To the University of Wyoming:

The members of the Committee approve the dissertation of Brian A. Lockwood presented on April 10, 2012.

Dimitri J. Mavriplis, Chairperson

Fred Furtado, External Department Member

Jay Sitaraman

Andrew C. Hansen

Jonathan W. Naughton

Mihai Anitescu

APPROVED:

Paul A. Dellenback, Head, Mechanical Engineering

Rob Ettema, Dean, College of Engineering and Applied Science

Lockwood, Brian A., Gradient-based Approaches for Sensitivity Analysis and Uncertainty Quantification within Hypersonic Flows, Ph.D., Mechanical Engineering, May, 2012.

With the proliferation of simulation within the design and analysis of engineering systems, uncertainty quantification and sensitivity analysis have taken on increased importance, providing valuable information for assessing the reliability of simulation outputs and a means for improving these results. In this work, uncertainty quantification and sensitivity analysis within the context of hypersonic computational fluid dynamics is examined. The simulation of hypersonic relies on numerous constitutive relations to account for chemical reactions, internal energy modes and molecular transport. Within these constitutive relations are hundreds of constants and parameters, which are often the result of experimental measurements. The goal of sensitivity analysis is determining the simulation parameters most affecting an output of interest, while the goal of uncertainty quantification is determining the variability of simulation outputs resulting from the uncertainty associated with model parameters. Traditional methods for uncertainty quantification and sensitivity analysis typically rely on exhaustive sampling, where hundreds to thousands of simulations are performed and relevant statistics are computed. For complex simulations, these exhaustive approaches are prohibitively expensive and well beyond the computational budget of most projects.

For this work, gradient-based methods are used to reduce the expense of uncertainty quantification and sensitivity analysis. Using an adjoint-based approach, the derivative of an output with respect to simulation parameters can be computed in a constant amount of work, providing more information about the simulation output without a significant increase in cost. This additional information can then be leveraged in novel ways, such as surrogate models or optimization, to accelerate the process of uncertainty quantification or sensitivity analysis. This dissertation demonstrates these gradient-based methods for sensitivity analysis and uncertainty quantification in hypersonic flow simulations and assesses the performance of these methods in terms of cost and accuracy.

GRADIENT-BASED APPROACHES FOR SENSITIVITY ANALYSIS AND UNCERTAINTY QUANTIFICATION WITHIN HYPERSONIC FLOWS

by

Brian A. Lockwood, B.S.M.E & M.S.N.E

A dissertation submitted to the
Mechanical Engineering
and the
University of Wyoming
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY
in
MECHANICAL ENGINEERING

Laramie, Wyoming
May 2012

This dissertation is dedicated to my grandfather, the late Arthur ”Bunky” Lowing, for leading me to a career in engineering and his guidance in life.

Contents

List of Figures

List of Tables

Acknowledgments

First, I would like to acknowledge the financial support of the Computational Science Graduate Fellowship administered by the Krell Institute under DOE Grant number DE-FG02-97ER25308. Additionally, I would like to thank my advisor, Dimitri Mavriplis, for his support during my studies. I would also like to acknowledge the contributions of my collaborators: Mihai Anitescu, Markus Rumpfkeil, Wataru Yamazaki, Karthik Mani, Nicholas Burgess, and Bryan Flynt. Finally, I would like to express my gratitude to the members of my committee for reviewing this dissertation and their feedback regarding my work.

Brian A. Lockwood

University of Wyoming
May 2012

Chapter 1

Introduction

Over the past decades, the increasing power of computational hardware has led to the expanded capability of computational modeling and an increased reliance on simulation for the design and analysis of complex engineering systems. This increased reliance has led to a desire to improve the efficiency of simulation, allowing for an acceleration of the design process and improved analysis capability. In general, this increased efficiency is achieved by increasing the ratio of simulation utility to the amount of work required to arrive at a result. In the field of computational fluid dynamics (CFD), improving this efficiency has taken the form of decreasing the time to solution or improving the value of the result without a corresponding increase in cost. This increased value can come from expanding the scope of the simulation, improved solution accuracy, or providing additional information about the simulation result. One means of providing this additional information is through sensitivity analysis (SA) and uncertainty quantification (UQ). The focus of this work are methods for accelerating the process of sensitivity analysis and uncertainty quantification within engineering CFD simulations, particularly the simulation of hypersonic flows.

For this work, engineering simulation is defined as simulation that seeks to characterize a physical system for the purposes of design or ensuring desired performance. The opposite of this type of simulation is scientific simulation, the purpose of which is to gain insight or knowledge into the underlying physics governing a system. Engineering simulations are typified by a large number of input parameters defining the system and relevant physics and the results typically consist of a relatively small number of design metrics used to characterize the performance of the system. Given this nomenclature, the purpose of sensitivity analysis is to determine the effect of changes to input parameters on the simulation outputs. The purpose of uncertainty quantification is to determine the uncertainty of simulation outputs due to the uncertainty within the simulation, either due to uncertainty associated with defining the system or inherent randomness within the simulation itself. Using the information gained through sensitivity analysis and uncertainty quantification, the quality of the simulation result can be assessed and a means for improving the simulation is provided. Additionally, the confidence level of the result can be quantified, allowing for further utilization of simulation within the design process.

The additional information provided by uncertainty quantification and sensitivity analysis is especially important for situations in which real world data is difficult or impossible to obtain. A prime example of this situation is hypersonic flow. The simulation of hypersonic flow is characterized by high velocities, leading to strong shocks, high temperatures and pressures, the excitation of internal energy modes and often chemical reactions. To simulate these physical phenomena, a number of constitutive relations and empirical models are required, each of which contain a large number of often experimentally derived inputs. For engineering systems featuring hypersonic fluid flow, such as atmospheric re-entry of spacecraft, acquiring experimental data is costly and, in some cases, impossible, leading to a heavy reliance on simulation for the design of these systems and for ensuring the proper performance. Because of this heavy reliance, uncertainty quantification and sensitivity analysis are vital tools for the simulation of hypersonic flow. Using uncertainty quantification, the quality of simulation results can be assessed and the reliability of the system can be characterized in a probabilistic sense. Using sensitivity analysis, the parameters within the simulation most affecting the result are identified, providing a means for improving the accuracy and precision of simulation results. Additionally, sensitivity analysis can provide a basis for optimizing the performance of the system.

Traditional methods for sensitivity analysis and uncertainty quantification are often expensive and heavily reliant on user experience. Methods, such as parameter studies [1], Monte Carlo sampling [2] and finite differences [3], have all been used in the past to assess the quality of simulation results and determine the optimal results. Although non-intrusive and easily applied, these methods are extremely time consuming and require significant manual input. In this work, intrusive gradient-based methods are explored to reduce the cost associated with uncertainty quantification and sensitivity analysis, enabling the use of this information throughout the design process.

1.1 Background

With the motivation for incorporating rapid uncertainty quantification and sensitivity analysis into the simulation of hypersonic flow established, previous work that has advanced this goal is given.

1.1.1 Sensitivity Analysis

Two approaches to sensitivity analysis are possible with in the context of computational fluid dynamics, local and global sensitivity analysis. For the local approach, the sensitivity represents the change in the objective due to an infinitesimal change in the design variable, while the global sensitivity represents the average change in the objective due to variations of the design variables over their possible values. For local analysis, the sensitivity represents the derivative of the output with respect to the input parameters of the simulation. In the past, these derivative values have been used to determine the relative importance of parameters within the code as well as a basis for gradient-based numerical optimization. This gradient can be calculated directly by differentiation of the analysis code or can be approximated via finite-differences. When computed analytically, an adjoint approach is typically used within the field of CFD. Historically, this adjoint approach has been used for the purposes of aerodynamic design via optimization. For fluid mechanics, the use of the adjoint was first proposed by Pironneau in 1974 [4]. This work was later extended to design optimization by Jameson in 1988 [5]. Using an adjoint approach, the derivative of a single output with respect to any number of input parameters, or design variables, can be computed with an essentially constant amount of work. For shape optimization purposes, these design variables have historically been related to the geometric definition of the system, such as parameters defining an airfoil. The approach pioneered by Jameson for aerodynamics is based on the continuous adjoint. This continuous adjoint is derived from the governing equations of the simulation and represents a PDE that can be discretized and solved to determine the required derivative values. In contrast to this continuous approach, a discrete adjoint approach has also been advanced by a number of researchers, including Anderson, Giles, Elliott, and Peraire [6–8]. While the continuous adjoint predicts the analytical derivative associated with the governing equations, the discrete approach predicts the derivative of the simulation output itself. Because the goal of sensitivity analysis in this work is to determine the effect of parameters on the simulation itself, the discrete adjoint is the natural choice. The discrete approach possesses a number of advantages when compared to the continuous approach, such as the ease with which boundary conditions are applied, the ease of verifying derivative predictions and the ability to utilize automatic differentiation to compute the necessary terms in the analysis code [9]. Outside of the field of CFD, the use of the adjoint has a long history applied to sensitivity analysis [10], especially within the context of neutron transport and reactor physics [11].

Although the localized sensitivity analysis is relatively easy to implement and computationally efficient when derivatives are calculated using an adjoint-based approach, it suffers from two significant limitations. First, because the localized sensitivity analysis is based on derivative values, it is only valid within a small vicinity of the original solution and may not provide an accurate estimate of the change in output if large perturbations of the design variables are considered. Second, the localized analysis is incapable of accounting for interference effects whereby the perturbations of one design variable may alter the derivative value associated with another design variable [2]. To overcome these difficulties, a global sensitivity analysis can be performed. This global sensitivity analysis is closely related to the problem of uncertainty quantification. As such, the parameters within a global sensitivity analysis have an associated interval or distribution, known as the uncertainty space or design space. Global sensitivity analysis therefore quantifies the effect each variable has on the output measured over the entire design space. Within the simulation of hypersonic flows, this global sensitivity analysis has been performed based on Monte Carlo sampling. Examples of this sampling-based global sensitivity analysis for hypersonic flows can be found in the works of Wright, Palmer and Kleb [2, 12, 13]. Based on the results of the Monte Carlo sampling, correlation coefficients between the input parameters and the output of interest are computed and used to determine the contribution to the output made by each variable. Because of the slow convergence of statistics associated with Monte Carlo sampling, these previous approaches required thousands of CFD simulations [2, 12, 13]. In order to reduce the cost associated with this sensitivity analysis, polynomial regression has been used to represent the simulation output, and correlation coefficients have been calculated based on this regression [14]. Although this regression was able to reduce the number of CFD results required for the sensitivity analysis, the it was limited to analyzing a small number of input parameters due to the large growth in the size of the basis as the dimension of the parameter space increases.

1.1.2 Uncertainty Quantification

Uncertainty quantification within the context of computational science is the process by which variability is assigned to the output of a numerical simulation based on the variability associated with input parameters as well as variability within the simulation itself. In the case of deterministic simulations, such as solving the Navier-stokes equations for aerodynamic applications, the process of uncertainty quantification consists of propagating the uncertainty associated with model input parameters to the output of a simulation. In this context, it is convenient to view the effect of the simulation as a function, $y = f(x)$, that maps inputs x to a simulation output y. Hence, y becomes uncertain only when considered with the uncertain input x. As a matter of terminology, the inputs x belong to an uncertainty space, or design space, and the goal of uncertainty quantification is to model the behavior of y over this space. The variability of simulation inputs can come in two forms: aleatory or epistemic. The quantification of each form, as well as the case of mixed form, requires a unique set of methods. In order to provide a basis for the acceleration techniques used throughout this work, previous work relating to uncertainty quantification in CFD simulations is given.

Aleatory Uncertainty

Aleatory uncertainties arise due to the inherent randomness of a variable and are characterized by a probability distribution [15]. For aleatory inputs, the goal of uncertainty quantification is to determine the distribution of an output quantity due to these input distributions. Depending on the application, this characterization may consist of constructing the full empirical distribution function or may be limited to calculating statistics of the distribution that can be used within design, such as the average, variance or a specified quantile. This characterization can be performed in a relatively straightforward, although expensive, manner using Monte Carlo sampling. In the works of Wright and Palmer [2,12], this approach has been used successfully within the context of hypersonic flows to build up the required statistics for relevant simulation outputs. For this type of sampling, computing an output requires a complete computational fluid dynamic (CFD) simulation, making exhaustive sampling expensive for complex problems. When only a limited number of simulation outputs are of practical interest, a typical approach for reducing the expense of Monte Carlo sampling is the use of an inexpensive surrogate. This surrogate approximates the relationship between the true function value and the input parameters and is built based on a limited number of function evaluations. Because the surrogate is inexpensive to evaluate, exhaustive sampling of this model can be performed to build the required statistics of the output. Surrogate models range in complexity from simple extrapolations [16,17] to more sophisticated models, such as least-squares polynomials [14,18–20], support vector regression [21], radial basis functions [22], and Kriging. In computational fluid dynamics (CFD), Kriging methods in particular have gained popularity [23–33]. For the quantification of uncertainty in hypersonic flows, surrogates based on polynomial chaos have been employed with success [14,18]. Despite this success, these polynomial approaches have been limited to a small number of variables.

One drawback of surrogate-based methods is the "curse of dimensionality", whereby the number of samples required for an accurate surrogate increases exponentially as the number of input parameters grows. This problem is typically overcome using two methods. First, a sensitivity analysis can be used to reduce the dimension modeled with the surrogate. As shown in other works, the variance of a typical simulation is often a function of a small fraction of the total number of variables [19]. By creating the surrogate only over these variables, it should be possible to account for the output variance with a lower dimensional surrogate model. The other method for overcoming this limitation is the incorporation of gradient information into the training of the surrogate [19, 27, 31–34]. When adjoint methods are employed, this gradient may be evaluated with a cost approximately equal to the simulation of the physical problem [4, 35, 36]. By incorporating derivative values, the cost associated with training an accurate surrogate can be greatly reduced. For the methods detailed here, a combination of both methods are used.

Epistemic Uncertainty

Epistemic uncertainty arises from a lack of knowledge regarding the true value of a parameter. Because of this lack of knowledge, the parameter has no associated probability distribution function and is typically only specified using an interval. The goal of uncertainty quantification for epistemic uncertainties is to determine the output interval of a quantity due to specified input intervals. The quantification of epistemic uncertainties has been scarcely explored in the context of hypersonic flows, despite their dominance in hypersonic flows. Previous studies assuming pure aleatory uncertainties, although important initial steps, have likely underestimated the uncertainty associated with simulation objectives [2, 12, 37]. Epistemic uncertainty may be quantified via sampling based approaches or via optimization. Typically, Latin hypercube sampling [38] is used for epistemic uncertainties, although other methods such as approaches based on random sampling and Dempster-Shafer evidence theory can be used [39–41]. For Latin hypercube sampling in particular, the required number of samples grows quickly as the dimension of the problem increases, making the quantification of epistemic uncertainties for large-dimension problems difficult [15]. As was the case with aleatory uncertainty, one possible solution is to replace sampling with a surrogate model; however, this approach will again eventually encounter the curse of dimensionality as the input dimension increases.

The other main approach for epistemic uncertainty quantification is to pose the problem as a bound-constrained optimization problem, defined as: given input parameters within specified ranges, determine the maximum and minimum values of an output function. Although this approach entails solving a complicated global optimization problem with the possibility of multiple extrema [42], the number of function evaluations to solve the optimization problem scales more readily to high-dimensional problems if a gradient-based optimizer is employed. For situations in which traditional gradient-based approaches, such as those built from Newton's method, are inadequate, efficient global optimization techniques based on Kriging surrogate models have been successfully demonstrated within the field of CFD [29]. For these methods, a Kriging surrogate is used to represent the design space, and traditional global optimization techniques, such as genetic algorithms, are applied to this surrogate. As is the case for aleatory uncertainty, these Kriging methods can be enhanced with derivative values to improve the performance of the model in higher dimensions [31,33].

Mixed Aleatory/Epistemic Uncertainty

The problem of epistemic uncertainty quantification is further complicated when contributions from aleatory sources are also considered. This mixed aleatory/epistemic uncertainty quantification typically relies on a nested sampling strategy. Although the required number of samples grows extremely fast, these strategies are conceptually easy to understand and are capable of separating the effects of each type of uncertainty [15, 43]. For nested strategies, samples are first drawn from the epistemic variables; and for each set of epistemic variables, the distribution of the output due to the aleatory variables is determined using sampling over the aleatory variables. Since the number of samples required for the epistemic uncertainty grows exponentially fast, the expense of nested sampling grows rapidly with respect to the number of epistemic variables [15]. For hypersonic flows, the number of epistemic variables is typically much greater than the number of aleatory variables. Hence, for complex models with many uncertain epistemic variables, nested approaches will quickly become prohibitively expensive. Here, too, surrogates can be created as a function of all variables and samples extracted according to a nested strategy.

For relatively low dimensions, this strategy has been shown to be effective [14] and, when combined with gradient-enhancement, could be applied to problems of moderate dimension. However, once the number of epistemic variables increases sufficiently, surrogate-based approaches will again become prohibitively expensive as the required number of samples for an accurate surrogate increases. In order to address this concern, combination sampling/optimization approaches have been explored [43, 44]. For mixed aleatory/epistemic problems, the goal of the uncertainty quantification is to produce a region in which the function is contained with a specific level of confidence, known as a P-Box [15]. Because the bounds of the P-Box are the desired results, the sampling over epistemic variables can be replaced by optimization. The work of Eldred and Swiler explores the use of optimization to propagate the epistemic uncertainty combined with a polynomial chaos expansion for the aleatory uncertainty. For this method, the outer loop for a nested strategy is replaced by optimization over the epistemic variables. The optimization itself is performed to determine the desired bound of the P-Box, making the objective of the optimization stochastic in nature based on the aleatory uncertainty. Hence, each function evaluation of the optimization requires an aleatory uncertainty quantification. This quantification is performed rapidly using a polynomial chaos expansion [43].

1.2 Issues with Previous Approaches

Although the previously described methods for uncertainty quantification and sensitivity analysis are well established for other types of problems, their application to hypersonic flow simulations, and CFD in general, is limited by two factors. For uncertainty quantification and global sensitivity analysis, the large number of simulation results required for sampling-based approaches limits their application to complex CFD simulations, which require hours to days of computing for a single result. Additionally, the rapid increase in cost associated with previous rapid uncertainty quantification techniques prevents these methods from being used for simulations with a large number of inputs. For the case of global sensitivity analysis and aleatory uncertainty quantification, previous works have reduced the required number of simulation results by utilizing surrogate models. Although effective in reducing the total number of simulation results, these works have been limited to low dimension. In order to apply these methods for hypersonic CFD simulations, strategies to extend these surrogates to higher dimension without a dramatic increase in expense must be devised. For epistemic and mixed form uncertainties, the expense associated with quantification is prohibitive for the number of inputs encountered in a typical CFD simulation, with a cost increasing exponentially as the dimension expands. Although previous works have explored optimization-based approaches for propagating these uncertainties, the use of optimization for complex engineering calculations has yet to be demonstrated. Given these issues, the goal of this work is to develop and implement uncertainty quantification and sensitivity analysis strategies that can accurately capture the output variability of a hypersonic CFD simulation with a small number of simulation results and that can readily scale to a large number of input parameters.

1.3 Contributions of Current Work

Based on the previous research performed in the area of uncertainty quantification and sensitivity analysis for CFD, the current work makes several contributions to the field by incorporating gradient-based methods into the previously described techniques and applying these techniques to hypersonic flows. Because this work relies on the efficient calculation of derivative values, an analysis tool suitable for hypersonic flows with an adjoint capability was required. Although a number of mature CFD tools exist for the simulation of hypersonic flows, the application of the discrete adjoint to these tools is an area of active research [45, 46]. Hence, in order to explore gradient-based approaches, a CFD solver capable of simulating hypersonic flows was developed and a discrete adjoint approach was implemented within this solver. Although this code is limited to two dimensions, it is capable of solving problems in thermal and chemical non-equilibrium, replicating a number of the challenges resulting from the simulation of hypersonic flow. Using this analysis code, the discrete adjoint was implemented using automatic differentiation. The details of the physical model and solver used in this work can be found in Chapters 2 and 3, while the details of the adjoint implementation are found in Chapter 4.

Using this solver and adjoint implementation, gradient-based strategies for sensitivity analysis and uncertainty quantification were applied to rapidly determine the importance of model parameters and to quantify the uncertainty in the simulation based on these parameters. For the case of sensitivity analysis, local sensitivity analysis is applied directly using the gradients produced by the discrete adjoint approach. In addition to local sensitivity analysis, a gradient-based approach for rapid global sensitivity analysis applicable to large input dimensions was developed by expanding the regression used in previous work [14] to include derivative values. The methods used for these sensitivity analyses and corresponding results are presented in Chapter 4.

Utilizing the adjoint capability of the solver and building upon the sensitivity capability, gradient-based methods for uncertainty quantification developed in previous work were applied to the simulation of hypersonic flows. For the case of aleatory uncertainty, the performance of gradient-enhanced surrogate models was tested for the quantification of uncertainty arising from model parameters within the CFD simulation. For epistemic uncertainty, the use of gradient-based optimization was demonstrated and compared to the traditional sampling-based method. Finally, for the case of mixed form uncertainty, a novel optimization/surrogate based approach was developed. Compared to the method of Eldred and Swiler [43,47], the order of optimization and surrogate approximation in this method is switched. By reversing the order, the new method is capable of predicting multiple statistics without an associated increase in cost. These gradient-based methods are detailed and demonstrated for a hypersonic test problem in Chapter 5 for aleatory uncertainties, Chapter 6 for epistemic uncertainties and Chapter 7 for mixed form uncertainties.

Chapter 2

Physical Models

In order to numerically solve problems featuring hypersonic flows, a number of physical models must be utilized. These models are detailed in this chapter. First, the characteristics of high temperature gases are introduced. With the required physics introduced, the equations solved for this work, the Navier-Stokes equations for reacting flow in thermal and chemical non-equilibrium, are developed, and the required constitutive laws are identified. Finally, the state relations required to close the equations, such as the caloric equation of state, thermal equation of state, reaction terms and necessary fluid properties, are given.

2.1 Characteristics of High Temperature Gases

In order to provide the necessary background for developing the continuous equations governing hypersonic fluid flow, the properties of high temperature gases are discussed. Because of the energy available to the molecules of the gas at high temperature, a number of unique characteristics are encountered that do not appear for room temperature flows. These characteristics include the excitation of internal energy modes, the dissociation of molecules and thermal non-equilibrium, where by energy modes of the molecule are described by multiple distinct temperatures.

2.1.1 Energy Modes

The energy of a molecule is composed of four components: translational (e_t), rotational (e_r), vibrational (e_v), and electronic energy (e_{el}). Typically, the energy of the molecule is assumed separable and the total energy is the sum of these four components [48].

$$e = e_t + e_r + e_v + e_{el} \tag{2.1}$$

The translational and rotational energy are present for molecules at room temperatures; however, the vibrational and electronic energy are only significant for high temperatures due to the fact that molecular energy is quantized. In order for molecules to occupy these states, the average energy must be comparable to the energy spacing for the vibrational and electronic energy modes, which only occurs for high temperatures. Although all energies at the molecular level are quantized, continuous descriptions of the average energy for a collection of molecules can be assumed in certain circumstances. In order to determine properties of the high temperature gas, the energy as a function of macroscopic thermodynamic variables must be defined. For this work, a thermally perfect gas is assumed. This assumption implies that the specific energy is a function only of temperature.

The translational energy represents the kinetic energy of the molecule due to its linear momentum, while rotational energy is the kinetic energy due to angular momentum. Because the energy spacing for these energy modes is small, the translational and rotational energy can be treated as continuous and fully excited for essentially all temperatures, with discretization of these modes important only for extremely low temperature flows. For a fully excited energy mode, the average energy for a molecule in a gas is a function of the degrees of freedom [49].

$$\varepsilon = \frac{d}{2} k_b T \tag{2.2}$$

Here, d is the degrees of freedom for an energy mode, k_b is Boltzmann's constant and T is the temperature. For the translational energy mode, the molecule has three degrees of freedom, corresponding to three coordinates required to specify its location. Based on equation (2.2), the translational energy is given as:

$$e_t(T) = \frac{3}{2} k_b T \tag{2.3}$$

This equation gives the average translational energy for a single molecule. This equation can be converted to the energy for a single mole of molecules by replacing Boltzmann's constant with the universal gas constant, $\bar{R}$. The equation for translational energy is the same for both atoms and polyatomic molecules. For the rotational energy, the degrees of freedom depend on the moments of inertia of the molecule. For a single atom, the principle moments of inertia are negligible, which makes the rotational energy zero. For a diatomic molecule, only one of the principle moments of inertia is negligible, giving a diatomic molecule two rotational degrees of freedom. Given these facts, the average rotational energy per molecule is given below.

$$e_r(T) = \begin{cases} 0 & \text{for atoms,} \\ k_b T & \text{for diatomic molecules} \end{cases} \tag{2.4}$$

Because of the energy spacing associated with the vibrational and electronic energy, these modes do not become significant until the energy of the molecule is comparable to this energy spacing. The vibrational energy mode appears only for polyatomic molecules and represents the energy contained in the vibrating of the atoms in the molecule relative to one another. For a diatomic species, two vibrational degrees of freedom are possible; however, the energy is not fully excited for most applications, meaning equation (2.2) can not be used. In order to compute the vibrational energy, the molecule is represented as a harmonic oscillator as given by equation (2.5) [50].

$$e_v(T) = \frac{k_b \theta_v}{e^{\theta_v / T} - 1} \tag{2.5}$$

Here, θ_v is the characteristic temperature for the vibration. This characteristic temperature is given in equation (2.6) and is unique for each molecule [49].

$$\theta_v = \frac{h\nu}{k_b} \tag{2.6}$$

Table 2.1: Vibrational Temperatures for Non-ionizing Dissociating Air

Species	θ_v (K)
N_2	3395.0
O_2	2239.0
NO	2817.0
N	-
O	-

Here, h is the Plank's constant and ν is the fundamental frequency of vibration for the molecule. The vibrational temperature for the molecules used in this work are given in Table 2.1 taken from reference [51]. Based on equation (2.5), the value of vibrational energy in the limit of high and low temperature relative to the vibrational temperature can be calculated as:

$$e_v(T) \approx 0 \quad \text{for } T << \theta_v \tag{2.7}$$

$$e_v(T) \approx \frac{k_b \theta_v}{(1 + \theta_v/T + ...) - 1} = k_b T \quad \text{for} \qquad T >> \theta_v \tag{2.8}$$

For the case of temperatures well below the vibrational temperature, the vibrational energy is essentially zero as the exponential in the denominator increases rapidly to a large value. For the case of temperatures much greater than the vibrational temperature, a Taylor series expansion is used for the exponential and the energy becomes linear in temperature. This limit is the case of full excitation of the vibrational energy and the energy relationship reverts to that found in equation (2.2) [49].

The electronic mode represents the energy contained in a molecule due to the excitation of electrons above the ground state to higher energy orbitals. Because this energy mode is atomic in nature, all species possess this energy mode. The representation of electronic energy is similar to that of vibrational energy, with a characteristic temperature for each mode. Because multiple electron transitions are possible, numerous levels exist for the electronic mode and the electronic energy is given by summing of these levels [51].

Table 2.2: Electronic Temperatures and degenericies for Non-ionizing Dissociating Air

Species	$\theta_{el,1}$ (K)	$\theta_{el,2}$ (K)	g_0	g_1	g_2
N_2	72,233.0	85,744.0	1	3	6
O_2	11,392.0	18,985.0	3	2	1
NO	55,835.0	63,258.0	4	8	2
N	27,665.0	41,495.0	4	10	6
O	22,831.0	48,620.0	9	5	1

$$e_{el} = k_b \frac{\sum_{i=1}^{\infty} g_i \theta_{el,i} e^{\theta_{el,i}/T}}{g_0 + \sum_{i=1}^{\infty} g_i \theta_{el,i} e^{\theta_{el,i}/T}} \tag{2.9}$$

Here, $\theta_{el,i}$ is the characteristic temperature for each electronic transition and g_i is the degeneracy for each transition. These values are given for the first and second electronic level in Table 2.2. As the table shows, the characteristic temperature for the electronic mode is much greater than the vibrational characteristic temperature. This fact signifies that the electronic energy mode is excited at higher temperature than the vibrational energy mode.

Although the equations for vibrational and electronic energy are not used explicitly in this work, the forms of these equations offer valuable insight into the energy modes present for high temperature molecules. For this work, the energy is calculated based on the integration of polynomial curve fits of specific heat for each species, which include the effects of vibrational and electronic energy at high temperatures. This energy calculation is detailed later in this chapter.

2.1.2 Molecular Dissociation

As the temperature of a molecule increases, the energy available eventually becomes large enough to break the chemical bonds holding the molecule together. Once the molecules dissociate, the composition of the gas mixture becomes variable and the effect of chemical reactions must be modeled. The dissociation of molecules can be modeled in either an equilibrium and non-equilibrium manner. An equilibrium treatment is appropriate when the time scale of the reaction is much smaller than the other time scales experienced by the molecule, such as the timescale associated with collisions with other molecules. For an equilibrium model, the properties of the reacting mixture are given by curve fits that account for the changing composition of the mixture. Because of the dissociation, this mixture becomes a non-ideal gas, making the properties of the mixture a function of two variables, such as density and temperature. An example of these curve fits can be found in Reference [52]. If the composition of the mixture is required, the equilibrium reaction rate combined with the partial pressures can be used and a system of equations can be solved. This process is demonstrated in Appendix D [53]. An equilibrium reaction model is used for the Fay-Riddell heating correlation [54] discussed later in this chapter.

When the time scale of the reaction is comparable to the time scale associated with molecule collisions, a non-equilibrium (or finite-rate) chemistry model is used. For this model, each species in the mixture must be modeled separately and the creation and destruction of each species is accounted for using a chemical kinetics model. This chemical kinetics model determines the appropriate reaction rate as a function of the mixture properties. The chemical kinetics models used in this work are detailed later in the chapter.

2.1.3 Thermal Non-equilibrium

For situations in which the energy modes are described by different representative temperatures, the system is said to be in thermal non-equilibrium. Thermal non-equilibrium occurs when the coupling between the energy modes is weak, typically because of the limited number of collisions caused by the low density of the mixture. For the thermal non-equilibrium model used in this work, two temperatures are used to model the energy modes and all the species in the mixture are described by the same representative temperatures. For systems with weak inter-species coupling, each species can be modeled by its own representative temperatures [55], but this type of model is not used in this work.

The two temperatures used for the model in this work are a translation-rotational temperature and a vibrational-electronic temperature. The choice of these two temperatures implies that the between translational and rotational energy modes is strong, as well as the coupling between the vibrational and electronic energy modes. This strong coupling causes these energy modes to be in equilibrium with one another and allows for a unified temperature to describe these energy modes. Using these two temperatures, the functional form of the total energy equation becomes:

$$e(T, T_v) = e_t(T) + e_r(T) + e_v(T_v) + e_{el}(T_v) \tag{2.10}$$

where T is the translation-rotational temperature and T_v is the vibrational-electronic temperature. The use of two temperatures necessitates the need to solve a separate conservation law for each unified-energy mode, requiring two energy equations for this model. The assumption of strong inter-species coupling and equilibrium between some energy modes implies that the energy transferred in a collision is preferentially transferred to the same energy mode in the colliding molecule. However, it is possible for energy to be transferred between the modes, requiring the coupling between energy modes to be modeled within the multiple energy modes. The exact form of the energy conservation equations and the coupling between modes is examined in the next section.

2.2 Multicomponent Navier-Stokes Equations

With the energy modes presented and the concepts of chemical and thermal non-equilibrium introduced, the five species, two temperature real gas model can be presented. In order to solve problems utilizing this model, the Navier Stokes equations for a gas in chemical and thermal non-equilibrium are solved. This model neglects the effect of ionization, accounting only for dissociation and the creation of other species based on these dissociated species. The equations for this model consist of a mass conservation equation for each species, a momentum equation for the mixture of species, and an energy equation for each energy mode of the mixture. In order to develop these equations, each species is assumed to be in equilibrium with itself. This equilibrium implies that each species can be described by its own set of continuous variables, namely a species density, velocity, and temperature for each energy mode. As equilibrium is assumed over the appropriate variables, the equations will be reduced to a manageable size.

The mass conservation equation for each species is given by [56]:

$$\frac{\partial \rho_s}{\partial t} + \nabla \cdot (\rho_s \vec{u}_s) = \omega_s \tag{2.11}$$

where ω_s is the net mass creation per unit time for species s. In order to reduce the required number of variables required to describe the flow, the species velocity is decomposed into a bulk velocity that is the same across all species and a diffusive velocity.

$$\vec{u}_s = \vec{U} + \vec{V}_s \tag{2.12}$$

The bulk velocity is defined as:

$$\vec{U} = \frac{\sum_s \rho_s \vec{u}_s}{\rho} \tag{2.13}$$

$$0 = \sum_s \rho_s \vec{V}_s \tag{2.14}$$

where ρ is the total density of the mixture, given as the sum of the species densities. By this definition, the sum of the diffusive velocity weighted by the species density over all species is equal to zero as it represents the departure from the bulk velocity. Using this velocity decomposition, the conservation of each species is given as:

$$\frac{\partial \rho_s}{\partial t} + \nabla \cdot (\rho_s \vec{U}) = -\nabla \cdot (\rho_s \vec{V}_s) + \omega_s \tag{2.15}$$

For this model, five mass equations must be solved. Instead of solving for each of the five individual species, the total density along with 4 of the species equations are solved, as this set of variables allows for convenient calculation of mixture properties. The total mass equation is found by adding the individual species equations:

$$\sum_s \frac{\partial \rho_s}{\partial t} + \sum_s \nabla \cdot (\rho_s \vec{U}) = -\sum_s \nabla \cdot (\rho_s \vec{V}_s) + \sum_s \omega_s \tag{2.16}$$

leading to the mass conservation equation for the mixture:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{U}) = 0 \tag{2.17}$$

Here, the fact that the sum over the reaction source term is zero when summed over all species by conservation of mass has been utilized. The resulting equation is the total mass equation typical of the Navier-Stokes equations.

Based on this velocity decomposition, a single momentum equation is solved for this model. This equation is arrived at by summing the individual species momentum equations. For a single species in a non-ionizing model, the species momentum equation is given as [57]:

$$\frac{\partial \rho_s \vec{u}_s}{\partial t} + \nabla \cdot (\rho_s \vec{u}_s \otimes \vec{u}_s) = \vec{Q}_{m,s} \tag{2.18}$$

where $\vec{Q}_{m,s}$ is the rate of momentum transfer to species s. Summing this equation over all species gives the momentum equation for the mixture, given as [51]:

$$\frac{\partial \rho \vec{U}}{\partial t} + \nabla \cdot (\sum_s \rho_s \vec{u}_s \otimes \vec{u}_s) = 0 \tag{2.19}$$

Using the velocity decomposition given in equation (2.13), the momentum flux can be decomposed into a convective term, pressure term and shear stress term.

$$\sum_s \rho_s \vec{u}_s \otimes \vec{u}_s = \sum_s \left[\rho_s \vec{U} \otimes \vec{U} + \rho_s U \otimes \vec{V}_s + \rho_s \vec{V}_s \otimes U + \rho_s \vec{V}_s \otimes \vec{V}_s\right] \tag{2.20}$$

$$\sum_s \rho_s \vec{u}_s \otimes \vec{u}_s = \rho \vec{U} \otimes \vec{U} + \sum_s \rho_s \vec{V}_s \otimes \vec{V}_s \tag{2.21}$$

Here, the fact that the sum of the diffusive velocity weighted by the species density is zero (equation (2.14)) has been utilized to eliminate the two middle terms. By definition, the diffusive term can now be decomposed into pressure and shear stress components, given in equation (2.22).

$$\sum_s \rho_s \vec{V}_s \otimes \vec{V}_s = \sum_s p_s \underline{I} - \sum_s \underline{\tau_s} \tag{2.22}$$

$$\sum_s \rho_s \vec{V}_s \otimes \vec{V}_s = P \underline{I} - \underline{\tau} \tag{2.23}$$

Here, $\underline{I}$ is the identity matrix, $\underline{\tau}$ is the shear stress tensor for the mixture and P is the total pressure, which is the sum of the partial pressure of each species. The final momentum equation for the mixture is the same momentum equation encountered for a non-reacting mixture or single isolated species.

$$\frac{\partial \rho \vec{U}}{\partial t} + \nabla \cdot (\rho \vec{U} \otimes \vec{U}) = -\nabla P + \nabla \cdot \underline{\tau} \tag{2.24}$$

Finally, the two energy equations are required for this model. These energy equations are the total energy equation and a separate vibrational-electronic energy equation. To derive the total energy equation, the species total energy equations are added to give an equation for the mixture total energy. For a single species, the total energy equation is given below [57].

$$\frac{\partial \rho_s e_{t,s}}{\partial t} + \nabla \cdot (\rho_s e_{t,s} \vec{u}_s) = Q_s \tag{2.25}$$

Here, $e_{t,s}$ is the total energy for the species, which is the sum of the molecular energy, e and the macroscopic kinetic energy, $\frac{1}{2}\vec{u}_s \cdot \vec{u}_s$. The source term Q_s is the total energy transfer into species s. Summing this equation over all the species gives the total energy equation for the mixture:

$$\frac{\partial \rho e_t}{\partial t} + \nabla \cdot (\rho h_t U) = \nabla \cdot (\underline{\tau} \vec{u}) - \nabla \cdot \vec{q}_t \tag{2.26}$$

The details of constructing the total energy equation are omitted here as the velocity decomposition leads to a large number of terms. The omitted details can be found in Reference [51]. In this equation, e_t is the total energy of the mixture, given as the sum of the mixture molecular energy and the bulk kinetic energy $\frac{1}{2}\vec{U} \cdot \vec{U}$. The total enthalpy h_t is given as the sum of internal energy and pressure and the total heat flux $\vec{q}_t$ represents the transport of total energy by the diffusive species velocity. This total heat flux is the sum of the multiple energy components as well as the flux of enthalpy due species diffusion.

$$h_t = e_t + \frac{P}{\rho} \tag{2.27}$$

$$\vec{q}_t = \vec{q} + \vec{q}_v + \sum_s h_{t,s} \rho_s \tilde{V}_s \tag{2.28}$$

Here, $\vec{q}$ is the diffusion of translation-rotational energy and q_v is the diffusion of vibrational-electronic energy.

