel rich and the single initiation reaction feeds directly on O_2. The rate of the initiation reaction determines the slope of the inner edge of the flame. The faster the reaction rate, the steeper the slope. Once the premixed oxygen is depleted, the overall reaction stagnates until enough oxygen can diffuse into the flame to react

with the remaining fuel. Since the rate of oxygen diffusion is roughly the same for each simulation, the point at which the fuel runs out is nearly constant.

The length of the internal cool cone for the detailed GRI 3.0 mechanism is longer than for the reduced mechanisms since the initiation of the overall combustion is much more complex than for the reduced mechanisms. Unlike the reduced mechanisms which initiate with the reaction of CH_4 and O_2, the elementary initiation steps in the detailed mechanism require radical species to start the chain reaction. Since no radical species are present in the premixed fuel flow, the radical pool must first be established before the detailed mechanism can combust. Not until high enough oxygen concentrations diffuse into the flame can the radical pool grow to a sufficient size for the propagation and chain branching steps to drive the overall reaction to termination.

In previous studies the reduced mechanisms were made to behave more like the detailed mechanism by eliminating the oxidizer in the fuel flow [3, pp. 86-92]. In an attempt to get a three-step mechanism simulation to run properly, Jones experimented with removing the premixed oxygen from the fuel and replacing it with nitrogen. The resulting flame had an elongated interior cool section which better matched experimental data and more closely resembled the GRI 3.0 mechanism from Fig. 7.1. This result was due to the lack of oxidizers in the fuel stream to prematurely start the reactions. The reactions could only begin once enough oxygen had diffused into the flame. Since the diffusion rate was unaltered between these simulations, the resulting solutions appeared to match experimental data better. However, this was not due to mechanisms performance, but rather because the flame was diffusion limited and was being denied a premixed supply of oxygen.

7.1.2 Comparison to Experimental Data

Profiles of temperature and species mass fractions were taken for each of the five mechanisms at 25, 50, and 100 mm downstream of the fuel tube. These profiles were then plotted against the experimental Sandia flame A data shown in figures 7.3-7.5.

The GRI 3.0 mechanism, being the only detailed mechanism tested, was expected to match the experimental data the closest. This expectation was verified as shown in figures 7.3-7.5 below. Along the inner edge of the flame the GRI 3.0 mechanism matches the experimental data very well. This is particularly emphasized in Fig. 7.4 where the consumption of CH_4 is shown. At all three axial locations the curve for the GRI 3.0 mechanism falls almost directly on top of the experimental data. Very close agreement is found between the GRI 3.0 mechanism and the experimental data along the inner edge of the flame. This result is a good verification for the detailed GRI 3.0 mechanism being used as the optimization target for the Arrhenius parameter fitting presented in chapter 6.

There is a wide variation in the structure of the inner edge of the flame predicted by the various reduced mechanisms. In the last section it was theorized that this was due to a premature combustion of the premixed fuel. For the Sandia flame A experiment the fuel mixture was chosen to be outside the flammability limits of the methane fuel. However, the flammability limits of the reduced mechanisms may not match the physical values. The optimized two- and three-step mechanisms developed herein, along with Jones Mech 2, were created to minimize the number of violations of the DEI seen in reduced mechanisms. Never in their creation were flammability limits considered. Therefore, a study must be carried out to determine if the flammability limits are well predicted by these mechanisms.

Flammability limits are typically given in terms of the equivalence ratio, ϕ, which

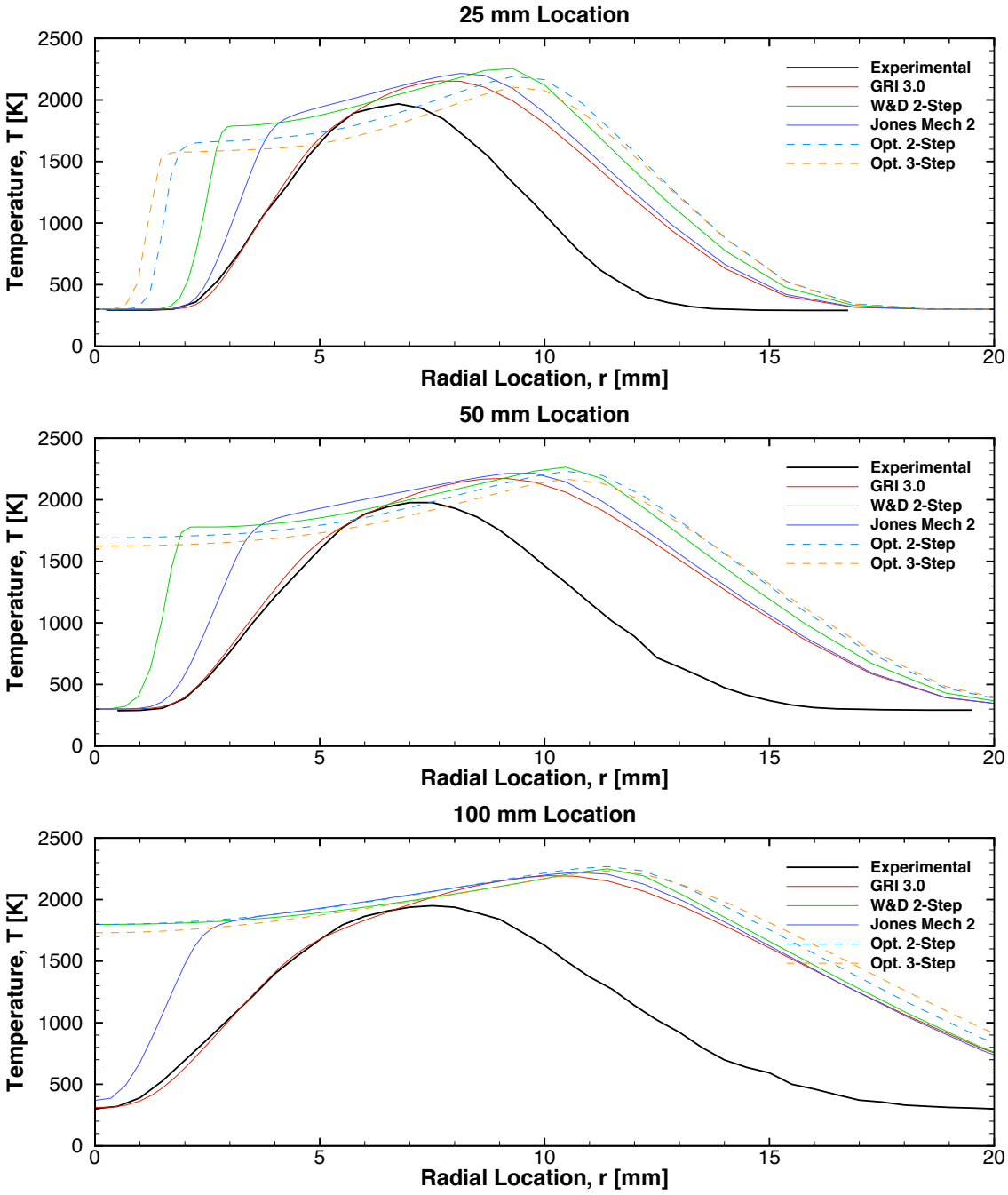

Figure 7.3: Comparison of temperature profiles to experimental data for all mechanisms. Profiles shown from top to bottom at: 25 mm, 50 mm, and 100 mm locations.

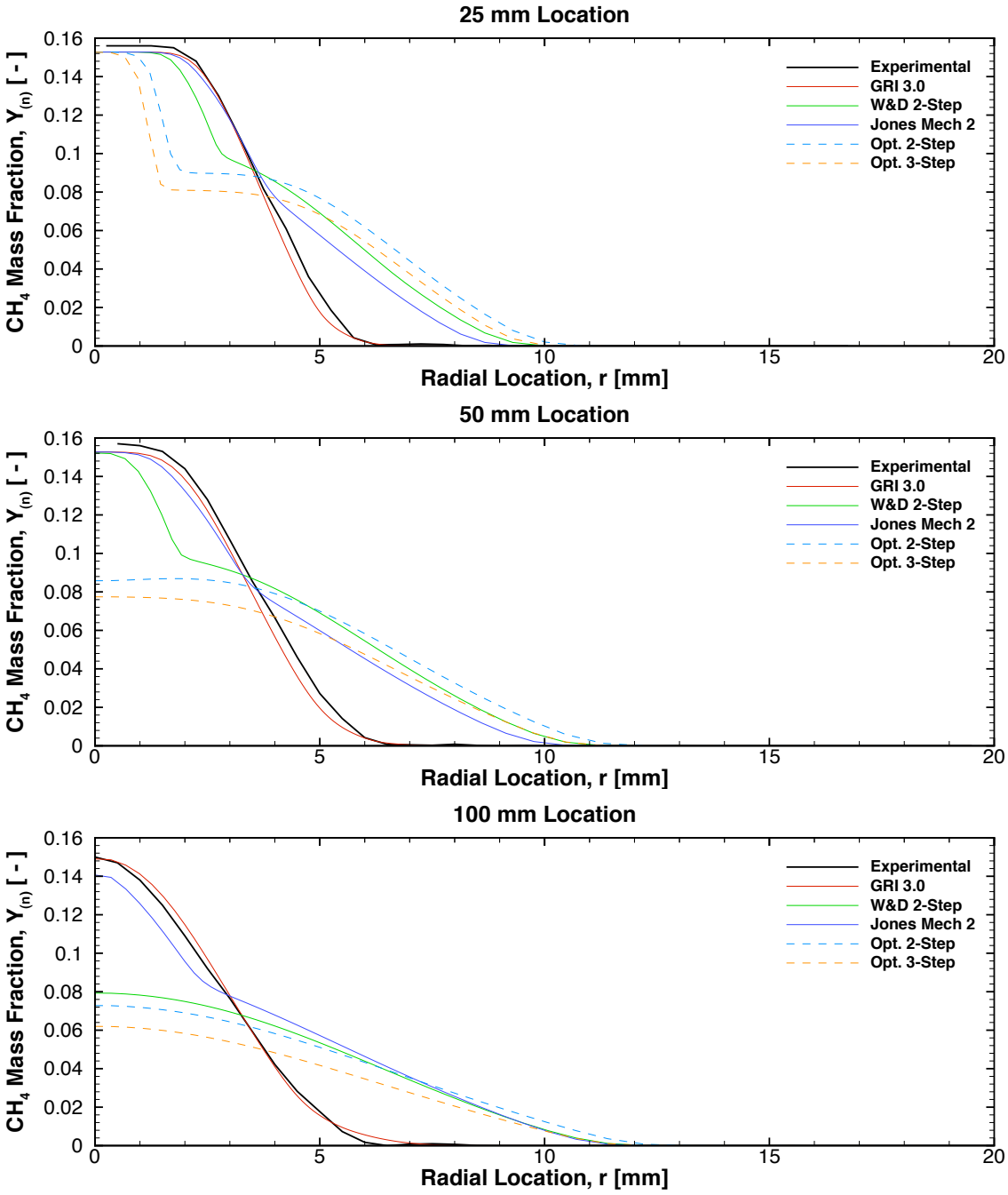

Figure 7.4: Comparison of CH_4 mass fraction profiles to experimental data for all mechanisms. Profiles shown from top to bottom at: 25 mm, 50 mm, and 100 mm locations.