The vibrational-electronic energy equation is derived based on the conservation of internal energy modes for a single species and summing these equations over the species. The conservation of an internal energy mode, i, is given as [55]:

$$\frac{\partial \rho_s e_{i,s}}{\partial t} + \nabla \cdot (\rho_s e_{i,s} \vec{u}_s + q_{i,s}) = Q_{i,s} + e_{i,s}\omega_s \tag{2.29}$$

where $e_{i,s}$ is the internal energy mode for species s, $q_{i,s}$ is the heat flux for the energy mode, $Q_{i,s}$ is the net rate of energy transfer into the energy mode and $e_{i,s}\omega_s$ represents the net internal energy created from chemical reactions. The internal energy equation for the mixture is again found by summing over the species. The details of this summation are again omitted and can be found in Reference [51]. The final vibrational-electronic energy is given as:

$$\frac{\partial \rho e_v}{\partial t} + \nabla \cdot (\rho e_v U) = Q_{T-V} + \sum_s e_{v,s}\omega_s - \nabla \cdot \left(\sum_s h_{v,s}\rho_s \tilde{V}_s \right) - \nabla \cdot \vec{q}_v \tag{2.30}$$

where Q_{T-V} represents the coupling between the translational and vibrational energy modes.

The final equations in vector form are given as:

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \vec{F}(\mathbf{U}) = \nabla \cdot \vec{F}_v(\mathbf{U}) + \mathbf{S}(\mathbf{U}) \tag{2.31}$$

The terms in the equation are defined as:

$$\mathbf{U} = \left\{ \begin{array}{c} \rho_s \\ \rho\vec{u} \\ \rho e_t \\ \rho e_v \end{array} \right\} \quad \vec{F} = \left\{ \begin{array}{c} \rho_s \vec{u} \\ \rho\vec{u} \otimes \vec{u} + P \\ \rho\vec{u} h_t \\ \rho\vec{u} h_v \end{array} \right\} \quad \vec{F}_v = \left\{ \begin{array}{c} -\rho_s \tilde{V}_s \\ \underline{\tau} \\ \underline{\tau} \cdot \vec{u} - \vec{q} - \vec{q}_v - \sum_s h_{t,s}\rho_s \tilde{V}_s \\ -\sum_s h_{v,s}\rho_s \tilde{V}_s - \vec{q}_v \end{array} \right\}$$

In two dimensions, this model contains nine conserved variables. The first variables, ρ_s for $s = 1, 2, .., 5$, are the five species densities. The other variables are the bulk momentum, $\rho\vec{u}$, the total energy ρe_t, and the vibrational-electronic energy, ρe_v. In the absence of body forces and internal heat generation, the source vector for this model can be represented by:

$$\mathbf{S} = \left\{ \begin{array}{c} \omega_s \\ 0 \\ 0 \\ \sum_s \omega_s \hat{e}_{v,s} + Q_{T-V} \end{array} \right\} \tag{2.32}$$

With these equations established, constitutive laws are used to define the heat flux, stress tensor and diffusive velocity in terms of macroscopic flow properties. The species diffusion velocities,$\tilde{V}_s$, are determined using Fick's law.

$$\rho_s \tilde{V}_s = -\rho D_s \nabla c_s \tag{2.33}$$

where c_s is the species mass fraction and D_s is the species diffusion coefficient. The shear stress tensor, $\underline{\tau}$, is given by the relation for a Newtonian fluid and the heat flux for each mode is represented by Fourier's law of conduction.

$$\underline{\tau} = \mu(\nabla\vec{u} + \vec{u}\nabla) - \frac{2}{3}\mu\nabla\cdot\vec{u}\underline{I} \tag{2.34}$$

$$\vec{q} = -k\nabla T \tag{2.35}$$

$$\vec{q_v} = -k_v\nabla T_v \tag{2.36}$$

With the equations defined, state relationships are used to close the equations.

2.3 State Relationships

In this section, the state relations required to close the equations are detailed. These relationships consist of the caloric equation of state, the thermal equation of state, the transport coefficients, reaction models and energy coupling relationships.

2.3.1 Caloric Equation of State

The caloric equation of state defines the relationship between the internal energy and the thermodynamic properties. The caloric equation of state is specified by the specific heat at constant volume c_v. The variation of the specific heat as a function of temperature is given by a set of fourth order polynomials for each species [56].

$$\frac{C_v^s(T)M_s}{\bar{R}} = A_{o,s}^i + A_{1,s}^i T + A_{2,s}^i T^2 + A_{3,s}^i T^3 + A_{4,s}^i T^4 \tag{2.37}$$

Here, A_s^i are the polynomial coefficients for each species over a range of temperature i. Because the curve fit specifies the specific heat on a molar basis, the species mass M_s is required and the universal gas constant is used for normalization purposes. The coefficients themselves vary depending on the temperature, with different coefficients defined for different temperature ranges between 300 K up to 35000 K. The coefficients are given in Table A.2 of Appendix of A. To ensure the specific heat is smooth as a function of temperature, the coefficients between two ranges are given by the temperature-weighted average of the coefficients in the lower and upper temperature range. This averaging is defined in equation (2.38).

$$A(T)_s = \frac{T_r - T}{T_r - T_l} A_s^i + \frac{T - T_l}{T_r - T_l} A_s^{i+1} \tag{2.38}$$

In this equation, A_s^i represents the coefficients from the lower temperature range, A_s^{i+1} represents the coefficients from the higher temperature range, T_l represents the highest temperature from range i and T_r represents the lowest temperature from range $i+1$. This interpolation between coefficients occurs within $\pm 50K$ of the boundary between temperature ranges.

These specific heat curves represent the total specific heat and have contributions from all energy modes. Because the different energy modes are described by two different temperatures, the different energy components must be separated in some way. Because the translational and rotational energy modes are fully excited for the flows considered in this work, the specific heat corresponding to these energy modes is constant. Hence, the vibrational/electronic specific heat can be determined by subtracting the translation-rotational specific heat from the total [56].

$$C_{v,v-e}^s(T_v) = C_v^s(T_v) - C_{v,t-r}^s \tag{2.39}$$

Because the vibrational/electronic specific heat is a function of the vibration/electronic temperature, the curve fit is evaluated based on this temperature and the constant translational-rotational component is subtracted from this value. Based on the results from Section 2.1, the translation-rotational specific heat is given on a per mass basis as:

$$C^s_{v,t-r} = \begin{cases} \frac{5}{2}\frac{\bar{R}}{M_s} & \text{for diatomic species} \\ \frac{3}{2}\frac{\bar{R}}{M_s} & \end{cases} \tag{2.40}$$

The total internal energy is given by integrating the curve fits for each species, represented in equation (2.41).

$$e^s(T) = \int_{T_o}^{T} C_v(T)dT + e_o \tag{2.41}$$

Here, e_o is the reference energy, which includes the heat of formation, and T_o is the temperature at which e_o is evaluated. These reference values are given in Table A. Using the previously defined curve fits, the internal energy is given as:

$$\bar{e}^s(T) = A^i_{o,s}T + \frac{1}{2}A^i_{1,s}T^2 + \frac{1}{3}A^i_{2,s}T^3 + \frac{1}{4}A^i_{3,s}T^4 + \frac{1}{5}A^i_{4,s}T^5 + A^i_{6,s} \tag{2.42}$$

where $A^i_{6,s}$ accounts for the reference temperature and energy. In order to determine the individual energy components, the assumption of fully excited translational-rotational energy is again utilized. Hence, the vibrational-electronic is given as:

$$e^s_{v-e}(T_v) = \bar{e}^s(T_v) - C_{v,t-r}(T_v - T_o) - e_o \tag{2.43}$$

The translational-rotational energy is given by integrating the translational-rotational specific heat. On a per mass basis, this energy is given as:

$$e^s_{t-r}(T) = C_{v,t-r}(T - T_o) \tag{2.44}$$

The total internal energy using the two temperature model is now the sum of the vibration-electronic energy with the translational-rotational and the reference energy.

$$e^s(T,T_v) = e^s_{v-e}(T_v) + e^s_{t-r}(T) + e_o = \bar{e}^s(T_v) + C_{v,t-r}(T - T_v) \tag{2.45}$$

Finally, in order to determine the internal energy or specific heat of the mixture, the species values are averaged using the mass fraction.

$$e(T) = \sum_s c_s e^s(T) \tag{2.46}$$

$$C_v(T) = \sum_s c_s C_v^s(T) \tag{2.47}$$

2.3.2 Thermal Equation of State

The thermal equation defines the relationship between the pressure and the thermodynamic variables. For fluid dynamics, this equation of state is typically a function of density and temperature. For this work, each species is assumed to obey an ideal gas equation of state. From this assumption, the partial pressure is given by equation (2.48) [51].

$$p_s = \rho_s \frac{\bar{R}}{M_s} T \tag{2.48}$$

Here, ρ_s is the species density, M_s is the molecular mass and $\bar{R}$ is the universal gas constant. Because this pressure is the result of atomic collisions, the pressure is evaluated using the translational-rotational temperature, T. The mixture pressure is the sum of the partial pressures given as:

$$P(\rho,T) = \sum_s \rho_s \frac{\bar{R}}{M_s} T = \rho \sum_s c_s \frac{\bar{R}}{M_s} T \tag{2.49}$$

In addition to defining the relationship between thermodynamic variables and pressure, the thermal equation of state also defines the speed of sound. By definition, the speed of sound is defined as the derivative of pressure with respect to density evaluated isentropically.

$$c^2 = \left.\frac{\partial P}{\partial \rho}\right|_{isentropic} \tag{2.50}$$

Assuming the equation of state is defined based on density and internal energy, the derivative can be evaluated as:

$$\frac{\partial P}{\partial \rho} = \left.\frac{\partial P}{\partial \rho}\right|_{e} + \left.\frac{\partial P}{\partial e}\right|_{\rho} \frac{\partial e}{\partial \rho} \tag{2.51}$$

For an adiabatic flow, the derivative $\frac{\partial e}{\partial \rho}$ is given by [49]:

$$\left.\frac{\partial e}{\partial \rho}\right|_{adiabatic} = \frac{P}{\rho^2} \tag{2.52}$$

By definition, a reversible (inviscid), adiabatic flow is isentropic. Hence, the isentropic derivative is given by substituting the above expression into equation (2.51).

$$\left.\frac{\partial P}{\partial \rho}\right|_{isentropic} = \left.\frac{\partial P}{\partial \rho}\right|_{e} + \left.\frac{\partial P}{\partial e}\right|_{\rho} \frac{P}{\rho^2} \tag{2.53}$$

For an equation of state defined based on temperature, the chain rule can be used to construct the necessary derivative of pressure w.r.t energy.

$$\left.\frac{\partial P}{\partial e}\right|_{\rho} = \left.\frac{\partial P}{\partial T}\right|_{\rho} \left.\frac{\partial T}{\partial e}\right|_{\rho} = \left.\frac{\partial P}{\partial T}\right|_{\rho} \frac{1}{C_{v,t-r}} \tag{2.54}$$

The pressure relationship defined in equation (2.49) is a function of all of the species densities. In order to evaluate the required derivatives, this dependency must be reduced to a single density. This reduction is achieved by assuming a fixed chemical composition for the derivative evaluation. This fixed composition assumption yields what is known as the frozen speed of sound. The practical results of this assumption is that the mass fraction is assumed fixed for the differentiation. Using the frozen assumption, the required derivatives for equation (2.49) are given as:

$$\left.\frac{\partial P}{\partial \rho}\right|_{e} = \sum_s c_s \frac{\bar{R}}{M_s} T \tag{2.55}$$

$$\left.\frac{\partial P}{\partial T}\right|_{\rho} = \rho \sum_s c_s \frac{\bar{R}}{M_s} \tag{2.56}$$

$$\left.\frac{\partial e}{\partial T}\right|_{\rho} = \frac{1}{C_{v,t-r}} \tag{2.57}$$

Because the translational-rotational temperature is used in the equation of state, the translation-rotational specific heat of the mixture is used in the derivative of energy with respect to temperature. The specific heat in this equation is the specific heat of the mixture and can be calculated using equation (2.46). Using these derivatives, the final frozen speed of sound is given by [56]:

$$c^2 = \sum_s c_s \frac{\bar{R}}{M_s} T + \frac{\rho \sum_s c_s \frac{\bar{R}}{M_s}}{C_{v,t-r}} \frac{P}{\rho^2} \tag{2.58}$$

Using the definition of the mixture pressure, the first term can be replaced by the pressure divided by density to give an expression similar to the speed of sound equation typically seen for a perfect gas.

$$c^2 = \frac{P}{\rho} + \frac{\sum_s c_s \frac{\bar{R}}{M_s}}{C_{v,t-r}} \frac{P}{\rho} \tag{2.59}$$

$$c^2 = \frac{P}{\rho} \left(1 + \frac{\sum_s c_s \frac{\bar{R}}{M_s}}{C_{v,t-r}} \right) \tag{2.60}$$

Here, the term in parentheses can be viewed as the effective ratio of specific heats, γ_{eff}, for the mixture.

2.3.3 Transport Coefficients

In order to determine the viscosity, thermal conductivity and mass diffusivity, a transport model is required. For this work, a transport model based on collision integrals is used. These collision integrals account for the interaction of the individual species at the molecular level. The basis of the transport model is the variation of the collision integrals with temperature. This variation is modeled using the collision integrals measured at two temperatures and interpolating between these temperatures for other variables. For each temperature, two collision integrals are specified for each unique interaction. For a five species model, 15 interactions are possible, requiring a total of 30 collision integrals to be specified at each temperature (60 total collision integrals). The collision integrals are specified at 2000 K and

4000 K and interpolation of the base-10 logarithm of the collision integral is used. This interpolation is found in equation (2.61) [56].

$$\log_{10}(\Omega^{k,k}_{s,r}) = \log_{10}(\Omega^{k,k}_{s,r})_{2000} + \left[\log_{10}(\Omega^{k,k}_{s,r})_{4000} - \log_{10}(\Omega^{k,k}_{s,r})_{2000}\right] \frac{ln(T) - ln(2000)}{ln(4000) - ln(2000)} \quad (2.61)$$

The collision integrals for each interaction are given in Table A in Appendix A in units of cm^2. In this equation, s and r represent the species participating in the collision and k is an index running from 1 to 2. The different values for k are used to construct the different transport terms required for the model. With the collision integral calculated at the appropriate temperature, a modified cross-section is calculated, given by equation (2.62) [50].

$$\Delta^1_{s,r} = \frac{8}{3}\sqrt{\frac{2M_sM_r}{\pi\bar{R}T(M_s+M_r)}}\Omega^{1,1}_{s,r} \quad (2.62)$$

$$\Delta^2_{s,r} = \frac{16}{5}\sqrt{\frac{2M_sM_r}{\pi\bar{R}T(M_s+M_r)}}\Omega^{2,2}_{s,r} \quad (2.63)$$

With the modified collision integrals calculated, the viscosity and thermal conductivity of the mixture can be calculated as:

$$\mu = \sum_s \frac{m_s\gamma_s}{\sum_r \gamma_r\Delta^{(2)}_{s,r}} \quad (2.64)$$

$$k_t = \frac{15}{4}k_b\sum_s \frac{\gamma_s}{\sum_r \alpha_{s,r}\gamma_r\Delta^{(2)}_{s,r}} \quad (2.65)$$

$$k_r = k_b \sum_{molecules} \frac{\gamma_s}{\sum_r \gamma_r\Delta^{(1)}_{s,r}} \quad (2.66)$$

$$k_v = k_b \sum_{molecules} \frac{\frac{C^s_{v,v}M_s}{\bar{R}}}{\sum_r \gamma_r\Delta^{(1)}_{s,r}}. \quad (2.67)$$

In these equations, k_b is Boltzmann's constant, and m_s is the species molecular mass (as opposed to molar mass, M_s). The quantities $\alpha_{s,r}$ and γ_s are defined as:

$$\gamma_s = \frac{\rho_s}{\rho M_s} = \frac{c_s}{M_s} \tag{2.68}$$

$$\alpha_{s,r} = 1 + \frac{\left[1 - \left(\frac{M_s}{M_r}\right)\right]\left[0.45 - 2.54\frac{M_s}{M_r}\right]}{1 + \left(\frac{M_s}{M_r}\right)^2} \tag{2.69}$$

Finally, the mass diffusivity is calculated using the previously defined modified collision cross-section and molar concentration. First, the binary diffusion coefficient is calculated for the collisions between each species [56].

$$D_{s,r} = \frac{k_b T}{p\Delta^{(1)}_{s,r}} \tag{2.70}$$

Here, T is the translation-rotational temperature and p is the pressure. Using this binary coefficient, the effective diffusion coefficient for each species is calculated as:

$$D_s = \frac{\gamma_t^2 M_s(1 - M_s\gamma_s)}{\sum_{r\neq s}(\gamma_r/D_{s,r})} \tag{2.71}$$

Here, the variable γ_t is the sum of the molar concentration over all species, defined as:

$$\gamma_t = \sum_s \gamma_s \tag{2.72}$$

With the transport quantities defined, the viscous fluxes can now be calculated based on the local flow properties.

2.3.4 Reaction Rates

The source term ω_s represents the creation or destruction of species s per unit time. From the principle of conservation of mass, this source term must sum to zero over all species. The source term is constructed by summing the contribution from each reaction and applying conservation for each reaction. For a generic equilibrium reaction, the contribution to the source is given as follows:

$$n[A] + m[B] \leftrightarrows p[C] \tag{2.73}$$

$$\frac{\partial A}{\partial t} = n(R_b - R_f) \tag{2.74}$$

$$\frac{\partial B}{\partial t} = m(R_b - R_f) \tag{2.75}$$

$$\frac{\partial C}{\partial t} = p(R_f - R_b) \tag{2.76}$$

where $[A], [B], [C]$ represent the molar concentrations of the reactants and products respectively, and R_f and R_b represent the forward and backward reaction rates. The rate of change of the concentration is converted to a mass-based source term by multiplying by the molar mass. Based on the above individual reaction results, the source term for each species is given for a general set of reactions by the following equation:

$$\omega_s = M_s \sum_r (\beta_{s,r} - \alpha_{s,r})(R_{f,r} - R_{b,r}) \tag{2.77}$$

Here, r represents a particular reaction, $\alpha_{s,r}$ represents the stoichiometric coefficient of species s in the reactants of reaction r and $\beta_{s,r}$ is the stoichiometric coefficient of species s in the products of reaction r.

The reaction rates are dictated by the law of mass action, whereby the reaction rate is proportional to the molar concentrations of the products or reactants. The reaction rates are defined as:

$$R_{f,r} = 1000 \left[k_{f,r} \prod_s (0.001\rho_s / M_s)^{\alpha_{s,r}} \right] \tag{2.78}$$

$$R_{b,r} = 1000 \left[k_{b,r} \prod_s (0.001\rho_s / M_s)^{\beta_{s,r}} \right] \tag{2.79}$$

Here, $k_{f,r}$ and $k_{b,r}$ are the forward and backward reaction rate coefficients. The factors 1000 and 0.001 are required because the rate coefficients are in terms of cgs units for most chemical kinetics models, including the models specified in Appendix A.

For this work, two different models are used to specify the forward and backward reaction rate coefficients. The Park model [58] is widely used in similar CFD simulations and appears to give more accurate than competing reaction rate specifications [56]. For the Park model, the forward reaction rate coefficient is specified directly with an Arrhenius relationship, and the backward reaction rate coefficient is calculated based on the equilibrium constant, $K_{eq,r}$.

$$k_{f,r} = C_{f,r} T_a^{\eta_{f,r}} e^{-\frac{E_{f,r}}{k_B T_a}} \tag{2.80}$$

$$K_{eq,r} = e^{B_1^r + B_2^r lnZ + B_3^r Z + B_4^r Z^2 + B_5^r Z^3} \tag{2.81}$$

$$k_{b,f} = \frac{k_{f,r}}{K_{eq,r}} \tag{2.82}$$

In these equations, k_b is Boltzmann's constant, T_a represents the rate controlling temperature, which is a combination of the translation-rotational and vibrational-electronic temperature. The other terms, such as the required constants $C_{f,r}$, $\eta_{f,r}$ and B_r, and activation energy $E_{f,r}$, are specified by the Park model and given in Table A of Appendix of A. Finally, Z is given as:

$$Z = \frac{10,000}{T_a} \tag{2.83}$$

where T_a is given in Kelvin. As a matter of implementation, the reaction rates for the Park Model are set to zero when the equilibrium constant is sufficiently small. For each reaction, the temperature at which the equilibrium constant is below machine zero ($1E-14$ for double precision) is determined. For temperatures below this value, the reaction rate is set to zero, signifying that the reaction is negligible.

In addition to the Park model, the Dunn-Kang model [59] is also used for this work. For this model, the forward and backward reaction rate coefficients are specified directly through Arrhenius relationships, given in equation (2.84).

$$k_{f,r} = C_{f,r} T_a^{\eta_{f,r}} e^{-\frac{E_{f,r}}{k_B T_a}} \tag{2.84}$$

$$k_{b,r} = C_{b,r} T_a^{\eta_{b,r}} e^{-\frac{E_{b,r}}{k_B T_a}} \tag{2.85}$$

The variables required for the Dunn-Kang model are given in Table A of Appendix of A [56].

For both models, the rate controlling temperature is a combination of the translation-rotational temperature and vibrational-electronic temperature, given as [51]:

$$T_a = T^{0.7} T_v^{0.3} \tag{2.86}$$

This rate controlling temperature is used for the dissociation reactions while the translation-rotational temperature is used for all other reactions, including recombination.

Because of its consistency with similar CFD tools, the Park model is used for the validation flow results presented in this section, while the Dunn-Kang model is used for the flow solutions and gradient evaluations required for the subsequent sensitivity analysis and uncertainty quantification results due to the ease with which uncertainty parameters can be specified for both the forward and backward rates.

2.3.5 Energy Coupling

The relaxation between translation energy and vibrational energy is modeled using the energy coupling source term, Q_{T-V}. This term accounts for the exchange between translation energy and vibrational energy. The exchange rate is approximately equal to the difference between the two energies divided by a characteristic relaxation time. The expression used for this exchange rate is given in equation (2.87) [56].

$$Q_{t-v} = \sum_{molecules} \frac{C_{v,v}^s (T - T_v)}{< \tau_s >} \tag{2.87}$$

Here, $C_{v,v}^s$ is the vibrational-electronic specific heat for each species, and $< \tau_s >$ is the relaxation time for each species. Because this coupling is only for vibrational energy, the sum is only over the polyatomic molecules. The calculation of the relaxation time is based on a blending between two different relaxation times. The first is due to Millikan and White and is given by equation (2.88) [60].

$$\tau_s^{MW} = \frac{\sum_r n_r e^{A_s (T^{-1/3} - 0.015 \mu_{s,r}^{1/4}) - 18.42}}{p \sum_r n_r} \tag{2.88}$$

In this equation, n_r is the number density of species r and p is the pressure measured in atmospheres. A_s is given in Table A of Appendix A. The reduced mass, $\mu_{s,r}$, is defined as:

$$\mu_{s,r} = \frac{M_s M_r}{M_s + M_r} \tag{2.89}$$

The second relaxation time is due to Park [58] and given in equation (2.90).

$$\tau_s^P = (\sigma \bar{c_s} n_s)^{-1} \tag{2.90}$$

In this equation, n_s is again the number density, σ is the collision cross-section and $\bar{c_s}$ is the average molecular velocity. This velocity is given by:

$$\bar{c_s} = \sqrt{\frac{8\bar{R}T}{\pi M_s}} \tag{2.91}$$

and the cross-section is modeled as [51]:

$$\sigma = 10^{-21} \left(\frac{50,000}{T} \right)^2 \tag{2.92}$$

where σ is assumed constant over all species. The final relaxation time is simply the sum of the two different models.

$$< \tau_s >= \tau_s^{MW} + \tau_s^P \tag{2.93}$$

2.4 Simplified Models

In order to provide inexpensive computational models, simplified versions of the above real gas model can be constructed. To provide a simplified model for the CFD solver, a perfect gas model can be constructed by neglecting reaction rates and internal energy modes. To provide an explicit function capable of approximating hypersonic flow results but suitable for testing uncertainty quantification strategies, the Fay-Riddell stagnation heating correlation is used.

2.4.1 Perfect Gas Idealization

To provide a simplified physical model for the CFD solver, the previously described real gas can be reduced to a perfect gas model. Although none of the results in this dissertation use this model, the perfect gas model was used to prototype the CFD solver and the gradient-based methods for uncertainty quantification and sensitivity analysis. For completeness, the reduction of the real gas model to a perfect gas model is given. In order to arrive at the perfect gas model, several assumptions must be made. First, chemical reactions and dissociation are ignored and the flow is modeled as a mixture of molecules whose composition does not change. With this assumption, the species mass equations can be combined into a single conservation equation for the mixture. Second, the vibrational-electronic energy modes are assumed to be unexcited and thus contain zero energy ($e_{v,e} = 0$). This assumption has two consequences. First, the second energy equation is no longer required as this energy is assumed zero. Second, the specific heat will consist exclusively of the translational and rotational components. Because these modes are fully excited at the temperatures typical of most flows, this fact implies that the specific heat is constant.

From these assumptions, the perfect gas variables and fluxes are given as:

$$\mathbf{U} = \left\{ \begin{array}{c} \rho \\ \rho\vec{u} \\ \rho e_t \end{array} \right\} \quad \vec{F} = \left\{ \begin{array}{c} \rho\vec{u} \\ \rho\vec{u}\otimes\vec{u} + P \\ \rho\vec{u}h_t \end{array} \right\} \quad \vec{F}_v = \left\{ \begin{array}{c} 0 \\ \underline{\tau} \\ \underline{\tau}\cdot\vec{u} - \vec{q} \end{array} \right\}$$

In addition to simplifying the equations, many constitutive relations become simplified. In particular, when a constant specific heat is assumed, the speed of sound becomes:

$$c^2 = (1 + \frac{R}{C_v})\frac{P}{\rho} \tag{2.94}$$

$$c^2 = \frac{C_v + R}{C_v}\frac{P}{\rho} \tag{2.95}$$

$$c^2 = \gamma\frac{P}{\rho} \tag{2.96}$$

where R is the gas constant of the mixture and C_v is the translation-rotational specific heat of the mixture. The transport model must also be adjusted for a perfect gas flow. Because the composition of the mixture is constant, the mass diffusivity is no longer required as the effect of species diffusion is no longer important. Additionally, only the translation-rotational thermal conductivity is required as vibrational-electronic energy modes are assumed to be zero. Given these drastic differences, the collision integral based transport model is overly complicated for perfect gas flows and Sutherland's law is instead used.

For Sutherland's law, the viscosity obeys the following equation:

$$\frac{\mu}{\mu_{ref}} = \frac{C_1 T^{3/2}}{T + S} \tag{2.97}$$

where μ_{ref} is the reference viscosity and C_1 and S are model parameters found in any fluid dynamics textbook [49].

Using this viscosity, the thermal conductivity is found by assuming a constant Prandtl number for the flow, with air having a Prandtl number of approximately 0.7.

$$k_t = \frac{C_p \mu}{Pr} \tag{2.98}$$

Here, C_p is the specific heat at constant pressure.

2.4.2 Fay-Riddell Stagnation Heating Correlation

In order to provide a simplified, computationally inexpensive test problem that can mimic the characteristics of hypersonic CFD simulations with respect to model parameters and design space complexity, the Fay-Riddell stagnation heating correlation is examined [54]. This model is an analytic function, permitting the use of exhaustive sampling to validate the uncertainty quantification strategies examined in this paper. The equations for this model are as follows.

$$q" = 0.76\frac{(\rho_w\mu_w)^{0.1}(\rho_e\mu_e)^{0.4}}{Pr_w^{0.6}}\sqrt{\left(\frac{dU_e}{dx}\right)}(h_{o,e} - h_w)\left[1 + (Le^{0.52} - 1)\left(\frac{h_D}{h_{o,e}}\right)\right] \tag{2.99}$$

$$\left(\frac{dU_e}{dx}\right) = \frac{1}{R_N}\sqrt{2\frac{p_e - p_\infty}{\rho_e}} \tag{2.100}$$

$$h_D = \sum_i C_{i,e}\Delta h^o_{f,i} \tag{2.101}$$

In these equations, $q"$ is the heat flux at the stagnation point, Pr represents the Prandtl number, Le is the Lewis number, R_N is the radius of curvature, $\Delta h^o_{f,i}$ is the heat of formation for each species, h is the internal enthalpy and h_o is the total enthalpy. For this equation, the subscript e represents properties at the edge of the boundary layer, and w represents values at the wall. The properties at the edge of the boundary layer are found by solving the normal shock problem using equilibrium properties for air and determining the composition after the shock. From the curve fits for the equilibrium properties of air, the enthalpy downstream of the shock is first written in terms of the density and pressure. A root-finder is then used to determine the downstream density and pressure required to satisfy the Rankine-Hugoniot jump conditions. With this density and pressure determined, all other bulk properties are calculated via curve fits [52]. The composition after the shock is determined via statistical thermodynamics. Through analytic expressions of each species partition function, the equilibrium constant for each dissociation reaction is calculated. These equilibrium constants are then related to the partial pressure of each species and a Newton solver is used to solve for the partial pressure of each species [48, 53]. Using these partial pressures, the mass fraction for each species can be found. Further details of the solution process used for the Fay-Riddell heating correlation can be found in Reference [61] and in Appendix D. The required viscosities and thermal conductivity are evaluated by using the collision integral transport model described in the previous section. This model was chosen in order to maintain the same input parameters between the Fay-Riddell model and the real gas CFD code. The Prandtl number at the wall is computed based on the transport values evaluated at the wall, and the Lewis number is evaluated based on the Prandtl number and assuming a constant Schmidt number of 0.5 [51].

2.5 Summary

In this chapter, the details of the real gas model used throughout this work were given. First, the prerequisite knowledge required from gas kinetics was presented and the properties of high temperature gases were introduced. Next, the continuous equations solved for this model were developed and the required constitutive laws were introduced. Finally, the equations of state for the model as well as the transport, chemical kinetics and energy coupling models were introduced. Contained in these models is a large number of experimentally derived constants and parameters. The effect of these parameters on relevant simulation outputs will be examined within the uncertainty quantification and sensitivity analysis presented in this work. In order to provide simplified computational models to test the proposed uncertainty quantification techniques, a perfect gas model and the Fay-Riddell stagnation heating correlation were also presented. In the next chapter, the numerical methods used to approximate the solution to the governing equations for this model will be detailed.

Chapter 3

Numerical Implementation of Real Gas Model

For this work, a two dimensional finite volume solver utilizing the previously described five species/two temperature real gas model was developed. This solver was developed from the ground up both for practical and pedagogical reasons. The development of a solver from scratch allowed for experimentation with regard to choice of flux function, physical models, and linear and nonlinear solver. Additionally, because it is central to this work, the solver was built to accommodate the implementation and solution of the flow adjoint either through manual or automatic differentiation.

In this chapter, the details of the solver are described. First, the spatial discretization is examined and details of the unstructured finite volume method are given. With the spatial discretization presented, the non-linear solvers used to determine the flow solution is explained. Finally, the linear solvers employed within the two non-linear solvers are given.

3.1 Spatial Discretization

In order to solve problems using the previously described models, the governing equations are first discretized in space and the solution is advanced in time using a non-linear solver. In semi-discrete form, the unsteady equations have the following form:

$$\frac{\partial \mathbf{V}\mathbf{U}}{\partial t} + \mathbf{R}(\mathbf{U}) = 0 \tag{3.1}$$

The term $\mathbf{R}(\mathbf{U})$ is referred to as the residual and represents the result of the spatial discretization. For this work, the finite volume method is used. To apply this method, a mesh is created and the governing equations (2.31) are integrated over the volumes defined by this mesh, denoted as cells.

$$\int_{V_i} \frac{\partial \mathbf{U}}{\partial t} dV + \int_{V_i} \nabla \cdot \vec{F}(\mathbf{U}) dV = \int_{V_i} \nabla \cdot \vec{F}_v(\mathbf{U}) dV + \int_{V_i} \mathbf{S}(\mathbf{U}) dV \tag{3.2}$$

Using the divergence theorem, the integration of the flux terms is converted from an integration over the cell volume to an integration over the surface of the volume, given as equation (3.3).

$$\int_{V_i} \frac{\partial \mathbf{U}}{\partial t} dV + \int_{S_i} \vec{F}(\mathbf{U}) \cdot \vec{n} dA = \int_{S_i} \vec{F}_v(\mathbf{U}) \cdot \vec{n} dA + \int_{V_i} \mathbf{S}(\mathbf{U}) dV \tag{3.3}$$

With the equations in this form, several finite volume approximations are applied. First, the integration over the volume is approximated as a representative cell-centered value (located at the volume centroid) multiplied by the cell volume. Second, the integration over the surface is represented as a sum over the faces of the cell. For volumes that are fixed over time, these approximations result in the following equation:

$$V_i \frac{\partial \mathbf{U}_i}{\partial t} + \sum_k \vec{F}^k(\mathbf{U}) \cdot \vec{N}_k = \sum_k \vec{F}_v^k(\mathbf{U}) \cdot \vec{N} + V_i \mathbf{S}_i(\mathbf{U}) \tag{3.4}$$

where k is a sum over the faces and $\vec{N}_k$ is the dimensional normal for the face. The final discretized equations used in this work are arrived at by first reducing the dimension to two, resulting in the volume V_i being replaced by the area A_i. Next, the face-centered flux is replaced by a flux function that approximates the solution to the Riemann problem at the interface based on the values on either side of the face. Finally, despite its nonlinearity, the cell-centered source term is assumed to be the source evaluated using the cell-centered flow variables. Based on these approximations, the final residual within each cell is given by the sum of the normal inviscid and viscous flux over all faces plus a cell centered contribution

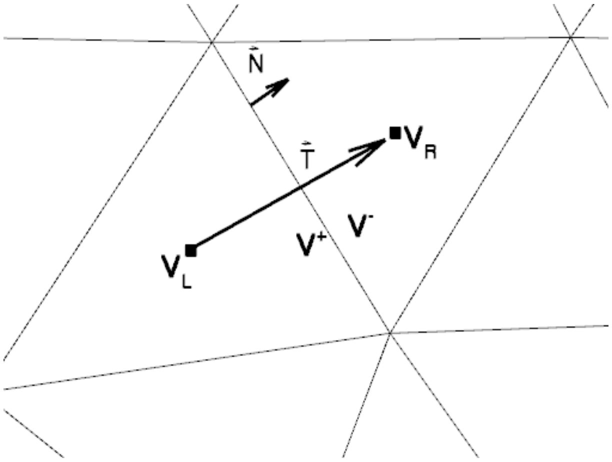


Figure 3.1: Variable definitions at mesh face

due to source terms. The residual equation for cell i is given in equation ((3.5)) along with the functional dependence of each flux function.

$$\mathbf{R}_i(\mathbf{U}) = \sum_k \left[F_n^k(\mathbf{V}^+, \mathbf{V}^-, \vec{N}) - F_{v,n}^k(\mathbf{V}_L, \mathbf{V}_R, \nabla\mathbf{V}_L, \nabla\mathbf{V}_R, \vec{N}, \vec{T}) \right] - A_i \mathbf{S}_i(\mathbf{V}_i) \tag{3.5}$$

In this equation, $\mathbf{V}_L$ and $\mathbf{V}_R$ represent cell centered primitive values for the cell to the left and the right of the face respectively. The variables $\mathbf{V}^+$ and $\mathbf{V}^-$ represent reconstructed primitive variables to the left and right of the face. The variables $\mathbf{S}_i$ represent the source terms computed using the cell-centered primitive variables of cell i. Finally, $\vec{N}$ is the dimensional face normal and $\vec{T}$ represents the vector connecting the two cell centers on either side of the face. These terms are clarified in Figure 3.1.

The inviscid flux is defined exclusively in terms of the primitive variables, $\mathbf{V}$, instead of the conserved variables, $\mathbf{U}$. The conserved variables for this model are the 5 species densities (ρ_s), the two components of linear momentum (ρu and ρv), and the total and vibrational energy (ρe_t and ρe_v). In contrast to these conserved variables, a set of primitive variables is also used. These primitive variables are the five species mass fractions c_s, the two components of velocity (u and v), and the two temperatures (T and T_v). This choice of variables has two major advantages. For the diffusive fluxes, the gradients of these variables are required; hence, performing reconstruction strictly on the primitive variables eliminates the need for two sets of gradients. Reconstruction of primitive variables also increases the robustness of the solver as it ensures the appropriate variables, such as pressure, remain positive throughout the calculation.