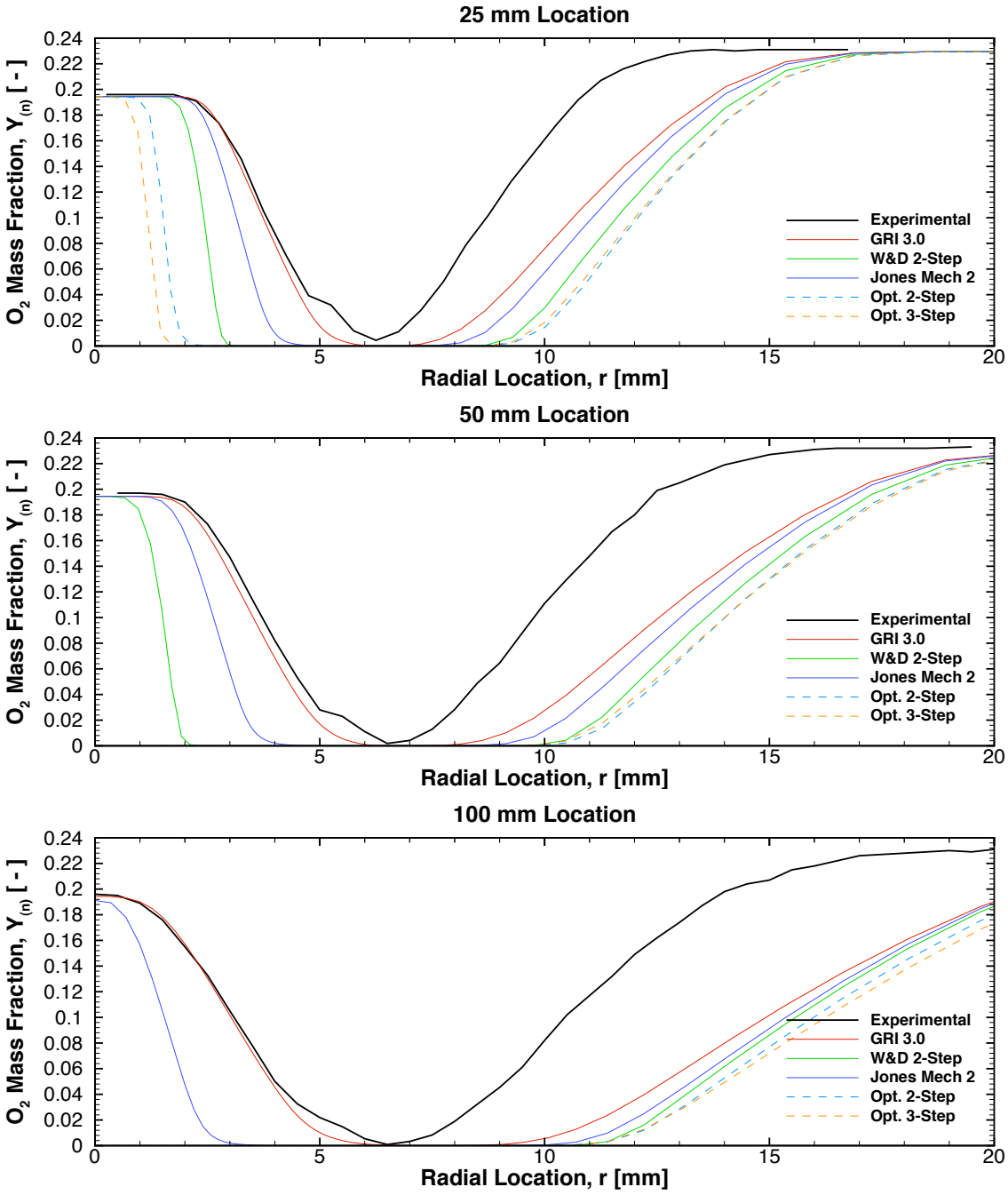

Figure 7.5: Comparison of O_2 mass fraction profiles to experimental data for all mechanisms. Profiles shown from top to bottom at: 25 mm, 50 mm, and 100 mm locations.

is defined as

$$\phi = \frac{n_{(fuel)}/n_{(oxidizer)}}{\left(n_{(fuel)}/n_{(oxidizer)}\right)_{stoi}}. \tag{7.1}$$

Here $n_{(m)}$ is the number of moles of species m. The equivalence ratio relates the current fuel-to-air ratio to the stoichiometric fuel-to-air ratio. Table 7.2 shows a numerical approximation of the upper and lower flammability limits for the optimized two- and three-step mechanisms and Jones Mech 2. These values are compared to experimental values supplied by Westbrook and Dryer [1]. The limits for the reduced mechanisms were approximated using the Cantera one-dimensional free flame utility. The equivalence ratio was incrementally adjusted until the simulation failed to converge. The simulation was run to an upper limit of 10 in accordance with the values used by Westbrook and Dryer.

Table 7.2: Approximation of Flammability Limits for Reduced Mechanisms

Mechanism	$\phi < 1$	$\phi > 1$
Experimental	0.5	1.6
Jones Mech 2	0.6	5.4
Optimized 2-step	0.2	> 10
Optimized 3-step	0.3	> 10

The equivalence ratio of the premixed fuel flow for the Sandia flame A experiment was 3.17. While this value was outside the flammability range of an actual flame, it was well within the ranges of the reduced mechanisms shown in Table 7.2. Therefore, along the inner edge of the flame, premature combustion was expected. The premature combustion is clearly shown in the contours of each of the reduced mechanisms

in figures 7.1 and 7.2 as a shortened interior cone of unburnt fuel compared to the GRI 3.0 mechanism.

The degree to which the interior cone of unburnt fuel was shortened for the reduced mechanisms was determined by the reaction rate of the fuel decomposition reaction. A faster reaction rate led to a shorter cone. It was predicted that the flame speed is proportional to the square root of the reaction rate [1]. Based on this relation, and the predicted flame speeds from Table 6.5, the optimized two- and three-step mechanisms were predicted to exhibit the shortest sections of unburnt fuel. Another way to view this is in the temperature profiles of Fig. 7.3. For the optimized two- and three-step mechanisms the temperature does not dip back to ambient at the center of the flame for the 50 mm and 100 mm sampling locations due to the shortened interior cone. Therefore, while the optimized two- and three-step mechanisms predict the flame speed most accurately, Jones Mech 2 appears to match the inner surface of the Sandia flame A more closely.

Interestingly, none of the five mechanisms tested show good agreement with the experimental data on the outside edge of the flame. For all mechanisms tested, the overall width of the flame was over-predicted by approximately 25%. While the detailed GRI 3.0 mechanism consistently showed the closest match to the experimental data on the outside edge of the flame, all of the reduced mechanisms also predicted very similar results to the detailed mechanism along the outer edge. This is best shown in the temperature profiles in Fig. 7.3 and the oxygen mass fraction profiles in Fig. 7.5. At the outside edge of the flame the profiles for all of the mechanisms seem to cluster together into one consistent curve which sits a considerable distance from the experimental data.

The similarities between all the solutions at the outside edge of the flame are due to the flame being diffusion rate limited. Since the fuel and oxygen supplies are on

opposite sides of the flame, combustion can only occur when the fuel and oxygen have sufficiently diffused together. Diffusion occurs on a slower time scale than the reaction rates and the same diffusion model is used on all simulations. Therefore, the solutions at the outside edge of the flame should be very similar, regardless of reaction mechanism used.

The reason for the difference between the numerical simulations and the experimental data at the outer edge of the flame is most likely due to an error in the co-flow velocity. In all simulations the co-flow was taken to be 0.4 m/s. This value was first used by Chambers [4] and was established through an undocumented personal correspondence. However, documentation from Sandia [35] suggests that the actual co-flow velocity was 0.9 m/s. This is a substantial difference which could account for error between the numerical simulations and the experimental data. Further testing is required to determine if simulations with a co-flow of 0.9 m/s decreases the error at the outer edge of the flame.

To summarize the results of the flame structure, it was determined that, for the reduced mechanisms, the flame was allowed to prematurely ignite since the premixed fuel was within their flammability limits. Once the premixed oxygen was depleted the reactions stagnated until enough oxygen diffused into the flame for the fuel to fully combust. The initial oxygen depletion is clearly shown in Fig. 7.5. Along the inner edge of the flame the premixed oxygen concentration is quickly consumed. In Fig. 7.4 the corresponding drop in fuel mass fraction is shown. After a quick burst of fuel consumption the profile flattens out until more oxygen can diffuse into the flame and allow the fuel decomposition to progress to completion. Since the diffusion rate is the same for each simulation, the profiles along the outer edge of the flame are all very closely clustered. The reaction rates in this section of the flame are faster than the diffusion rates so the flame structure is determined by the slower species

diffusion rates. Therefore, despite the differences in mechanisms, all the simulations show very good agreement along the outer edge of the flame.

7.1.3 Plateaued Structure of Reduced Mechanisms

In each of the plots in Fig. 7.3, there is a characteristic plateaued structure exhibited by each of the reduced mechanisms. Where the experimental data and the GRI 3.0 simulation have rounded temperature profiles, the reduced mechanisms are defined by nearly flat, slightly sloping temperature plateaus in the middle of the flame. This structure can be explained by the low number of species and lack of competing reactions in the reduced mechanism.

In a detailed mechanism, when the initiation reactions occur, the heat released goes into supplying the activation energy to fuel further reactions. Gradually, as the chain reactions progress, the overall temperature increases until a maximum is reached. For the reduced mechanisms, the heat released by the initiation reaction has no other reactions to supply so the temperature increases sharply. The temperature then remains relatively constant until the next reaction occurs. For many of the reduced mechanisms shown in Fig. 7.3 there are only two reactions so the temperature profiles are defined by two sharp gradients.

Figure 7.6 shows the reaction rates overlaid on the temperature profiles for the Westbrook and Dryer two-step mechanisms. It is clear from these plots that edges of the plateaued structure correspond with peaks in reaction rate. From the inside of the flame outward, the initial jump in temperature occurs where the first reaction spikes. At this location, the forward and reverse rates of the second reaction start to slowly increase until a peak in rates of production at the outer edge of the flame. The slight slope in the middle of the temperature profile is due to the heat release from the building forward and reverse rates of the second reaction.

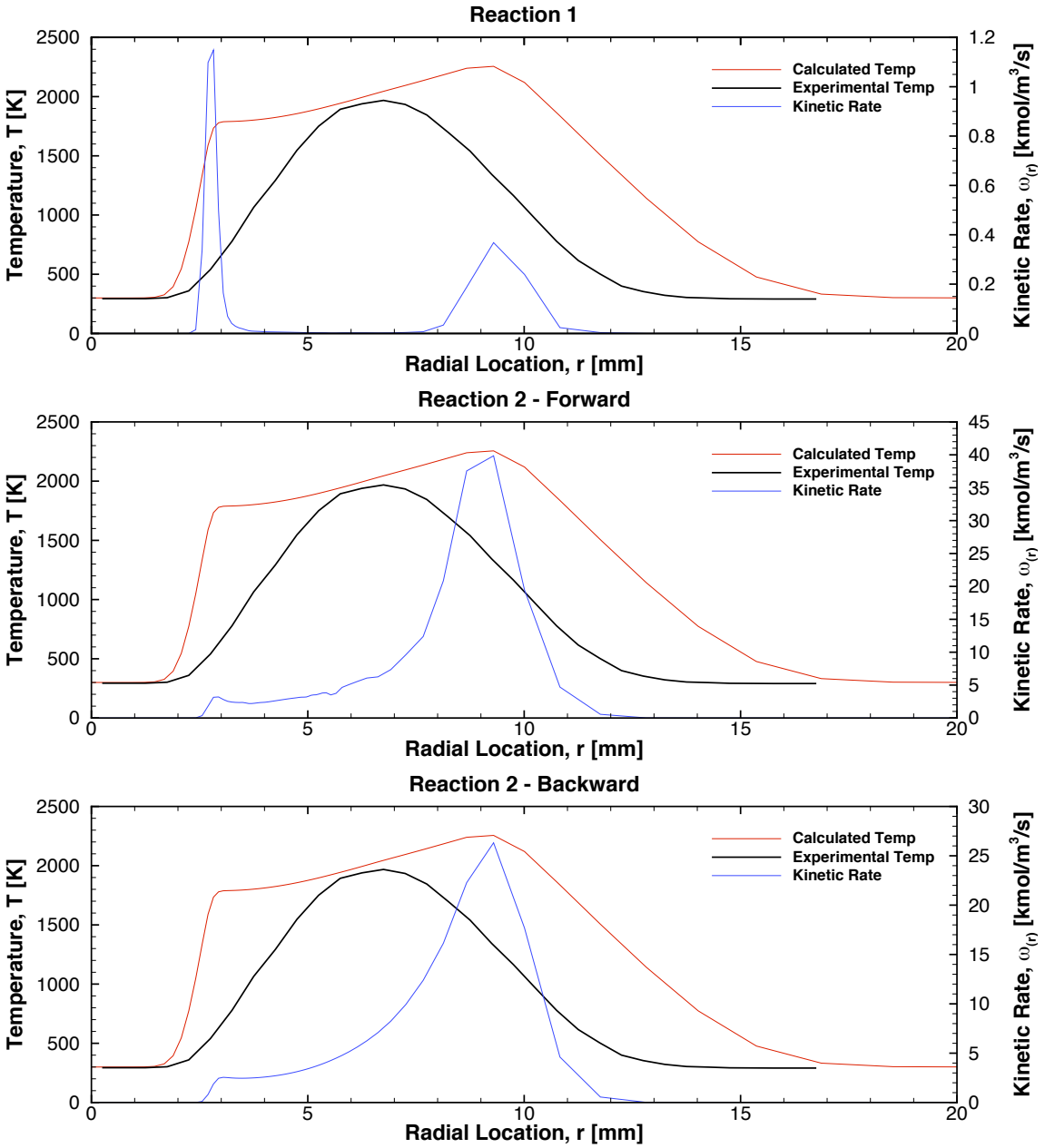

Figure 7.6: Plateaued temperature profile plotted next to kinetic rate for each reaction of Westbrook and Dryer two-step mechanism [1]. All data is captured at the 25 mm location.

7.2 DEI Violations

The DEI, as outlined in chapter 4, represents a local form of the second law of thermodynamics. For a CFD simulation of a reacting flow to be physically accurate it must satisfy the DEI at all points within the computational domain. However, as first noticed by Chambers [4, pp. 126-32], the DEI is violated at numerous locations for common reduced combustion mechanisms. No current CFD reacting flow solver implements the DEI as a solution limiter. Thus, combustion mechanisms should be used which automatically satisfy the DEI. The optimized two- and three-step mechanisms created in chapter 6 were made in accordance with a theorem for automatic satisfaction given by Slattery et al. [18].

In the following sections, solutions calculated using various combustion mechanisms are analyzed for violations of the DEI. The mechanisms used are the optimized two- and three-step mechanisms, the Westbrook and Dryer two-step mechanism, Jones Mech 2, and the detailed GRI 3.0 mechanism. All terms of the DEI are investigated separately to ensure that the necessary conditions for automatic satisfaction are being met.

7.2.1 First Term

The first term of the DEI represents entropy generation due to viscous stresses and is repeated here for convenience

$$-\mathrm{tr}\left[\left(\overline{\overline{\Pi}} + P\overline{\overline{I}}\right) \cdot \nabla \vec{v}\right].$$

For the Sandia flame A, since the domain is of constant pressure, the greatest viscous stresses will be encountered along the solid wall surfaces of the fuel tube and wind tunnel. Since the velocity of the fuel is much greater than the co-flow, and the

diameter of the fuel tube is considerably smaller than the tunnel, the largest velocity gradients are expected to be seen within the fuel tube. Therefore, Fig. 7.7 shows a contour plot of the first term of the DEI inside the fuel tube. The values of the first term are all negative, however, the distribution lends itself to a logarithmic scaling. Thus, Fig. 7.7 is actually a plot of the negative values of the first term.

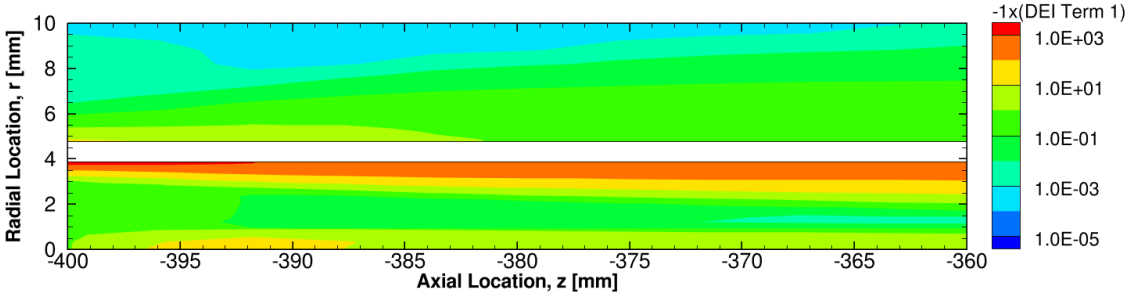

Figure 7.7: Typical profile for first term of the DEI. Calculated for the Westbrook and Dryer two-step mechanism [1]. Values of largest magnitude occurred along wall surfaces where shear stresses were the greatest.

The condition for automatic satisfaction of the first term is that the fluid behaves as a compressible Newtonian fluid. No non-linear viscosity relationships were used in the numerical model so the first term was automatically satisfied in the solutions for each of the five mechanisms. It is interesting to note that the values of greatest magnitude for the first term were found along the tube wall at the inlet boundary. This is due to the constant velocity profile that was used for this boundary condition. Boundary layers start to form along the inner wall of the tube which gives rise to large velocity gradients and viscous stresses at the velocity inlet. The large values of the first terms seen at this point are due to the boundary condition and are not completely physical. More reasonable values are found once the boundary layers have had time to develop.

7.2.2 Second Term

The second term of the DEI gives the entropy generation due to species diffusion and is written as

$$c\hat{R}T \sum_{n=1}^{N_s} \vec{J}_{(n)} \cdot \frac{\vec{d}_{(n)}}{\rho_{(n)}}.$$

The second term is primarily influenced by the temperature, gradients in composition, and the diffusion model used. The calculation of the diffusive mass flux vector, $\vec{J}_{(n)}$, is where the diffusion model makes an impact. For Fickian diffusion $\vec{J}_{(n)}$ is calculated using (4.15). However, if multicomponent and thermal diffusion coefficients are used, $\vec{J}_{(n)}$ is calculated using (4.16) and temperature gradients must also be considered.

Entropy production from the second term is expected to occur all throughout the flame where the temperature is elevated and composition gradients are large due to chemical reactions. Figure 7.8 shows a typical profile of the second term of the DEI. As in Fig. 7.7, this plot is logarithmically scaled so the negative of the values are plotted. The greatest rate entropy generation from species diffusion are found along the edges of the flame where reactions are occurring most rapidly. This is due to the large heat release and change in chemical composition at these locations.