For the inviscid flux, F_n, the AUSM+UP flux function is used [62]. This flux function is chosen based on its applicability to a wide range of Mach numbers and the ease with which it can be extended to additional equations. For the real gas model, a frozen speed of sound is used within the inviscid flux function [63]. In order to achieve second-order accuracy, the flux function is evaluated using the reconstructed primitive variables on each side of the face, denoted previously as $\mathbf{V}^+$ and $\mathbf{V}^-$. This reconstruction requires the cell-centered gradient of the primitive variables. This gradient is calculated based on Green-Gauss contour integration. The gradient of a generic variable ζ is given by:

$$\frac{\partial \zeta}{\partial x}_i = \frac{\sum_k \frac{\zeta_L + \zeta_R}{2} N_{x,k}}{A_i} \tag{3.6}$$

$$\frac{\partial \zeta}{\partial y}_i = \frac{\sum_k \frac{\zeta_L + \zeta_R}{2} N_{y,k}}{A_i} \tag{3.7}$$

where the subscript i represents the cell-centered value, k is a loop over the faces, and L and R represent the cell centered value on the "left" and "right" side of the face. The variables N_x and N_y represent the components of the dimensional normal $\vec{N}$. In order to maintain stability, a limiter is used in this reconstruction process. The limiter used within this code is a combination of a pressure switch and smooth Van Albada limiter, inspired by the experiences in references [51] [50], and [63]. The pressure switch in each cell is defined by: [64]

$$\nu_i = \frac{|\sum_k P_R - P_L|}{\sum_k P_R + P_L} \tag{3.8}$$

The limiter value at each face is defined by the continuous function Φ which assigns a limiter value given inputs dependent on the reconstructed value and neighboring cell-centered values [65]. The value at face k for cell i is given as:

$$\Phi(\Delta^+, \Delta^-) = \frac{1}{\Delta^-} \frac{({\Delta^+}^2 + \varepsilon^2)\Delta^- + 2{\Delta^-}^2 \Delta^+}{{\Delta^+}^2 + 2\Delta^- + \Delta^- \Delta^+ + \varepsilon^2} \tag{3.9}$$

where Δ^- and Δ^+ are defined as:

$$\Delta^- = \mathbf{V}_k - \mathbf{V}_i \tag{3.10}$$

$$\Delta^+ = \begin{cases} \mathbf{V}_{max} - \mathbf{V}_i & \text{if } \mathbf{V}_k - \mathbf{V}_i > 0 \\ \mathbf{V}_{min} - \mathbf{V}_i & \text{if } \mathbf{V}_k - \mathbf{V}_i < 0 \end{cases} \tag{3.11}$$

. Here, $\mathbf{V}_k$ is the unlimited reconstructed face value, $\mathbf{V}_i$ is the cell centered value and $\mathbf{V}_{min}$ and $\mathbf{V}_{max}$ represent the minimum and maximum values surrounding cell i. The parameter ε reduces the limiter sensitivity to small changes in the flow, effectively forcing the limiter to unity in smooth regions. This reduced sensitivity allows the limiter to achieve convergence to machine zero provided the parameter ε is high enough [65]. Unfortunately, this improved convergence tends to come at the price of robustness. Hence, the pressure switch is included in the limiter to ensure the solution is first order in the presence of strong shocks. Outside of this small region, the limiter is allowed to adjust the reconstruction. In functional form, the final limiter at each face takes the form:

$$\boldsymbol{\Psi}_k = max(0, 1 - Kmax(\nu_L, \nu_R))\tilde{\boldsymbol{\Psi}}_k \tag{3.12}$$

where $\tilde{\boldsymbol{\Psi}}_k$ is the result of applying the function Φ at each face with the proper inputs and K is a prescribed constant used to set the minimum value of the pressure switch required to force the limiter to zero. The cell centered limiter value is taken to be the minimum of the limiters calculated at each face of the cell. With the limiter calculated, the reconstructed value at each face is given by:

$$\mathbf{V}^+ = \mathbf{V}_L + \boldsymbol{\Psi}_L \nabla \mathbf{V}_L \cdot \Delta\vec{X}_L \tag{3.13}$$

$$\mathbf{V}^- = \mathbf{V}_R + \boldsymbol{\Psi}_R \nabla \mathbf{V}_R \cdot \Delta\vec{X}_R \tag{3.14}$$

where $\Delta\vec{X}_L$ and $\Delta\vec{X}_R$ are the position vectors connecting the face center with the left and right cell centers respectively.

The viscous flux is a function of both the cell centered primitive variables as well as the cell centered gradients of the primitive variables. The calculation of the viscous flux is relatively straight forward. Each term in the viscous flux from equation ((3.5)) is calculated using face centered quantities. In the case of any primitive variables, a simple average of the left and right states is used and transport quantities are calculated using this averaged state. Face-based gradients are calculated through the following equation which ensures a coupling between neighboring cells and a straight forward approximate linearization: [66]

$$\nabla \mathbf{V}_k = \tilde{\nabla \mathbf{V}} + \frac{\mathbf{V}_R - \mathbf{V}_L - \tilde{\nabla \mathbf{V}} \cdot \Delta T}{|\Delta T|} \frac{\Delta T}{|\Delta T|} \tag{3.15}$$

In this equation, $\tilde{\nabla \mathbf{V}}$ is a simple average of the left and right cell centered gradients and the vector ΔT is defined as before.

The source term is calculated using the cell-centered primitive variables, evaluating the reaction and energy coupling terms using these variables. The cell-centered source term is then weighted by the cell area (or volume in 3D) to account for the integration over the volume of the source term.

3.2 Nonlinear Solver

In this section, the non-linear solvers used to determine the solution of the spatial discretization are outlined. Two different non-linear solvers are required to determine the flow solution: a start-up solver and a full-convergence solver. The start-up solver is a pseudo-time unsteady solver used to overcome start-up transients caused by a poor initial flow field. Once these transients have been overcome, a steady-state solver based on Newton's method is used to rapidly drive the residual to zero.

3.2.1 Pseudo-time Solver

The result of the previously described spatial discretization is a system of coupled ordinary differential equations. These ODE's are advanced toward steady-state using the first-order backward difference formula (BDF1 or Backward Euler) given by:

$$\frac{\partial \mathbf{U}}{\partial t} \approx \frac{\mathbf{U}^n - \mathbf{U}^{n-1}}{\Delta t} \tag{3.16}$$

The result of this temporal discretization is a system of nonlinear equations that must be solved at each time step. These nonlinear equations are represented by the unsteady residual **J**. Applying this temporal discretization to equation (3.1), the unsteady residual is given by:

$$\mathbf{J}(\mathbf{U}^n, \mathbf{U}^{n-1}) = \frac{\mathbf{U}^n - \mathbf{U}^{n-1}}{\Delta t} + \mathbf{R}(\mathbf{U}^n) = 0 \tag{3.17}$$

To solve this nonlinear equation, an approximate Netwon's method is employed. In general, a Newton's method for this problem takes the following form:

$$\delta\mathbf{U}^k = -\left[\frac{\partial\mathbf{J}(\mathbf{U}^k, \mathbf{U}^{n-1})}{\partial\mathbf{U}^k}\right]^{-1} \mathbf{J}(\mathbf{U}^k, \mathbf{U}^{n-1}) \tag{3.18}$$

$$\mathbf{U}^{k+1} = \mathbf{U}^k + \delta\mathbf{U}^k \tag{3.19}$$

Within this general form, several approximations are introduced to increase the robustness of the solver and improve its applicability to steady problems. First, instead of fully converging the nonlinear problem, only a set number of Newton iterations is performed per time step (typically 10). Additionally, local time-stepping is used within the formulation. Using local time-stepping, the solution in each cell is allowed to advance with a different time-step, accelerating convergence to steady-state. To decrease the time required per Newton iteration, the exact Jacobian matrix is replaced by a preconditioning matrix that is calculated once at the beginning of the time step and frozen for the duration of the time step. Typically, this preconditioning matrix is based upon a robust and easy to calculate simplified Jacobian. Finally, transport quantities such as viscosity, thermal conductivity and species diffusion coefficients can be frozen within the Newton iterations to further decrease computational cost. Incorporating these simplifications, the sub-iterations within each timestep take the following form.

$$\delta\mathbf{U}^k = -[\tilde{P}]^{-1} J(\mathbf{U}^k, \mathbf{U}^{n-1}) \tag{3.20}$$

$$\mathbf{U}^{k+1} = \mathbf{U}^k + \lambda\delta\mathbf{U}^k \tag{3.21}$$

In these equations, $[\tilde{P}]$ is the frozen preconditioning matrix and the variable λ is used to ensure that updates remain within a specified percentage of the previous solution value [67].

The preconditioner matrix consists of an unsteady term and contributions from the spatial residual linearization. In order to preserve a nearest-neighbor sparsity pattern, only the first-order Jacobian is used. For the inviscid flux Jacobian, the Van-Leer-Hänel flux function [68] is linearized, as opposed to the AUSM+UP flux function. The presence of additional dissipation in the Van-Leer-Hänel flux function yields a Jacobian with increased diagonal dominance. For the viscous Jacobian, only the nearest neighbor edge terms in equation (3.15) for the face-centered gradient are linearized. By ignoring the derivative of the gradient with respect to the flow variables, the stencil of the viscous Jacobian remains nearest-neighbor. Additionally, the transport term linearization is neglected to reduce the cost of constructing the viscous Jacobian. Because the source term is exclusively a cell-centered quantity, the linearization can be performed exactly; however, due to the complexity, the Jacobian is computed using automatic differentiation (AD) with the AD engine Tapenade [69]. More information regarding the use of automatic differentiation is given in Chapter 4.

For the sake of stability, a relaxation term is applied to each component of the Jacobian. Hence, the final preconditioner matrix is given as:

$$[P] = \frac{A}{\Delta t} + r_i \left[\frac{\partial \mathbf{R}_i}{\partial \mathbf{U}}\right]_{1o-VLH} + r_v \left[\frac{\partial \mathbf{R}_v}{\partial \mathbf{U}}\right]_{edge} + \left[\frac{\partial \mathbf{R}_s}{\partial \mathbf{U}}\right] \tag{3.22}$$

where the subscript i corresponds to inviscid components, v corresponds to the viscous components and s stands for the source terms. The relaxation factors r_i and r_v are added to increase the stability of the non-linear solver and should take on values greater than 1.5 for the inviscid component and 0.5 for the viscous component. Because these values represent minimum requirements, relaxation factors of 5 and 3 for the inviscid and viscous terms respectively are typical for the solver [70].

The pseudo-timestep is determined based on a CFL number uniform over the domain and a local timestep is computed based on the inviscid stability criteria. The timestep for cell i for a given CFL number is calculated as [64]:

$$\Delta t_i = CFL \frac{A_i}{\sum_k |\vec{U}_k \cdot \vec{N}_k| + C_k |\vec{N}_k|} \tag{3.23}$$

where $\vec{N}_k$ is the dimensional face normal, and $\vec{U}_k$ and C_k are the face centered velocity and sound speed respectively, evaluated by simply averaging the cell-centered quantities on each side of the face.

The CFL itself is varied during the simulation, increasing with the number of timesteps performed. This increase takes the form of a power-law whereby the next CFL is the previous CFL multiplied by some specified rate. The CFL variation as a function of pseudo-timestep is therefore given as:

$$CFL(n) = min(\alpha^n CFL_o, CFL_{max}) \tag{3.24}$$

where CFL_o is the starting CFL,α is the rate of increase and CFL_{max} is the maximum CFL allowed to ensure diagonal dominance of the linear system. Because of the presence of strong starting transients, the initial CFL is typically taken to be 1×10^{-2} and the rate is typically between 1.001 and 1.01. The maximum CFL depends on the linear solver used within the solver. For linear solvers requiring diagonal dominance, the CFL must be capped, with a typical value for the Jacobi or Gauss-Seidel linear solver equal to 100 and for the line-implicit linear solver as 1000. These linear solvers are described in detail in Section 3.3.

3.2.2 Newton Solver

Once start up transients have been overcome, an exact Newton solver is used to accelerate the convergence to machine zero. This solver can be arrived at from the pseudo-unsteady solver by increasing the pseudo-time step to a large value ($CFL \sim 10^{12}$), causing the unsteady terms in the residual to reduce to approximately zero. The steps performed at iteration k are given in equation (3.25).

$$\left[\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right] \delta \mathbf{U}^k = -\mathbf{R}(\mathbf{U}^k) \tag{3.25}$$

$$\mathbf{U}^{k+1} = \mathbf{U}^k + \delta \mathbf{U}^k \tag{3.26}$$

For a Newton solver, the exact second-order Jacobian must be inverted. This inversion is performed using a preconditioned GMRES solver. For this solver, only Jacobian vector products are required; hence, the exact Jacobian does not need to be explicitly calculated and stored and the effect of the Jacobian can be built up on the fly. The steps in the preconditioned GMRES algorithm are given as [71]:

$$[P]\,\mathbf{q}_j = \mathbf{v}_j \qquad \text{Preconditioning} \tag{3.27}$$

$$\mathbf{w}_j = \left[\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right] \mathbf{q}_j \qquad \text{Matrix-vector product} \tag{3.28}$$

$$H_{i,j} = \mathbf{w}_j \cdot \mathbf{v}_i \qquad \text{Hessenberg Matrix} \tag{3.29}$$

Here, $[P]$ is the same preconditioner used in the start-up solver, namely the first-order Van-Leer-Hänel Jacobian inverted using either Jacobi, Gauss-Seidel or line smoothing. The Jacobian-vector product is built-up in one of three ways for this work: finite-differences, complex differentiation and automatic differentiation. Each of these approaches have different trade-offs regarding speed, ease of implementation and accuracy with automatic differentiation preferred throughout this work.

For the finite-difference approach, the Jacobian-vector product is constructed by equating it with the appropriate Frechet derivative and approximating the derivative through finite difference.

$$\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\mathbf{q} = \lim_{h \to 0} \frac{\mathbf{R}(\mathbf{U} + h\mathbf{q}) - \mathbf{R}(\mathbf{U})}{h} \tag{3.30}$$

$$\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\mathbf{q} \approx \frac{\mathbf{R}(\mathbf{U} + \epsilon\mathbf{q}) - \mathbf{R}(\mathbf{U})}{\epsilon} \tag{3.31}$$

Here, ϵ is the finite-difference step size. This step size affects the overall accuracy of the finite difference approximation and must be chosen heuristically. The step size must be chosen such that it is small enough to provide a sufficiently accurate approximation of the derivative but large enough such that finite precision arithmetic does not corrupt the approximation. The step size used for this work is given by:

$$\epsilon = \frac{\sqrt{(1 + |\mathbf{U}|_2)\epsilon_m}}{|\mathbf{q}|_2} \tag{3.32}$$

where ϵ_m is the tolerance to which R can be evaluated (5×10^{-10} for these calculations) [72].

The implementation of the finite-difference GMRES solver is relatively straight forward and requires nothing more than the code required to evaluate the residual. To construct the Jacobian-vector product, the flow variables are perturbed by the preconditioned Krylov vector q weighted by the finite difference step ϵ. A new residual is then evaluated using these perturbed flow variables and the difference between the two residual values is the Jacobian-vector product. Despite the ease of implementation, the choice of proper step size as the problem approaches full convergence can be difficult, jeopardizing the desirable properties of the exact Newton method.

In order to improve on the finite-difference approach, complex differentiation is used to construct the Jacobian-vector product. For complex differentiation, the subroutines used to compute the residual are modified to operate on complex numbers instead of real. When this substitution is made and the proper inputs are supplied, the complex part of these variables can be used to obtain the derivative of the variable. Further information on complex differentiation is given in Chapter 4 and Appendix B. For the GMRES solver, the Jacobian-vector product can be calculated by first creating complex flow variables with the real part corresponding to the flow variables $\mathbf{U}$ and the imaginary part proportional to the preconditioned Kylov vector $\mathbf{q}$. The complex flow variables are then used as an input to the complex residual subroutines. The imaginary part of the output of these complex residual subroutines is the desired Jacobian-vector product. These steps are given in equation (3.33).

$$\tilde{\mathbf{U}} = \mathbf{U} + i\epsilon\mathbf{q} \tag{3.33}$$

$$\tilde{\mathbf{R}}(\tilde{\mathbf{U}}) = \mathbf{R}(\mathbf{U}) + i\epsilon \left[\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right] \mathbf{q} \tag{3.34}$$

Here, i is the imaginary number and ϵ is the step size for the differentiation. In order to calculate the Jacobian-vector product accurately, the step size should be as small as possible. As opposed to finite difference, there is no lower limit for the choice of this step size as no arithmetic is required for the derivative prediction, as the Jacobian-vector product is the imaginary part of the complex residual divided by the complex step ϵ. A step size of 10^{-200} is used throughout this work. Although not as easy to implement as the finite-difference approach, the complex-differentiation approach is still relatively easy to implement as it simply requires the recasting of variables from type "real" to type "complex" and the overloading of some operators. Additionally, the increased accuracy provided by the complex differentiation, due to the flexible choice of step size, provides better convergence to machine zero than the finite difference approach. The downside of the complex differentiation approach is the increased computational overhead associated with complex operators.

Finally, in order to increase the accuracy of the Jacobian-vector product without the dramatic increase in the cost associated with the complex approach, automatic differentiation can be used to construct the derivative. Automatic differentiation is the use of a "compiler-like" tool to differentiate the individual operators of a code to produce a new code that is able to compute the derivative of the original code outputs. Automatic differentiation is especially suited for applications requiring derivative-vector products. As part of this work, forward differentiated subroutines for the residual calculation are generated for the purposes of sensitivity analysis. Because these subroutines are already available from the sensitivity analysis, they can be easily used within the GMRES-based flow solver. The details of the forward differentiation are given in Chapter 4. When the forward linearization of the residual is available for the purposes of sensitivity analysis, the use of these subroutines in GMRES represents the best combination of accuracy and computational speed as the Jacobian-vector product is exact and the increase in cost is only associated with calculating the linearization, which is typically only 2-3 times the cost of the original operator [73]. For these reasons, the AD-based GMRES is used exclusively for the presented results.

3.3 Preconditioner Techniques

Until now, the techniques used to invert the approximate Jacobian matrix have been listed but not explained. In this section, the techniques used to invert the Jacobian matrix are given. Because these techniques are used to precondition the GMRES solver, the various solvers are referred to as preconditioners.

The preconditioners can generally be divided into point and line solvers. To examine these preconditioners, they are cast as linear solvers. The general form of these preconditioners is found by placing the linear equation in the following form:

$$[A]\, x = b \tag{3.35}$$

$$[A]\left(x^{k+1} - x^{k}\right) = b - [A]\, x^{k} \tag{3.36}$$

$$[P]\, \delta x^{k} = b - [A]\, x^{k} = -r^{k} \tag{3.37}$$

In order to be an effective preconditioner, the matrix P^{-1} must approximate the matrix A^{-1}. For the preconditioners used for this work, a matrix is computed and stored. The first-order Jacobian given in equation (3.22) has a nearest neighbor stencil, allowing it to be calculated and stored easily. Additionally, the Van-Leer-Hänel flux function is used to construct the Jacobian, as this flux function gives enhanced diagonal dominance and allows for larger pseudo-time steps to be used.

Each of the preconditioners presented in this section represent different approximations applied to the Jacobian given in equation (3.22).

3.3.1 Point Solvers

The first preconditioner corresponds to a Jacobi solver. For this preconditioner, the block diagonal elements of the first-order Jacobian are inverted to determine the solution update. When this preconditioner is applied multiple times to the linear system within the Newton iteration, it is referred to as Jacobi. When it is applied once before moving to the next non-linear iteration, it is often referred to as a point-implicit solver (or non-linear Jacobi). The Jacobi preconditioner is given below.

$$[D]\,\delta x^k = -r^k \tag{3.38}$$

Because a system of equations is solved for each cell, the matrix D is organized into blocks and these blocks are inverted to yield the solution update. The block size for the five species, two temperature model is 9×9 for each cell and is inverted using LU factorization. For the point-implicit solver, the right-hand side of equation (3.38) is the non-linear residual, while the Jacobi linear solver requires the linear residual on the right-hand side, given as:

$$r^k = [A]x^k - b \tag{3.39}$$

The Jacobi preconditioner can be improved by incorporating the lower triangular part of the Jacobian matrix. This preconditioner is referred to as Gauss-Seidel and often will give twice the convergence rate of Jacobi [74].

$$[L]\,\delta x^k = -r^k \tag{3.40}$$

Here, $[L]$ is a lower triangular block matrix. Using forward substitution, this matrix can be inverted exactly in a single pass. Assuming $[L]$ is sparse, inverting this lower triangular matrix is only marginally more expensive than inverting the Jacobi preconditioner. Algorithmically, the Gauss-Seidel solver is given as:

$$\delta x_i^k = -\left[D_i\right]^{-1}\left(r_i^k - \sum_{j<i}\left[O_{j,i}\right]\delta x_j^k\right) \tag{3.41}$$

where $[D]_i$ represent the diagonal blocks of the Jacobian and $[O]$ are the off-diagonal blocks of the Jacobian matrix. As this equation demonstrates, the Gauss-Seidel solver is inherently sequential, with the update for cell i requiring the updates for the cells that have already been computed ($j < i$).

In order to allow the preconditioner to be inverted in parallel, a cell coloring strategy is employed. For this strategy, the cells are divided into groups (or colors) such that the elements in each group share no neighbors with other elements in the group. With the cells colored in this way, the inverse is performed on a color by color basis and the off-diagonal

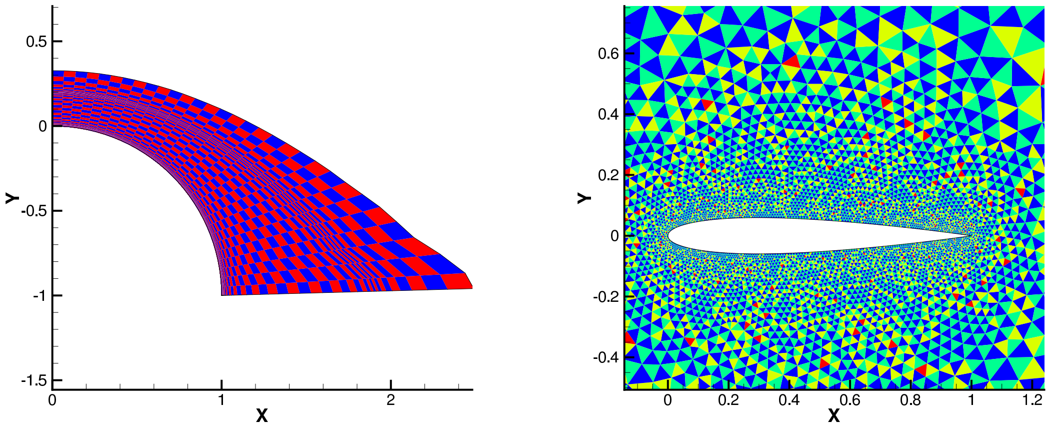


Figure 3.2: Cell coloring for (Left) structured quadrilateral mesh, and (Right) unstructured triangular airfoil mesh.

elements consist of the previously calculated colors. The form of the preconditioner matrix for colored Gauss-Seidel is depicted for three colors below:

$$\begin{bmatrix} [D]_r & 0 & 0 \\ [O]_{b,r} & [D]_b & 0 \\ [O]_{g,r} & [O]_{g,b} & [D]_g \end{bmatrix} \begin{bmatrix} \delta x_r^k \\ \delta x_b^k \\ \delta x_g^k \end{bmatrix} = \begin{bmatrix} -r_r^k \\ -r_b^k \\ -r_g^k \end{bmatrix} \tag{3.42}$$

Here, $[D]$ is a diagonal block matrix for each color and $O_{s,r}$ is a matrix consisting of the off-diagonal elements for color s that come from color r. The colored Gauss-Seidel solver is implemented by determining the update for the first color and using this solution to build up the subsequent updates for the other colors. Due to the nature of the coloring and the matrix $[D]$, the update for a given color can be computed in parallel.

The number of colors required for the solver is dependent on the topology of the mesh used for the solution. For a structured quadrilateral grid, only two colors are required and the solver is often referred to as red-black colored Gauss-Seidel. For an unstructured triangular grid, four colors are required to divide the mesh into appropriate groups. Representative colored meshes can be found in Figure 3.2.

3.3.2 Line Solver

In order to overcome the stiffness caused by the anisotropic grid stretching encountered in boundary layer flows, a line relaxation algorithm is employed in this work. For this algorithm, lines are constructed across the boundary layer and these lines are treated as the fundamental units of the linear solver. In this way, the Jacobian matrix is divided into blocks corresponding to each line. With the basic blocks of the solver identified, Jacobi is used to invert the matrix once it has been partitioned into line blocks. The Jacobi line solver is represented as:

$$[M]\,\delta x^k = -r^k \tag{3.43}$$

where $[M]$ is a diagonal block matrix with the blocks corresponding to each line. The structure of this line block matrix is itself a tridiagonal block matrix that is inverted to determine the update for each line.

The lines are constructed by first starting at the boundaries of the mesh where a no-slip boundary condition is enforced. These areas should also correspond to the boundaries from which the boundary layer mesh is grown. For each cell on this boundary, a line is initialized using the face on the boundary and the first cell off the boundary. With the line initialized, the algorithm for adding new cells is the following. Using the last face added to the line, the dot product between the unit normal of this face and the other faces in the current cell is calculated. The face with the largest dot product and the corresponding cell across this face are then added to the line. The process is then continued using the newly added face and cell as the basis for adding additional cells. The line creation process is terminated once the angle between the face normals is less than 0.7 radians. At this point, any cells not incorporated into a line are treated as their own lines with length 1. For these cells, the line solver merely reduces to a point Jacobi scheme. The lines constructed for a quadrilateral mesh used for hypersonic calculations and an airfoil mesh are depicted in Figure 3.3. Within these figures, the lines are denoted by the thick black lines.

With the lines created, the line connectivity can be used to partition the Jacobian matrix into a series of blocks, the diagonal of which is a tridiagonal block matrix. The Jacobi scheme

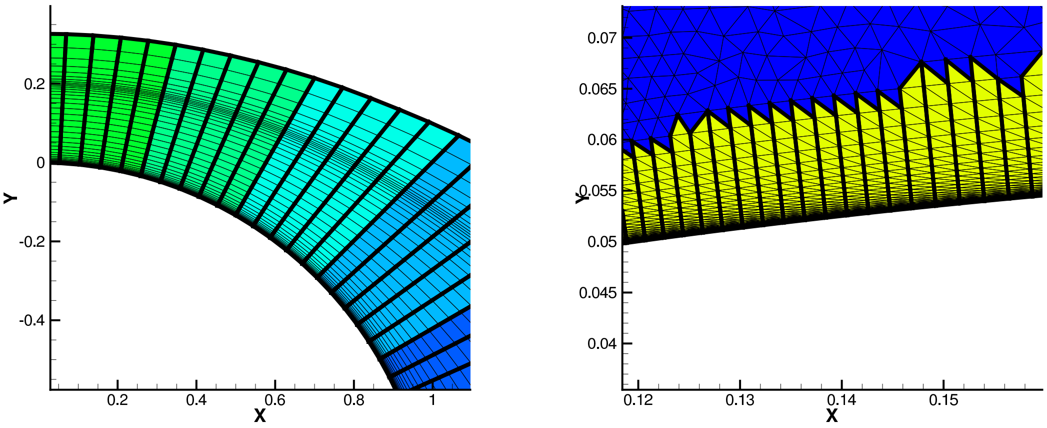


Figure 3.3: Lines constructed for (Left) structured quadrilateral mesh, and (Right) unstructured triangular airfoil mesh.

is applied to this partitioned Jacobian whereby only the tridiagonal component is inverted to advance the solution. When a single iteration of this line relaxation is applied, the scheme is referred to as line implicit. The tridiagonal matrix is inverted using the Thomas algorithm, which can perform LU factorization on the tridiagonal matrix with a cost directly proportional to the length of the line [74]. In addition to applying the Jacobi scheme, a Gauss-Seidel line solver can also be constructed. By constructing a connectivity between the line elements, the lines can be colored using an algorithm similar to the one presented for the cells. Using these colored lines, the Gauss-Seidel algorithm can be implemented by incorporating the previously updated lines on the right hand side of equation (3.43). Although the use of a Gauss-Seidel line solver would improve the convergence rate of the preconditioner, only a Jacobi line solver is used for this work due to the complexity of constructing the connectivity between line elements.

3.4 Demonstration Flow Results

To demonstrate the capability of the solver and provide validation for both the model and numerics, the $5km/s$ standard test case is used [45]. For this case, the flow over a cylinder is considered in two dimensions. A fixed temperature is applied to the surface of the cylinder and the cylinder is assumed to be super-catalytic, meaning that the surface induces the species to recombine. In practice, this boundary condition is equivalent to enforcing the species concentrations at the wall to match the freestream concentration. In addition to being super-catalytic, the surface also induces thermal equilibrium, meaning that the same temperature is enforced for the translation-rotational temperature and the vibrational-electronic temperature [45]. The results produced by the developed finite volume solver were compared against those produced by the thoroughly validated NASA code LAURA [70]. The flow conditions for this test case are given in Table 3.1.

Table 3.1: Benchmark flow conditions

$$\begin{aligned} V_\infty &= \text{5 km/s} \\ \rho_\infty &= 0.001\ kg/m^3 \\ T_\infty &= \text{200 K} \\ T_{wall} &= \text{500 K} \\ M_\infty &= 17.605 \\ Re_\infty &= 376{,}930 \\ Pr_\infty &= 0.72 \end{aligned}$$

The chemical kinetics model used for this flow solution is the Park model [58] as this model gave results in better agreement with those from LAURA. The mesh used for this test case was the grid generated by LAURA and used for the validation results produced by that code. Although the mesh used in LAURA is structured, the mesh is treated as unstructured within the developed finite-volume framework. The mesh itself consists of 64 cells normal to the cylinder surface and 60 tangential to the surface. Using the mesh generated by LAURA had two advantages. First, it eliminated the variable of mesh resolution and quality when comparing the finite volume results to those of LAURA. Hence, discrepancies in the results are due to modeling differences and differences in spatial discretization, such as flux function, gradient calculations and viscous discretization. The second advantage of using the same mesh is that the mesh produced by LAURA is aligned with the shock. For hypersonic flows, calculations on unstructured grids will typically contain spurious oscillations in the surface heating. The non-smooth surface heating can be addressed by aligning the edges in the mesh with the bow shock. For structured grids, this shock alignment is relatively well established;

however, for unstructured grids, this shock alignment is a topic of active research, especially in three dimensions [75,76]. Because this alignment is beyond the scope of this work, shock aligned meshes produced by LAURA have been employed exclusively.

To solve this problem, the start-up pseudo-time solver is first used. For this solver, the Jacobi point preconditioner is used. The Jacobian matrix is frozen and 10 quasi-non-linear steps are performed for each pseudo-time step. The constant used within the pressure switch is $K = 2$ and the constant within the limiter is 50. The CFL for this start-up problem was specified by the power law, with a starting value of 1×10^{-2}, a maximum value of 100 and a growth rate of 1.001 per timestep. Using this CFL, a local timestep for each cell was calculated based on the inviscid eigenvalue. Finally, the relaxation factors on the inviscid and viscous Jacobians were 5 and 3 respectively.

Once the residual was reduced to 1×10^{-4}, the exact Newton solver was engaged. At this point, the number of subiterations within a pseudo-timestep was reduced to 1 and the CFL was allowed to grow to a maximum of 1×10^{12} with a growth factor of 10 per iteration. To solve the linear system at each nonlinear iteration, the line preconditioned GMRES solver was used. For the preconditioner, a single iteration of line relaxation was performed using the first-order Van-Leer-Hänel Jacobian. Given the small problem size, a maximum of 500 GMRES vectors was allowed; however, this limit was never reached in the calculation and in practice, approximately 100 GMRES vectors were required to invert the linear system when the line-implicit solver was used as a preconditioner.

The convergence of the flow solution can be found in Figure 3.4 as a function of iteration and Figure 3.5 as a function of CPU time. As the results show, the solver convergence is irregular for the first 10000 iterations as start-up transients are overcome and the CFL is slowly allowed to grow to a reasonable value. Once the convergence has become regular and the exact Newton solver is engaged, the residual initially spikes due to the sharp growth in the CFL. After this spike, the solver encounters another mild start-up transient where the residual is flat, followed by rapid convergence in the residual once the solution is within the vicinity of the true solution.

In Figure 3.6, a temperature contour for the flow field is plotted. Additionally, the tem-

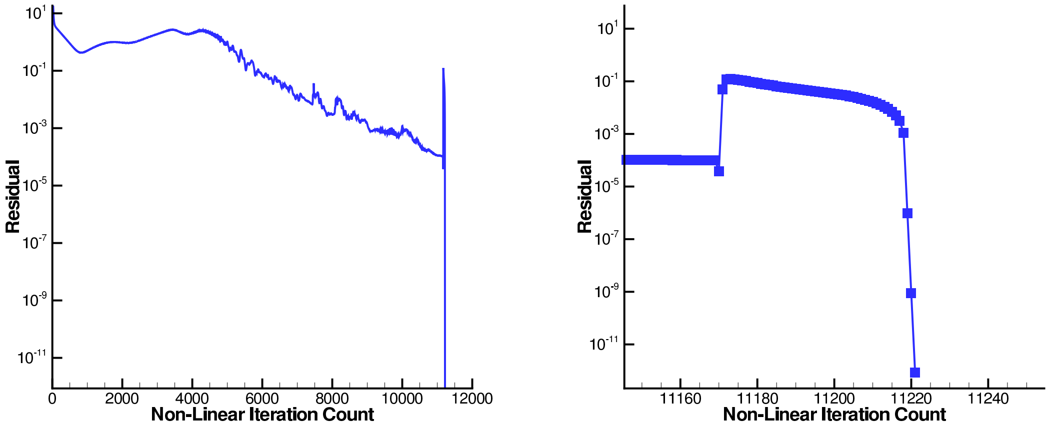


Figure 3.4: Left: Full solver convergence for $5km/s$ flow case using $34,560$ unknowns as a function of non-linear iteration count. Right: Exact newton solver convergence for same test case.

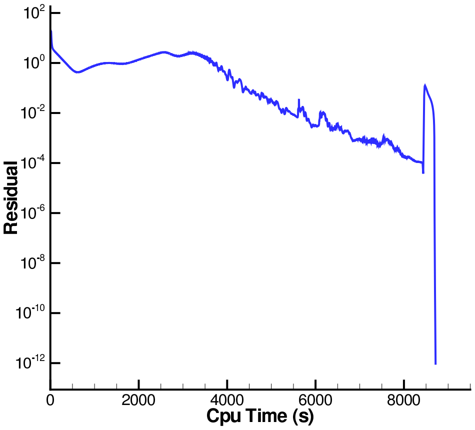


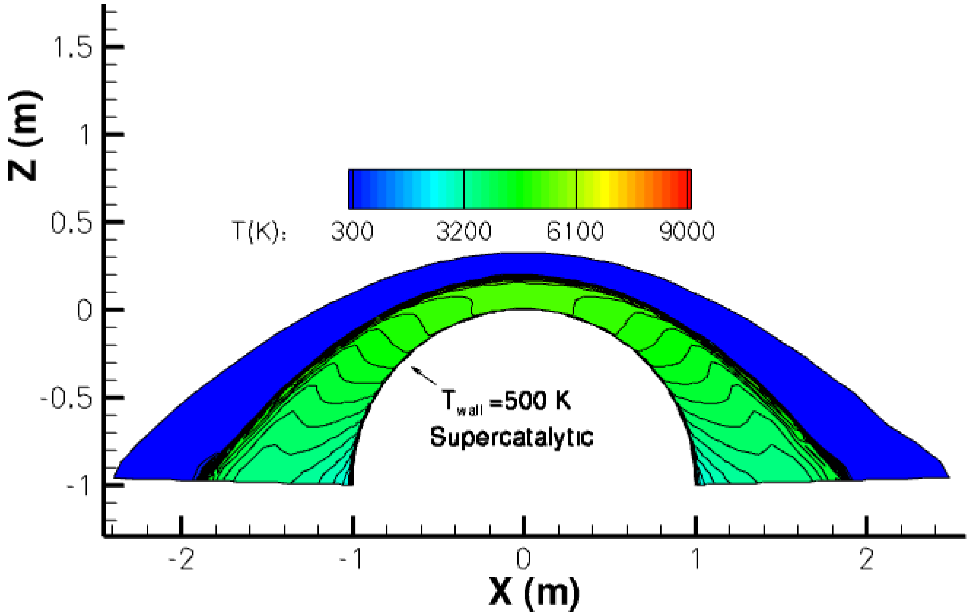
Figure 3.5: Solver convergence for $5km/s$ flow case using $34,560$ unknowns as a function of CPU wall time executed sequentially on an Intel core i7-870 clocked at 2.93 GHz.

perature along the stagnation streamline is plotted and compared to the temperature profile from LAURA. In Figure 3.7, the surface heating distribution and skin friction distribution are plotted and compared to the surface distributions from LAURA.

As the results show, the finite-volume code produces results in good agreement with those from LAURA. The differences in the results can reasonably be attributed to the various differences between the two codes. While the same mesh is used for the two simulations, the finite volume code uses a slightly different chemical kinetics model and a different spatial discretization. While the Park model is used in both codes, LAURA uses a more up to date version of the model as the coefficients in the Park model are continuously improved to match new experimental results. Additionally, the gradient calculation, viscous discretization and flux function differ between the two codes. While this work uses the AUSM+UP flux function [62], the LAURA solver uses Roe's approximate Riemann solver [77]. The LAURA solver is also structured, giving inherently different gradient calculations and viscous discretizations when compared to the unstructured finite-volume solver developed for this work. Despite these differences, the two codes give similar results, especially for the surface distributions, giving an indication that the physical model has been implemented and solved correctly.

3.5 Summary

In this chapter, the solver used to approximate the solution of the governing equations for the real gas fluid model described in Chapter 2 was presented. First, the application of the finite volume method to the governing equations was presented. Next, the details of the spatial discretization were given. The result of this spatial discretization is a set of couple ordinary differential equations. These ODE's are advanced using a pseudo-time stepping scheme to overcome start-up transients and a Newton solver to accelerate the convergence to steady-state. Additionally, the preconditioners used to advance the linear systems encountered in the nonlinear solver were explained. Finally, The performance of the developed CFD solver was demonstrated for the $5km/s$ flow over a cylinder.