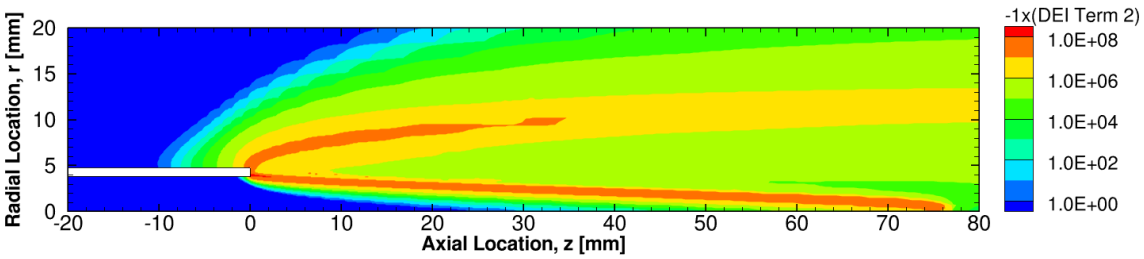

Figure 7.8: Typical profile for second term of the DEI. Calculated for the Westbrook and Dryer two-step mechanism [1]. The second term was heavily influenced by gradients in temperature and composition which were the most severe at the outer edges of the flame.

The necessary conditions for automatic satisfaction of the second term are for the fluid to be a mixture of dilute gases which obey Fick's Law. Using Fickian diffusion coefficients and calculating $\vec{J}_{(n)}$ with (4.15) resulted in negative values in all cells of the computation domain. Since all necessary conditions for automatic satisfaction were met by the model used, no positive cells should have been encountered. This result was a good verification that the diffusion coefficients were being calculated correctly.

A separate set of simulations were run using multicomponent and thermal diffusion. Multicomponent and thermal diffusion coefficients are a higher order model of diffusion which should provide more physical results. However, these models do not meet the necessary conditions for automatic satisfaction of the DEI. Therefore, it was expected that positive values would be found. Table 7.3 gives a summary of the cells which are positive for the second term for each simulation where multicomponent and thermal diffusion coefficients were used. Figure 7.9 shows a typical plot of where positive second term cells are found in these simulations.

Table 7.3: Positive 2^{nd} Term Values for Soret Diffusion Coefficients

Mechanism	Number of Cells	Maximum Value
GRI 3.0	6712	1.38×10^{-5}
Westbrook & Dryer	9792	1.36×10^{-5}
Jones Mech 2	9969	1.36×10^{-5}
Optimized 2-step	6880	1.36×10^{-5}
Optimized 3-step	4100	1.36×10^{-5}

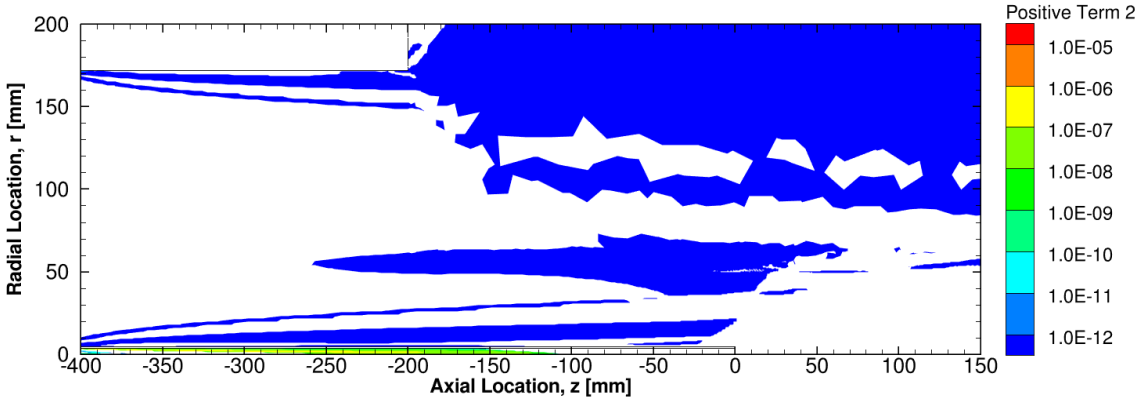

Figure 7.9: Typical profile of positive cells for the second term of the DEI in simulations using thermal diffusion coefficients. Calculated for the Westbrook and Dryer two-step mechanism [1]. Positive valued cells occur where temperature and composition gradients are insignificant and the domain is not perfectly isobaric.

Figure 7.9 shows that all positive cells were found well away from the reaction zone. In these areas the gradients in temperature and concentration were very small and entropy generation was negligible. Therefore, the small influence of the thermal diffusion coefficients allowed the second term to become slightly positive in these low gradient areas. The presence of positive values in the second term was due to the inclusion of the multicomponent and thermal diffusion coefficients. However, the

magnitude of the positive values was influenced by two additional sources.

First, since this was a numerical simulation, there were bound to be minor inaccuracies in the results. In areas where the gradients were small the error in the calculated gradients could be of the same order of magnitude as the gradients themselves leading to inaccuracies in the second term calculation. The bulk of the positive cells were found downstream of the wind tunnel edge. Figure 5.4 shows that, at that location, the mesh transitions from structured to unstructured and the aspect ratios of the cells experience a sharp transition. Numerically this was not ideal and errors could manifest in the solution. In Table 7.3 the number of cells and maximum value were approximately equal for all simulations. The reason was because the second term is related to species diffusion and not chemical reactions. Since the same diffusion model was used for each simulation, the results should have been the same. The differences in number of positive cells was also due to the slight numerical errors in the low gradient areas of the grid.

Second, the greatest magnitude positive values found when using multicomponent and thermal diffusion coefficients were located in the fuel tube. In the tube there was a slight pressure gradient due to the viscosity of the fuel flow. The calculation of the driving force for mass transfer vector, $\vec{d}$, assumed the flow to be isobaric. The small pressure gradient in the tube could have caused the calculated values to be slightly incorrect.

To quickly summarize the second term results: when Fickian diffusion coefficients were used, all cells automatically satisfied the second term of the DEI. However, where higher order multicomponent and thermal diffusion coefficients were used, slight positive values were found. While the positive values were due to the inclusion of the thermal diffusion coefficients, the magnitude of the positive values may have been influenced by the numerical grid and locally incorrect isobaric assumptions.

7.2.3 Third Term

The third term of the DEI is given by

$$\sum_{r=1}^{N_r} \sum_{n=1}^{N_s} \mu_{(n)} R_{(n)(r)},$$

and shows the entropy generation due to chemical reactions. The requirements for automatic satisfaction are that the fluid is a mixture of dilute gases and that all reactions in the mechanism are reversible and conform to the law of mass action. For reduced mechanisms, this is generally not the case, so it is expected that the third term could have positive values. Table 7.4 summarizes the values of the third term for each mechanism and Fig. 7.10 shows contour plots of the positive cells for each of the mechanisms.

Table 7.4: Positive 3^{rd} Term Values for All Mechanisms

Mechanism	Number of Cells	Maximum Value
GRI 3.0	6713	5.81×10^7
Westbrook & Dryer	23887	1.40×10^9
Jones Mech 2	12679	1.86×10^6
Optimized 2-step	3479	1.42×10^1
Optimized 3-step	0	-

The profiles of positive values for each of the mechanisms shown in Fig. 7.10 all share a similar structure. Positive values are found all throughout the center of the flame with sharp boundaries at the inside and outside edges. Positive values

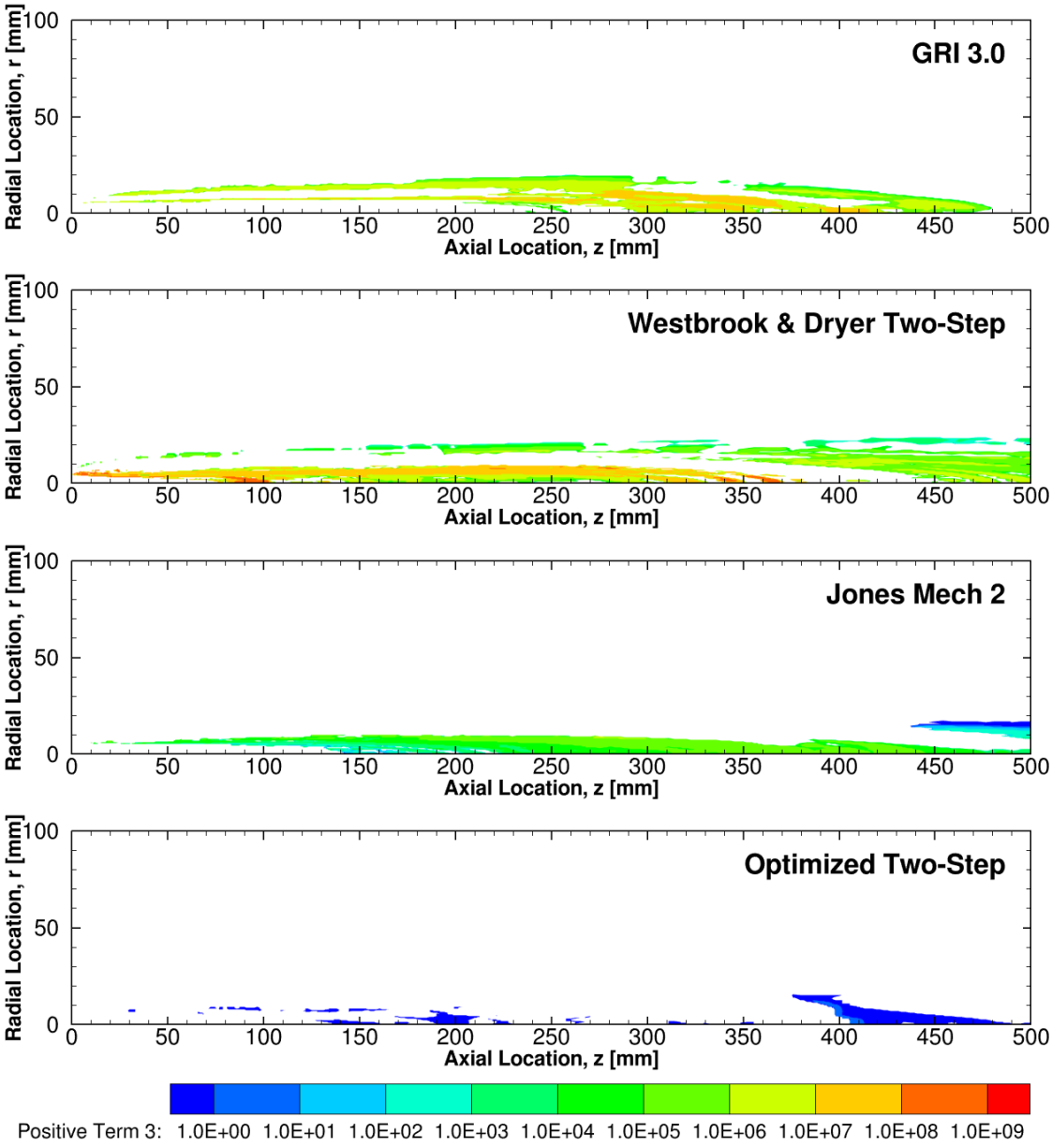

Figure 7.10: Profiles of positive third term values for various mechanisms. From top to bottom: GRI 3.0 [2], Westbrook and Dryer two-step [1], Jones Mech 2 [3], optimized two-step. The optimized three-step mechanism had no cells with positive third terms.

only occur where reactions are taking place and appear to be of largest magnitude near the central axis of the flame. The profile for the optimized two-step mechanism is noticeably more sparse than the other mechanisms. This is also reflected in the values from Table 7.4.