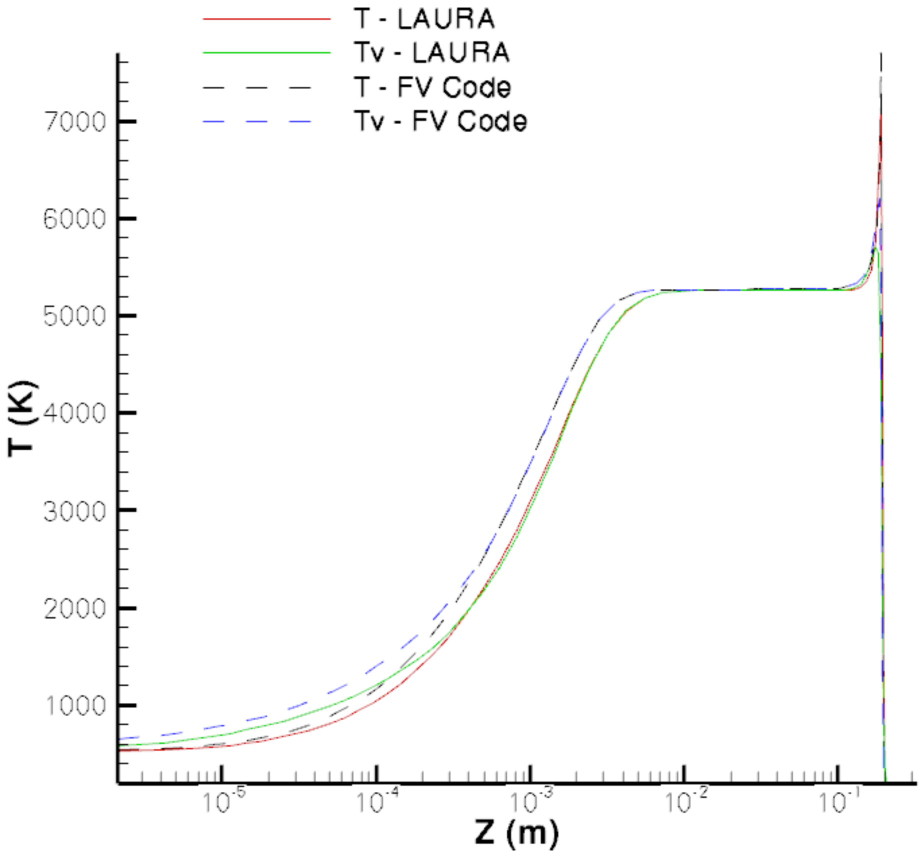


Figure 3.6: Validation of solver for 5 km/s flow over circular cylinder. Top: Computed flow field temperature contours. Bottom: Comparison of temperatures along centerline with LAURA [70] results running on equivalent mesh.

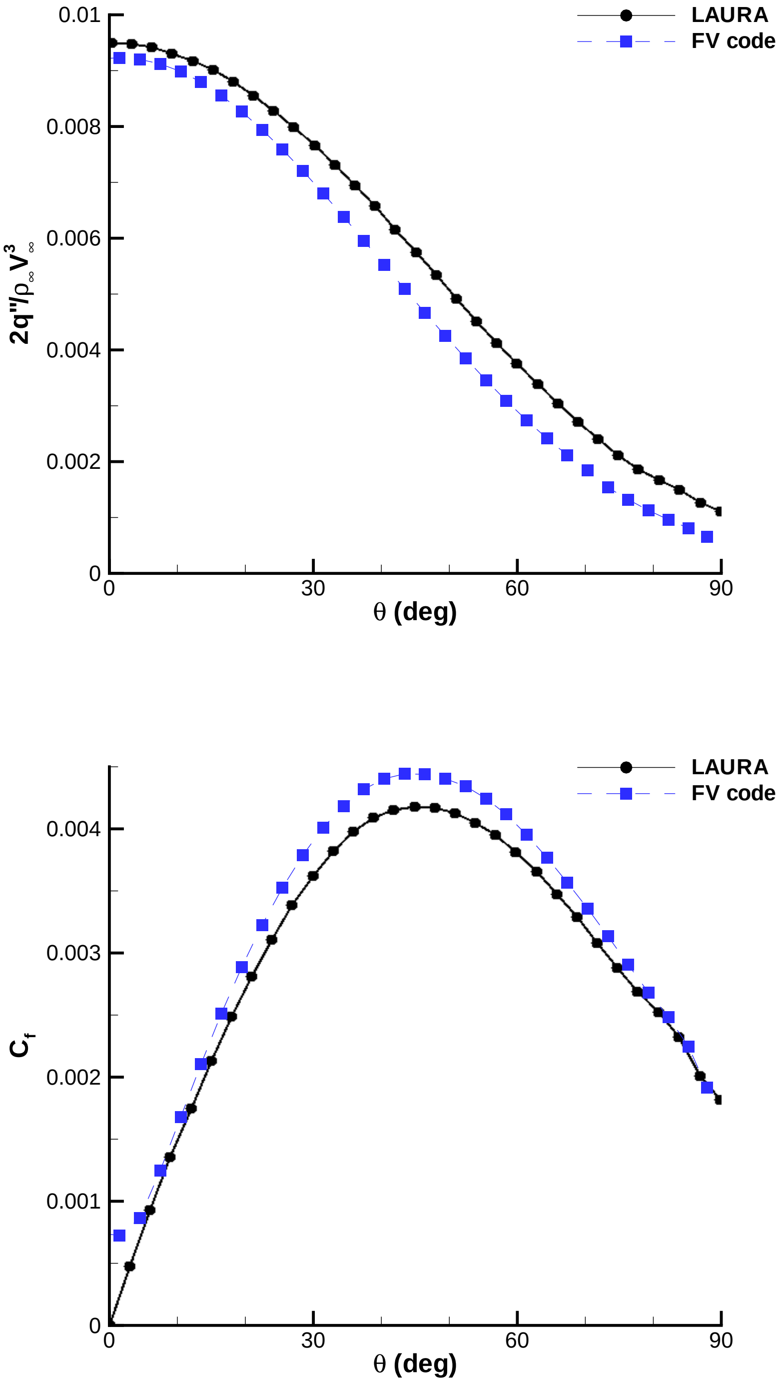


Figure 3.7: Validation of solver for 5 km/s flow over circular cylinder. Top: Surface heating distribution. Bottom: Surface Skin friction.

Chapter 4

Adjoint Derivation and Sensitivity Analysis

In this chapter, the derivation of the adjoint sensitivity equations and the details of the sensitivity implementation are given. With the calculation of the sensitivity derivatives outlined, the partial derivative values are used for the purposes of sensitivity analysis. As a starting point, the gradient values are used to rank the importance of each input parameter. Because this approach is inherently localized in nature, the gradient-based approach is compared with a global approach based on Monte Carlo sampling. Due to the expense of this sampling, a gradient-based strategy for global sensitivity analysis is explored.

4.1 Derivation of Sensitivity Derivatives

Because the calculation of derivatives is central to the sensitivity analysis and eventual uncertainty quantification central to this work, the derivation of the sensitivity derivative procedure is given. For this derivation, the gradient of an objective, L, with respect to a set of input parameters, D, is constructed. To determine the sensitivity of an objective to the input parameters, the code is differentiated and the final sensitivity is constructed using the chain rule. To illustrate this process, the following objective dependence is considered:

$$L = L(D, \mathbf{U}(D)) \tag{4.1}$$

In addition to this objective, a constraint is needed. For the steady problems considered in this work, the constraint is that the spatial residual must equal zero.

$$\mathbf{R}(D, \mathbf{U}(D)) = 0 \tag{4.2}$$

Both the constraint and the residual have an explicit dependence on the input parameter, or design variable D, and an implicit dependence through the flow variables $\mathbf{U}$. In order to determine the sensitivity derivative, the objective can be differentiated using the chain rule as: [78]

$$\frac{dL}{dD} = \frac{\partial L}{\partial D} + \frac{\partial L}{\partial \mathbf{U}} \frac{\partial \mathbf{U}}{\partial D} \tag{4.3}$$

The constraint may be differentiated in a similar manner. In this case, the derivative is equal to zero as the constraint must be satisfied for all admissible values of D and $\mathbf{U}$:

$$\frac{\partial \mathbf{R}}{\partial D} + \frac{\partial \mathbf{R}}{\partial \mathbf{U}} \frac{\partial \mathbf{U}}{\partial D} = 0 \tag{4.4}$$

Solving for $\frac{\partial \mathbf{U}}{\partial D}$ in the above equation and substituting into the objective derivative gives the forward sensitivity equation.

$$\frac{dL}{dD} = \frac{\partial L}{\partial D} - \frac{\partial L}{\partial \mathbf{U}} \frac{\partial \mathbf{R}}{\partial \mathbf{U}}^{-1} \frac{\partial \mathbf{R}}{\partial D} \tag{4.5}$$

The adjoint sensitivity equation is found by taking the transpose of the forward equation.

$$\frac{dL}{dD}^{T} = \frac{\partial L}{\partial D}^{T} - \frac{\partial \mathbf{R}}{\partial D}^{T} \frac{\partial \mathbf{R}}{\partial \mathbf{U}}^{-T} \frac{\partial L}{\partial \mathbf{U}}^{T} \tag{4.6}$$

Here, the last two terms can be replaced by the adjoint variable $\mathbf{\Lambda}$, defined as:

$$\frac{\partial \mathbf{R}}{\partial \mathbf{U}}^{T} \mathbf{\Lambda} = -\frac{\partial L}{\partial \mathbf{U}}^{T} \tag{4.7}$$

Finally, with the flow adjoint computed and the appropriate partial derivatives determined, the final sensitivity can be calculated with the equation:

$$\frac{dL^T}{dD} = \frac{\partial L^T}{\partial D} + \frac{\partial \mathbf{R}}{\partial D}^T \mathbf{\Lambda} \tag{4.8}$$

Based on these equations, several conclusions can be drawn. Because of the dimension of the matrix, the inversion of the flow Jacobian, $\frac{\partial \mathbf{R}}{\partial \mathbf{U}}$, represents the dominant cost in the sensitivity equations. As such, the number of Jacobian inversions required to yield the desired derivative values should be minimized. For the forward sensitivity equation, a new matrix inversion is required to determine the flow field sensitivity to each design variable; however, with this sensitivity in hand, the derivative with respect to any number of objectives can be calculated via derivative-vector products. Hence, the forward sensitivity equations should be used for situations with a large number of objectives and a limited number of design variables. For the adjoint sensitivity equation, a single adjoint can be used to compute the derivatives of a single objective with respect to all of the design variables, but a new adjoint solution is required for each additional objective. Because solving for the flow adjoint requires a Jacobian inversion, the adjoint approach is best suited for applications with a limited number of objectives and a large number of design variables. For typical engineering calculations, the number of input variables is typically much larger than the number of objectives, making the adjoint approach preferable. Although this work uses the adjoint approach exclusively, the forward sensitivity is detailed as its development is required to verify the adjoint sensitivity.

For this work, the sensitivity derivatives with respect to model input parameters are required. Let $\boldsymbol{\alpha}$ represent a set of model parameters in the code. Although the derivative with respect to model parameters is desired for sensitivity analysis, it is convenient to allow the model parameters to be a function of a set of design variables $\boldsymbol{\alpha}(\boldsymbol{D})$ for the purposes of uncertainty quantification. This abstraction allows for correlation to be considered between design variables as well as more realistic specification of the uncertainty in the model parameters.

To determine the sensitivity of an objective to any of the variables in the parameter vector, the following functional dependence should be considered:

$$L = L(\mathbf{U}(\alpha(D)), \boldsymbol{\alpha}(\boldsymbol{D})) \tag{4.9}$$

$$\mathbf{R} = \mathbf{R}(\mathbf{U}(\boldsymbol{\alpha}(\boldsymbol{D})), \boldsymbol{\alpha}(\boldsymbol{D})) \tag{4.10}$$

The forward sensitivity associated with this functional dependence is given by:

$$\frac{dL}{d\boldsymbol{\alpha}} = \frac{\partial L}{\partial \boldsymbol{\alpha}}\frac{\partial \boldsymbol{\alpha}}{\partial D} + \frac{\partial L}{\partial \mathbf{U}}\frac{\partial \mathbf{U}}{\partial D} \tag{4.11}$$

where $\frac{\partial \mathbf{U}}{\partial D}$ is determined by solving the equation:

$$\left[\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right]\frac{\partial \mathbf{U}}{\partial D} = -\frac{\partial \mathbf{R}}{\partial \boldsymbol{\alpha}}\frac{\partial \boldsymbol{\alpha}}{\partial D} \tag{4.12}$$

Because the model parameter abstraction only affects the specification of design variables, the equation for the flow adjoint (i.e. equation (4.7)) is unchanged. Using this adjoint, the final parameter sensitivity is given by:

$$\frac{dL}{dD}^T = \frac{\partial \boldsymbol{\alpha}}{\partial D}^T \left[\frac{\partial L}{\partial \boldsymbol{\alpha}}^T + \frac{\partial \mathbf{R}}{\partial \boldsymbol{\alpha}}^T \boldsymbol{\Lambda}\right] \tag{4.13}$$

For the case where the model parameters are identical to the design variables, the matrix $\frac{\partial \boldsymbol{\alpha}}{\partial D}$ becomes the identity matrix. For all other cases, the matrix $\frac{\partial \boldsymbol{\alpha}}{\partial D}$ is determined by differentiating the desired relationship between input parameters and design variables.

4.2 Sensitivity Implementation

In order to solve the forward and adjoint sensitivity equations, the equations are placed in defect correction form to aide with the construction and inversion of the flow Jacobian [78]. For the forward sensitivity equation, the defect correction form is given as:

$$[A]\,\delta\mathbf{U}_D^k = -\frac{\partial \mathbf{R}}{\partial D} - \frac{\partial \mathbf{R}}{\partial \mathbf{U}}\frac{\partial \mathbf{U}}{\partial D}^k \tag{4.14}$$

$$\frac{\partial \mathbf{U}}{\partial D}^{k+1} = \frac{\partial \mathbf{U}}{\partial D}^k + \delta\mathbf{U}_D^k \tag{4.15}$$

while the defect correction form of the adjoint equation is given by:

$$[A]^T \delta \mathbf{\Lambda}^k = -\frac{\partial L}{\partial \mathbf{U}}^T - \frac{\partial \mathbf{R}}{\partial \mathbf{U}}^T \mathbf{\Lambda}^k \tag{4.16}$$

$$\mathbf{\Lambda}^{k+1} = \mathbf{\Lambda}^k + \delta \mathbf{\Lambda}^k \tag{4.17}$$

In the above equations, the iteration matrix, $[A]$, is used to advance the solution toward convergence (as $k \to \infty$). This form of the equations has two advantages. First, the choice of the iteration matrix $[A]$ is general and can incorporate diagonal padding for cases where diagonal dominance of the matrix is required for the iterative solver. Second, the exact flow Jacobian only appears as a matrix-vector product. This fact allows the effect of the exact Jacobian to be built up piece by piece as opposed to explicitly computed and stored.

For this work, a line-preconditioned GMRES solver is used to invert the iteration matrix at each step. Because GMRES requires only matrix-vector products, the use of this solver allows the exact flow Jacobian with no diagonal padding to be used as the iteration matrix ($[A] = \left[\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right]$). Additionally, the full second-order Jacobian does not need to be explicitly computed and stored. The steps performed for each iteration of GMRES applied to the flow adjoint problem are enumerated below [71]:

$$[P]^T \mathbf{q}_j = \mathbf{v}_j \qquad \text{Preconditioning} \tag{4.18}$$

$$\mathbf{w}_j = \left[\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right]^T \mathbf{q}_j \qquad \text{Matrix-vector product} \tag{4.19}$$

$$H_{i,j} = \mathbf{w}_j \cdot \mathbf{v}_i \qquad \text{Hessenberg Matrix} \tag{4.20}$$

The preconditioner, $[P]$, is the same preconditioner used in the flow solver, namely the first-order flow Jacobian calculated using the Van-Leer-Hanel flux function. The preconditioner is inverted using the line-Jacobi solver described in Chapter 3.

In addition to using the line-preconditioned GMRES solver, the preconditioners used for the flow solver can be used directly to solve for the flow adjoint. Because the iterative solvers outlined in Chapter 3 require diagonal dominance, a pseudo-timestep is added to the diagonal of the preconditioner matrix for both the adjoint and forward sensitivity equations.

$$\left[P + \frac{V}{\Delta t_k}\right] \delta \mathbf{U}_D^k = -\frac{\partial \mathbf{R}}{\partial D} - \frac{\partial \mathbf{R}}{\partial \mathbf{U}} \frac{\partial \mathbf{U}}{\partial D}^{k} \tag{4.21}$$

$$\frac{\partial \mathbf{U}}{\partial D}^{k+1} = \frac{\partial \mathbf{U}}{\partial D}^{k} + \delta \mathbf{U}_D^k \tag{4.22}$$

$$\left[P^T + \frac{V}{\Delta t_k}\right] \delta \boldsymbol{\Lambda}^k = -\frac{\partial L}{\partial \mathbf{U}}^{T} - \frac{\partial \mathbf{R}}{\partial \mathbf{U}}^{T} \boldsymbol{\Lambda}^k \tag{4.23}$$

$$\boldsymbol{\Lambda}^{k+1} = \boldsymbol{\Lambda}^k + \delta \boldsymbol{\Lambda}^k \tag{4.24}$$

Because the purpose of this pseudo-timestep is used to ensure diagonal dominance, the CFL number or timestep size, Δt_k, for the defect-correction solve can be fixed at its maximum value. In some circumstances, the CFL ramping used in the flow solver may be required during the defect-correction process in order to get over start-up problems caused by a bad initial guess for the iterative solver. Although this ramping will not affect the final forward-sensitivity or adjoint solutions, it does break duality of the solution process, a property which is explained later in the section.

In order to construct the derivatives required for the flow Jacobian, as well as, constructing the final sensitivity, automatic differentiation is applied on a subroutine by subroutine basis to the code required to compute the flow residual and objective. The automatic differentiation engine used throughout this work is Tapenade version 3.2 [79]. Because both the forward and adjoint sensitivity procedure can be written exclusively as a series of Jacobian-vector products, the automatically differentiated subroutines build up the effect of the Jacobian matrix on an input vector in a piece-wise manner. The process by which individually differentiated subroutines are constructed and validated is given by the following three steps: complexify, differentiate, transpose.

The first step of the differentiation requires the generation of a complex version of the original subroutine. This process consists of converting the variables in the subroutine from type "real" to type "complex". With the variables in complex form, the real part of the variable will remain unchanged and the complex part can be used to approximate the derivative of the variable with respect to a single design variable. The derivative with respect to a single input variable is computed by adding an imaginary perturbation to the desired complex input variable and determining the corresponding imaginary part of the output. With the variables in complex form, the operators within the subroutine need to be overloaded to ensure the desired performance when applied to complex inputs. A majority of the intrinsic functions automatically possess the desired properties for complex inputs. The exceptions to this rule are comparative operators as well as non-differentiable operators such as the min/max functions and the absolute value function. For these operators, the desired behavior must be overloaded. The definition of these overloaded operators may be found in Appendix B. The veracity of the complex subroutines are confirmed in two ways. First, the output of the complex subroutine is compared with the output of the original subroutine. For this comparison, the output of the original subroutine should match the real part of the complex subroutine to essentially machine zero in a relative sense (meaning 13-14 digits of agreement for double precision). Because of the nature of the intrinsic complex operators, the complex part of a variable can induce a perturbation in the real part of the variable; however, this perturbation can be minimized by using a sufficiently small complex step for the input variables. Unlike finite difference, the step used for complex differentiation can be arbitrarily small. As a second check on the complex subroutines, the accuracy of the imaginary part of the subroutine (representing the derivative w.r.t a single design variable) can be compared with finite difference applied to the original subroutine.

With the complex subroutine generated and validated, the forward differentiation can be generated. This forward differentiation is performed using the automatic differentiation engine Tapenade, although it could be carried out by hand. The first step of differentiating the subroutine is properly identifying the input parameters that depend on the design variable of interest, either explicitly or implicitly. In addition to variable identification, the original source code must be modified to remove any pieces of code that the AD tool cannot process. For the code written in this work, only the openMP directives must be removed to differentiate the code correctly. With the set of input variables identified and offending code removed, the AD tool can then proceed through the subroutine, differentiating each operator. The resulting source code produces both the output of the original code as well as the derivative of the output with respect to a single, generic design variable. Although the derivative with respect to multiple design variables can be generated using Tapenade, this approach is seldom efficient from a memory point of view. In the event that multiple matrix-vector products are required, the single variable differentiation is merely called multiple times with different inputs.

With the forward differentiation generated, it can be verified using the complex subroutine. Because automatic differentiation is employed in this work, this verification is a formality and simply serves the purpose of ensuring that any modifications to the original subroutine required for the AD tool did not alter the output or linearization of the differentiated subroutine. This verification against complex differentiation is more critical when the differentiated subroutine is implemented by hand.

Once the forward differentiation is generated and validated, the transpose operator can be implemented. The transpose implementation consists of two steps. First, the state about which the linearization is performed is computed and stored. Second, with the state computed, the steps of the differentiation are performed in reverse order with each operator transposed. This transposing has the effect of interchanging the input and outputs of the differentiated operator. Hence, if the operator is defined as $f(x)$ and the forward differentiation takes in a δx and produces a δf, the resulting transpose operator will take in a δf and produce a corresponding δx. Because this transpose operator is again constructed using automatic differentiation, the details of the transpose implementation are handled by the AD tool. In order to apply the AD tool for the transpose operator, the original source subroutine must again be modified to ensure compatibility with the AD tool, namely removal of openMP directives. Once the AD tool has been applied to the original source code, the resulting transpose linearization often must be slightly modified to ensure proper performance when coupled with other operators. First, the initialization of all input variables within the linearized subroutine must be removed to allow multiple subroutines to contribute to the same variable. Second, because the input to the subroutine is often irrelevant once it has been propagated to the output, Tapenade typically sets this variable to zero. When a subroutine is considered in isolation, this behavior has no effect on the overall behavior; however, this zeroing must be removed when multiple subroutines are to be considered together.

The validity of the transpose operator is verified by checking duality between the transpose and forward linearization. Duality is best demonstrated through a series of equations. With the operator defined as $y = f(x)$, the derivative and transpose are represented as:

$$\delta y = \frac{\partial f}{\partial x}\delta x \tag{4.25}$$

$$\delta \bar{x} = \frac{\partial f}{\partial x}^T \delta \bar{y} \tag{4.26}$$

Given these problem definitions, duality requires the following relation to hold:

$$\delta y \cdot \delta \bar{y} = \delta x \cdot \delta \bar{x} \tag{4.27}$$

By checking this duality relationship, the validity of the transpose operator can be tested. This duality relationship can be extended to multiple inputs and outputs demonstrated below. With the operators defined as $y_1 = f(x_1, x_2, x_3)$ and $y_2 = g(x_1, x_2, x_3)$, the derivative and transpose become:

$$\delta y_1 = \frac{\partial f}{\partial x_1}\delta x_1 + \frac{\partial f}{\partial x_2}\delta x_2 + \frac{\partial f}{\partial x_3}\delta x_3 \tag{4.28}$$

$$\delta y_2 = \frac{\partial g}{\partial x_1}\delta x_1 + \frac{\partial g}{\partial x_2}\delta x_2 + \frac{\partial g}{\partial x_3}\delta x_3 \tag{4.29}$$

$$\delta \bar{x}_1 = \frac{\partial f}{\partial x_1}^T \delta \bar{y}_1 + \frac{\partial g}{\partial x_1}\delta \bar{y}_2 \tag{4.30}$$

$$\delta \bar{x}_2 = \frac{\partial f}{\partial x_2}^T \delta \bar{y}_1 + \frac{\partial g}{\partial x_2}\delta \bar{y}_2 \tag{4.31}$$

$$\delta \bar{x}_3 = \frac{\partial f}{\partial x_3}^T \delta \bar{y}_1 + \frac{\partial g}{\partial x_3}\delta \bar{y}_2 \tag{4.32}$$

For these functional forms, the duality relationship is given by:

$$\delta y_1 \cdot \delta \bar{y}_1 + \delta y_2 \cdot \delta \bar{y}_2 = \delta x_1 \cdot \delta \bar{x}_1 + \delta x_2 \cdot \delta \bar{x}_2 + \delta x_3 \cdot \delta \bar{x}_3 \tag{4.33}$$

In practice, to test the transpose of individual operators, random inputs are supplied to the input derivatives of the forward and adjoint subroutines and duality is tested using the subroutine outputs. Obviously for the case of multiple inputs, equation (4.27) can be recovered by supplying the forward and adjoint subroutines with a single non-zero input each.

In addition to applying duality to individual operators, the property can be applied to the solution of the flow adjoint and forward sensitivity equations. By definition of the duality property, the following relationship must be satisfied for fully converged sensitivity and adjoint solutions:

$$\frac{\partial \mathbf{L}}{\partial \mathbf{U}} \frac{\partial \mathbf{U}}{\partial D} = \frac{\partial \mathbf{R}}{\partial D}^T \mathbf{\Lambda} \tag{4.34}$$

This requirement can be used to ensure the correctness of the transposed flow Jacobian as well as the validity of the fully converged sensitivity and adjoint values. Because the forward sensitivity and flow adjoint equations are solved using a defect-correction scheme, duality can also be evaluated for each solver iteration. When duality holds at each iteration, the solution method is denoted as duality preserving. For a duality preserving scheme, the convergence rate of the forward sensitivity equation will match that of the adjoint problem; implying that if a forward sensitivity solution can be found, the adjoint solution can be found with a similar amount of work.

$$\frac{\partial \mathbf{L}}{\partial \mathbf{U}} \frac{\partial \mathbf{U}^k}{\partial D} = \frac{\partial \mathbf{R}}{\partial D}^T \mathbf{\Lambda^k} \qquad \text{for all } k \tag{4.35}$$

For a scheme to be duality preserving, the iteration matrix used in the adjoint solution process must be the exact transpose of the iteration matrix, A, used for the forward sensitivity equation. In addition to transposing the iteration matrix, the exact steps of the iterative solver must be reversed. For example, for a Gauss-Seidel solver, the order in which solution updates are calculated must be reversed. Hence, if the order of the loop over elements is in increasing order for the forward sensitivity equation, the transpose operator must loop over the elements in decreasing order. For Jacobi-type iterative solvers, no special consideration is required to ensure duality. Further information regarding the construction

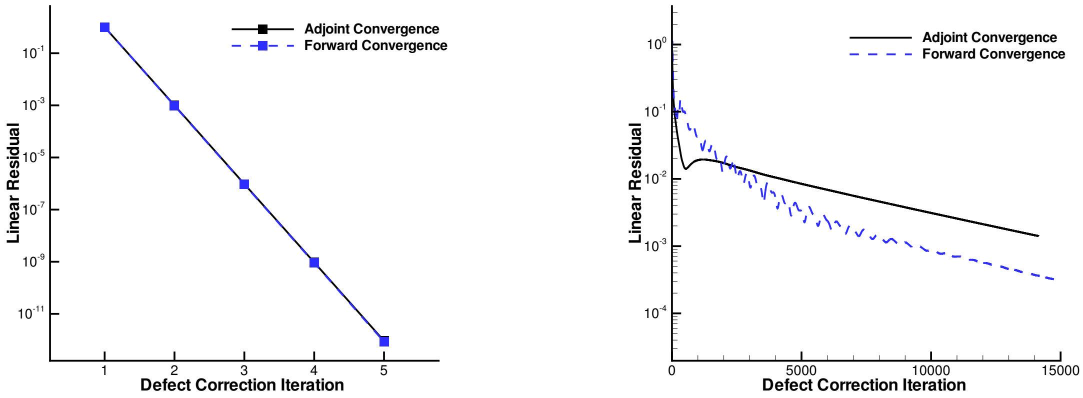


Figure 4.1: Convergence of forward sensitivity and adjoint solution using (left) Jacobi-line preconditioned GMRES and (right) Colored Gauss-Seidel with forward ordering

of duality preserving iteration strategies can be found in References [78, 80, 81]. Because the preconditioning in the GMRES solver is a Jacobi-line solver, the GMRES solver is automatically duality preserving. The practical impact of duality preservation can be seen by examining the convergence of the forward sensitivity and adjoint solution for both a duality preserving and non-preserving scheme. Figure 4.1 shows the convergence of the forward and adjoint solution using the Jacobi-line preconditioned GMRES solver as well as a colored Gauss-Seidel solver looping over the elements in increasing order for both the forward and adjoint problem.

As the figure demonstrates, the convergence rate for the duality preserving scheme is identical for the forward sensitivity and adjoint solution. When duality preservation is not enforced on the iteration scheme, the forward and adjoint problems may converge at significantly different rates. Because the iteration matrices are not exact tranposes of one another, the adjoint solution is not guaranteed to converge under the same circumstances as the forward sensitivity equation and different solver settings for the two problems, such as the amount of diagonal padding or linear solver subiterations, may be required for the adjoint solution.

With each operator independently verified, the final sensitivity derivative must be verified to ensure the operators have been assembled correctly. The verification of the final derivative can be performed in two ways: finite difference or complex differentiation. For the complex differentiation, the entire solution process must be complexified, including the linear and nonlinear solver used to determine the flow solution. The advantage of this method is that the derivative of every step in the solution process may be verified. The drawback of this approach is that a wider range of operators must be overloaded including file I/O and simulation monitoring subroutines. For situations in which complexification is not viable, finite difference can be applied to the entire code. Although finite-difference gradients will have an associated approximation error, the implementation of finite difference to the entire solution process is straight forward, simply requiring the calculation of a solution at perturbed input parameter values. Because the derivative of each individual operator used to calculate the gradient has been verified using complex differentiation, the use of finite difference should be sufficient to verify the fully assembled sensitivity derivative.

4.3 Localized Sensitivity Analysis

Using the previously described sensitivity process, the derivative of an objective with respect to model input parameters is calculated to determine the parameters most affecting the output. This sensitivity analysis is performed within the context of the real gas CFD solver described in Chapter 3. The flow conditions for this sensitivity analysis correspond to the 5 km/s flow over a cylinder test case described previously. The objective for this sensitivity analysis is the integrated surface heating over the surface of the cylinder, given by equation (4.36).

$$L = -\frac{\int_{\partial\Omega} \left(k\nabla T \cdot \vec{n} + k_v \nabla T_v \cdot \vec{n}\right) dA}{\frac{1}{2}\rho_\infty V_\infty^3} \tag{4.36}$$

In this equation, T is the translational-rotational temperature, k is the translational-rotational thermal conductivity, T_v is the vibrational temperature, and k_v is the vibrational thermal conductivity.

A total of 66 model parameters are examined for this sensitivity analysis. These parameters cover freestream conditions, transport properties and the chemical kinetics model. Although other model parameters exist within the real gas model, these additional parame-

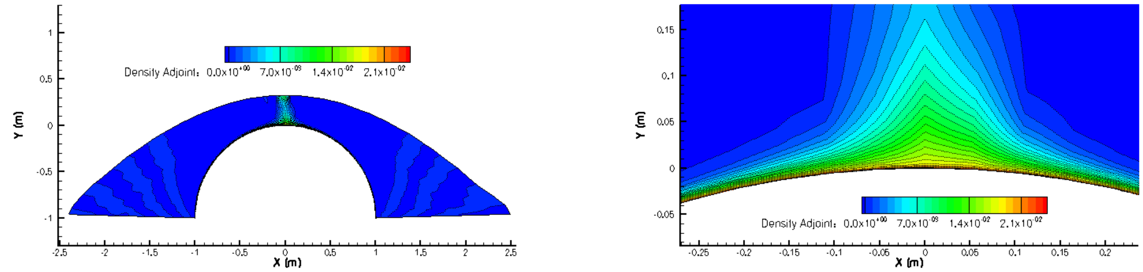


Figure 4.2: Left: Density adjoint for integrated surface heating (left) throughout flow field and (right) around stagnation point.

ters will not be examined for this work. These parameters are left out either because previous sensitivity and uncertainty analyses have indicated the parameters to be unimportant or reliable uncertainty specifications are unavailable [2]. The derivatives produced by the discrete adjoint were confirmed using finite difference. The comparison with finite difference is given in Appendix C. Although only a single step size is presented in Appendix C, a number of different step sizes were analyzed to ensure these results were representative. A representative adjoint field solution for this problem is given in Figure 4.2. Depicted in this figure is the adjoint variable corresponding to the density. The large value of the adjoint near the stagnation point and along the surface of the cylinder indicates that the density has a large effect on the objective, integrated surface heating, in that region. The adjoint problem itself was solved using the line-preconditioned GMRES solver. The solver required 527 Jacobian-vector products and 5 GMRES restarts to arrive at the solution. The total time to solution was 208 seconds. Compared with the 8616 seconds required for the flow solution, the adjoint was over a factor of 40 times faster than the flow solution when serial run-times are compared. Because the adjoint implementation lacks shared memory parallelization, this ratio would obviously decrease when the solver is run in parallel. However, because subsequent testing employed a form of embarrassingly parallel sampling, the typical simulation was run sequentially and parallelization was performed over the independent simulations required for sampling.

The first set of parameters considered in the sensitivity analysis relate to the specification of transport quantities (specified in Chapter 2). The transport model used for this work requires 60 input parameters. For the five species model, 15 unique collision interactions are possible. For each interaction, two collision integrals must be specified for both 2000 K and 4000 K. Although the transport model requires 60 inputs, these inputs are likely not independent for the purposes of uncertainty quantification. Because the inputs at 2000 K and 4000 K are correlated, a single parameter is prescribed that accounts for shifts in the cross-section across the entire temperature range. This parameter modifies the collision integrals as follows:

$$\Omega^{k,k}_{s,r}(T) = A^{k}_{s,r}\hat{\Omega}^{k,k}_{s,r}(T) \tag{4.37}$$

where $\hat{\Omega}^{k,k}_{s,r}(T)$ is the unperturbed collision integral and $A^{k}_{s,r}$ is a multiplicative constant for each unique collision integral (30 total). In terms of equation (4.9), the design variable D are the multiplicative constants $A^{k}_{s,r}$ while the input parameters into the code $\boldsymbol{\alpha}$ are the base-10 logarithm for each collision integral at 2000 K and 4000 K. By adding the same base-10 logarithm of $A_{s,r}$ to the input collision integrals at both 2000 K and 4000 K, the functional relationship above can be implemented without any modification to the analysis code. This fact is demonstrated below.

$$\begin{aligned}
\log_{10}(\Omega^{k,k}_{s,r})_{2000} &= \log_{10}(\hat{\Omega}^{k,k}_{s,r})_{2000} + \log_{10} A^{k}_{s,r} && (4.38)\\
\log_{10}(\Omega^{k,k}_{s,r})_{4000} &= \log_{10}(\hat{\Omega}^{k,k}_{s,r})_{4000} + \log_{10} A^{k}_{s,r} && (4.39)\\
\log_{10}(\Omega^{k,k}_{s,r}(T)) &= \log_{10}(\hat{\Omega}^{k,k}_{s,r})_{2000} + && (4.40)\\
&\quad \left[\log_{10}(\hat{\Omega}^{k,k}_{s,r})_{4000} - \log_{10}(\hat{\Omega}^{k,k}_{s,r})_{2000}\right] \frac{ln(T) - ln(2000)}{ln(4000) - ln(2000)} &&\\
&\quad + \log_{10} A^{k}_{s,r} &&\\
\log_{10}(\Omega^{k,k}_{s,r}(T)) &= \log_{10}(\hat{\Omega}^{k,k}_{s,r}(T)) + \log_{10} A^{k}_{s,r} && (4.41)\\
\Omega^{k,k}_{s,r}(T) &= A^{k}_{s,r}\hat{\Omega}^{k,k}_{s,r}(T) && (4.42)
\end{aligned}$$

Hence, the desired parameter relationship (given in equation (4.37)) is implemented simply by adding the base-10 logarithm of $A_{s,r}$ to the input collision integrals at 2000 K and 4000 K.

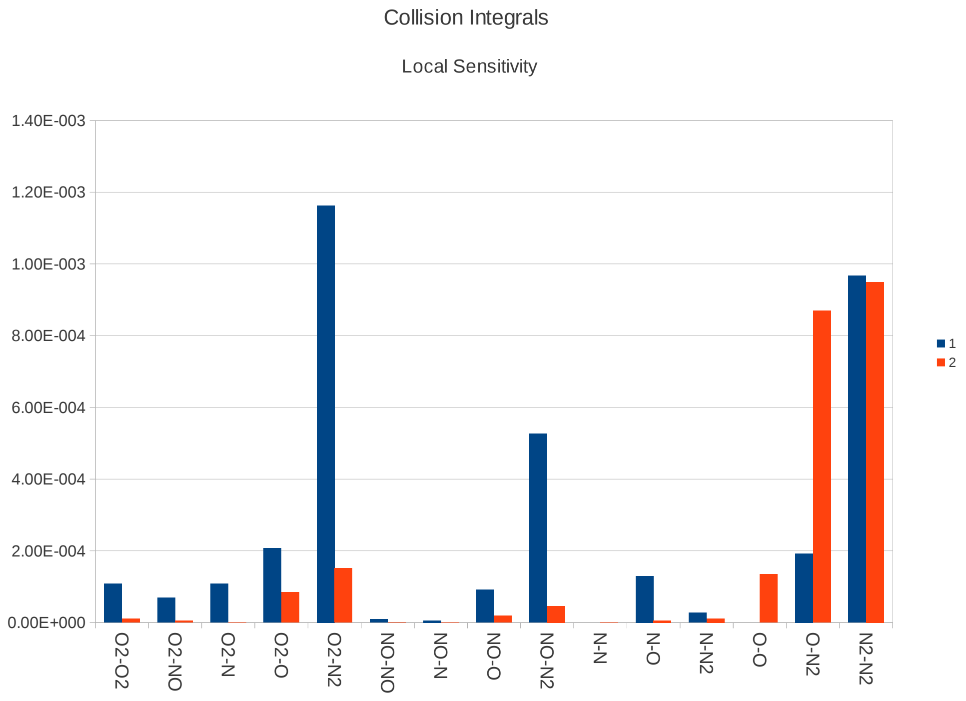


Figure 4.3: Sensitivity of surface heating to collision integrals parameters

Given this specification of the model parameter, the gradient of integrated surface heating is constructed using the adjoint sensitivity framework. In order to rank the importance of each collision interaction to integrated surface heating, the magnitude of the derivative is used. The associated sensitivity for each interaction can be found in Figure 4.3.

As the figure demonstrates, the interactions associated with diatomic nitrogen have the greatest effect on integrated surface heating. This result is unsurprising as diatomic nitrogen is the dominant species near the surface of the cylinder due to the super-catalytic boundary condition.

The next set of parameters relate to the chemical kinetics model. Specifically, the parameters in the Dunn-Kang chemical kinetics are examined. For the Dunn-Kang model, the forward and backward reaction rates are specified using Arrhenius relations, given by equation (4.43).

$$K_f = C_f T_a^{\eta_f} e^{-\frac{E_{a,f}}{k_B T_a}} \tag{4.43}$$

$$K_b = C_b T_a^{\eta_b} e^{-\frac{E_{a,b}}{k_B T_a}} \tag{4.44}$$

Here $E_{a,f}$ and $E_{a,b}$ represent the activation energy for the forward and backward reactions, k_B is Boltzmann's constant, and T_a is the rate controlling temperature for the reaction. From previous uncertainty and sensitivity studies, the specification of the uncertainty for the reaction rate is again given by a multiplicative constant applied over the entire temperature range. Because of the large uncertainty typical of reaction rates, the uncertain reaction parameter is written as [2]:

$$\log_{10}\left(\frac{k_r}{k_o}\right) = \xi \tag{4.45}$$

where k_r is either the forward or backward reaction rate, k_o is the corresponding unperturbed reaction rate and ξ is the design variable examined for this sensitivity study. A separate parameter is applied to each reaction rate, resulting in a total of 34 total parameters when both the forward and backward reaction rates are considered. Transforming equation (4.45), the reaction rates are specified as:

$$K_f = 10^{\xi_f} C_{f,o} T_a^{\eta_f} e^{-\frac{E_{a,f}}{k_B T_a}} \tag{4.46}$$

$$K_b = 10^{\xi_b} C_{b,o} T_a^{\eta_b} e^{-\frac{E_{a,b}}{k_B T_a}} \tag{4.47}$$

where $C_{f,o}$ and $C_{b,o}$ represent the unperturbed coefficients, and ξ_f and ξ_b represent the corresponding model parameters for the forward and backward rates. These design variables are implemented in the solver by perturbing the inputs C_f and C_b to the model and gradients are computed via the chain rule.

$$C_r = 10^{\xi_r} C_{r,o} \tag{4.48}$$

$$\frac{\partial C_r}{\partial \xi_r} = 10^{\xi_r} C_{r,o} ln(10) \tag{4.49}$$

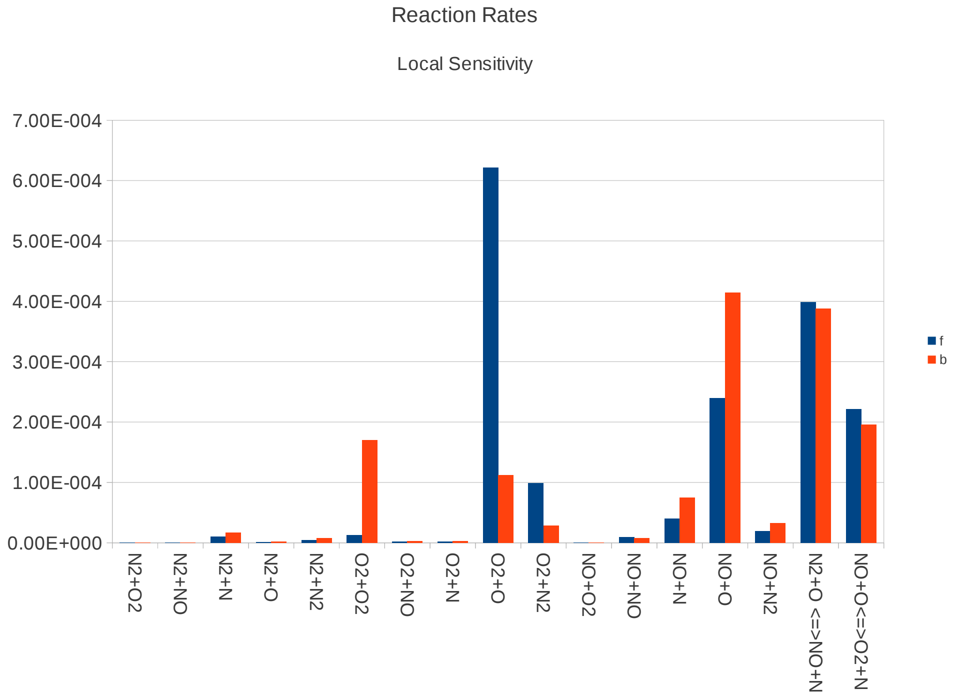


Figure 4.4: Sensitivity of surface heating to reaction rate parameters

In addition to being consistent with the uncertainty for these parameters, this specification gives the resulting gradient value a convenient normalization. The Arrhenius coefficients are large numbers and have variations of many orders of magnitude between them. This wide variation in magnitude causes corresponding magnitude variations in the components of the gradient $\frac{\partial L}{\partial C_r}$. Using the above parameter specification, the resulting gradient values $\frac{\partial L}{\partial \xi_r}$ can be directly compared against each other and other model parameters.

In order to assess the sensitivity of integrated surface heating to the reaction rates, the magnitude of the gradient is once again used. The magnitude of $\frac{\partial L}{\partial \xi}$ is plotted in Figure 4.4. As the results show, the dissociation of diatomic oxygen and nitric oxide with oxygen have the greatest effect on surface heating as well as the creation and decomposition of nitric oxide. In addition to identifying the most sensitive reactions, these results demonstrate that based on this localized analysis, the collision integrals in general have a greater effect on the surface heating than the reaction rates.

Table 4.1: Ranking of input parameters based on derivative values

Number	Variable	Derivative Value
1	ρ_∞	4.887
2	O2-N2 (k=1)	1.1630×10^{-3}
3	N2-N2 (k=1)	9.6708×10^{-4}
4	N2-N2 (k=2)	9.4977×10^{-4}
5	O-N2 (k=2)	8.6974×10^{-4}
6	$O_2 + O \leftrightarrows 2O + O$ (f)	6.2120×10^{-4}
7	NO-N2 (k=2)	5.2660×10^{-4}
8	$NO + O \leftrightarrows N + O + O$ (b)	4.1447×10^{-4}
9	$N_2 + O \leftrightarrows NO + N$ (f)	3.9861×10^{-4}
10	$N_2 + O \leftrightarrows NO + N$ (b)	3.8755×10^{-4}

Finally, in order to determine the relative importance of parameters drawn from the various parts of the real gas model, the derivative magnitude over all the model parameters are directly compared. In addition to the parameters for the transport model and chemical kinetics model, the freestream density and velocity are also included in the sensitivity analysis to measure the relative importance of freestream conditions. The top ten parameters based on the derivative magnitude are given in Table 4.1. As these results demonstrate, the freestream density has the greatest effect on surface heating by several orders of magnitude. The rest of the top parameters are composed of the previously identified important collision integrals and reaction rates with the top collision integrals ranking above the majority of the top reaction rates.

4.4 Global Sensitivity Analysis

The previously presented derivative sensitivity analysis is inherently localized in nature and may give inaccurate results for large perturbations of the input parameters or due to interference effects between variables [2]. To overcome these limitations, a global sensitivity analysis can be performed and the importance of each variable can be estimated based on its contribution to the simulation output. This global sensitivity analysis is performed using Monte Carlo sampling. For this method, design variable values are drawn from the associated input distribution and a simulation is run to acquire an output sample. Statistics are then calculated based on these output samples. Although the work associated with this sampling is independent of the number of design variables, the expense of this sampling is often prohibitively high due to the slow convergence of output statistics. The sensitivity itself can be quantified using a linear regression analysis. This linear analysis calculates the correlation coefficient for each variable based on the Monte Carlo results. Although this method cannot fully separate the effect of each variable on the output, it can provide valuable information on the overall effect each variable has on the output measured over the entire design space. It can also account for interference effects between the variables and give a measure of the contribution each variable's uncertainty makes to the overall output uncertainty. The correlation coefficient for variable D_i is given by the following [2]:

$$r_i = \frac{cov(D_i, y)}{\sigma_{D_i}\sigma_y} \tag{4.50}$$

Here, y represents the output of interest from the simulation, σ_{D_i} represents the standard deviation of the input design variable and σ_y represents the standard deviation of the output. The standard deviation of the input design variable is a quantity that must be taken from the relevant literature or estimated based on some expert judgment or experience [2, 15]. The quantity σ_y is measured empirically from the Monte Carlo data set. Because the output variance is required, uncertainty quantification is intricately tied to global sensitivity analysis. Gradient-based techniques for rapid uncertainty quantification will be covered extensively in Chapters 5 through 7; however, to aide in the discussion of global sensitivity analysis, relevant concepts will be introduced as needed.

In practice, the correlation coefficient must be estimated based on the Monte Carlo data using estimators for each of the statistical quantities in equation (4.50). The correlation coefficient can therefore be estimated as [2, 82]:

$$r_i = \frac{\sum_k (D_{i,k} - \bar{D}_i)(y_k - \bar{y})}{\sqrt{\sum_k (D_{i,k} - \bar{D}_i)^2 \sum_k (y_k - \bar{y})^2}} \tag{4.51}$$

where the summation over k indicates a sum over the Monte Carlo sample points.

Because the statistics required for the correlation coefficient are estimated based on

Table 4.2: Input Model Parameters with Uncertainty

Number	Variable	Mean	Standard Deviations
1	ρ_∞ (kg/m^3)	1×10^{-3}	5%
2	$V_\infty(m/s)$	5000	15.42
3-17	A^1_{s-r}	1	5%
18-32	A^2_{s-r}	1	5%
33-49	ξ_f	0	0.25
50-66	ξ_b	0	0.25

Monte Carlo sampling, there will be an associated error due to the finite size of the data set. This error will be inversely proportional to the number of Monte Carlo samples (N), decreasing as $\frac{1}{\sqrt{N}}$. For the purposes of ranking the sensitivities, the square of the correlation coefficient is used. In addition to acting as a proxy for the magnitude of the sensitivity, the square of the correlation coefficient represents the fraction of the output variance from each of the input parameters [2].

Within this section, a global sensitivity analysis for the 66 model parameters identified previously is performed and compared to the results of the localized sensitivity analysis. This comparison is performed to assess the practical effect of the theoretical limitations to localized sensitivity analysis on a real world simulation. In order to perform the sensitivity analysis, the uncertainty for each model parameter must be specified to define the space over which the sensitivity is assessed. The uncertainty for each parameter is given in Table 4.2.

The correlation coefficients for each variable were computed based on Monte Carlo sampling using 6331 simulation results. The error in the square of the correlation coefficient is estimated at 0.02 based on the results of a similar global sensitivity analysis with a similar number of samples [2]. Hence, when comparing coefficients, only differences greater than 0.04 should be considered when ranking parameters.

The results of the local and global sensitivity analysis are compared based on the square of the correlation coefficient. Because this metric represents the fraction of the output variance based on a particular input parameter, a corresponding number for the localized sensitivity analysis must be constructed. Based on the moment method, the fraction of the output variance based on an input can be computed as:

$$f_i = \frac{\frac{\partial L}{\partial D_i}^2 \sigma_{D_i}^2}{\sum_k \frac{\partial L}{\partial D_k}^2 \sigma_{D_k}^2} \tag{4.52}$$

The global and local sensitivity analysis results are first compared for the collision integral model parameters. This comparison is made in Figure 4.5. As the figure demonstrates, both the local and global sensitivity analysis identify the same high importance collision integrals although the magnitude of the uncertainty contribution show some disagreement. Hence, the local sensitivity analysis appears to provide the appropriate importance rankings; however, if exact contributions to uncertainty are required, the localized analysis is not sufficient.

In order to further investigate the differences between global and local sensitivity analysis, the results for the reaction rates are compared. Figure 4.6 shows this comparison. Compared to the collision integral parameters, the results for the reaction rate parameters show greater discrepancy between the two sensitivity analyses. These discrepancies occur in terms of predicting the contribution to output variance as well as the relative importance of each parameter. In addition to misidentifying the top reaction rates, the uncertainty contribution estimates vary dramatically between the two methods. This behavior is predicted in the literature when the input uncertainties increase [2]. For the reaction rates, the uncertainty is specified as plus/minus an order of magnitude, giving an indication that the linear analysis used in the local analysis may break down.

Finally, the rankings of model parameters between the two methods are compared. For this comparison, the top parameters from the local sensitivity analysis are identified and compared with their ranking from the global sensitivity analysis. This comparison is made in Table 4.3. In addition to comparing the ranking of each parameter, the uncertainty contribution is also compared. The inverse of Table 4.3 is given in Table 4.4 where the top parameters from the global sensitivity are given with their associated local ranks.

As this comparison shows, the local sensitivity analysis does a reasonable job identifying the most sensitive parameters with the global and local analysis. The two analyses identify the same top 3 parameters and identify 7 of the same parameters within their respective

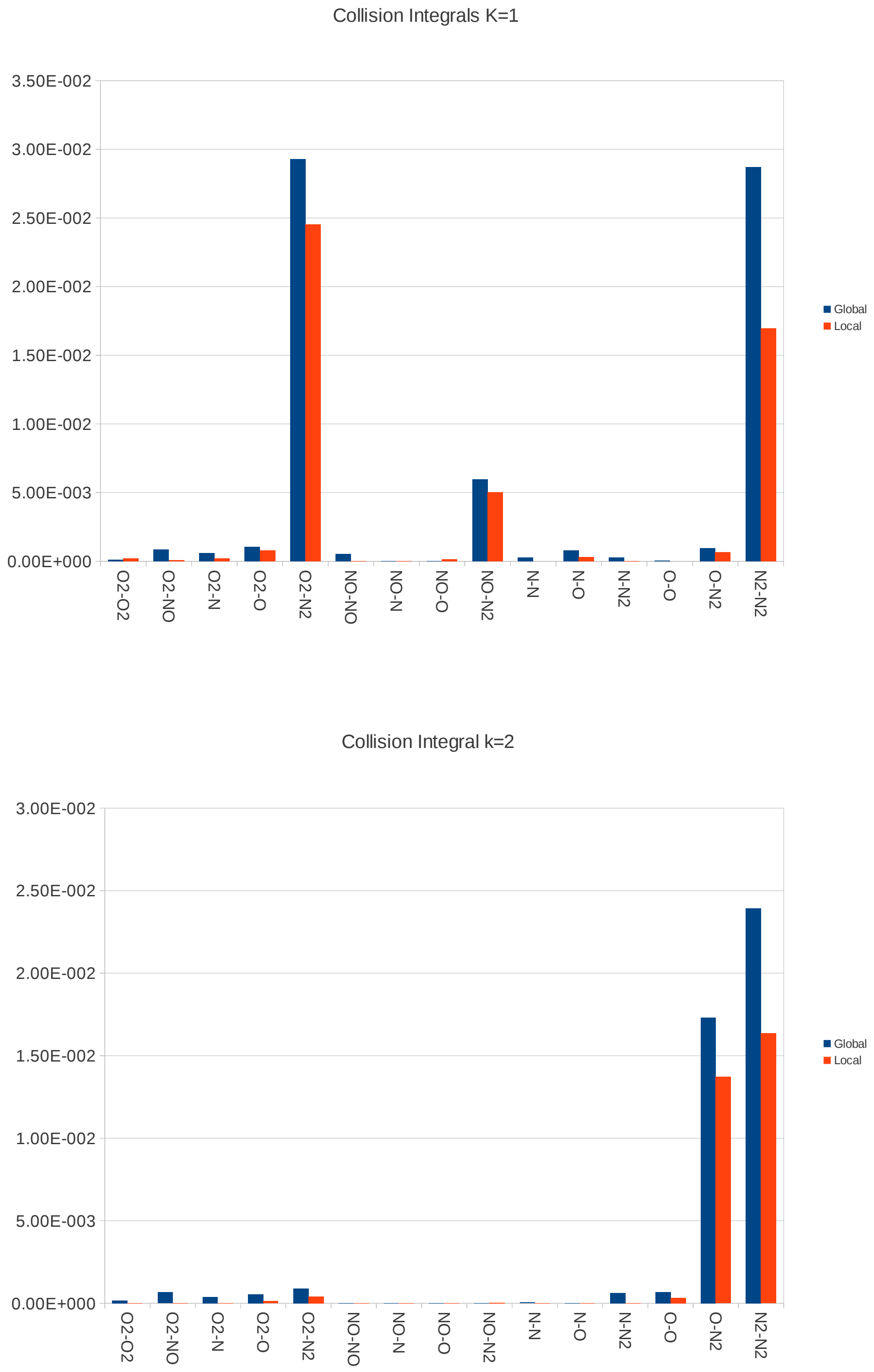


Figure 4.5: Comparison of local and global sensitivity analysis for collision integrals for (top) k=1 and (bottom) k=2.

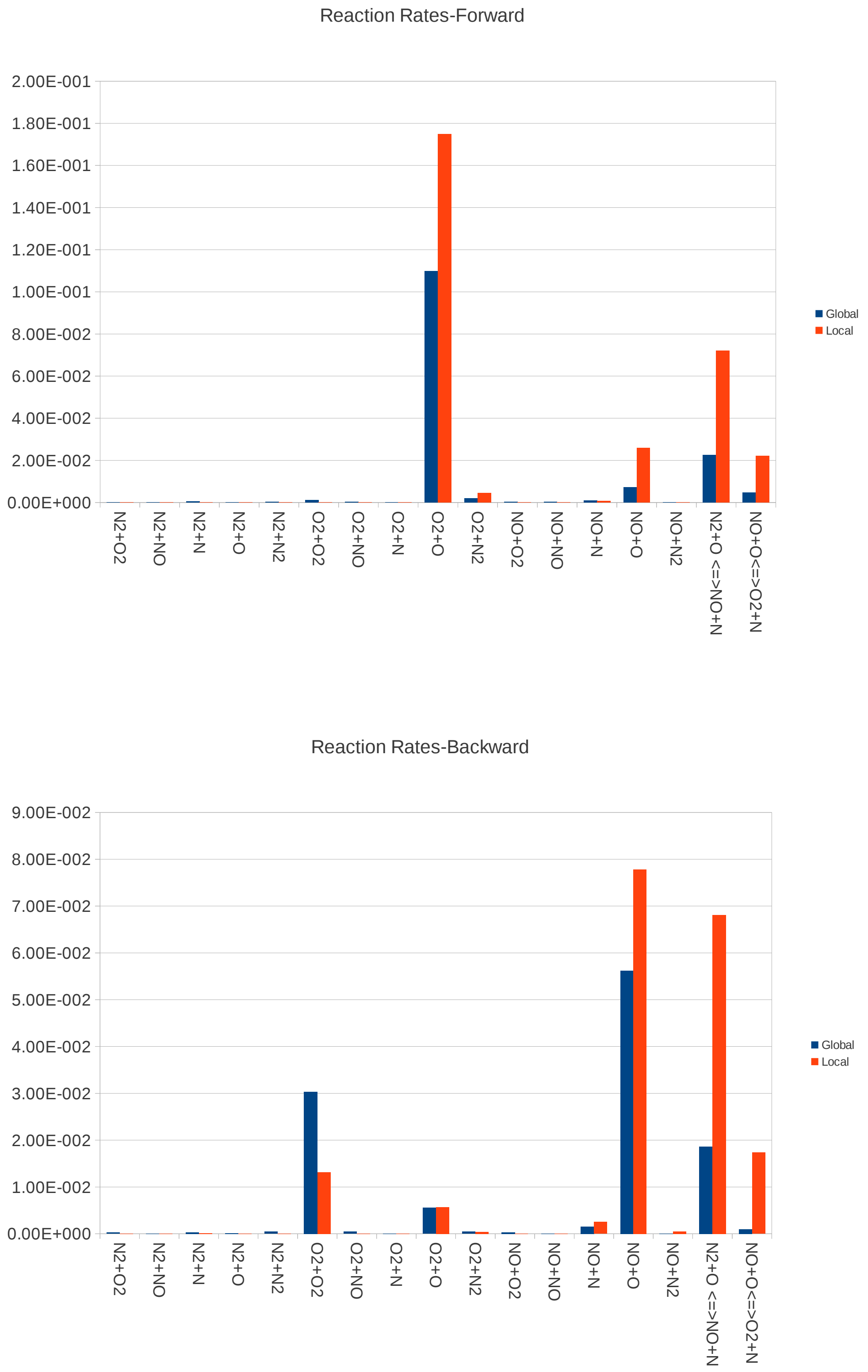


Figure 4.6: Comparison of local and global sensitivity analysis for reaction rates for (top) forward reaction and (bottom) backward reaction.

Table 4.3: Top 10 parameters from local sensitivity analysis compared with global

Rank	Variable	Global Rank	Local Contribution	Global Contribution
1	ρ_∞	1	0.43230	0.60055
2	$O_2 + O \leftrightarrows 2O + O$ (f)	2	1.7490×10^{-1}	1.0610×10^{-1}
3	$NO + O \leftrightarrows N + O + O$ (b)	3	7.7860×10^{-2}	5.1914×10^{-2}
4	$N_2 + O \leftrightarrows NO + N$ (f)	7	7.2017×10^{-2}	2.0647×10^{-2}
5	$N_2 + O \leftrightarrows NO + N$ (b)	10	6.8076×10^{-2}	1.2155×10^{-2}
6	$NO + O \leftrightarrows N + O + O$ (f)	11	2.5979×10^{-2}	8.3596×10^{-3}
7	O2-N2 (k=1)	4	2.4524×10^{-2}	4.2121×10^{-2}
8	$NO + O \leftrightarrows O_2 + N$ (f)	16	2.2155×10^{-2}	1.4609×10^{-3}
9	$NO + O \leftrightarrows O_2 + N$ (b)	63	1.7353×10^{-2}	3.7488×10^{-6}
10	N2-N2 (k=1)	6	1.6956×10^{-2}	3.1617×10^{-2}

Table 4.4: Top 10 parameters from global sensitivity analysis compared with local

Rank	Variable	Local Rank	Global Contribution	Local Contribution
1	ρ_∞	1	0.60055	0.43230
2	$O_2 + O \leftrightarrows 2O + O$ (f)	2	1.0610×10^{-1}	1.7490×10^{-1}
3	$NO + O \leftrightarrows N + O + O$ (b)	3	5.1914×10^{-2}	7.7560×10^{-2}
4	O2-N2 (k=1)	7	4.2121×10^{-2}	2.4524×10^{-2}
5	N2-N2 (k=1)	10	3.1617×10^{-2}	1.6956×10^{-2}
6	$O_2 + O_2 \leftrightarrows 2O + O_2$ (b)	13	2.1621×10^{-2}	1.3120×10^{-2}
7	$N_2 + O \leftrightarrows NO + N$ (f)	4	2.0647×10^{-2}	7.2017×10^{-2}
8	N2-N2 (k=2)	11	1.9019×10^{-2}	1.6354×10^{-2}
9	O-N2 (k=2)	12	1.3874×10^{-2}	1.3714×10^{-2}
10	$N_2 + O \leftrightarrows NO + N$ (b)	5	1.2155×10^{-2}	6.8076×10^{-2}

top 10 lists. Unfortunately, outside of the top three, the two types of analysis begin giving dramatically different rankings and the local sensitivity analysis seems to miss a number of the important collision integrals. Based on the uncertainty in the global sensitivity results, the disagreement in rankings is only important for variables with correlation coefficients significantly greater than 0.02, namely $O2 - N2(k = 1)$ and $N2 - N2(k = 1)$. The fact that the local analysis misidentifies the ranking of these parameters demonstrates a significant disagreement between the two forms of sensitivity analysis. Interestingly, these collision integrals are identified as important when the derivative magnitude is used and not weighted with the input uncertainty. When the rankings in Table 4.4 are compared with those from Table 4.1, there is more overlap between the two groups even though these rankings are based on different measures of sensitivity (derivative magnitude vs. correlation). When compared with the results in Table 4.1, 9 of the top 10 parameters are in agreement between the two analysis, even though the relative ranking of the parameters does vary, indicating the agreement may be the result of chance.

4.4.1 Gradient Accelerated Global Sensitivity Analysis

Based on the previous comparison, it is clear that the localized sensitivity analysis has significant deficiencies when compared to the global method when it comes to identifying the most important parameters and their contribution to the output uncertainty. In order to combine the speed of the localized analysis with the utility of the global sensitivity analysis, techniques based on rapid uncertainty quantification, namely surrogate models, can be incorporated into the Global sensitivity analysis. A surrogate model is a function that can be used to inexpensively approximate the output of a simulation based on a handful of simulation results. The main drawback with surrogate models is that the cost associated with constructing an accurate surrogate typically grows exponentially as the number of design variables increases. To address this limitation, two strategies are typically employed. First, a sensitivity analysis is performed to reduce the dimension of the space the surrogate must represent. Obviously, this approach cannot be used within this context as global sensitivity analysis is the purpose of the surrogate. The second approach for addressing the issue of dimensionality is the incorporation of gradient information into the surrogate model. By providing more information to the surrogate model, the cost associated with training an accurate surrogate can be reduced. Because an adjoint-based approach gives a cost independent of dimension, the gradient can provide additional information for the surrogate at a fixed cost. Hence, by calculating the simulation output and gradient at a number of points throughout the design space, an accurate surrogate may be constructed and used as a basis for global sensitivity analysis [19, 33, 34].

Two requirements will be imposed on the surrogate used for global sensitivity analysis. First, the surrogate must easily incorporate gradient information. Second, the cost associated with training the surrogate must increase minimally as the input dimension increases. As a practical limitation, linear growth in the cost as the dimension increases will be the largest allowed growth rate, with a cost independent of dimension preferred. The most simple surrogate that meets these requirements is least-squared polynomial regression incorporating gradient information [19].

The cost of constructing a polynomial regression model is dictated by the number of terms required in the regression, as the number of simulation results required for the surrogate must be greater than or equal to the number of terms in the regression. The number of terms in the regression is given by the following relation:

$$S = \frac{(d+p)!}{d!p!} \tag{4.53}$$

where d is the dimension of the space and p is the highest polynomial order. When gradient information is incorporated into the training of the surrogate, each simulation result provides $d+1$ pieces of information. Hence, the number of simulation results required for a surrogate using gradient information is given by:

$$N \geq \left\lceil \frac{(d+p)!}{d!p!(d+1)} \right\rceil \tag{4.54}$$

In order to satisfy the requirement of worst-case linear growth in the number of required simulations, only regression orders of $p = 1$ and $p = 2$ are considered. With this constraint, the required number of simulation results is given as:

$$N \geq \begin{cases} 1 & \text{for } p = 1, \\ \left\lceil \frac{(d+2)}{2} \right\rceil & \text{for } p = 2. \end{cases} \tag{4.55}$$

As these equations show, when gradient information is used, there is no constraint on the number of simulations required to construct a linear model. For a quadratic model, the expense associated with training the model increases only linearly. Equation (4.55) places a lower bound on the number of simulations required to construct the regression model. In practice, the regression model is typically over-determined and the coefficients are determined in a least-squares sense.

The regression model is based on the assumption that the output is approximated by a linear combination of polynomials given by equation:

$$y(D) = \sum_s \beta_s \Psi_s(D) \tag{4.56}$$

where $\Psi(D)$ represent a series of polynomials in D with degree less than p and β are a set of undetermined coefficients. The coefficients are determined by finding the set that best describe the simulation results at various values of D. Based on the results of N simulations, a system of equations can be solved to determine β.

$$\begin{bmatrix} \Psi_1(D_1) & \Psi_2(D_1) & \cdots & \Psi_s(D_1) \\ \Psi_1(D_2) & \Psi_2(D_2) & \cdots & \Psi_s(D_2) \\ \vdots & \ddots & \ddots & \vdots \\ \Psi_1(D_{N-1}) & \Psi_2(D_{N-1}) & \cdots & \Psi_s(D_{N-1}) \\ \Psi_1(D_N) & \Psi_2(D_N) & \cdots & \Psi_s(D_N) \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_s \end{bmatrix} = \begin{bmatrix} y(D_1) \\ y(D_2) \\ \vdots \\ y(D_{N-1}) \\ y(D_N) \end{bmatrix} \tag{4.57}$$

Here, the matrix on the left-hand side of the equation is defined as the collocation matrix H. Because N is typically greater than S (normally by a factor of two), this collocation matrix is inverted in a least-squared sense. This process is demonstrated below. Let the vector Y represent the results of N simulations using the input parameters D_i for $i = 1, 2, ..., N$. Additionally, define the collocation matrix H as the rectangular matrix (dimension $N \times S$)

whose elements represent the basis functions evaluated at the input parameters D_i. The coefficients β are determined by solving a set of linear equations, represented below.

$$H\beta = Y \tag{4.58}$$

Because H is a rectangular matrix, the coefficients are determined in a least-squares sense. To solve the problem in this manner, both sides of equation (4.58) are multiplied by the transpose of the collocation matrix. The product of the collocation matrix with its transpose is denoted as the regression matrix, A.

$$H^T H\beta = A\beta = H^T Y \tag{4.59}$$

Because the regression matrix A is square, it can be inverted to determine the coefficients that best describe the simulation outputs Y. These coefficients are given as:

$$\beta = A^{-1} H^T Y \tag{4.60}$$

Derivative observations can be incorporated into the regression model by differentiating equation (4.56).

$$\frac{\partial y(D)}{\partial D_k} = \sum_s \beta_s \frac{\partial \Psi_s(D)}{\partial D_k} \tag{4.61}$$

These derivative observations can be incorporated into the collocation matrix, providing additional equations without increasing the number of samples required for the regression.

$$\begin{bmatrix} \Psi_1(D_1) & \Psi_2(D_1) & \cdots & \Psi_s(D_1) \\ \frac{\partial \Psi_1(D_1)}{\partial D_1} & \frac{\partial \Psi_2(D_1)}{\partial D_1} & \cdots & \frac{\partial \Psi_s(D_1)}{\partial D_1} \\ \vdots & \ddots & \ddots & \vdots \\ \frac{\partial \Psi_1(D_1)}{\partial D_d} & \frac{\partial \Psi_2(D_1)}{\partial D_d} & \cdots & \frac{\partial \Psi_s(D_1)}{\partial D_d} \\ \vdots & \ddots & \ddots & \vdots \\ \Psi_1(D_N) & \Psi_2(D_N) & \cdots & \Psi_s(D_N) \\ \frac{\partial \Psi_1(D_N)}{\partial D_1} & \frac{\partial \Psi_2(D_N)}{\partial D_1} & \cdots & \frac{\partial \Psi_s(D_N)}{\partial D_1} \\ \vdots & \ddots & \ddots & \vdots \\ \frac{\partial \Psi_1(D_N)}{\partial D_d} & \frac{\partial \Psi_2(D_N)}{\partial D_d} & \cdots & \frac{\partial \Psi_s(D_N)}{\partial D_d} \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_s \end{bmatrix} = \begin{bmatrix} y(D_1) \\ \frac{\partial y(D_1)}{\partial D_1} \\ \vdots \\ \frac{\partial y(D_1)}{\partial D_d} \\ \vdots \\ y(D_N) \\ \frac{\partial y(D_N)}{\partial D_1} \\ \vdots \\ \frac{\partial y(D_N)}{\partial D_d} \end{bmatrix} \tag{4.62}$$

With the collocation defined as in equation (4.62), the regression coefficients are determined using the least-squares procedure given in equations (4.58) through (4.60).

The choice of basis functions Ψ_D used for this regression is general. In previous work, Hermite polynomials have been used within gradient-enhanced regression models [19] and multidimensional basis have been constructed by means of a tensor product of one dimensional polynomials [20].

$$\Psi_k(D) = \prod_{i=1}^{d} H_{m_i^j}(D_i) \tag{4.63}$$

Here, m_i^j is a multi-index dictating the order of the polynomial H. The basis of a given order p will contain products of polynomials less than or equal to degree p. In this way, the basis is hierarchical in that the basis of degree p contains all the terms of basis of degree $p-1$. Because this work is limited to $p = 1$ and $p = 2$ to satisfy the scaling requirements, an example basis of degree 2 in 3 dimensions is given below.

$$\Psi(D) = \begin{bmatrix} 1, \\ H_1(D_1), H_1(D_2), H_1(D_3) \\ H_1(D_1) \times H_1(D_2), H_1(D_1) \times H_1(D_3), H_1(D_2) \times H_1(D_3), \\ H_2(D_1), H_2(D_2), H_2(D_3) \end{bmatrix} \tag{4.64}$$

Here, H_k is the hermite polynomial of degree k. As the above set demonstrates, the number of terms in the regression grows rapidly as order and/or dimension is increased.

The gradient-enhanced regression model detailed here can provide a basis for rapid global sensitivity analysis even in the limit of large input dimension when the degree of the regression is limited to 2. To demonstrate this capability, a global sensitivity analysis is performed on the real gas solver using the previously enumerated 66 model parameters based on a polynomial regression of degree 1 and 2. The results of this regression-based analysis are then compared with the Monte Carlo global sensitivity analysis results.

For 66 dimensions, the size of the regression basis is 67 terms for $p = 1$ and 2278 terms for $p = 2$. When gradient values are included, these facts indicate a single simulation result can be used for a $p = 1$ regression and 34 simulation results are required for a $p = 2$ regression. These numbers represent the minimum number of simulations and in practice additional simulations can be incorporated into the regression by additional over-determination of the system. The training data for the regression is found by performing simulations using randomly sampled inputs. The random inputs are generated through uniform Latin Hypercube sampling and the total number of simulations is determined by the computational budget.

In order to demonstrate the advantage of the regression approach, the top parameters from a P=1 and P=2 regression are calculated and compared to the Monte Carlo global sensitivity results. The linear regression is built from 10 sample points with associated function/gradient values spread uniformly throughout the design space. The quadratic regression is built from 68 function/gradient evaluations. The number of training points for each regression was determined by monitoring the convergence of statistic predictions and is demonstrated in Chapter 5.

With this regression model built, samples are extracted from the model using the same sample points used for the Monte Carlo global sensitivity analysis. In Tables 4.5 and 4.6, the top 10 parameters from a P=1 and P=2 regression are given and compared to the global sensitivity results. As the Table demonstrates, the top parameters from the regression based analysis show greater overlap with the top parameters from the Monte Carlo global sensitivity analysis for both the $p = 1$ and $p = 2$ models. For the localized analysis, only seven of the top

Table 4.5: Top 10 parameters from P=1 regression sensitivity analysis compared with global

Rank	Variable	Global Rank	Regression Contribution	Global Contribution
1	ρ_∞	1	0.49814	0.60055
2	$O_2 + O \leftrightarrows 2O + O$ (f)	2	1.5388×10^{-1}	1.0610×10^{-1}
3	$NO + O \leftrightarrows N + O + O$ (b)	3	1.0557×10^{-1}	5.1914×10^{-2}
4	$O_2 + O_2 \leftrightarrows 2O + O_2$ (b)	6	5.0369×10^{-2}	2.1621×10^{-2}
5	O2-N2 (k=1)	4	4.3405×10^{-2}	4.2121×10^{-2}
6	N2-N2 (k=1)	5	3.6679×10^{-2}	3.1617×10^{-2}
7	$O_2 + O \leftrightarrows 2O + O_2$ (b)	13	2.4095×10^{-2}	7.4280×10^{-3}
8	N2-N2 (k=2)	8	1.5437×10^{-2}	1.9019×10^{-2}
9	O-N2 (k=2)	9	1.4627×10^{-2}	1.3874×10^{-2}
10	$O_2 + N_2 \leftrightarrows 2O + N_2$ (f)	15	1.2281×10^{-2}	3.5848×10^{-3}

Table 4.6: Top 10 parameters from P=2 regression sensitivity analysis compared with global

Rank	Variable	Global Rank	Regression Contribution	Global Contribution
1	ρ_∞	1	0.56879	0.60055
2	$O_2 + O \leftrightarrows 2O + O$ (f)	2	1.0002×10^{-1}	1.0610×10^{-1}
3	$O_2 + O_2 \leftrightarrows 2O + O_2$ (b)	6	5.7669×10^{-2}	2.1621×10^{-2}
4	$NO + O \leftrightarrows N + O + O$ (b)	3	4.0057×10^{-1}	5.1914×10^{-2}
5	N2-N2 (k=1)	5	3.7461×10^{-2}	3.1617×10^{-2}
6	O2-N2 (k=1)	4	3.3299×10^{-2}	4.2121×10^{-2}
7	N2-N2 (k=2)	8	2.1163×10^{-2}	1.9019×10^{-2}
8	O-N2 (k=2)	9	1.7395×10^{-2}	1.3874×10^{-2}
9	V_∞	14	1.3497×10^{-2}	4.8401×10^{-3}
10	$O_2 + O \leftrightarrows 2O + O$ (b)	13	1.1734×10^{-2}	7.4280×10^{-3}

ten parameters were in agreement when the comparison was made based on the contribution to uncertainty. When the regression model is used, eight of the top ten parameters are in agreement and the rankings produced by the methods are in greater agreement. In addition to identifying more of the important parameters from the global sensitivity analysis, the $p = 2$ regression shows good agreement in predicting the contribution to output uncertainty for most of the top parameters. This predictive capability is likely important for the dimension reduction required for many surrogate-based approaches to uncertainty quantification.