The Westbrook and Dryer two-step mechanism had the greatest number of positive cells with the largest positive values by two orders of magnitude. This was the expected result since the Westbrook and Dryer mechanism has only two reactions and the concentration exponents are not the stoichiometric coefficients. The optimized two- and three-step mechanisms had the least number of positive cells with the three-step mechanism having none at all. While the optimized two-step mechanism had some positive cells, the maximum value seen was five orders of magnitude smaller than any of the other mechanisms. This was an encouraging result since the optimized two- and three-step mechanisms were specifically created to minimize this value.

Unexpectedly, the detailed GRI 3.0 mechanism had a considerable number of positive third term cells. While positive values of the individual terms are allowed, as long as the sum total is negative, such large values for a single term are an indication that DEI violations may occur. One possible explanation for the positive cells is that the GRI 3.0 mechanism is not truly a complete mechanism. For the case tested here this is actually quite possible. FLUENT has an upper bound of 50 chemical species and the GRI 3.0 mechanism contains 53. Therefore, the minor species Ar, C_3H_7, and C_3H_8 were removed. These were chosen because they are either inert or related to propane combustion which was not found in flame A.

An incomplete mechanism could potentially cause the positive values seen, however, the magnitudes of the positive values for the optimized two- and three-step mechanisms, as well as Jones Mech 2, were considerably lower. All three of these

reduced mechanisms were clearly incomplete and had at least one non-reversible reaction. Therefore, another likely reason would be that the GRI 3.0 mechanism contains a few pressure-dependent reactions which do not meet the necessary conditions for automatic satisfaction of the third term. While only a small number of elementary reactions are pressure-dependent, those that are directly deal with the decomposition of the methane fuel and play a large role in the overall reaction.

Another cause of the positive third term values for the GRI 3.0 mechanism is that the rate parameters are only order-of-magnitude estimates for many of the elementary reactions which deal with minor species. The error in the rates which control the minor species, while individually small, could accumulate to a sizable value when all the reactions are considered. More testing is required to determine the leading cause of the non-satisfaction of the third term for the GRI 3.0 mechanism.

7.2.4 Fourth Term

The final term of the DEI shows the entropy generation due to heat transfer and is given by the expression

$$\frac{1}{T}\vec{\epsilon}\cdot\nabla T.$$

Since temperature gradients are what drive heat transfer, the greatest magnitude of entropy generation is anticipated to be found where the temperature gradients are the steepest. Figure 7.11 shows a typical contour of the fourth term calculated using the Westbrook and Dryer two-step mechanism. The greatest entropy generation is encountered along the inner surface of the flame. From Fig. 7.6 it is shown that, for this mechanism, the steepest temperature gradient is found in the same area, confirming the prediction. Along the outside edge of the flame the gradient is not as steep so the entropy generation is lessened.

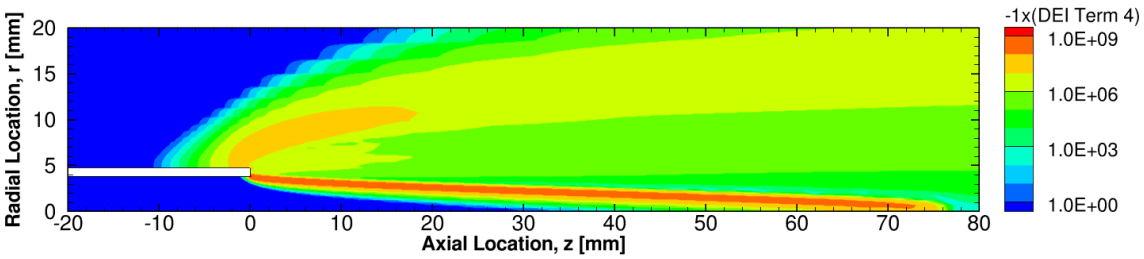

Figure 7.11: Typical profile for fourth term of the DEI. Calculated for the Westbrook and Dryer two-step mechanism [1]. The fourth term was of greatest magnitude where gradients were large and temperature was low. The most favorable conditions for this were at the inner edge of the flame.

The conditions for automatic satisfaction of the fourth term are for the fluid to obey Fourier's Law of heat conduction. The numerical model assumed this behavior for all species so no positive values were predicted. However, the thermal diffusion coefficients that caused the positive second term values are also used in the calculate of the energy flux vector, $\vec{\epsilon}$. The equation for the energy flux vector was (4.40). Since the thermal diffusion coefficients do not meet the conditions for automatic satisfaction, positive values could potentially be encountered.

In all simulations with the thermal diffusion coefficients turned off, each cell resulted in a negative value for the fourth term; all in accordance with the terms for automatic satisfaction. When the thermal diffusion coefficients were turned on, a few non-satisfying cells were found for each mechanism. A comparison of the number of non-satisfying cells, and magnitudes of the positive values, is shown in Table 7.5 for simulations using the thermal diffusion coefficients.

Table 7.5: Positive 4^{th} Term Values for Soret Diffusion Coefficients

Mechanism	Number of Cells	Maximum Value
GRI 3.0	830	1.26×10^{3}
Westbrook & Dryer	2	1.67×10^{1}
Jones Mech 2	13	3.29×10^{1}
Optimized 2-step	1	8.43×10^{-1}
Optimized 3-step	377	6.88×10^{1}

In each simulation using the thermal diffusion coefficients, the positive values for the fourth term are found along the inner edge of the flame as shown in Fig. 7.12. While some of the simulations only have one or two positive cells, they are always found at the forefront of the inner edge of the flame. Much like the second term, the positive values for the fourth term are all found in areas of low gradients where the slight effect of the thermal diffusion coefficients are most noticeable. The fourth term has an additional sensitivity to low temperatures due to the leading $1/T$ dependence which enhances the magnitude of the positive values of the fourth term.

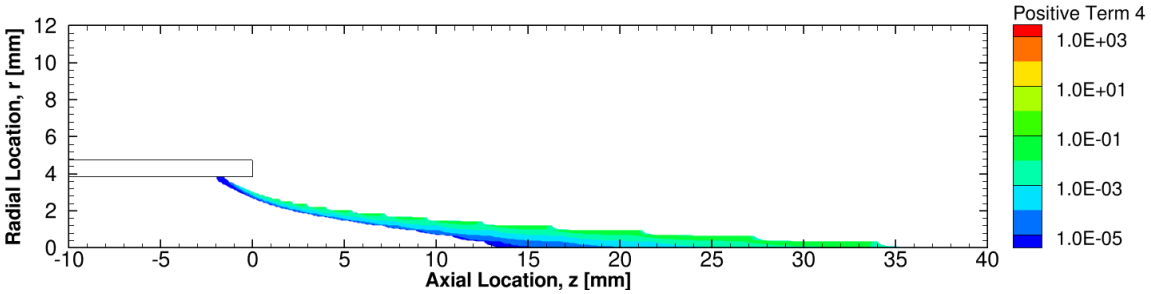

Figure 7.12: Typical profile for cells which did not automatically satisfy the fourth term of the DEI when using thermal diffusion coefficients. Calculated for the GRI 3.0 mechanism [2]. While some mechanisms had only a few non-automatically satisfying cells, they all occurred within this zone.

7.2.5 Entropy Violations

In the last few sections the distribution of values for each of the terms was explored. With the exception of the third term, when the necessary conditions for automatic satisfaction of each term were met were met, no positive values were found. For the second and fourth terms, a second set of simulations was run using thermal diffusion coefficients. Since the thermal diffusion coefficients did not automatically satisfy the DEI, positive values were found.

The method to fit Arrhenius parameters developed in chapter 6 was created with the intent of eliminating violations of the DEI due to third term non-satisfaction. To see the results of this most clearly, only simulations which satisfied all terms besides the third were included. Therefore, simulations using multicomponent and thermal diffusion coefficients are not presented in the following analysis.

The DEI was calculated by summing the contributions from each of the four terms at every grid point. Positive values of any term individually were acceptable as long as the sum total was negative. However, for three of the five mechanisms investigated, the non-satisfying values of the third term were too large to be balanced by the other terms and violations of the DEI were found. Table 7.6 outlines the violations of the DEI found for each chemical mechanism.

Table 7.6: Violations of the DEI for Various Mechanisms

Mechanism	Number of Cells	Volume Fraction [%]	Maximum Value
GRI 3.0	3513	3.98×10^{-3}	5.72×10^{7}
Westbrook & Dryer	20653	3.62×10^{-2}	1.40×10^{9}
Jones Mech 2	2075	1.15×10^{-3}	1.71×10^{6}
Optimized 2-step	0	0	-
Optimized 3-step	0	0	-

The optimized two- and three-step mechanisms performed exactly as they were intended, producing no violations of the DEI. Jones Mech 2 and the detailed GRI 3.0 mechanism both contained a small number of violating cells while, the Westbrook and Dryer two-step mechanism contained the most, and largest magnitude, violating cells. The Westbrook and Dryer mechanism was expected to produce the greatest number of violations since it met none of the criterion for automatic satisfaction of the third term.