4.5 Summary

In this chapter, procedures for sensitivity analysis were outlined. Because the calculation of derivatives was central to this sensitivity analysis, the calculation of gradients via an adjoint based approach was first detailed. With the adjoint calculation given, the application of derivatives for a localized sensitivity analysis was demonstrated for the $5km/s$ real gas flow over a cylinder. Using this localized sensitivity analysis, the important parameters from the collision integral and chemical kinetics model were identified. In order to investigate potential drawbacks of this localized analysis, a global sensitivity analysis was performed and the results were compared with the local analysis. Finally, a gradient-based approach for global sensitivity analysis based on polynomial regression was explored to combine the strengths of the localized and global methods.

To further compare the methods presented in this section, the cost of each of the sensitivity analyses in terms of cpu-hours should be presented. For the localized sensitivity analysis, a single function and gradient evaluation is required, giving a cost of 2.45 cpu-hours. For the Monte Carlo based global sensitivity analysis, 6331 function evaluations (corresponding to an equal number of flow solutions) were required. These 6331 function evaluations required 5090 cpu-hours with each function evaluation requiring an average of 2893 seconds. The average function time is less for the Monte Carlo sampling than the normal flow analysis as the flow solution with mean parameters can be used as the starting point for simulations with parameters away from the mean, reducing the time required for a flow solution with parameters away from their mean values. Although significantly more expensive than the local analysis, this global analysis gives a more widely accepted sensitivity for hypersonic flow simulations. The expense of global sensitivity analysis was reduced using a gradient-enhanced polynomial regression. For the two regression orders tested in this chapter, 10 and 68 function and gradient evaluations were required for the $P = 1$ and $P = 2$ respectively. These two regression orders required 11.01 cpu-hours and 60.97 cpu-hours respectively. All of the costs reported here are measured using a single core of an Intel Core i7-870 clocked at 2.93 GHz. Because the samples are independent in nature, the Monte Carlo sampling and acquisition of the training data can be performed in parallel to reduce the cost.

Comparing the cost of these methods, the localized analysis is clearly the least expensive; however, this analysis can be insufficient for nonlinear problems and large perturbations of the input parameters. For the global sensitivity analysis, the regression-based approach gives results similar to the Monte Carlo based approach but at a fraction of the cost. For the $P = 1$ regression, a factor of 462 cost savings was achieved, while the $P = 2$ regression achieved a factor of 83 savings. Hence, for problems in which the low order regression is sufficient for the sensitivity analysis, which appears to be the case for the test case presented here, the gradient-enhanced regression approach is capable of greatly reducing the cost associated with global sensitivity analysis. Further analysis of the costs associated with sampling and surrogate training are given in Chapter 5.

Chapter 5

Aleatory Uncertainty Quantification Results

In this chapter, the gradient-based aleatory uncertainty quantification strategies are outlined and applied to a test case using the real gas CFD solver. To provide the necessary background, each strategy is detailed. With the strategies specified, the performance of these methods is assessed based on accuracy and computational cost. To characterize the performance these strategies, a data set of simulation results is generated using Monte Carlo sampling. Based on this data, the statistic predictions and overall surrogate error for each gradient-based approach is judged. Despite the uncertainty associated with Monte Carlo sampling, the performance of the surrogates are validated against these results in the sense that consistency between the gradient-based strategies and Monte Carlo sampling is ensured, the latter method being a widely accepted approach for characterizing aleatory uncertainties.

In this chapter, the statistics based on the linear methods, such as the moment method and linear extrapolation, are first calculated and compared against those of the validation data. Next, the polynomial regression approaches outlined in Chapter 4 are used to represent the design space and the surrogate error as well as statistic predictions are calculated for each model. Finally, gradient-enhanced Kriging methods are used to approximate the design space. Because of the dimensional issues relating to Kriging models, the space approximated by the Kriging model is reduced in two different ways. The first method simply truncates the set of variables used to construct the model based on a global sensitivity analysis and the low importance variables are ignored. The second method relies on a low order model built over all variables that is then enhanced with a Kriging model over the important variables. The performance is again assessed based on the statistic predictions and surrogate approximation error for each model. The chapter concludes with a comparison of the results from each method and the savings associated with each model when compared to exhaustive Monte Carlo sampling.

5.1 Gradient-based Aleatory Uncertainty Quantification

Acceleration techniques for propagating aleatory uncertainties typically rely on assuming a functional representation of the output over the design space and sampling from this approximation. The most simple functional representation is a linear function over the design space. When gradient information is available, this model is possible with a single function/gradient evaluation. Using this information, a Taylor series approximation can be constructed using the function and gradient evaluated at the mean input parameters. In the case that only the mean and variance are desired, the moment method can be used. For a first-order moment method, the mean and variance can be given by [16]:

$$\mu = f(\bar{x}) \tag{5.1}$$

$$\sigma^2 = \sum_{i=1}^{d} \frac{\partial f}{\partial x_i}^2 \sigma_i^2 \tag{5.2}$$

where μ is the mean of the output, $\bar{x}$ is the mean of the inputs, σ_i is the standard deviation of the input x_i and σ is the standard deviation of the output. Although this method does not rely explicitly on a Taylor series approximation, it is implicitly dependent on this approximation.

If other statistics are desired, inexpensive Monte Carlo can be performed. In the case of the linear model, approximate function values are calculated for the randomly sampled inputs based on the Taylor series about the mean input values [16].

$$y_{lin}(x) = f(\bar{x}) + \left.\frac{\partial f}{\partial x_i}\right|_{\bar{x}} (x_i - \bar{x}_i) \tag{5.3}$$

Obviously, any functional representation can be used as a basis for modeling the variation of the function f throughout the design space, known as the surrogate model. As was presented in Chapter 4, the Taylor series can be replaced with polynomial regression. The main drawback of this functional representation is the "curse of dimensionality" with the cost increasing exponentially as the dimension of the space increases. For the global sensitivity analysis presented in Chapter 4, this problem was overcome by restricting the order of the polynomial to no higher than 2 and incorporating gradient information into the training of the regression model. Although this second order regression appeared accurate enough for sensitivity analysis, the limitation of second order may be overly restrictive for a general surrogate model.

For a general surrogate model, a two pronged approach is employed. First, the dimension of the space modeled by the surrogate model must be reduced. For this work, dimension reduction is based exclusively on the sensitivity analysis detailed in Chapter 4. As was demonstrated in Chapter 4, the variance in the output for this work is due to a small subset of the variables. Hence, constructing a surrogate over only this subset should be able to account for the majority of the variation in the output. Second, the surrogate is enhanced with gradient information to provide additional information to the surrogate at reduced cost. This two pronged approach is employed exclusively on Kriging surrogate models. Although it suffers from the curse of dimensionality, Kriging models can easily accommodate gradient information and perform well in the low sample density regime [83]. Additionally, the polynomial regression presented in Chapter 4 can be easily incorporated into the Kriging model, allowing for a unification of the various surrogate model discussed here.

A Kriging model is a form of regression that is able to account for correlation between the data points, enabling more accurate function representations. An overview of Kriging models is given here based on the descriptions in Reference [83]. The Kriging model is premised on the assumption that the output data y obey a Gaussian process, specified as:

$$y = N(m(x), K(x, x'; \theta)) \tag{5.4}$$

where $m(x)$ is the mean function, $K(x, x'; \theta)$ represents the covariance between data points, and θ represent parameters used to govern the covariance function, known as hyperparameters. The choice of mean function can vary widely based on the application of the Kriging model. Within machine learning applications where training data is abundant, a zero mean function is often used, known as simple Kriging. For the purposes of uncertainty quantification, a non-zero mean function is typically required. This mean function can be explicitly defined or incorporated as part of the Kriging construction. For an explicitly defined mean function, the output is represented as the sum of the mean function with a zero mean Gaussian process [83].

$$y = m(x) + N(0, K(x, x'; \theta)) \tag{5.5}$$

For an explicit mean function, the zero mean Kriging model is built based on the residual between the measured output y and the explicitly defined mean function $m(x)$.

To construct the Kriging model, it is first "trained" using a number of simulation results, represented as Y, evaluated at a set of input parameters, $\vec{X}$. Using these results, the parameters in the Gaussian process are fitted and predictions away from these training points can be made. The prediction from the Kriging model is itself a Gaussian process with an associated mean value and variance. Using the covariance between points in the domain, model predictions throughout the domain are determined by sampling from the conditional distribution $y_*|\vec{X}, Y$ where $\vec{X}, Y$ are the input and output training data. The posterior mean predictions for an explicit mean are given by the formula [83]:

$$y(\vec{x}_*)|\vec{X}, Y, m(x) = m(x) + k_*^T K^{-1}(Y - m(x)) \tag{5.6}$$

where k_*^T represents the covariance between the test point, $\vec{x}_*$, and the training points $\vec{X}$ (a row vector of length N). The term K is the covariance between the training data, represented by a matrix of dimension $N \times N$.

In addition to an explicitly defined mean function, a non-zero mean function can be incorporated into the construction of the Gaussian process. This non-zero mean function usually takes the form of a polynomial regression and the coefficients within the regression are informed by the correlation within the Kriging model. When a polynomial mean function is used, the Kriging model is referred to as Universal Kriging. The special case of a zeroth order regression, meaning a constant mean function, is referred to as ordinary Kriging. For a universal Kriging model, the functional form becomes [83]:

$$y(\vec{x}) = N(h(\vec{x})\beta, K(\vec{x}, \vec{x}; \theta)) \tag{5.7}$$

where $h(\vec{x})$ is a column vector containing the basis functions of the regression evaluated at the point $\vec{x}$ and β are the regression parameters. Using a regression-based mean function, predictions can be made based on this model using the formula:

$$y(\vec{x}_*)|\vec{X}, Y = h(\vec{x}_*)\beta + k_*^T K^{-1}(Y - H\beta) \tag{5.8}$$

where H is the collocation matrix of the regression, Y is the vector of training function values, K is the covariance matrix between the training points and k_* is the vector of covariances between the training points and the test point (x_*). Since the regression is built from a limited number of training points, it is prudent to assume that the regression parameters belong to a distribution of parameters. In the limit of zero knowledge of this distribution (vague prior assumption), the optimal regression parameters are given by [83]:

$$\hat{\beta} = (H^T K^{-1} H)^{-1} H^T K^{-1} Y = A^{-1} H^T K^{-1} Y \tag{5.9}$$

where A is the regression matrix defined as $H^T K^{-1} H$. Unlike the regression matrix used in polynomial regression, the regression matrix for the Kriging model includes the correlation between the data points, K. The case of polynomial regression is found by assuming no correlation between the data points, represented as K reducing to the identity matrix. Using this definition of regression parameters, mean predictions can be made using the following single formula:

$$y(\vec{x}_*)|\vec{X}, Y = k_*^T K^{-1} Y + (h(\vec{x}_*) - k_*^T K^{-1} H)\hat{\beta} = k_*^T K^{-1} Y + R(\vec{x}_*)\hat{\beta} \tag{5.10}$$

The elements of the covariance matrix represent the covariance between the function values. For Kriging, the covariance between function values is assumed to be a function of distance between the two data points. This functional form is known as a stationary covariance function [83].

$$K_{i,j} = cov(y_i, y_j) = k(|\vec{x}_i - \vec{x}_j|) + \sigma_n \delta_{i,j} \tag{5.11}$$

Here, $k(|\vec{x}_i - \vec{x}_j|)$ is the covariance between training points i and j and σ_n represents the noise in the training data. For machine learning applications, this noise is treated as a hyperparameter and fitted through the likelihood equation, which is described below. For the application of uncertainty quantification, this noise is a specified value and is set to ensure proper conditioning of the covariance matrix. The multidimensional covariance function is formed as the product of one dimensional covariance functions as this form produces better conditioned covariance matrices [84]. The covariance between two points (denoted as $\vec{x}$ and $\vec{x}'$ to avoid confusion) is given below.

$$k(\vec{x}, \vec{x}'; \theta) = \sigma^2 \prod_{i=1}^{d} k_i(x_i - x_i'; \theta_i) = \sigma^2 \prod_{i=1}^{d} k_i(r_i; \theta_i) \tag{5.12}$$

Here, σ represents the covariance magnitude and θ_i is the length scale for each dimension. The one dimensional covariance function used in this work is the Matern function with the parameter $\nu = 3/2$, given below [83].

$$k_i(x_i - x_i') = \left(1 + \sqrt{3}\left|\frac{x_i - x_i'}{\theta_i}\right|\right) e^{-\sqrt{3}\left|\frac{x_i - x_i'}{\theta_i}\right|} \tag{5.13}$$

The parameters in the covariance function are determined via the likelihood equation. The likelihood equation gives the probability that the data with a given set of parameters satisfy the assumed Gaussian process. By maximizing this probability, the optimal parameters can be determined. The log-likelihood equation for a universal Kriging model is given by [83]:

$$\log(p(y|X;\theta)) = -\frac{1}{2}(Y - H^T\beta)^T K^{-1}(Y - H^T\beta) - \frac{1}{2}\log|K| - \frac{N}{2}\log 2\pi \tag{5.14}$$

where N is the number of training points. Based on this likelihood equation, the optimal covariance magnitude can be found analytically. By differentiating the likelihood equation with respect to the covariance magnitude, the optimal magnitude can be determined. Denoting the covariance matrix with a magnitude of 1 as $\hat{K}$, the optimal magnitude is given as [84]:

$$\sigma(\theta)^2 = \frac{(Y - H^T\beta(\theta))\hat{K}(\theta)^{-1}(Y - H^T\beta(\theta))}{N} \tag{5.15}$$

Using this explicit equation for the magnitude, the likelihood equation can be rewritten as:

$$\log(p(y|X;\theta)) = -n\log\sigma(\theta)^2 - \log(|\hat{K}(\theta)|) - \frac{N}{2}\log 2\pi \tag{5.16}$$

The only parameters without analytic expressions are the length scales θ. These length scales are determined via numerical optimization. For this work, a pattern search [85] is used to determine the optimal parameters as a global deterministic optimization technique is preferable to non-deterministic methods, such as genetic algorithms, as deterministic methods allow for repeatability of results. This optimization represents the most expensive part of constructing the Kriging model, as the covariance matrix must be constructed and inverted for each function evaluation in the optimization. Additionally, the dimension of the optimization problem corresponds to the dimension of the space approximated by the model.

Like the polynomial regression presented in Chapter 4, the Kriging model is also susceptible to the “curse” of dimensionality. Although the Kriging model does not have an explicit relationship for the number of training points as the dimension of the space expands, the amount of data required for an accurate model typically increases exponentially fast as the number of parameters increases. This fact is because an accurate surrogate requires training points that fill the design space. To reduce the cost of training the model as the dimension expands, gradient information can be incorporated into the Kriging model. To include gradient information in the Kriging model, the covariance matrix must first be extended to include the covariance between derivative values. When these correlations are included, the covariance becomes a block matrix with the following components.

$$\underline{K} = \begin{bmatrix} cov(Y,Y) & cov(Y,\nabla Y) \\ cov(\nabla Y,Y) & cov(\nabla Y,\nabla Y) \end{bmatrix} \tag{5.17}$$

The covariance between function and gradient components is found by differentiating the covariance function [86]:

$$cov(\frac{\partial y}{\partial x_k}, y') = \frac{\partial}{\partial x_k} k(\vec{x}, \vec{x}') \tag{5.18}$$

Differentiating once more (now w.r.t to the second argument of the covariance function) gives the covariance between gradient components:

$$cov(\frac{\partial y}{\partial x_k}, \frac{\partial y'}{\partial x'_l}) = \frac{\partial^2}{\partial x_k \partial x'_l} k(\vec{x}, \vec{x}') \tag{5.19}$$

The gradient vector and Hessian matrix resulting from equations (5.18) and (5.19) can then be arranged into the matrices $cov(\nabla Y, Y)$ and $cov(\nabla Y, \nabla Y)$ respectively.

In addition to extending the covariance matrix, the training data must be redefined as:

$$Y = \begin{bmatrix} Y \\ \nabla Y \end{bmatrix} \tag{5.20}$$

and the collocation matrix must be extended to include the derivative of the basis function evaluated at the training points. This extension is performed by differentiation of the basis functions, as was demonstrated in Chapter 4.

5.2 Problem Statement and Validation Results

The problem examined throughout this chapter is determining the uncertainty in integrated surface heating for the $5km/s$ flow of the cylinder outlined in Chapter 3. The definition integrated surface heating, given in Chapter 4, is given by:

Table 5.1: Input model parameters with uncertainty

Number	Variable	Mean	Standard Deviations
1	$\rho_\infty \ (kg/m^3)$	1×10^{-3}	5%
2	$V_\infty (m/s)$	5000	15.42
3-17	A^1_{s-r}	1	5%
18-32	A^2_{s-r}	1	5%
33-49	ξ_f	0	0.25
50-66	ξ_b	0	0.25

$$L = -\frac{\int_{\partial\Omega} \left(k \nabla T \cdot \vec{n} + k_v \nabla T_v \cdot \vec{n} \right) dA}{\frac{1}{2} \rho_\infty V_\infty^3} \tag{5.21}$$

where T is the translational-rotational temperature, k is the translational-rotational thermal conductivity, T_v is the vibrational temperature, and k_v is the vibrational thermal conductivity.

The uncertainty in this output due to 66 uncertain input parameters is determined. These 66 parameters include collision integrals (A^1_{s-r} and A^2_{s-r}), reaction rates (ξ_f and ξ_b), and freestream parameters (ρ_∞ and V_∞). These parameters are described in detail in Chapter 4. For this study, these input parameters are assumed to be aleatory and described by a Gaussian distribution. These variables and the associated mean and standard deviations are given in Table 5.1.

Statistics for the output are estimated through Monte Carlo sampling. To implement this sampling, input samples are drawn from the corresponding input distribution. A CFD simulation is then performed using these sample points and statistics are estimated for the output. Because these samples are independent, the simulations are performed in parallel using MPI. For the validation results, 6331 samples were used corresponding to an equal number of CFD simulations. The convergence of the mean and variance estimate are found in Figure 5.1. The sample mean is given by:

$$\mu = \frac{1}{N} \sum_s y_i \tag{5.22}$$

while the sample variance is given by:

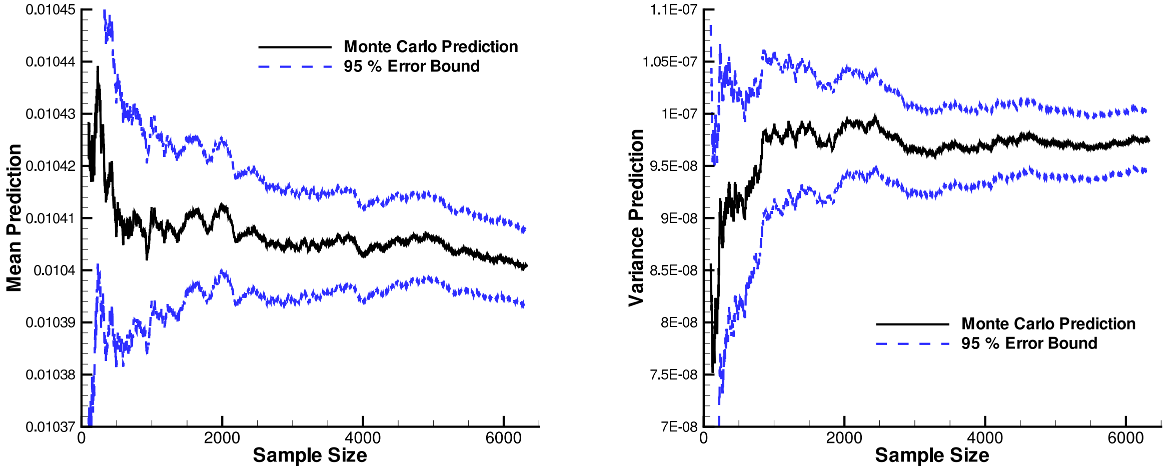


Figure 5.1: Monte Carlo convergence of surface heating for average and variance for increasing sample size.

$$S^2 = \frac{1}{N-1} \sum_{s} (y_i^2 - \mu^2) \tag{5.23}$$

The square root of the sample variance is the standard deviation and can be used to calculate a 95% confidence interval for the output, given by ±2 standard deviations around the mean when N is large enough.

Because the statistics are based on a limited number of samples, the Monte Carlo results require an error estimate so that the confidence level of the statistic predictions can be quantified and a sufficient number of samples for the desired level of accuracy can be chosen. For Monte Carlo sampling, the uncertainty in the statistic are quantified using a distribution and the confidence level for the prediction is quantified using the variance of this distribution. For Monte Carlo sampling, the statistics obeys a normal distribution with a variance dependent on the number of samples. For the mean, the standard deviation for this convergence distribution is given by [87]:

$$\sigma_N = \frac{\sigma}{\sqrt{N}} \tag{5.24}$$

where σ is the underlying standard deviation of the the simulation objective. This standard deviation is in turn estimated by the sample variance (S^2) of the Monte Carlo results. Using this standard deviation estimate, a 95% confidence interval for the average prediction can be established based on a two-σ bound, given as:

$$\mu_L = \mu - 2\frac{\sigma}{\sqrt{N}} \tag{5.25}$$

$$\mu_R = \mu + 2\frac{\sigma}{\sqrt{N}} \tag{5.26}$$

where μ is the sample mean and σ is approximated by the sample standard deviation, S.

To provide bounds for the sample variance, its variance must also be estimated. For independent, identically distributed samples, the variance of the sample variance can be calculated as [88]:

$$Var(S^2) = \sigma^4 \left(\frac{2}{N-1} + \frac{K}{N} \right) \tag{5.27}$$

where K is the excess kurtosis of the distribution being sampled. The kurtosis can be estimated using the sample kurtosis G, given as [89]:

$$G = \frac{\frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})^4}{\left(\frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})^2\right)^2} - 3 \tag{5.28}$$

The 95% confidence interval for the variance prediction can be calculated as:

$$S_L^2 = S^2 - 2\sigma^2\sqrt{\frac{2}{N-1} + \frac{K}{N}} \tag{5.29}$$

$$S_R^2 = S^2 + 2\sigma^2\sqrt{\frac{2}{N-1} + \frac{K}{N}} \tag{5.30}$$

where σ^2 is approximated by the sample variance, and K is approximated by the sample kurtosis, G. Due to the importance of tail statistics in calculating the higher order moments required in the Kurtosis, the sample Kurtosis can be unreliable for all but large numbers of samples. However, in this context, a highly accurate Kurtosis is not required as it is scaled by the number of samples point used for the Monte Carlo sampling, N. For hypersonic applications, reasonably converged statistics require N to be on the order of 1000 [2,13]. For

Table 5.2: Statistics based on Monte Carlo sampling using 6331 points.

Statistic	Prediction	Lower Limit	Upper Limit
Mean	1.0401E-002	1.0393E-002	1.0409E-002
Variance	9.7517E-008	9.3979E-008	1.0106E-007
Standard Deviation	3.1228E-004	3.0656E-004	3.1789E-004
95% Confidence Interval	±6.0049%	±5.8994%	±6.1083%

the data set generated in this work, the sample Kurtosis of the output was 0.083 compared to the 0 expected if the output obeyed an exact normal distribution. Hence, the effect of the Kurtosis on the variance error estimate was small for this case.

Table 5.2 shows the estimate of the average and variance based on the Monte Carlo results. Using the above formulas, the uncertainty in these statistics is also estimated and given in Table 5.2 in the form of a 95% confidence interval on each of the statistics.

In addition to these statistics, the entire cumulative distribution function (CDF) of the output is often desired. The CDF is the integral of the probability density function and is useful for determining the percentiles of the distribution and characterizing the tails of the distribution. In order to provide addition validation data, the CDF of the Monte Carlo results is constructed and shown in Figure 5.2.

Using these validation results, the performance of the proposed gradient-based uncertainty quantification strategies is assessed.

5.3 Linear Methods

The first gradient-based method tested for this problem is the first-order moment method. For this method, the average is estimated by the output evaluated at the average of the input parameters and the variance is given by the gradient evaluated at the average input parameters multiplied by the standard deviation of the input parameters. The average and variance estimates are given in Table 5.3 and compared with the Monte Carlo results. As the table demonstrates, the moment method overestimates the variance and underestimates the mean. For both statistics, the moment method produces estimates outside of the intervals

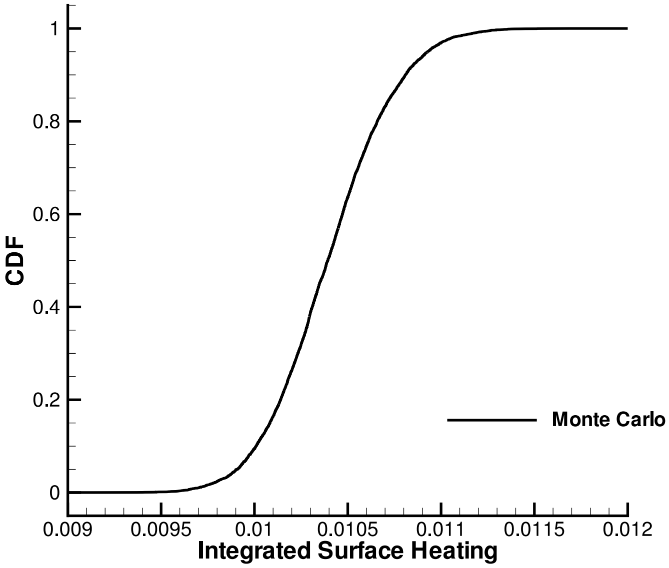


Figure 5.2: CDF curve for Monte Carlo results based on 6331 sample points

produced by the Monte Carlo sampling. Despite this disagreement, the method produces a variance estimate in the same range as Monte Carlo. For applications where rapid uncertainty estimates are required, such as robust optimization, the moment method is likely sufficient for this problem. The moment method is limited to predicting moments of the distribution, such as the statistics and variance. In order to predict other statistics, such as quantiles or a full CDF curve, linear extrapolation about the mean can be used. In order to test this functional form, approximate function values using the input samples from the Monte Carlo were calculated using equation (5.3) and statistics were predicted. The mean and variance prediction from this linear extrapolation are also given in Table 5.3. In addition to predicting specific statistics, an approximate CDF curve based on linear extrapolation is constructed. This approximate CDF is plotted in Figure 5.3 and compared to the CDF from Monte Carlo sampling. As the statistic predictions and CDF curve show, the linear approach does a reasonable job approximating the upper part of the CDF curve but deviates significantly for the bottom tail of the distribution. Although the results of the first-order moment method and linear extrapolation should be identical in theory, in practice the two methods will

Table 5.3: Statistics based on the moment method and linear extrapolation

Statistic	Moment Method	Linear Extrapolation	Monte Carlo Lower	Monte Carlo Upper
Mean	1.0370E-002	1.0369E-002	1.0393E-002	1.0409E-002
Variance	1.3790E-007	1.3412E-007	9.3979E-008	1.0106E-007
Standard Deviation	3.7134E-004	3.6622E-004	3.0656E-004	3.1789E-004
95% Confidence Interval	±7.1616%	±7.0638%	±5.8994%	±6.1083%

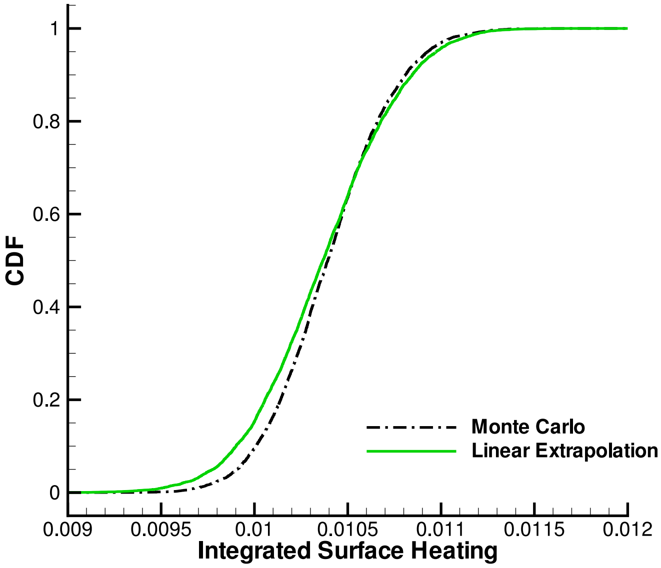


Figure 5.3: CDF curve from linear extrapolation compared to CDF curve from Monte Carlo sampling.

give different results due to the finite sample size used to acquire results from the linear extrapolation model. As the number of sample points increases, the linear extrapolation results will approach those of the moment method.

5.4 Regression

In order to improve upon the predictions of the moment method and linear extrapolation, these simplified relationships are replaced with polynomial regression. The details of these regression models were given in Chapter 4. As was the case for the sensitivity analysis, the regression is limited to first and second order. Because the coefficients in the regression model are solved in a least-squares sense, there is no upper limit on the number of sample points that can be used to construct the regression and only a minimum number of sample points is required. Hence, the optimal number of training points for the regression must first be determined. In order to assess the effect of the number of training points on the regression predictions, the surrogate errors based on increasing numbers of training points for a first and second order regression model were calculated. The surrogate error is calculate based on the difference between the Monte Carlo function values and those predicted by the surrogate at the same sample locations. For the results presented throughout, the RMS surrogate error is used. The variation of this error versus the number of simulations used to construct the regression is plotted in Figure 5.4 for both the first and second-order model. Figure 5.5 shows the convergence of the statistic predictions for the model. A simulation result in this context represents a function and gradient evaluation and provides $d+1$ pieces of information, where d is the dimension of the problem.

From the plot, both models show non-monotone variation in the error and the error does not appear to converge as the number of function/gradient evaluations (or training points) is increased. Additionally, the convergence of the statistics also shows erratic variation. In practice, the Monte Carlo results are not available, so these statistic predictions are the only measure of the solution that can be used to evaluate the number of training points needed for an accurate model. For the first-order model, the error initially drops rapidly, before settling into an oscillatory behavior. The first minimum in the error occurs for 10 sample points. Based on the convergence of the statistics, 10 function/gradient evaluations is also the point where the variance prediction settles into a range typical of the large number of sample points. Based on these observations, 10 will be the number of training points used throughout for the first order model. For the second order model, the variation of the error is slight with a minimum at 136 training points. Based on the statistic predictions, 136 training points also gives the most accurate variance prediction, although the variance prediction from 68 training points is more typical of the value seen for larger number of

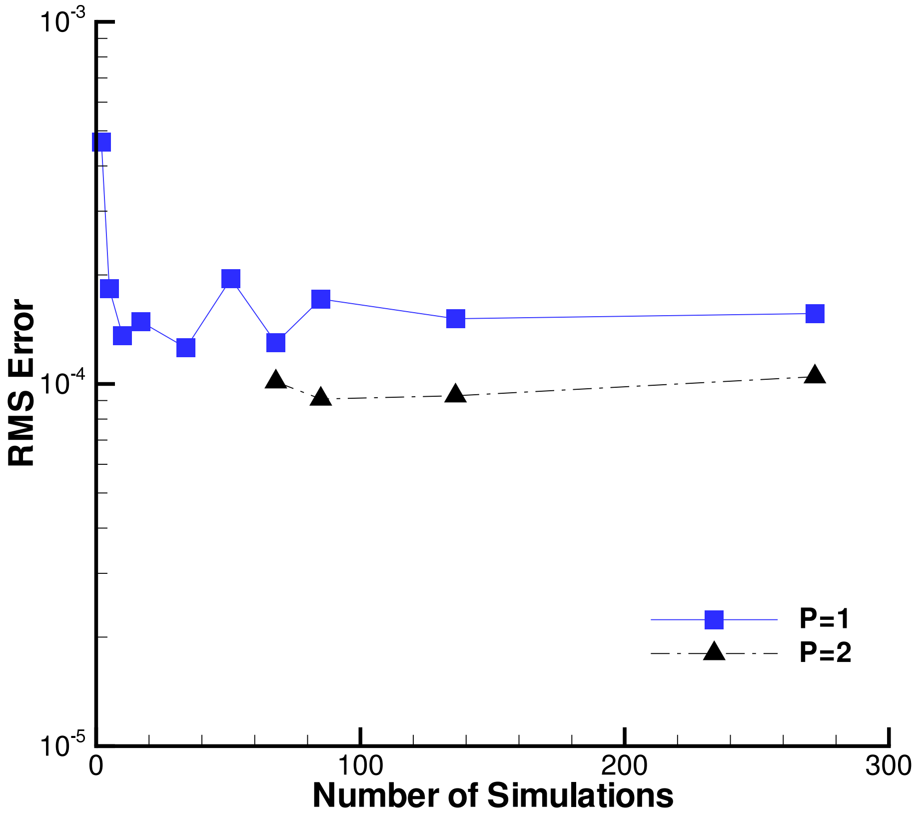


Figure 5.4: Variation of surrogate error for regression model as number of training points increases.

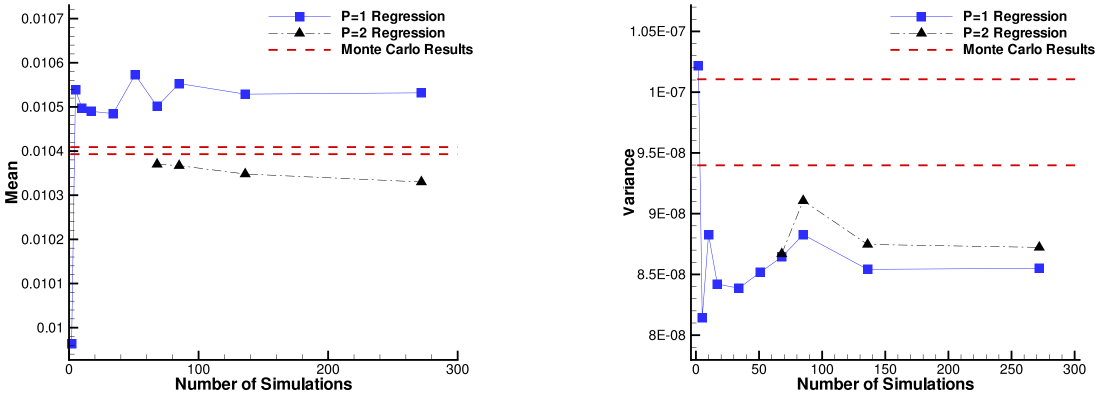


Figure 5.5: Variation of mean (left) and variance (right) predictions for regression model as number of training points increases.

Table 5.4: Statistics based on first (N=10) and second order (N=68) polynomial regression.

Statistic	First Order	Second Order	Monte Carlo-Lower	Monte Carlo-Upper
Mean	1.0497E-002	1.0370E-002	1.0393E-002	1.0409E-002
Variance	8.8273E-008	8.6692E-008	9.3979E-008	1.0106E-007
Standard Deviation	2.9711E-004	2.9444E-004	3.0656E-004	3.1789E-004
95% Confidence Interval	5.6610%	5.6786%	5.8994%	6.1083%

training point. Because of this fact and because over-determining the system by a factor of two is indicated in the literature as a good rule of thumb for regression models [20], 68 training points will be used for the second order regression models moving forward.

Based on these results, the statistic predictions of the first order model and second model are examined. Table 5.4 shows the mean and variance prediction for these two regression models. As the results show, both regression models give more accurate statistic predictions than the linear extrapolation based models. While the linear model gives a slightly improved variance prediction, the second order model produces more accurate predictions in general, showing better agreement for the mean value and reasonable variance predictions.

In addition to statistic predictions, an approximate CDF for both models is constructed and compared to that of Monte Carlo sampling. These CDF curves are plotted in Figure 5.6. As the plot shows, the $P = 1$ regression gives a CDF curve shifted to the right but with a generally correct spread. The $P = 2$ regression on the other hand matches the CDF from Monte Carlo well over the entire curve with slight departures near the upper tail of the distribution.

5.5 Kriging Results

In order to improve on the regression predictions, a Kriging model is used to represent the simulation output. Unless specified otherwise, a gradient-enhanced Kriging model is used throughout this chapter for aleatory uncertainty quantification. The performance of the Kriging model is assessed based on the error of the Kriging approximation and the

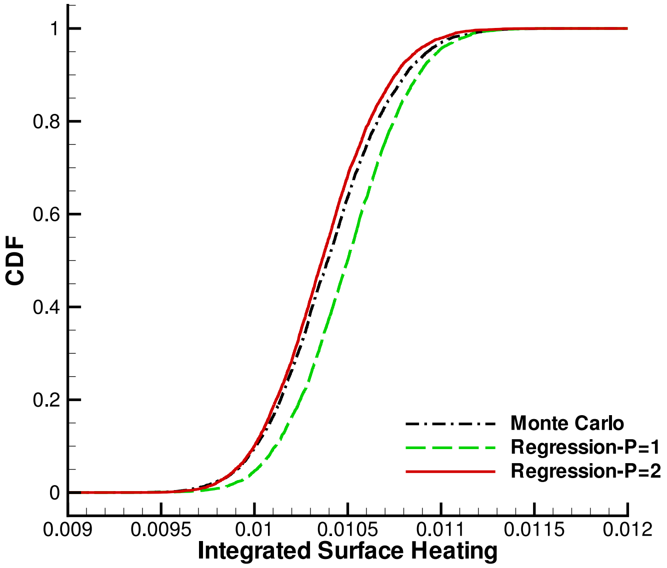


Figure 5.6: CDF prediction based on P=1 (N=10) and P=2 (N=68) regression models compared with Monte Carlo CDF.

statistic predictions. Because the number of training points required for an accurate Kriging model increases rapidly as the dimension increases, the number of variables over which the Kriging model is constructed must be reduced. This reduction is achieved by leveraging the information provided by the sensitivity analysis presented in Chapter 4. For this work, two dimension reduction techniques will be explored. For the first method, the output of the simulation is assumed to be only a function of the most sensitive variables and low importance variables are ignored. For the second method, the regression and Kriging models will be combined. For this method, a low-order regression model is first constructed over all of the input variables. Then, the surrogate is improved by creating a Kriging model based on the difference between the training data and the regression model. This Kriging model is based only on the most sensitive variables. In this way, a low-order model is used for the low importance variables and the Kriging model is used for the most sensitive variables.

5.5.1 Variable Truncation Results

From the global sensitivity analysis presented in Chapter 4, it was shown that the majority of the uncertainty is the result of a relatively small number of variables. Because of this fact, the dependence of the output on the lower importance variables can be neglected, allowing for the creation of a Kriging model with respect to the variables contributing the majority of the uncertainty. In order to demonstrate the performance of this strategy, three properties of the Kriging model are assessed. First, the error encountered in neglecting a number of the input variables is quantified. Second, because the Kriging model must accurately approximate the design space with a limited number of training points, the effect of number of training points on surrogate accuracy is examined. Third, in order to reduce the required number of points, gradient information is incorporated into the training of the Kriging model. The effect that the incorporation of gradients has on the accuracy of the model is also be quantified.

To quantify the effect of neglecting variables, the accuracy of the surrogate is measured. The performance of the model is assessed based on predictions of the average and variance, and the overall error of the surrogate. The error of the surrogate is calculated by sampling the Kriging model at the same points used for the Monte Carlo sampling and measure the difference between the Kriging predictions and the Monte Carlo results. Based on the global sensitivity analysis, variables are chosen such that a specified percentage of the uncertainty is accounted. In Chapter 4, a global sensitivity analysis was performed using both exhaustive Monte Carlo analysis as well as an inexpensive regression-based approach. In practice, the exact global sensitivity analysis based on Monte Carlo is not available, so results will be presented based on the exhaustive Monte Carlo global sensitivity analysis as well as the second-order regression based approach.

For the variable truncation based on the exhaustive Monte Carlo global sensitivity analysis, the Kriging model was created for sets of variables accounting for 80%, 85%, 90% and 95% of the total variance. This necessitated creating Kriging models in 4,6,9 and 15 dimensions. The variables used to create the Kriging model are given in Table 5.5. For the variable truncation method, an ordinary Kriging model is used throughout.

The Kriging model itself was created based on 68 simulation results including function

Table 5.5: Variables for the Kriging model based on Monte Carlo global sensitivity analysis

Rank	Variable	Uncertainty Contribution	Total Contribution
1	ρ_∞	0.60055	0.60055
2	$O_2 + O \leftrightarrows 2O + O$ (f)	1.0610×10^{-1}	0.70665
3	$NO + O \leftrightarrows N + O + O$ (b)	5.1914×10^{-2}	0.75857
4	O2-N2 (k=1)	4.2121×10^{-2}	0.80069
5	N2-N2 (k=1)	3.1617×10^{-2}	0.83231
6	$O_2 + O_2 \leftrightarrows 2O + O_2$ (b)	2.1621×10^{-2}	0.85393
7	$N_2 + O \leftrightarrows NO + N$ (f)	2.0647×10^{-2}	0.87457
8	N2-N2 (k=2)	1.9019×10^{-2}	0.89359
9	O-N2 (k=2)	1.3874×10^{-2}	0.90747
10	$N_2 + O \leftrightarrows NO + N$ (b)	1.2155×10^{-2}	0.91962
11	$NO + O \leftrightarrows N + O + O$ (f)	8.3596×10^{-3}	0.92798
12	NO-N2 (k=1)	7.6676×10^{-3}	0.93565
13	$O_2 + O \leftrightarrows 2O + N$ (b)	7.4280×10^{-3}	0.94308
14	V_∞	4.8401×10^{-3}	0.94792
15	$O_2 + N_2 \leftrightarrows 2O + N_2$ (f)	3.5848×10^{-3}	0.95150

and gradient values. Because gradient values are used in the training of the model, the dimension reduction was also applied to remove the unnecessary derivative values from the training data. Table 5.6 shows the error of the Kriging predictions for models with increasing dimension. Included in this table is the fraction of the variance accounted for by the subset of variables. This fraction is given by the sum of the square of the correlation coefficient for each subset of variables. The effect of increasing the dimension could have mixed effects on the accuracy of a general Kriging model. Although increasing the dimension allows for more variation to be accounted for when predicting function values, the increase in dimension when combined with a fixed number of training points reduces the sampling density, indicating that the quality of the Kriging model may not improve as dimension increases. However, for a gradient-enhanced model, as the dimension increases, more training data is introduced to the model, alleviating the issue of decreasing function density (although not eliminating it). As Table 5.6 shows, the error in the predictions steadily decreases as the dimension of the Kriging model increases to account for more input variables. In order to test the effect of gradients on this dimension expansion, a non-gradient enhanced model was also created using

Table 5.6: Surrogate error for gradient-enhanced Kriging model as dimension increases

Dimension	Variance Fraction	RMS Error
4	8.0069E-001	3.3045E-004
6	8.5393E-001	2.0711E-004
9	9.0747E-001	1.6189E-004
15	9.5150E-001	6.8632E-005

Table 5.7: Surrogate error for function-only Kriging model as dimension increases

Dimension	Variance Fraction	RMS Error
4	8.0069E-001	3.4776-004
6	8.5393E-001	2.5753E-004
9	9.0747E-001	2.6424E-004
15	9.5150E-001	1.3642E-004

only the function values at the same 68 training points. Table 5.7 shows the error resulting for the function-only Kriging model as dimension increases. As the table shows, the error generally decreases as the dimension expands, indicating that the increased flexibility of the model to account for more input variables outweighs the decrease in sampling density for this combination of training points and dimension. However, when compared with the gradient-enhanced Kriging model, the error for the function-only model reduces less significantly as the dimension increases, and the decrease in error is non-smooth, actually increasing for the 9 dimensional case. This fact indicates that the function-only model may be moving into a regime where the amount of training data is insufficient for the design space and that the addition of gradient-information significantly reduces the problem of declining sample density for this test case.

In addition to measuring the error, statistic predictions based on Kriging models of increasing dimension were calculated. These statistic predictions are given in Table 5.8. Because the global sensitivity analysis used to truncate the variables also provides an estimate for the percentage of the variance these variables contribute to the total uncertainty, the Kriging results can also be corrected based on this fraction to predict the total output

Table 5.8: Statistic predictions for gradient-enhanced Kriging model as dimension increases

Dimension	Mean	Variance	Variance Fraction	Corrected Variance
4	1.0408E-002	1.8544E-007	8.0069E-001	2.3160E-007
6	1.0324E-002	8.5423E-008	8.5393E-001	1.0004E-007
9	1.0301E-002	7.9160E-008	9.0747E-001	8.7232E-008
15	1.0416E-002	9.4618E-008	9.5150E-001	9.9441E-008

Monte Carlo Results	Mean	Variance
Mean	1.0401E-002	9.7517E-008
Lower Bound	1.0393E-002	9.3979E-008
Upper Bound	1.0409E-002	1.0106E-007

uncertainty for the full dimension model. This corrected variance is computed by simply dividing the variance predicted by the Kriging model by the fraction of the uncertainty accounted for by the choice of variables. The corrected variance is also given in Table 5.8.

As the table shows, the statistic predictions vary more widely than the surrogate error, which decreases monotonically. While the low dimension case predicts the average accurately, the variance prediction is nearly double the appropriate value. As the dimension increases, while the variance predictions improve, the average prediction begins loosing accuracy. The 15 dimensional case is the only scenario in which both statistics are reasonably predicted. It should be noted that with the exception of the 4 dimensional case, the corrected variance seems to provide an improved variance estimate. Because the Kriging model is built on a subset of variables accounting for a specified fraction of the variance, this corrected value will only provide improved estimates when the output variance based on the Kriging model is underestimated, as is the case for all dimensions above 4.

With the dimension reduction examined, the effect of number of training points is examined. This examination is performed based again on surrogate error and statistic predictions. For this test, gradient-enhanced Kriging models were created using training datasets of increasing sizes. These models were based on the 15 variables identified by the Monte Carlo based global sensitivity analysis as accounting for 95% of the variance, given in Table 5.5. In addition to creating gradient-enhanced Kriging models, function-only models were created to again examine the effect of including gradients as the number of training points increases.

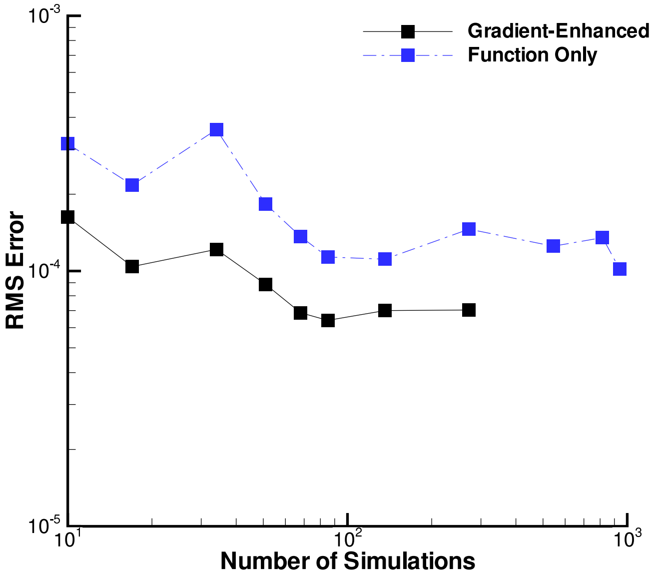


Figure 5.7: Variation of surrogate error as number of training points increases for gradient-enhanced and function-only Kriging models

The error as the number of training points increases is plotted in Figure 5.7 for both the gradient-enhanced and function-only models. As the figure shows, the gradient-enhanced model consistently produces a more accurate surrogate than the function-only model. Additionally, the error for the gradient-enhanced model quickly levels off while the function-only model shows large variation over all but the largest numbers of training points. This variation may indicate that the sampling density for the function-only model may not be sufficient until a large number of points is reached.

Figure 5.8 shows the average and variance predictions as the number of training points increases for both the gradient-enhanced and function only model. As the figure demonstrates, the average predictions for the gradient-enhanced model show less variation and have better agreement with the Monte Carlo results when compared to the function-only results. For the variance, both models show discrepancies with the Monte Carlo results but the predictions for gradient-enhanced model show less variation as the number of training points increases, indicating some level of convergence for the model training.

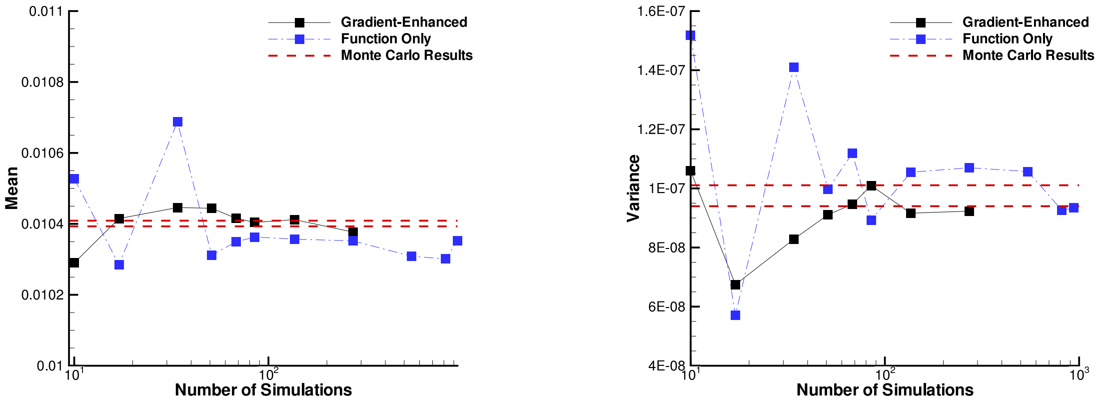


Figure 5.8: Variation of mean (Left) and variance (Right) as number of training points increases for gradient-enhanced and function-only Kriging models

In order to test the variable truncation technique under a more realistic set of circumstances, the correlation coefficients built from the regression-based global sensitivity analysis are used to identify the variables over which to create the Kriging model. The global sensitivity analysis was performed using the same 68 training points used to construct the Kriging model. Based on the function and gradient values for these points, a second order polynomial regression was created and correlation coefficients were computed using this functional representation. Hence, this method only relies on 68 function/gradient evaluations for both the sensitivity analysis and uncertainty quantification. The variables for this test were chosen to account for 80%, 85%, 90% and 93% of the uncertainty. The last percentage was altered because the regression-based global sensitivity analysis could not account for 95% of the uncertainty in a dimension comparable to the Monte Carlo global sensitivity analysis. These uncertainty percentages required Kriging models to be constructed over 5, 7,10 and 17 dimensions. The variables used to construct these models are given in Table 5.9.

Based on the regression-based global sensitivity analysis and variable truncation, the performance of Kriging models using these sets of variables was again assessed using surrogate error as well as statistic prediction. The statistic predictions and surrogate error are given in Table 5.10. As the table shows, the error steadily decreases as the dimension of the surrogate expands, indicating that the regression-based global sensitivity analysis identifies the appropriate variables in which to expand the Kriging model. However, the regression-

Table 5.9: Variables for the Kriging model based on regression global sensitivity analysis

Rank	Variable	Uncertainty Contribution	Total Contribution
1	ρ_∞	0.56879	0.56879
2	$O_2 + O \leftrightarrows 2O + O$ (f)	1.0002×10^{-1}	0.66882
3	$O_2 + O_2 \leftrightarrows 2O + O_2$ (b)	5.7669×10^{-2}	0.72649
4	$NO + O \leftrightarrows N + O + O$ (b)	4.0057×10^{-2}	0.76654
5	N2-N2 (k=1)	3.7461×10^{-2}	0.80400
6	O2-N2 (k=1)	3.3299×10^{-2}	0.83730
7	N2-N2 (k=2)	2.1163×10^{-2}	0.85847
8	O-N2 (k=2)	1.7395×10^{-2}	0.87586
9	V_∞	1.3497×10^{-2}	0.88936
10	$O_2 + O \leftrightarrows 2O + O$ (b)	1.1734×10^{-2}	0.90109
11	NO-N2 (k=1)	9.0220×10^{-3}	0.91011
12	$O_2 + N_2 \leftrightarrows 2O + N_2$ (f)	5.5034×10^{-3}	0.91562
13	$NO + O \leftrightarrows N + O + O$ (f)	5.1194×10^{-3}	0.92074
14	$NO + N_2 \leftrightarrows N + O + N_2$ (b)	2.9103×10^{-3}	0.92365
15	$NO + O \leftrightarrows O_2 + N$	2.6108×10^{-3}	0.92626
16	$NO + N \leftrightarrows N + O + N$ (b)	2.0229×10^{-3}	0.92828
17	$NO + N \leftrightarrows N + O + N$ (f)	1.9061×10^{-3}	0.93019

based global sensitivity analysis fails to reduce the error as significantly as the Monte Carlo based truncation, due to discrepancies between the two sensitivity analyses. As the dimension of the models increase, the statistic predictions again show wide variation as more simulation results are included. However, the predictions resemble those from the models produced using the Monte Carlo sensitivity analysis, and the models in 10 and 17 dimensions give reasonable predictions for both the mean and variance. Additionally, the corrected variance provide good estimates for the full dimensional variance for all but the lowest dimension.

Finally, to examine the prediction of statistics beside the mean and variance based on a Kriging model, the CDF of the output based on the 17 variables identified in Table 5.9 was constructed using 68 training points. This CDF is plot in Figure 5.9 along with the CDF from the Monte Carlo results. The CDF curve produced by the Kriging model matches the CDF from the Monte Carlo results well, falling somewhere between the first and second order regressions in terms of quality of fit. This fact is in spite of the fact that a majority of the input variables have been ignored within the construction of the Kriging model.

Table 5.10: Statistic predictions/surrogate error for gradient-enhanced Kriging model as dimension increases based on regression global sensitivity analysis

Dimension	Mean	Variance	RMS Error	Variance Fraction	Corrected Variance
5	1.0237E-002	9.7286E-008	2.7561E-004	0.8040036369	1.2100E-007
7	1.0309E-002	8.3252E-008	1.8621E-004	0.8584660143	9.6978E-008
10	1.0406E-002	8.7731E-008	1.3462E-004	0.9010918226	9.7361E-008
17	1.0446E-002	1.0227E-007	9.8999E-005	0.9301866679	1.0994E-007

Monte Carlo Results	Mean	Variance
Mean	1.0401E-002	9.7517E-008
Lower Bound	1.0393E-002	9.3979E-008
Upper Bound	1.0409E-002	1.0106E-007

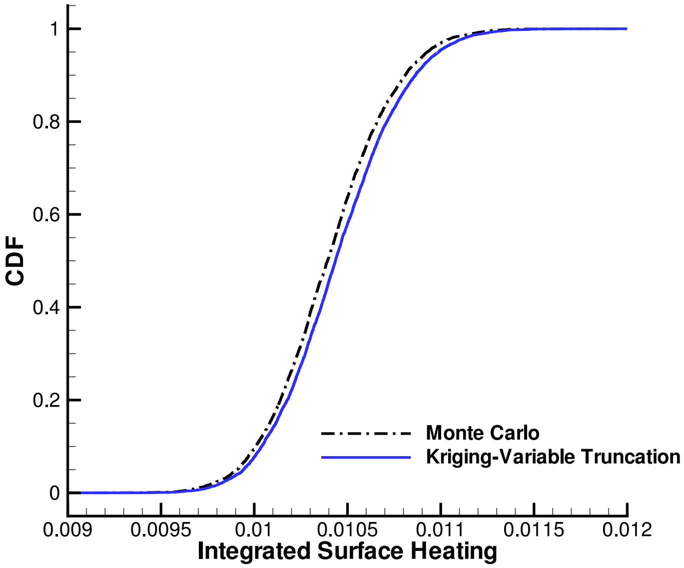


Figure 5.9: CDF prediction for Kriging model based on 68 training points in 17 dimensions compared with Monte Carlo CDF.

5.5.2 Regression/Kriging results

Instead of merely discarding the variables identified as ”low importance”, these variables can be accounted for in a low order manner, reserving the Kriging model for only the most important variables. This low order model is incorporated into the Kriging model by using an explicitly defined mean function. For an explicitly defined mean function, the output predictions are assumed to be the sum of a zero mean Gaussian process and the mean function.

$$y(x) = m(x) + N(0, K(x, x')) \tag{5.31}$$

Hence, the Kriging model is created based on the difference between the measured training data and the prediction of the training data from the mean function. The explicit mean function used here is the second-order regression used in Chapter 4 for the global sensitivity analysis. This regression is gradient-enhanced and includes all of the input variables in the code. Using this mean function, the residual between the actual function and gradient values with the regression-predicted values is used to construct the surrogate model.

$$\begin{bmatrix} R(x) \\ \nabla R(x) \end{bmatrix} = \begin{bmatrix} y(x) - m(x) \\ \nabla y(x) - \nabla m(x) \end{bmatrix} = N(0, \underline{K}(x, x')) \tag{5.32}$$

To test the ability of this method to account for the low-importance variables, the dimension was again increased based on the global sensitivity analysis and the surrogate error and statistic predictions were measured. The variables for the Kriging model are identified in Table 5.9 and the models again range from 5 to 17 dimensions. It should be noted that the choices of which parameters to treat as important were based on the regression-based global sensitivity analysis. This global sensitivity analysis uses the same mean function that is used to enhance the Kriging model. Both the sensitivity analysis and Kriging model are built from the same 68 function/gradient evaluations and the regression is second order. In Table 5.11, the statistic predictions and surrogate error for gradient-enhanced regression/Kriging models are given as the dimension is increased. For comparison, the results of Table 5.10 are repeated. As the results demonstrate, the surrogate error for the

Table 5.11: Statistic predictions/surrogate error for gradient-enhanced Kriging model combined with regression mean function

Dimension	Mean -Reg	Mean Trunc	Variance Reg.	Variance Trunc.
5	1.0361E-002	1.0237E-002	8.9684E-008	9.7286E-008
7	1.0396E-002	1.0309E-002	9.5146E-008	8.3252E-008
10	1.0412E-002	1.0406E-002	9.3201E-008	8.7731E-008
17	1.0384E-002	1.0446E-002	9.2394E-008	1.0227E-007

Dimension	Error Reg.	Error Trunc.
5	1.1250E-004	2.7561E-004
7	9.6692E-005	1.8621E-004
10	9.4842E-005	1.3462E-004
17	9.3034E-005	9.8999E-005

Monte Carlo Results	Mean	Variance
Mean	1.0401E-002	9.7517E-008
Lower Bound	1.0393E-002	9.3979E-008
Upper Bound	1.0409E-002	1.0106E-007

combined regression/Kriging model is consistently lower than the corresponding model based on variable truncation. This decrease in error is more pronounced for the lower dimensions and disappears once the highest dimension is reached. As a result of this decreased error, the statistic predictions for the lower dimensions, in particular the mean predictions, are improved by incorporating the regression model.

In addition to examining individual statistic predictions, a CDF curve based on the prediction of the 17 dimensional Kriging model using the regression mean function was created. This CDF is plotted in Figure 5.10. As the plot demonstrates, the CDF produced by the combined Kriging/regression model matches the Monte Carlo results incredibly well, producing the best fit of all the methods. It should be noted that the $P = 2$ regression that is enhanced by the Kriging model already produced a relatively accurate CDF curve; however, the Kriging/regression approach is able to improve the predicted CDF toward the upper tail of the distribution.

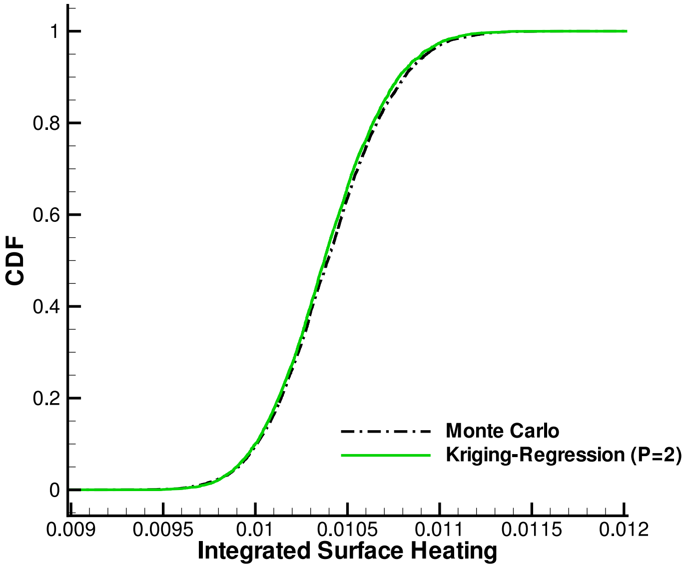


Figure 5.10: CDF prediction for $P = 2$ regression enhanced with 17 dimensional Kriging model using 68 function/gradient evaluations.

5.6 Method Comparison

In order to directly compare the various methods presented in this section, the statistic predictions from each method are summarized. In addition to comparing the statistic predictions, the number of simulations required for each statistic prediction is also compared, such that possible trade-offs between accuracy and expense can be judged. Tables 5.12 and 5.13 contain a summary of the results from the various strategies employed in this chapter and compares these to the Monte Carlo results. For the regression and Kriging methods, a single optimal result is used for each. A description for each of the methods compared in Tables 5.12 and 5.13 is given below:

- Moment Method: First order moment method using a single function/gradient evaluated at mean.
- Linear Extrapolation: Inexpensive Monte Carlo based on linear extrapolation about mean value.

Table 5.12: Summary of statistic predictions for gradient-based uncertainty quantification strategies

Method	Mean	Variance	95% CI	F/G Cost
Moment Method	1.0370E-002	1.3790E-007	7.1616%	1
Linear Extrapolation	1.0369E-002	1.3412E-007	7.0638%	1
P=1 Regression	1.0497E-002	8.8273E-008	5.6610%	10
P=2 Regression	1.0370E-002	8.6692E-008	5.6786%	68
Kriging-Var. Trunc.-17D	1.0446E-002	1.0227E-007	6.1228%	68
Kriging-Regression-17D	1.0384E-002	9.2394E-008	5.8543%	68
Monte Carlo	**1.0401E-002**	**9.7517E-008**	**6.0049 %**	**6331 (f only)**

- P=1 Regression: Inexpensive Monte Carlo based on first order polynomial regression using 10 function/gradient evaluations.

- P=2 Regression: Inexpensive Monte Carlo based on second order polynomial regression using 68 function/gradient evaluations.

- Kriging-Variable Truncation-17D: Sampling based on Gradient-enhanced Kriging model of top 17 parameters using 68 function/gradient evaluations. Important parameters identified using global sensitivity analysis based on P=2 polynomial regression.

- Kriging-Regression-17D: Sampling based on $P = 2$ regression augmented with Gradient-enhanced Kriging model over the top 17 parameters using 68 function/gradient evaluations. Important parameters again identified using global sensitivity analysis based on P=2 polynomial regression.

As the table shows, all the methods examined for this work produce reasonable approximations for the mean, matching the Monte Carlo results to within 1% of the exact value. For the variance, the Kriging models produce the best estimates followed by the regression models. The linear methods produce the worst predictions, significantly over estimating the variance. Given the low computational cost associated with these linear methods, these methods may be sufficient for optimization under uncertainty methods, where the uncertainty quantification is merely a step within a broader optimization; however, even for these

Table 5.13: Error in statistic prediction for gradient-based uncertainty quantification strategies

	Relative Error of Average	Relative Error of Variance
Moment Method	0.29%	41.41%
Linear Extrapolation	0.30%	37.53%
P=1 Regression	0.92%	9.48%
P=2 Regression	0.29%	11.10%
Kriging-Variable Truncation-17D	0.44%	4.87%
Kriging-Regression-17D	0.16%	5.25%

applications, a $P = 1$ gradient-enhanced regression model may be preferable as the cost can be varied to match the computational budget, with increased function/gradient evaluations improving the variance predictions. For accurate variance predictions, the Kriging models are the clear choice, producing variance predictions just outside the error bounds of the Monte Carlo result. Between the variable truncation method or combined Kriging-Regression method, the optimal strategy is unclear from these results with each method giving a superior estimate for one of the statistics. In this case, the CDF curves should also be compared to gauge the prediction of distribution quantiles given by each method. This comparison is most easily made using a quantile-quantile (QQ) plot where the quantiles predicted by each of the Kriging methods are plotted against the quantiles of the Monte Carlo results. The closer this plot approaches a line of slope 1, the better the agreement between the Kriging predictions and the Monte Carlo results.

As Figure 5.11 shows, although both method perform reasonably well, the Kriging-regression method gives predictions most in agreement with the Monte Carlo results. From a cost point of view, the Kriging-regression approach requires the same number of function/gradient evaluations, making the increased accuracy well worth the extra complexity required for Kriging. Comparing the Kriging-regression approach to the variable truncation approach, the Kriging-regression approach does have a lower limit for the number of function/gradient evaluations required to construct the model as a regression must first be constructed. For situations where this lower limit is unacceptably high, variable truncation

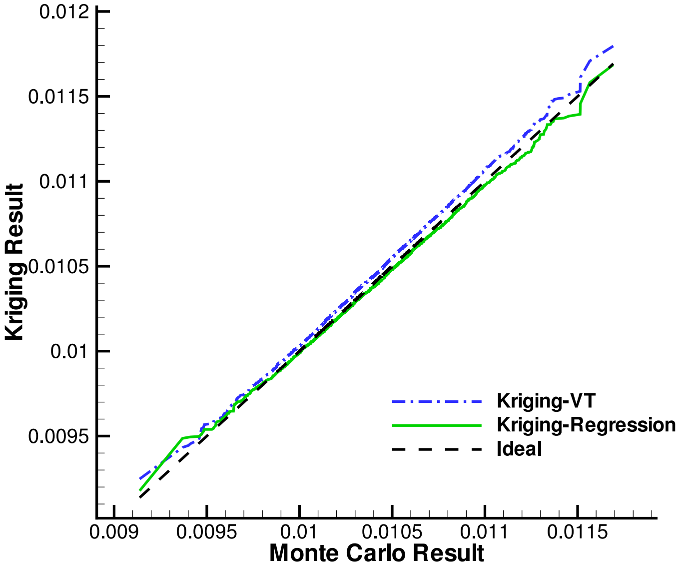


Figure 5.11: QQ Plot for Variable truncation Kriging and regression/Kriging models with the Monte Carlo results. Line with Slope 1 indicates perfect agreement with Monte Carlo.

method may be preferable; however, given the results presented in Figure 5.7, it is likely that the lower limit required for regression may be a good rule of thumb for achieving an accurate Kriging model even when no regression model is used.

In Table 5.12, the cost of the methods is compared in terms of function and gradient evaluations. Comparing on this basis allows more general conclusions to be drawn from these results. When comparing these methods for a specific application, the dimension of the uncertainty problem as well as the cost of the adjoint solution relative to the primal solution. The scaling of these methods as the dimension increases has been examined within the main body of this chapter. For the linear methods requiring a single flow and adjoint solution, the total computational cost was equal to 2.45 cpu-hours with the adjoint solution solving approximately 40 times faster than the flow solution. For the Monte Carlo sampling, 6331 function evaluations were required. Because multiple flow solutions were required for this sampling, the flow solution using mean parameter values can be used as the starting point for additional flow solutions using off-mean parameters. Using this mean solution as

Table 5.14: Cost comparison for gradient-based uncertainty quantification strategies

	Simulation results	Cost per simulation (hrs)	Total Cost (hrs)	Ratio to MC cost
Moment Method	1	2.45	2.45	2077
Linear Extrapolation	1	2.45	2.45	2077
P=1 Regression	1	0.861	11.01	462
P=2 Regression	68	0.861	60.97	83
Kriging-Variable Truncation-17D	68	0.861	60.97	83
Kriging-Regression-17D	68	0.861	60.97	83
Monte Carlo Sampling	**6331**	**0.804**	**5090**	**1**

the starting point eliminated a portion of the start-up transient for the other flow solutions, reducing the time required for a fully converged solution from 2.39 to 0.804 hours on average. Hence, the total cost for the Monte Carlo sampling was 5090 cpu-hours. To acquire the function/gradient values used to train the regression and Kriging models, multiple flow and adjoint solutions were required. Utilizing the same start-up procedure used for the Monte Carlo sampling, the average time for a flow solution was again 0.804 hours and the cost of an adjoint solution remained unchanged at 0.058 hours. Based on these facts, the ratio of flow solution to adjoint is reduced to 14 when multiple flow and adjoint solutions are computed. Including to cost of acquiring the initial flow solution, the cost associated with acquiring the 10 and 68 function/gradient values used in the regression and Kriging models is 11.01 cpu-hours and 60.97 cpu-hours respectively. Neglected in these costs is the cost associated with building the regression or Kriging model. For this case (and for most complex simulations), the cost associated with building these models is insignificant compared with the cost associated with generating the training data. The cost comparison is summarized in Table 5.14. Based on these computational costs, the cost savings for these methods range from over a factor of 2000 for the linear methods to a factor of 83 for the Kriging model and second order regression. Depending on the desired level of accuracy, it is clear that the gradient-based methods examined in this chapter show great potential in reducing the cost associated with aleatory uncertainty quantification.

5.7 Summary

In this chapter, gradient-based strategies for aleatory uncertainty quantification were presented. At the heart of these strategies are a surrogate models that can be built from a small number of simulation results and can inexpensively approximate the output of a simulation, serving as a basis for inexpensive Monte Carlo sampling. These surrogates ranged in complexity from simple linear extrapolation to polynomial regression and Kriging models. To reduce the cost associated with the construction of these surrogate models as the number of input parameters expanded, gradient values were incorporated into the training of the surrogate, providing additional information for a reduced cost relative to a full simulation result. These strategies were tested using the $5km/s$ flow test case and the output uncertainty resulting from 66 uncertain model parameters was quantified. Compared with the results from traditional sampling methods, the gradient-based strategies presented in this chapter were able to accurately approximate the statistics of the output using a fraction of the flow results required for Monte Carlo sampling.

Chapter 6

Epistemic Uncertainty Quantification Results

In this chapter, gradient-based approaches for epistemic uncertainty quantification are presented. Because of the expense associated with traditional methods of episetmic uncertainty, a subset of the input variables used in Chapter 5 is chosen and treated as epistemic. Using this variable subset, the output interval for a simulation is predicted using a linear surrogate and an optimization approach. The performance of these methods is measured by comparing against the bounds produced by exhaustive sampling.

6.1 Gradient-based Epistemic Uncertainty Quantification

For epistemic uncertainty, the goal is to determine the interval of the output given intervals for the input parameters. As was the case with aleatory uncertainty, this output interval can be approximated by assuming a functional form of the output over the design space. Using a linear functional representation, the output interval width can be approximated using a moment-method type formula, given as:

$$y_o = f(x_o) \tag{6.1}$$

$$\Delta_y = \sum_{i=1}^{d} \left| \frac{\partial f}{\partial x_i} \Delta_{x_j} \right| \tag{6.2}$$

where x_o represents the design variables at the center of the input interval and y_o represents the center of the output interval. Because epistemic uncertainties are often specified using a plus/minus, x_o and y_o can be thought of as the unperturbed values. Although these values are analogous to mean values for the aleatory case, the use of mean implies an associated probability distribution function so the term unperturbed will be used throughout this section.

For cases in which the linear approximation is insufficient, the determination of the output interval can be recast as a bound-constrained optimization problem. The optimization problem is specified as: given a set of intervals for the input parameters, determine the minimum and maximum possible output values. This problem statement gives two optimization problems, given in equation (6.3).

$$y_{min} = \min_{x \in I} f(x) \tag{6.3}$$

$$y_{max} = \max_{x \in I} f(x) \tag{6.4}$$

Here, the input intervals are given by the space I and the minimum and maximum values are determined based on inputs in this space. The process by which these minimum and maximum values are determined can vary widely in expense and complexity. The most straight-forward method for determining the extrema is by exhaustive sampling of the function. Typically, uniform Latin hypercube sampling is performed using a fixed number of samples, usually 3, for each variable. For this method, the required number of samples grows exponentially fast as the number of input parameters is increased, with the formula given as $N = 3^d$. For complex simulations, this exhaustive sampling is prohibitively expensive. As was the case with aleatory uncertainty, the simulation output can once again be replaced with a gradient-enhanced surrogate model and the Latin hypercube sampling can be performed on this surrogate. As was discussed previously, the expense associated with these surrogate models increases dramatically with dimension; hence, for large numbers of epistemic variables, the expense of a surrogate based approach may become prohibitively large.

In addition to exhaustive sampling approaches, more sophisticated optimization methods can be applied directly to the problems posed in equation (6.3). In particular, this work will focus on gradient-based methods for optimization as these method can scale to large input dimension without an exponential increase in cost. The optimization method used for this work is the limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm [90]. The L-BFGS algorithm is a quasi-Newton solver, meaning that only the function and gradient at each iteration is required and an approximate Hessian is constructed based on the convergence history of the solver. Using this approximate Hessian, Newton's method is used to drive the gradient to zero. For a BFGS method, the design space is approximated at iteration k as [85]:

$$m_k(p) = y_k + \nabla y_k^T p + \frac{1}{2} p^T B_k p \tag{6.5}$$

where the subscript k indicates current iteration values, B_k is the approximate Hessian and p represents the search direction. With this functional form, the direction of the minimum location is given by:

$$p_k = -B_k \nabla y_k \tag{6.6}$$

With the search direction determined, a line search is performed in this direction to determine a step size that gives a sufficient decrease in the function. The approximate Hessian is updated at each iteration based on the requirement that the new approximate Hessian B_{k+1} must accurately predict the difference between the current gradient value ∇y_{k+1} and the previous gradient ∇y_k when used in a Taylor series approximation. The exact details of the construction of this approximate Hessian can be found in Reference [85]. For the limited-memory BFGS algorithm, this approximate Hessian is never explicitly constructed and the effect of the approximate Hessian is built up using only a limited number of previous gradient values. The limited storage requirements of this method make it particularly suited for high dimensional problems.

The main drawback of gradient-based optimization is that the methods are inherently localized in nature. For epistemic uncertainty propagation, the global maximum and minimum are required. For this work, the localized nature of gradient-based optimization did not prove to be a constraint as the optimization results were in agreement with the results of exhaustive sampling, often located at the bounds of the design space. For problems where gradient-based optimization is insufficient, a global optimization method must be employed. Popular global optimization methods include genetic algorithms, pattern searches, Monte-Carlo sampling and swarm-based algorithms. Global optimization methods typically require a large number of function evaluations. In light of this drawback, efficient global optimization is an area of active research. One such efficient global optimization technique often used in CFD is based on the previously outlined Kriging model [29, 31, 33]. Although no results will be shown using this method, the method will be outlined due to its close relation to the other techniques outlined in this chapter.

The process of Kriging-based global optimization is given by the following steps. This process is outlined only for minimization as the case of maximization can be found by reversing the sign on the function. First, a handful of function evaluations are performed and a Kriging surface is constructed based on these results. Second, based on the Kriging surface, promising candidate locations for the minimum value are determined. Finally, additional function evaluations are performed at these candidate locations and a new Kriging surface is constructed with these new function values added to the surrogate training data. This process is repeated until a termination criteria is met, typically a relative convergence, gradient norm or maximum function evaluation criteria.