Figure 7.13 shows contours of violations of the DEI for the three violating mechanisms. Since only the third term had non-automatically satisfying cells, the plots in Fig. 7.13 look remarkable similarity to those in Fig. 7.10, where the non-satisfaction of the third term is shown. Comparing the number of DEI violations in Table 7.6 to the number of positive third term cells in Table 7.4, it is seen that the number of DEI violations was less than the number of positive third term cells. This indication that some of the non-satisfying cells for the third term were balanced by the negative values of the other terms. However, the magnitude of the third term values were simply too great to be balanced in all cells.

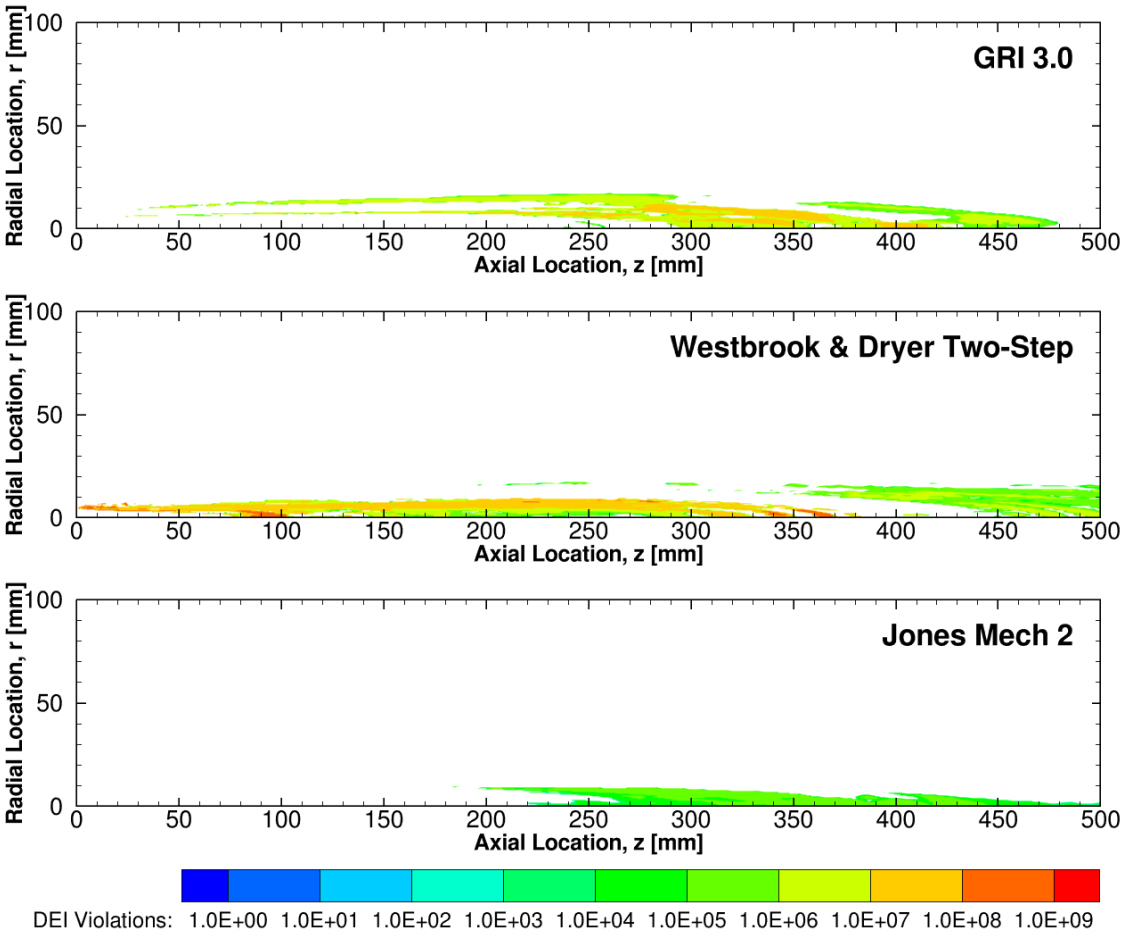

Figure 7.13: Profiles of entropy violations for various mechanisms. From top to bottom: GRI 3.0 [2], Westbrook and Dryer two-step [1], Jones Mech 2 [3]. The optimized two- and three-step mechanisms had no violating cells.

An interesting phenomenon is found in the contours of third term and DEI violations. For each of the mechanisms there is a channel, centrally located in the flame, where violations do not occur. The top pane of Fig. 7.14 shows the DEI violations for the Westbrook and Dryer mechanism where this channel is most prominently displayed. A bounding line has been drawn around the contours to show where the values transition from negative to positive. In the second and third panels this bounding line has been superimposed over mass fraction contours for CO_2 and H_2O.

Interestingly, the gap in the DEI violations line up exceedingly well with the areas where these product species at at their highest concentrations. Violations occur much more prevalently where reactions are occurring, or in other words, where the global reaction has not moved completely to product species. This could also support the theory that violations of the GRI 3.0 mechanism are due to the inaccurate rate parameters for the minor species reactions. However, more analysis is required to verify this claim.

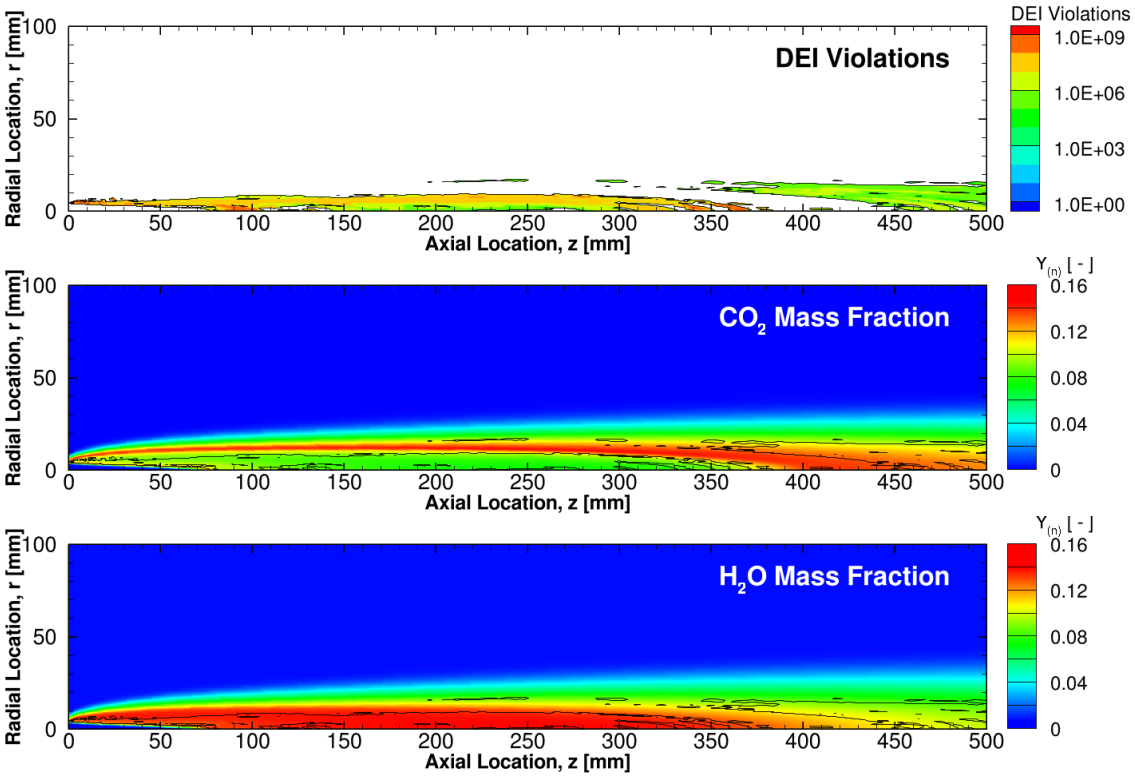

Figure 7.14: For the characteristic case of the Westbrook and Dryer two-step mechanism [1], the outline of the entropy violations was superimposed over contour plots of the major species product species CO_2 and H_2O to demonstrate that the gap in DEI violations occurred where the major species were at their greatest mass fractions.

The most surprising result from this study was that the detailed GRI 3.0 mech-

anism still produced violations of the DEI. Equally intriguing was that the GRI 3.0 mechanism was used as the optimization target for creating the optimized two- and three-step mechanisms; both of which produced no violations. Previously discussed were the possibilities that the violations from the GRI 3.0 mechanism could be due to an incomplete combustion mechanism, pressure-dependent elementary reaction rates, or inaccurate rate parameters. Another possible cause could be that the numerical solution scheme imparted these violations. In previous studies where the Stiff Chemistry Solver was not used, Jones Mech 2 produced no violations of the DEI [3, p. 113]. In the current study, using the Stiff Chemistry Solver, Jones Mech 2 was found to produce 2075 violating cells. While this was still a very low number of violations, it was on par with the 3513 violations of the GRI 3.0 mechanism. Unfortunately, it was impossible to produce a converged solution of the GRI 3.0 mechanism without using the Stiff Chemistry Solver. Therefore, a more rigorous study is needed to determine the influence of the numerical scheme on the violations of the DEI, and to ascertain the root cause of violations when using the detailed GRI 3.0 mechanism.

7.3 Summary

The structure of the Sandia flame A predicted by the detailed GRI 3.0 mechanism was found to match the experimental results quite well. The only exception was that the flame width was over-predicted by approximately 25%. This incorrect width could be due to the co-flow velocity used. Further investigation is required to determine the actual co-flow velocity from the Sandia experiments. The reduced mechanisms did a poor job of matching the experimental data due to excessively high upper flammability limits. The premixed fuel supply, while outside the flammability limits of actual methane fuel, was well within the acceptable range of the reduced mechanisms. Therefore, premature ignition was experienced which caused the inner

surface of the flame to be artificially advanced towards the fuel tube. Once the initial oxygen supply was depleted, the reduced mechanism simulations exhibited similar responses to the GRI 3.0 mechanism since the flame is diffusion rate limited. The plateaued temperature profile of the reduced mechanisms was shown to be due to a lack of competing chemical reactions.

The DEI UDF was run for the simulations of each chemical mechanism. Non-automatic satisfaction was limited to only the third term, as long as Fickian diffusion was used. Positive values for the second and fourth terms were found when using Soret thermal diffusion coefficients. Soret diffusion accounts for counter gradient effects and does not meet the conditions for automatic satisfaction of the second and fourth terms. No violations of the DEI were found for the optimized two- and three-step mechanisms, just as they were designed to perform. Surprisingly, violations were found for the detailed GRI 3.0 mechanism. Potential sources could include an incomplete set of reactions, the inclusion of pressure-dependent elementary reactions, inaccurate rate parameters, or even the numerical scheme used. More testing is required to determine the leading cause.

8. CONCLUSIONS

Simulating the flow of reacting fluids is an exceptionally difficult problem which incorporates the studies of fluid dynamics, thermodynamics, chemistry, and numerical techniques. In many cases, the computational demand to fully simulate a reacting flow is prohibitive, even for a modern supercomputer. Therefore, it is necessary to simplify the numerical models so that results can be obtained in a reasonable amount of time.

Combustion mechanisms are one area in which considerable solution speed up can be obtained through simplifying assumptions. For the combustion of methane, the most simple hydrocarbon, a complete combustion mechanism contains 53 chemical species and 325 reversible elementary reactions. In a typical simulation, most of these species and reactions only play a very minor role. Since solution time scales with the number of species, there is the potential for substantial speed up if all of the minor species were to be removed. There is an entire field of combustion research dedicated to the creation of reduced chemical mechanisms.

One issue with the use of reduced chemical mechanisms is that the simplifying assumptions used in their creation ignore the laws of thermodynamics. To fully trust the results of a reduced mechanism simulation, it becomes necessary to impose thermodynamic limits on the solution. The differential entropy inequality (DEI) represents a local form of the second law of thermodynamics which lends itself as a natural limiter for flow solutions. It consists of four terms which individually represent the entropy generation in a flow due to viscous stresses, species diffusion, reactions, and heat transfer. Unfortunately, the DEI is never used in modern flow solvers. Therefore, efforts must be taken to ensure that the reduced mechanisms used

automatically satisfy the DEI so that it does not need to be directly implemented.

A new method of determining rate parameters for reduced mechanisms was developed for this thesis. It was used to create two reduced mechanisms which produced no violations of the DEI for a sample simulation. The process involved using the species production rates of a detailed mechanism simulation to estimate the reaction rates of a reduced mechanism. A non-linear least squares curve fitting routine was then performed to find the Arrhenius rate parameters which best reproduce the estimate reaction rates. This method was inspired by the work of Jones [3]. The notable divergence from Jones' method was the initial estimation of the reduced mechanism reaction rates which allowed for the reactions be be decoupled. This had many benefits, including: less user input and intuition required, decreased solution time, and the ability to preferentially weight individual species. The resulting mechanisms were also shown to produce a more accurate estimation of the flame speed. Finally, the new method produced mechanisms which were able to capture both the creation and destruction of minor species where only creation could be modeled before. The two reduced mechanisms created using this process were named the optimized two- and three-step mechanisms.

To test the newly created mechanisms for violations of the DEI, two steps were taken. First, a program was created which could calculate the value of the DEI for any mechanism. Since the CFD program FLUENT was used for all flow solutions, the DEI was implemented as a user defined function (UDF) to post-process the results. Fortunately, a similar UDF had already been created by Chambers [4] which could be modified to fit the current purpose. The original UDF was hard-coded for one specific reduced mechanism so considerable effort was expended generalizing it for any mechanism or numerical options. The only part that had to remain hard-coded was the calculation of the binary diffusion coefficients since FLUENT had no

functionality to extract these values.

The second task was to create a flow domain and choose numerical options for calculating the solution. Experimental data was available for the laminar Sandia flame A, so a mesh was created to models this domain. A grid convergence study was carried out to ensure that the mesh produced results which were grid independent. A new numerical scheme was implemented to allow for the detailed GRI 3.0 mechanism [2] to be simulated. The new scheme was a Stiff Chemistry Solver which helped with the large disparity between the time scales of the flow and of the reaction rates. An additional benefit of the Stiff Chemistry Solver was that it removed a numerical instability in the solution which eliminated the need for a temperature limiter to keep the solution bounded.

Five chemical mechanisms were applied to the Sandia flame A simulation. These included the optimized two- and three-step mechanisms, Jones Mech 2 [3, p. 103], the Westbrook and Dryer two-step mechanism [1], and the detailed GRI 3.0 mechanism [2]. The reduced mechanisms were found to prematurely ignite the premixed fuel flow since they incorrectly model the flammability limits of the methane fuel. This causes substantial differences from the experimental data on the inside edge of the flame. The GRI 3.0 mechanism modeled the inside edge of the flame very well. All five mechanisms show close agreement on the outside edge of the flame, however, none are close to the experimental values. The flame width error could be due to the co-flow velocity differing from the experimental value. Further investigation is required to determine the correct co-flow velocity. The simulations of all mechanisms had similar profiles at the outside edge since the flame was diffusion rate limited. For diffusion flames the reaction mechanism is much less important.

Looking at the individual terms of the DEI, positive values were found for terms two, three, and four. The positive values for the second and fourth terms were due

to the inclusion of thermal diffusion coefficients which did not satisfy the conditions for automatic satisfaction of the DEI. Removing the thermal diffusion coefficients eliminated any positive cells for the second and fourth terms. Only simulations without thermal diffusion coefficients were used to investigate DEI violations.

No violations of the DEI were found for the optimized two- and three-step mechanisms. However, violations were found in the Westbrook and Dryer two-step mechanism, Jones Mech 2, and the detailed GRI 3.0 mechanism. For the reduced mechanisms violations were expected since they did not meet the terms for automatic satisfaction. Since the detailed GRI 3.0 mechanism was considered complete, no violations were anticipated. It was postulated that violations of the DEI for the GRI 3.0 mechanisms were due to missing species from the mechanism used, pressure-dependent elementary reactions, the Arrhenius rate coefficients being incorrect, or the numerical scheme imparting errors.

Potential topics for future study would be to quantify the root cause of the GRI 3.0 mechanism DEI violations. Additionally, extra work could be done to improve the method for fitting Arrhenius parameters to reduced mechanisms. A weighting scheme could be implemented to improve the resolution of minor species in the results. The method to estimate the reduced reaction rates from the detailed simulation could benefit from a positively bounded algorithm to aid in the estimation of the reverse rates. Also, a more efficient method could be implemented to reduce the time it takes to find an initial guess for the non-linear least squares method. Finally, the Sandia flame A is the only case that the optimized two- and three-step mechanisms have been applied to. More simulations are necessary to prove that they produce no violations of the DEI for any combustion simulation.

REFERENCES

[1] C. K. Westbrook and F. L. Dryer, "Simplified Reaction Mechanisms for the Oxidation of Hydrocarbon Fuels in Flames," *Combustion Science and Technology*, vol. 27, pp. 31–43, 1981.

[2] G. P. Smith, D. M. Golden, M. Frenklach, B. Eiteener, M. Goldenberg, C. T. Bowman, R. K. Hanson, W. C. Gardiner, V. V. Lissianski, and Z. W. Qin, "GRI-Mech 3.0," 2000. http://combustion.berkeley.edu/gri-mech/.

[3] N. H. Jones, "The Importance of the Entropy Inequality on Numerical Simulations Using Reduced Methane-Air Reaction Mechanisms," Master's thesis, Texas A&M University, August 2012.

[4] S. B. Chambers, "Investigation of Combustive Flows and Dynamic Meshing in Computational Fluid Dynamics," Master's thesis, Texas A&M University, December 2004.

[5] A. A. Adamczyk and G. A. Lavoie, "Laminar Head-On Flame Quenching a Theoretical Study," tech. rep., SAE Technical Paper, 1978.

[6] F. Dryer and I. Glassman, "High-Temperature Oxidation of CO and CH_4," in *Symposium (International) on combustion*, vol. 14, pp. 987–1003, Elsevier, 1973.

[7] ANSYS Fluent, Release 15.0, ANSYS Inc., Nov. 2013.

[8] N. Peters, "Numerical and Asymptotic Analysis of Systematically Reduced Reaction Schemes for Hydrocarbon Flames," in *Numerical simulation of combustion phenomena*, pp. 90–109, Springer, 1985.

[9] S. Lam, "Using CSP to Understand Complex Chemical Kinetics," *Combustion Science and Technology*, vol. 89, no. 5-6, pp. 375–404, 1993.

[10] N. Peters and F. A. Williams, "The Asymptotic Structure of Stoichiometric

Methane-Air Flames," *Combustion and Flame*, vol. 68, pp. 185–207, 1987.

[11] R. W. Bilger, S. H. Starner, and R. J. Kee, "On Reduced Mechanisms for Methane-Air Combustion in Nonpremixed Flames," *Combustion and Flame*, vol. 80, pp. 135–149, 1990.

[12] K. Seshadri, X. S. Bai, H. Pitsch, and N. Peters, "Asymptotic Analysis of the Structure of Moderately Rich Methane-Air Flames," *Combustion and Flame*, vol. 113, pp. 589–602, 1998.

[13] K. Seshadri, X. S. Bai, and H. Pitsch, "Asymptotic Structure of Rich Methane-Air Flames," *Combustion and Flame*, vol. 127, pp. 2265–2277, 2002.

[14] S. Lam and D. Goussis, "The CSP method for simplifying kinetics," *International Journal of Chemical Kinetics*, vol. 26, no. 4, pp. 461–486, 1994.

[15] U. Maas and S. B. Pope, "Simplifying Chemical Kinetics: Intrinsic Low-Dimensional Manifolds in Composition Space," *Combustion and flame*, vol. 88, no. 3, pp. 239–264, 1992.

[16] M. R. Roussel and T. Tang, "The Functional Equation Truncation Method for Approximating Slow Invariant Manifolds: A Rapid Method for Computing Intrinsic Low-Dimensional Manifolds," *The Journal of chemical physics*, vol. 125, no. 21, 2006.

[17] S. S. Girimaji and A. A. Ibrahim, "Advanced Quasi-Steady State Approximation for Chemical Kinetics," *Journal of Fluids Engineering*, vol. 136, no. 3, 2014.

[18] J. C. Slattery, P. G. A. Cizmas, A. N. Karpetis, and S. B. Chambers, "Role of Differential Entropy Inequality in Chemically Reacting Flows," *Chemical Engineering Science*, vol. 66, pp. 5236–5243, 2011.

[19] N. H. Jones, P. G. Cizmas, and J. C. Slattery, "Creating Reduced Kinetics Models that Satisfy the Entropy Inequality," *Journal of Engineering for Gas Turbines and Power*, vol. 137, no. 7, 2015.

[20] R. Aris, *Vectors, Tensors and the Basic Equations of Fluid Mechanics*. Courier Corporation, 2012.

[21] J. D. Anderson, *Modern Compressible Flow: With Historical Perspective*. McGraw Hill Higher Education, 2nd ed., 1990.

[22] F. M. White and I. Corfield, *Viscous Fluid Flow*. McGraw-Hill New York, 3rd ed., 2006.

[23] ANSYS Inc., *ANSYS Fluent User's Guide*, 15.0 ed., 2013.

[24] R. C. Reid, J. M. Prausnitz, and B. E. Poling, *The Properties of Gases and Liquids*. New York: McGraw-Hill Inc., 4th ed., 1987.

[25] ANSYS Inc., *ANSYS Fluent Theory Guide*, 15.0 ed., 2013.

[26] B. J. McBride, S. Gordon, and M. A. Reno, "Coefficients for Calculating Thermodynamic and Transport Properties of Individual Species," 1993.

[27] R. Kee, F. Rupley, J. Miller, M. Coltrin, J. Grcar, E. Meeks, H. Moffat, A. Lutz, G. Dixon-Lewis, M. Smooke, *et al.*, "CHEMKIN Collection, Release 3.6, Reaction Design," *Inc., San Diego, CA*, 2000.

[28] F. J. Kelecy, "Coupling Momentum and Continuity Increases CFD Robustness," *ANSYS Advantage*, vol. 2, 2008.

[29] F. A. Williams, *Combustion Theory*. The Benjamin/Cummings Publishing Company, Inc., 2nd ed., 1985.

[30] J. C. Slattery, *Advanced Transport Phenomena*. Cambridge University Press, Cambridge, UK, 1999.

[31] J. M. Prausnitz, R. N. Lichtenthaler, and E. G. de Azevedo, *Molecular Thermodynamics of Fluid-Phase Equilibria*. Pearson Education, 1998.

[32] ANSYS Inc., *ANSYS Fluent UDF Manual*, 15.0 ed., 2013.

[33] R. Barlow and J. Frank, "Effects of Turbulence on Species Mass Fractions in Methane/Air Jet Flames," in *Symposium (International) on Combustion*,

vol. 27, pp. 1087–1095, Elsevier, 1998.

[34] R. Barlow, J. Frank, A. Karpetis, and J.-Y. Chen, "Piloted Methane/Air Jet Flames: Transport Effects and Aspects of Scalar Structure," *Combustion and flame*, vol. 143, no. 4, pp. 433–449, 2005.

[35] Sandia National Laboratories, "International Workshop on Measurement and Computation of Turbulent and Nonpremixed Flames," 2014, http://www.sandia.gov/TNF/abstract.

[36] A. N. Karpetis, T. B. Settersten, R. W. Schefer, and R. S. Barlow, "Laser Imaging System for Determination of Three-Dimensional Scalar Gradients in Turbulent Flames," *Optics letters*, vol. 29, no. 4, pp. 355–357, 2004.

[37] J. E. Dennis Jr and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, vol. 16. SIAM, 1996.

[38] A. Björck, *Numerical Methods for Least Squares Problems*. SIAM, 1996.

[39] The MathWorks, Inc., "lsqcurvefit", 2015, http://www.mathworks.com/help/optim/ug/lsqcurvefit.html.

[40] K. Levenberg, "A Method for the Solution of Certain Problems in Least Squares," *Quarterly of applied mathematics*, vol. 2, pp. 164–168, 1944.

[41] D. W. Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," *Journal of the Society for Industrial & Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[42] The MathWorks, Inc., "Equation Solving Algorithms", 2015, http://www.mathworks.com/help/optim/ug/equation-solving-algorithms.html.

[43] J. J. Moré and D. C. Sorensen, "Computing a Trust Region Step," *SIAM Journal on Scientific and Statistical Computing*, vol. 4, no. 3, pp. 553–572, 1983.

[44] M. A. Branch, T. F. Coleman, and Y. Li, "A Subspace, Interior, and Conjugate Gradient Method for Large-Scale Bound-Constrained Minimization Problems,"

SIAM Journal on Scientific Computing, vol. 21, no. 1, pp. 1–23, 1999.

[45] D. Goodwin, "Cantera: An object-oriented software toolkit for chemical kinetics, thermodynamics, and transport processes," 2013. http://www.cantera.org/docs/sphinx/html/index.html.

APPENDIX A

ADDITIONAL PROOFS

A.1 DEI Second Term Alternate Form

The second term of the DEI is given by

$$c\hat{R}T\sum_{n=1}^{N_s}\vec{J}_{(n)}\cdot\frac{\vec{d}_{(n)}}{\rho_{(n)}},$$

and can be rearranged as

$$c\hat{R}T\left[\vec{J}_{(N_s)}\cdot\frac{\vec{d}_{(N_s)}}{\rho_{(N_s)}}+\sum_{n=1}^{N_s-1}\vec{J}_{(n)}\cdot\frac{\vec{d}_{(n)}}{\rho_{(n)}}\right].$$

The sum of the diffusive mass flux vectors, as given in (4.17), is

$$\sum_{n=1}^{N_s}\vec{J}_{(n)}=0.$$

Solving for the diffusive mass flux vector of the final species gives

$$\vec{J}_{(N_s)}=-\sum_{n=1}^{N_s-1}\vec{J}_{(n)}.$$

Inserting into the rearranged form of the DEI yields

$$c\hat{R}T\left[-\sum_{n=1}^{N_s-1}\vec{J}_{(n)}\cdot\frac{\vec{d}_{(N_s)}}{\rho_{(N_s)}}+\sum_{n=1}^{N_s-1}\vec{J}_{(n)}\cdot\frac{\vec{d}_{(n)}}{\rho_{(n)}}\right].$$

Finally, the alternate form of the second term of the DEI can be found by simplification

$$c\hat{R}T \sum_{n=1}^{N_s-1} \vec{J}_{(n)} \cdot \left(\frac{\vec{d}_{(n)}}{\rho_{(n)}} - \frac{\vec{d}_{(N_s)}}{\rho_{(N_s)}} \right).$$

A.2 Linear Least Squares Function Simplification

In least squares curve fitting, when the function to fit to the data set is linear in the parameters it can be rewritten as

$$f\left(x_i, \vec{\eta}\right) = \sum_{j=1}^{N_\eta} g_j\left(x_i\right) \eta_j, \quad i = 1, \ldots, N_p.$$

Here $g_j\left(x_i\right)$ are functions of only the independent variable. Recall from (6.5) that the components of the Jacobian are the partial derivatives of f with respect to the parameters. For the function linear in the parameters this yields

$$J_{ij} = \frac{\partial f\left(x_i, \vec{\eta}\right)}{\partial \eta_j} = g_j\left(x_i\right), \qquad \begin{matrix} i = 1, \ldots, N_p \\ j = 1, \ldots, N_\eta \end{matrix}.$$

Back substitution results in the final form

$$f\left(x_i, \vec{\eta}\right) = \sum_{j=1}^{N_\eta} J_{ij} \eta_j, \quad i = 1, \ldots, N_p.$$

APPENDIX B

CODES

B.1 Python Script for Cantera Free Flame Simulation

```
import cantera as ct

# Simulation parameters
p = ct.one_atm  # pressure [Pa]
Tin = 300.0  # unburned gas temperature [K]
reactants = 'CH4:1, O2:2, N2:7.52'  # premixed gas composition

initial_grid = [0.0, 0.001, 0.01, 0.02, 0.029, 0.03]  # m
tol_ss = [1.0e-5, 1.0e-13]  # [rtol atol] for steady-state problem
tol_ts = [1.0e-4, 1.0e-13]  # [rtol atol] for time stepping
loglevel = 1  # amount of diagnostic output (0 to 8)
refine_grid = True  # 'True' to enable refinement, 'False' to disable

# IdealGasMix object used to compute mixture properties
gas = ct.Solution('gri30.xml', 'gri30_mix')
gas.TPX = Tin, p, reactants

# Outputs of interest
nsp = gas.n_species
print 'Number of Species'
print nsp

nre = gas.n_reactions
print 'Number of Reactions'
print nre

eqn = gas.reaction_equations()
print 'Reaction Equations'
print eqn

# Flame object
f = ct.FreeFlame(gas, initial_grid)
f.flame.set_steady_tolerances(default=tol_ss)
f.flame.set_transient_tolerances(default=tol_ts)
```

```
# Set properties of the upstream fuel-air mixture
f.inlet.T = Tin
f.inlet.X = reactants

f.show_solution()

# Solve with the energy equation disabled
f.energy_enabled = False
f.set_max_jac_age(10, 10)
f.set_time_step(1e-5, [2, 5, 10, 20])
f.solve(loglevel=loglevel, refine_grid=False)
f.save('ch4_adiabatic_GRI.xml', 'no_energy',
       'solution with the energy equation disabled')

# Solve with the energy equation enabled
f.set_refine_criteria(ratio=3, slope=0.06, curve=0.12)
f.energy_enabled = True
f.solve(loglevel=loglevel, refine_grid=refine_grid)
f.save('ch4_adiabatic_GRI.xml', 'energy',
       'solution with mixture-averaged transport')
f.show_solution()
print 'mixture-averaged flamespeed = {0:7f} m/s'.format(f.u[0])

# write the velocity, temperature, density,
# and mole fractions to a CSV file
f.write_csv('ch4_adiabatic_GRI_mix.csv', quiet=False)

# Solve with multi-component transport properties
f.transport_model = 'Multi'
f.solve(loglevel, refine_grid)
f.show_solution()
print 'multicomponent flamespeed = {0:7f} m/s'.format(f.u[0])
f.save('ch4_adiabatic_GRI.xml','energy_multi',
       'solution with multicomponent transport')

# write the velocity, temperature, density,
# and mole fractions to a CSV file
f.write_csv('ch4_adiabatic_GRI_multi.csv', quiet=False)
```