An effective way of determining promising candidate locations is based on the underlying Gaussian process representation of the Kriging surface. Because the Kriging surrogate is stochastic, function predictions have an associated variance. Hence, the surrogate predictions have an associated uncertainty representing the accuracy of the surrogate at a particular location. For optimization applications, this variance, along with the mean predictions, can be used to determine the location in the design space with the highest probability of containing a new minimum value. This criteria is known as the expected improvement (EI) method and is given by the following formula [29]:

$$EI(x) = \begin{cases} (y_{min} - y^*(x))\Phi\left(\frac{y_{min}-y^*(x)}{s(x)}\right) + s_x\phi\left(\frac{y_{min}-y^*(x)}{s(x)}\right) & \text{if } s(x) > 0, \\ 0 & \text{if } s(x) = 0 \end{cases} \tag{6.7}$$

where y_{min} is the minimum value from the previous optimization iteration, $y^*(x)$ is the mean prediction from the Kriging model, $s(x)$ is the Kriging standard deviation prediction, Φ is the normal cumulative distribution function and ϕ is the normal probability density function. As the formula shows, the EI criteria contains two terms. The first term is proportional to the mean Kriging prediction and grows when the mean prediction is less than the current minimum value. The second term is proportional to the variance of the Kriging prediction and grows for x values away from existing sample points. Hence, the EI criteria maintains a balance between a local search, where points around the minimum of the surrogate model are added, and a global search, where points are added in unexplored areas of the design space. The candidate locations are the points in the design space where the expected improvement function is maximized. Because this maximization is performed on the Kriging model, any global optimization technique can be used. In previous works, genetic algorithms have been used to determine the location with highest expected improvement [29]. Obviously, because this optimization technique is dependent on the construction of a Kriging model, gradient information can easily be incorporated into the training of the surrogate, improving the accuracy of the Kriging surface [33].

6.2 Problem Statement and Validation Results

In order to test the proposed epistemic uncertainty quantification strategies, the output interval for integrated surface heating for the $5km/s$ flow is determined. For convenience, this objective is repeated here:

$$L = -\frac{\int_{\partial\Omega} \left(k \nabla T \cdot \vec{n} + k_v \nabla T_v \cdot \vec{n}\right) dA}{\frac{1}{2}\rho_\infty V_\infty^3} \tag{6.8}$$

where T is the translational-rotational temperature, k is the translational-rotational thermal conductivity, T_v is the vibrational temperature, and k_v is the vibrational thermal conductivity.

The tradition method for quantifying epistemic uncertainty is Latin Hypercube sampling. This sampling is performed using a fixed number of samples (~ 3) in each direction (with a total number of samples increasing as 3^d with d representing the number of epistemic variables). Due to the expense of sampling approaches, the dimension for this problem is reduced relative to the aleatory uncertainty problem presented in Chapter 5. For the purposes of validation, the set of 66 variables used for the aleatory uncertainty problem is reduced to a set of 8 variables for the epistemic problem. Limiting the validation to 8 variables allowed the output interval to be computed in a cost equivalent to that of an aleatory problem. Using 3 sample points in each direction, as indicated in Reference [15], the choice of 8 epistemic variables requires $6,561$ simulation results to provide the appropriate interval, a cost similar to the $6,331$ results used to validate the aleatory results. The choice of which 8 parameters to include in the epistemic study was motivated by both theoretical and practical concerns. Of the 66 parameters used in the sensitivity analysis and aleatory uncertainty quantification, the parameters corresponding to the collision integrals and chemical kinetics model are most accurately described by epistemic uncertainty due to their experimental nature and the lack of information in the literature regarding the distribution of these values.

In addition to this insight, the set of variables was further reduced based on the solver robustness. Because of the large uncertainty in the chemical kinetics model, the convergence of the solver will often stall for certain combinations of reaction parameters far from their mean values. For the Monte Carlo sampling used to validate the aleatory uncertainty, the parameter combinations far from the mean occur rarely due to the assumed Gaussian nature of the uncertainty. When this assumption is relaxed for the case of epistemic uncertainty, these reaction combinations occur frequently and disrupt the sampling process. This problem is exasperated by the nature of the design space associated with the $5 km/s$ simulation. For

Table 6.1: Epistemic Model Parameters

Variable	Unperturbed Value	Lower Bound	Upper Bound
A^1_{N2-N2}, A^2_{N2-N2}	1	0.8	1.2
A^1_{N2-N}, A^2_{N2-N}	1	0.8	1.2
A^1_{N2-O}, A^2_{N2-O}	1	0.8	1.2
A^1_{N2-O2}, A^2_{N2-O2}	1	0.8	1.2

this space, the minimum and maximum values typically occur at or near the bounds of the space. Because this region is the area with the least solver robustness in terms of the reaction parameters, the interval predictions are drastically affected by solver failure. Although solver failure (by means of non-convergence, not floating point exception) did occur in some cases for the aleatory sampling, these failures likely did not significantly effect the statistic predictions as their occurrence was infrequent and the contribution to output statistics by a single point is small for Monte Carlo sampling. Because of the issue of solver robustness to large perturbations of the reaction parameters, the set of variables examined for the epistemic validation problem was limited to transport parameters exclusively.

With the possible set of epistemic variables limited to the 30 transport parameters, the final set of 8 parameters was chosen based on the sensitivity analysis presented in Chapter 4. From this analysis, the collisions involving diatomic nitrogen have the greatest effect on surface heating. The final set of 8 variables are chosen as the collision integrals of diatomic nitrogen with itself and the other predominant species near the wall, namely O_2, O, and N, for k equal to 1 and 2. These variables are given in Table 6.1 with their associated intervals. The assumption of ±20% uncertainty for the collision integrals is taken from Reference [12], although that work assumed a Gaussian for the uncertainty while an interval is used in this work. It should be noted that the parameters chosen for the epistemic problem do not correspond exactly with the 8 most important collision integrals from the sensitivity analysis, although that criteria would indicate 6 of the same parameters. Although the sensitivity analysis played a role in selecting the parameters used for the epistemic problem, the variable set was modified slightly to more closely match the experiences of Reference [13]. Had similar studies not been available, the sensitivity analysis could have been used exclusively.

The output interval due to variations in these 8 parameters was determined based on Latin Hypercube sampling using 3 samples in each dimension ($6,561$ total samples). For this sampling, variations were only allowed for these 8 variables and the other variables were frozen at their unperturbed or mean values. With the samples determined, the minimum and maximum objective values from the data set are determined. Because no distribution can be associated with epistemic uncertainty, this output interval is the only useful result of the sampling, as the epistemic nature of this problem prevents inferring any other properties. The choice of using 3 samples in each dimension was made based on Reference [15]. As will be shown in the next chapter, this choice may lead to underestimation of the output interval. Hence, when comparing the results of other methods to the interval predicted by sampling, an interval prediction larger than the sampling result is preferred, as the sampling result is limited by the number of samples used and is used to guard against the possibility of the optimization falling into a local optimum. Using these validation results, the performance of the gradient-based methods can be measured.

In addition to this 8 dimensional problem, the gradient-based methods are used to predict the output interval due to uncertainty in all 30 transport parameters. Although interval predictions are made, no validation of the interval based on sampling is possible due to the prohibitively high cost associated with sampling. Using 3 samples in each direction for 30 dimensions would require over 100 trillion samples. Instead of comparing the results of the gradient-based method to a sampling based approach, the methods can only be compared with one another. The interval for all 30 transport parameters was $[0.8, 1.2]$ and again corresponds to ±20% uncertainty [12].

6.3 Linear Results

When gradient values are available, the width of the output interval can be predicted based on a single function/gradient evaluation at the unperturbed parameter values. The half width of the interval is predicted using the gradient values (according to equation (6.1)) and the lower and upper bounds of the output interval are predicted by subtracting and adding this half width to the function prediction of the simulation. The results of this linear approximation for the 8 dimensional case are given in Table 6.2 and compared to the interval produced by Latin Hypercube sampling.

Table 6.2: Interval prediction from moment method compared with LHS result

	Linear Method	LHS interval
Center	1.0370E-002	1.0449E-002
Interval Half Width	8.6634E-004	7.1266E-004
Upper	1.1237E-002	1.1161E-002
Lower	9.5040E-003	9.7361E-003
Percentage	±8.35%	±6.82%

As this table demonstrates, the linear method gives a conservative estimate for both the lower bound and the upper bound of the interval. Also included in Table 6.2 is the center of the interval predicted by the LHS sampling and linear method. As the table shows, the center of the interval from sampling deviates significantly from the function value evaluated using parameters at the center of the interval. This fact gives an indication of the non-linearity of the simulation result. Additionally, it demonstrates the difficulty in determining a representative result for the simulation value. While the center of the interval prediction may be an intuitive representative result, any value inside the interval bounds is equally valid. Despite this difficulty, the center of the interval is used as a representative value so the uncertainty can be represented as a plus/minus value about this value. When this percentage error about the center is computed, it is clear that the linear method produces a significantly wider output uncertainty when compared to the sampling results.

Because a single function/gradient evaluation is used to predict the output interval, this method can be applied to the 30 dimensional problem without increasing the cost of the prediction. Although no validation results are presented for this case, the 30 dimensional interval prediction is given in Table 6.3. As expected, the size of the output interval, relative to the 8 dimension case, is larger as a result of adding more uncertain parameters. Although no validation of this result is possible, it will be compared to the corresponding optimization result in the next section.

Table 6.3: Interval prediction using moment method for 30 dimensional case.

	Linear Method
Center	1.0370E-002
Interval Half Width	1.1787E-003
Upper	1.1549E-002
Lower	9.1916E-003
Percentage	11.37%

6.4 Optimization results

In order to provide an inexpensive interval prediction even as the problem dimension increases, gradient-based optimization is used to determine the minimum and maximum values possible given specified intervals on the inputs. Specifically, the L-BFGS algorithm is used for the optimization. The optimization approach is first tested for the 8 dimensional test case. The convergence for the minimum and maximum optimization results is shown in Figure 6.1 in terms of function/gradient evaluations. Plotted with the convergence are the minimum and maximum values from the LHS results. As the figure demonstrates, the interval produced by optimization is larger than the sampling result, producing conservative estimates for the minimum and maximum values.

Because the optimization result produces interval bounds beyond the values produced by sampling, the optimization results should be viewed as the correct result. Because the results of the optimizations represent actual function values achieved with inputs contained in the specified intervals, the bounds produced by optimization are the correct solution to equations (6.3) provided the minimum and maximum values are the global extrema. Hence, the sampling results are used to ensure the optimization has not fallen into any local extrema. Although the sampling does not ensure that the optimization gives the global minimum and maximum values, the sampling results provide no evidence to undermine the optimization results. Even though L-BFGS is an inherently local optimization, in this case it appears to produce the global minimum and maximum values. In addition to producing the more accurate interval estimate, the optimization produces it at significantly reduced cost, requiring only 43 function/gradient evaluations. The rapid convergence of these optimization problems is likely due to the relatively smooth behavior of the design space and the fact that the optimal values typically occur at the bounds for most variables. Although a smooth design space is typical for most outputs of engineering calculations [19, 34], it is likely more complex simulations will require more work to solve the required optimization problems.

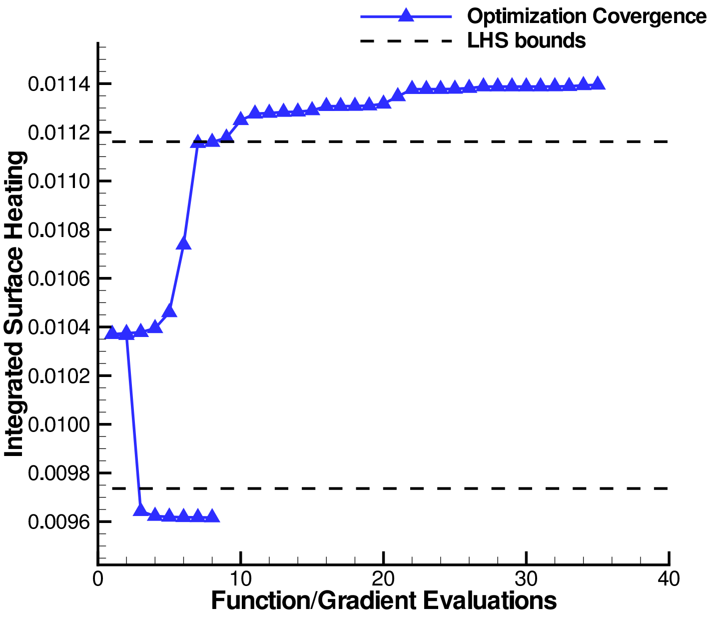


Figure 6.1: Convergence of optimization over epistemic variables for fixed aleatory variables compared with bounds from Latin hypercube sampling.

In order to further compare the optimization to the other results, its bounds are compared with sampling and the linear method in Table 6.4. In addition to comparing the bounds, the center of the interval as well as a symmetric uncertainty percentage are given. Based on these results, the optimization and linear method produce similar measures of uncertainty as a percentage of the center of the interval. Despite this fact, the bounds produced by the linear method show disagreement with the optimization result, with the linear method underestimating the maximum and producing an overly conservative minimum bound.

Because the gradient-based optimization can scale to larger dimension, the bounds for the 30 dimensional problem can be determined. These bounds are compared with the results from the linear method. Although the agreement of these two methods does not imply that

Table 6.4: Comparison of Interval prediction for Optimization, Sampling and Linear Method

	Linear Method	LHS interval	Optimization
Center	1.0370E-002	1.0449E-002	1.0506E-002
Interval Half Width	8.6634E-004	7.1266E-004	8.8912E-004
Upper	1.1237E-002	1.1161E-002	1.1395E-002
Lower	9.5040E-003	9.7361E-003	9.6168E-003
Percentage	8.35%	6.82%	8.46%

the bound is necessarily correct, the fact that the predictions are based on differing methods should give an indication of the quality of the result. Because the optimization result is based on actual function values, instead of extrapolation, the optimization result is likely the more correct result. The convergence of the 30 dimensional optimization problems is given in Figure 6.2. Using 83 function/gradient evaluations, the optimization method is able to produce the output interval for the 30 dimensional problem. Despite the large increase in the dimension of the problem, the cost of the optimization only increased by approximately a factor of two. This fact is likely because the optimal values are located at the bounds of the domain for most variables, allowing the optimizer to easily find the optimum. Additionally, because the sensitivity of many collision integrals is small, the addition of these variables to the optimization does not drastically effect the overall design space. The interval produced by optimization on the 30 dimensional problem is compared to the linear method in Table 6.5. As the table shows, the linear method again underestimates the maximum value but is overly conservative for the minimum value. Although the width of the interval and center of the interval do not agree with the optimization, the uncertainty expressed as a percentage of the center value agree remarkably well for the two methods, as was the case for the 8 dimensional case. Given the difference in bound predictions, this fact is likely coincidental.

6.5 Summary

In this chapter, the application of a linear surrogate and gradient-based optimization to the problem of episetmic uncertainty quantification was outlined. For a rapid quantification

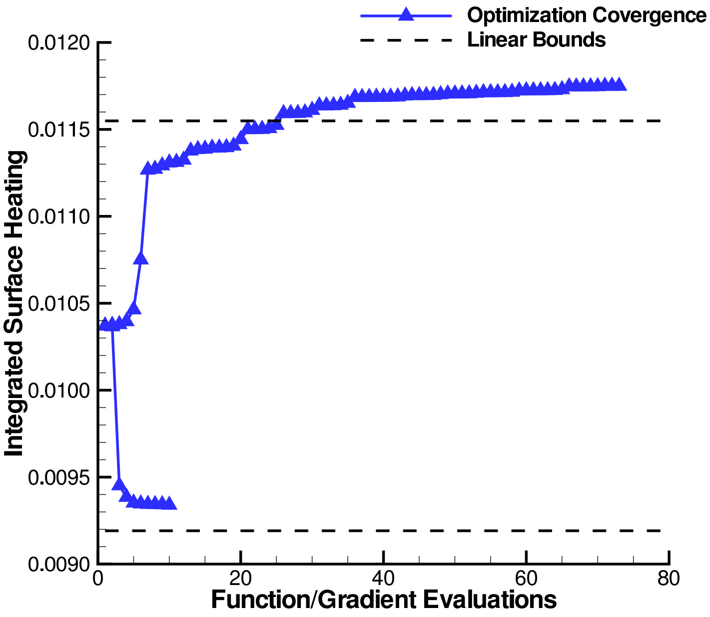


Figure 6.2: Convergence of optimization over epistemic variables for fixed aleatory variables.

Table 6.5: Interval prediction for optimization method for 30 dimensional problem.

	Linear Method	Optimization
Center	1.0370E-002	1.0543E-002
Interval Half Width	1.1787E-003	1.2031E-003
Upper	1.1549E-002	1.1746E-002
Lower	9.1916E-003	9.3400E-003
Percentage	11.37%	11.41%

of the output interval due to epistemic sources, a linear method similar to the moment method was presented. To improve on this method, a gradient-based method using the L-BFGS algorithm was given. Comparing the results against Latin Hypercube sampling, both methods produced reasonable estimates of the output interval, with the optimization providing a more accurate interval prediction than even the sampling result. Additionally, the gradient-based methods were able to produce interval estimates for cases where sampling is prohibitively expensive.

Chapter 7

Mixed Results

With gradient-based optimization demonstrated as a viable means of epistemic uncertainty, the problem of mixed aleatory/epistemic uncertainty can be addressed. This form of uncertainty is quantified using a combined surrogate-optimization approach. Two combined approaches are explored in this work, a statistics-of-intervals approach and an uncertain optimization approach. Due to the expense associated with traditional methods for mixed uncertainty, the Fay-Riddell stagnation heating correlation is used to validate these combined approaches and demonstrate their advantageous properties. With the properties of the combined methods demonstrated on this explicit function, the statistics-of-interval approach is demonstrated for the real gas problem. Because of the expense of validation, the mixed results for the real gas solver can not be validated directly, and accuracy is inferred based on validation of each individual component of the combined method and convergence of the results.

7.1 Gradient-based Mixed Aleatory/Epistemic Uncertainty

The quantification of uncertainty arising from both aleatory and epistemic sources is particularly challenging, especially as the dimension of the problem increases. For these types of problems, the input parameters are divided into two sets: a set of variables with only aleatory uncertainty and a set of variables with only epistemic uncertainty. The traditional strategy for mixed aleatory/epistemic uncertainty is known as nested sampling. For nested sampling, sampling is performed over the epistemic samples and for each epistemic sample, an aleatory uncertainty quantification problem is solved. The sampling over epistemic samples is performed using Latin Hypercube sampling with a fixed number of samples (~ 3) in each direction (with a total number of samples increasing as 3^d with d representing the number of epistemic variables). For the aleatory uncertainty problem, Monte Carlo sampling is typically used, requiring several thousands of samples. Hence, the expense of nested sampling increases exponentially fast and easily requires millions of samples even for a low dimensional problems. In order to reduce the cost associated with nested sampling, a surrogate model can be constructed over all the input parameters and samples can be extracted from it according to a nested strategy.

The main limitation of this surrogate based approach is again the curse of dimensionality. Using the strategies outlined for aleatory uncertainty, such as gradient-enhancement and dimension reduction, it is possible to construct a surrogate over all variables and sampling from this surrogate according to a nested strategy. However, due to the extreme number of points required for nested sampling, even sampling from a surrogate can pose problems. For example, if approximately 6000 sample points are used for the aleatory uncertainty quantification ($\sim 3^8$) and 3 sample points are used for each epistemic variable, the total number of samples required for the nested strategy is 3^{d+8}. For hypersonic flows, the number of epistemic variables is much larger than the number of aleatory variables [13]. For the 66 dimensional problem used in Chapter 5, 64 of the parameters are more accurately characterized as epistemic, corresponding to the transport and chemical kinetic parameters. For this variable partitioning, the total number of samples required for the nested approach is over 10^{34}. For comparison, the total number of stars in the observable universe is between 3×10^{22} and 100×10^{22} and the total number of bacterial cells on earth is estimated as 5×10^{30} [91, 92]. Clearly, acquiring and processing this amount of data is infeasible, regardless of how inexpensive a function evaluation is.

Because the goal of mixed aleatory/epistemic uncertainty is the determination of a P-box (a region in which the function exists with a specified probability), only the minimum and maximum values over the epistemic variables are required. Hence, optimization can replace the sampling over epistemic variables. To account for the aleatory uncertainty in an inexpensive manner, this optimization can be combined with a surrogate model. Because the number of aleatory variables in hypersonic flow is limited, the curse of dimensionality is likely not a problem for this surrogate. By using a gradient-based optimization technique, the scaling of the combined method as a function epistemic variables can be optimal when combined with a Newton or quasi-Newton optimizer.

Optimization and surrogate modeling can be combined in one of two ways. The approach most analogous with nested sampling will be referred to as the uncertain optimization approach. For this method, the surrogate model is constructed within the optimization process to account for aleatory uncertainty. The second method is referred to as the statistics-of-intervals (SOI) approach. For this method, a surrogate is constructed based on the results of multiple optimizations carried out at different aleatory values.

Statistics-of-Intervals Approach

For the statistics-of-intervals approach to the mixed uncertainty quantification strategy, the bounds of the output interval due to the epistemic uncertainty are treated as random variables, and the variability of these bounds due to the aleatory uncertainties in the problem are characterized. This process is best demonstrated with a set of equations. Let α represent the variables with associated aleatory uncertainties, β represent variables with epistemic uncertainties, and y represent the output of interest for a simulation. For this work, the variables in each group are assumed to have only aleatory or only epistemic uncertainty. In order to account for both types of uncertainty, sampling is performed for the aleatory variables while optimization is performed over the epistemic variables. This combination can be represented mathematically as follows.

$$y = f(\alpha, \beta) \tag{7.1}$$

$$y_{max}(\alpha) = \max_{\beta} f(\alpha, \beta) \tag{7.2}$$

$$y_{min}(\alpha) = \min_{\beta} f(\alpha, \beta) \tag{7.3}$$

The variables y_{max} and y_{min} can now be treated as random variables, since the inputs, α, are random variables with associated distributions. To characterize the distribution of y_{max} and y_{min}, one must extract repeated samples of y_{max} and y_{min}. These extractions entail running the optimization problem for the specified variables α. Because of the expense of these optimizations, strategies used to reduce the computational cost associated with sampling must be employed. For this work, separate surrogates are created for y_{max} and y_{min} as a function of the aleatory variables, and samples are extracted from these surrogates. Because the number of aleatory variables in this work is relatively small for this work compared with the number of epistemic variables, the required number of samples for this surrogate is small, necessitating few optimization results. Additionally, because of the small dimension, gradient information is not required for the surrogate. This fact is fortunate as a gradient-enhanced surrogate for this application would required the differentiation of the extrema of the optimization with respect to the aleatory variables, a quantity which is difficult to calculate.

Because the optimization results are viewed as general random variables, any surrogate can be used to represent the aleatory dependence of the variables. For this work, a Kriging surrogate is used. (When the Kriging model is used in conjunction with the statistics-of-intervals approach, the method is referred to as SOI-Kriging). With the surrogate constructed over the optimization results, any statistic of the output can be predicted and a full CDF for each bound can be constructed.

Uncertain Optimization Approach

In addition to the proposed method, an uncertain optimization (UQOPT) approach is considered for comparison [43]. In this approach, a single pair of optimization problems (one minimization and maximization) is performed over the epistemic variables. The objective for this optimization is a statistical quantity due to the aleatory uncertainty present in the problem. This process is represented by using the following equations. Within these equations, β again represents the set of epistemic variables, and α represents the aleatory variables. In contrast to the statistics-of-intervals approach, the statistical metric of interest must be chosen before the optimization is performed. Let the variable $J(\beta)$ represent a statistical metric based on the simulation output $f(\alpha, \beta)$, such as a variance or reliability metric. The optimization problem is now posed as follows.

$$J_{max} = \max_{\beta} J(\beta) \tag{7.4}$$

$$J_{min} = \min_{\beta} J(\beta) \tag{7.5}$$

An example of $J(\beta)$ is the reliability metric seen in equation (7.6). This reliability metric approximates the quantiles of the distribution based on moments of the distribution.

$$J(\beta) = \mu(\beta) - c\sigma(\beta) \tag{7.6}$$

Here, μ is the mean of the distribution, σ is the standard deviation and c is the specified reliability. For a 99% reliability metric, c is -2.33.

In order to determine statistics required for $J(\beta)$, the distribution of $y(\alpha, \beta)$ based on variations in α must be characterized for each β encountered during the optimization process. In order to accelerate this process, a surrogate of $y(\alpha, \beta)$ with respect to α is created, and statistics are calculated based on this surrogate. Because of the ease with which statistics can be calculated in closed form, a polynomial chaos expansion is used to calculate $J(\beta)$, yielding the method denoted as UQOPT-PC [43]. The polynomial chaos expansion is similar to the polynomial regression given in Chapter 4; however, the choice of basis functions and nature of the desired outputs allows for several time saving simplifications. For a polynomial chaos expansion, this expansion is represented as follows [20].

$$y(\alpha, \beta) = \sum_{i=0}^{N_p} \hat{y}_i(\beta)\phi_i(\alpha) \tag{7.7}$$

Here, $\hat{y}$ are the expansion coefficients, and $\phi(\alpha)$ is a polynomial in the random variable. The exact choice of polynomial depends on the input distribution with Hermite polynomials used for normal distributions and Legendre polynomials for uniform distributions [20]. Since the expansion polynomials are orthogonal, the determination of the expansion coefficients requires a simple inner production that can be computed by quadrature.

$$\hat{y}_i(\beta) = \int_R y(\alpha', \beta)\phi_i(\alpha')d\alpha = \sum_{k=0}^{N_q} w_k y(\alpha_k, \beta)\phi_i(\alpha_k) \tag{7.8}$$

Here, R is the support of the basis function ϕ, w_k are the quadrature weights, and N_q is the total number of quadrature points. Once the coefficients are determined, the mean and variance are given by the following.

$$\mu_y = \hat{y}_0 \tag{7.9}$$

$$\sigma_y^2 = \sum_{i=1}^{N_p} \hat{y}_i^2 \int_R \phi_i(\alpha')^2 d\alpha' \tag{7.10}$$

Because of these explicit relations, the derivatives required for a gradient-based optimizer are easily computable for many statistical metrics. Because the optimization is performed over the epistemic variables, only the derivative with respect to β is required. This gradient is easily computed using the following equations [43].

$$\frac{\partial y(\alpha, \beta)}{\partial \beta} = \sum_{i=0}^{N_p} \frac{\partial \hat{y}_i(\beta)}{\partial \beta}\phi_i(\alpha) \tag{7.11}$$

$$\frac{\partial \hat{y}_i(\beta)}{\partial \beta} = \int_R \frac{\partial y(\alpha', \beta)}{\partial \beta}\phi_i(\alpha')d\alpha' = \sum_{k=0}^{N_q} w_k \frac{\partial y(\alpha_k, \beta)}{\partial \beta}\phi_i(\alpha_k) \tag{7.12}$$

Although technically any surrogate or statistical metric $J(\beta)$ can be used with this method, the desire to use a gradient-based optimizer limits the type of surrogate and statistical metric that can be used. In order to use a gradient-based optimizer, both the statistical metric and the surrogate process used to produce the metric must be differentiable, limiting the choice of statistical metric to distribution moments and choice of surrogate to deterministic models.

Method Comparison

The two combined optimization/surrogate-based mixed uncertainty quantification methods outlined here approach the UQ problem from different directions, with the order of surrogate modeling and optimization interchanged between the two methods. In comparing the two methods, the question of whether the two approaches yield equivalent results must be addressed. For a given statistic, the uncertain optimization produces the minimum and maximum values of the statistic, while the statistics-of-intervals approach produces the corresponding statistic of the minimum and maximum values from the optimization. In mathematical terms, let the function $F(\mathbf{y}(\beta), \beta)$ map the samples over the aleatory variables from a given set of episetmic variables to the statistic of interest. The results from the uncertainty optimization are given as:

$$J_{min} = \min_{\beta} F(\mathbf{y}(\beta), \beta) \tag{7.13}$$

$$J_{max} = \max_{\beta} F(\mathbf{y}(\beta), \beta) \tag{7.14}$$

while the results of the statistics-of-intervals approach gives:

$$J_{min} = F(\mathbf{y}_{min}, \beta) \tag{7.15}$$

$$J_{max} = F(\mathbf{y}_{max}, \beta) \tag{7.16}$$

From these results, it is clear that if the statistic of interest, F, is monotone, the results will be identical as: $\min_{\beta} F(\mathbf{y}) = F(\mathbf{y}_{min})$ and similarly for the maximum. For most statistics of interest, this condition is met, meaning the two approaches will produce identical results [44].

In addition to theoretical concerns, several practical differences exist between the two methods. The main practical difference between the two approaches is that the uncertain optimization requires a new pair of optimization results for each statistic of interest, while the statistics-of-intervals approach can produce any number of statistics. Even though the

Table 7.1: Uncertain Model Parameters

Variable	Type	Uncertainty
ρ_∞ (kg/m^3)	Aleatory	±10% ($\sigma = 5\%$)
$V_\infty (m/s)$	Aleatory	±30.84 ($\sigma = 15.42$) m/s
$\Omega^{1,1}_{N2-N2}, \Omega^{2,2}_{N2-N2}$	Epistemic	±20%
$\Omega^{1,1}_{N2-N}, \Omega^{2,2}_{N2-N}$	Epistemic	±20%
$\Omega^{1,1}_{N2-O}, \Omega^{2,2}_{N2-O}$	Epistemic	±20%
$\Omega^{1,1}_{N2-O2}, \Omega^{2,2}_{N2-O2}$	Epistemic	±20%

statistics-of-intervals approach can predict any number of statistics with a single set of optimization results, the overall expense of the two methods is comparable. Although a single pair of optimizations is required for the uncertain optimization, each iteration of the optimization requires a surrogate to be constructed, necessitating numerous function evaluations. For the statistics-of-intervals approach, multiple optimizations are required; however, each optimization is straight-forward, requiring a single function/gradient evaluation per iteration. In addition to issues of cost, the use of a gradient-based optimizer requires that the metric $J(\beta)$ used with the UQOPT method be differentiable. Additionally, the process used to inexpensively predict $J(\beta)$ based on a surrogate must also be differentiable. The statistics-of-intervals approach has no such limitations on the statistics that can be computed or the surrogate that can be used to model the aleatory variability. Because the surrogate is applied merely to a collection of optimization sampling results, any surrogate may be used to represent the aleatory distribution and any statistic predicted.

7.2 Problem Definition

For these mixed results, the epistemic variables were the same 8 variables used for the validation and testing in Chapter 6. The aleatory variables were the freestream density and velocity. These variables are listed in Table 7.1. For the episetmic variables, the uncertainty is specified by an interval with a half width of 20%, while the aleatory variables were assumed to follow a Gaussian distribution, specified with the mean and standard deviation.

7.3 Fay-Riddell Heating Results

In order to provide a baseline result for the combined surrogate-optimization methods illustrated in this work, exhaustive nested sampling is performed, and a series of cumulative distribution function (CDF) curves is constructed (known as a horse-tail plot). This plot is shown in Figure 7.1. For this baseline result, 5,000 aleatory samples were performed for each epistemic sample. For the epistemic variables, three samples were taken for each variable, following the experiences of other mixed aleatory/epistemic uncertain studies [15]. This sampling strategy gave a total of 6,561 (3^8) epistemic samples. Because a full set of aleatory samples is required for each epistemic sample, the total number of samples was 3.28×10^7. Clearly, for anything other than an analytic function, this nested sampling is prohibitively expensive. For the Fay-Riddell heating correlation, this nested sampling required approximately 82 minutes to perform. In contrast, performing 3.28×10^7 samples using the CFD solver would require 3010 cpu-**years** for this test case. Even if performed in parallel, this amount of computing is far beyond the budget of nearly all modeling projects. This fact is in spite of the limited dimension of the problem. The samples for both the epistemic and aleatory variables were extracted using Latin hypercube sampling, uniform sampling in the case of epistemic variables and transformed to normal sampling by means of the inverse cumulative distribution function for the aleatory variables.

Because the epistemic uncertainty comes with no associated statistical distribution, each of the CDF curves in Figure 7.1 is equally valid. In order to provide some manageable metric, the bounds of the CDF curves can be found such that all samples lie between these two curves. One method for finding these bounding curves is to generate a CDF for the minimum and maximum values produced by sampling over the epistemic variables. In terms of nested sampling, for a given set of aleatory variables, the maximum and minimum values generated through sampling over the epistemic variables are determined and a CDF of these variables is constructed. These bounds are plotted in Figure 7.2.

With these baseline results established, the performance of optimization for the epistemic uncertainty within the mixed problem is demonstrated. With optimization demonstrated as a viable strategy for mixed problems, the statistics-of-intervals approach will be

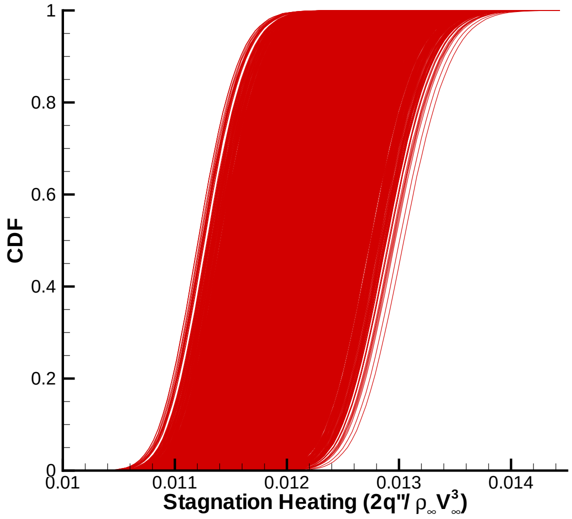


Figure 7.1: Horse-tail plot for Fay-Riddell heating with 6,561 epistemic samples and 5,000 aleatory samples.

applied for this problem. In addition to providing demonstration results for this method, the statistics-of-intervals approach will be compared with uncertain optimization.

7.3.1 Performance of Optimization

In order to show that optimization can be used for the epistemic component of the mixed problem, the distribution associated with the optimization results is constructed through exhaustive sampling of the optimization. For this test, a pair of maximization and minimization problems over the epistemic variables is performed for each set of aleatory variables. In order to accurately characterize the distribution of these bounds, 5,000 pairs of optimization problems were performed corresponding to the aleatory samples from the nested sampling approach. The optimization method used throughout this test is again L-BFGS [93]. Approximately 40 function/gradient evaluations were required for each pair of optimizations, giving a total of 1.96×10^5 function/gradient evaluations. The CDF curves of these optimization results should agree with the bounding CDF curves from nested sampling. The horsetail plot for nested sampling and the CDF curves for the optimization results are plotted

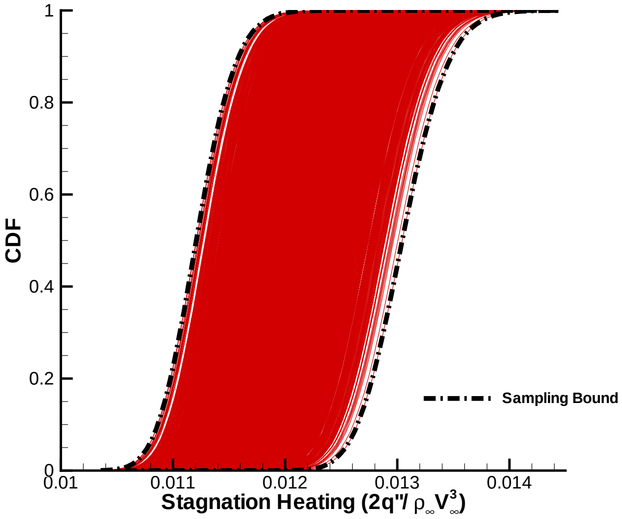


Figure 7.2: Horsetail plot with CDF curves for bounds from sampling.

in Figure 7.3.

From these results, the optimization seems to give overly conservative predictions for the combined epistemic/aleatory uncertainty. This behavior is similar to the result seen for the pure epistemic uncertainty case. Because the horsetail plot is constructed through exhaustive sampling, the minimum and maximum predictions for a given set of aleatory variables are limited by the extent to which sampling in the epistemic variables has been performed. Because optimization does not suffer from this limitation, it is reasonable to expect that the optimization approach gives the more accurate uncertainty prediction and that more extensive sampling of epistemic variables within the nested sampling approach should cause the sampling-based results to approach the optimization bounds.

In order to demonstrate that the optimization results give the proper bounding CDF curves, nested sampling was performed with increasing numbers of epistemic samples. Because the number of samples over the epistemic variables increases rapidly with dimension, the dimension of the problem was reduced to 6, and the number of samples for each epistemic variable was increased. As was the case with the 10-dimensional problem, the freestream density and velocity were again treated as aleatory. The set of epistemic variables was re-

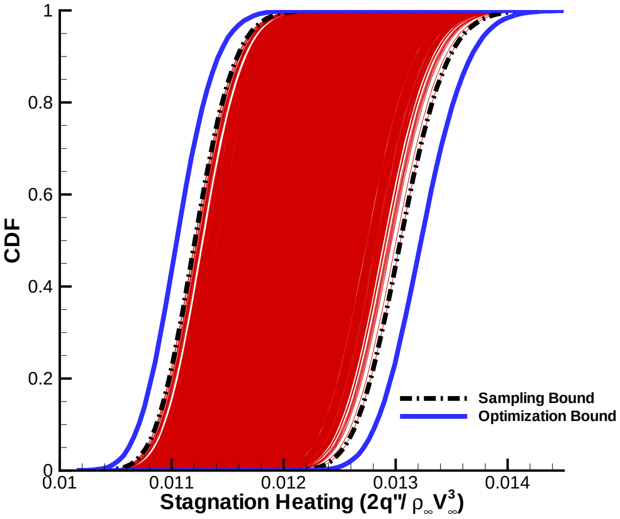


Figure 7.3: Horsetail plot with optimization based bounding CDF curves.

duced to the first four collision integrals in Table 7.1, and nested sampling was performed using 3, 5 and 10 samples in each dimension. Figure 7.4 shows the bounding CDF curves for each of these cases. Plotted with these sampling-based CDF curves are the CDF curves based on optimization.

As the plot demonstrates, the bounding CDF curves for nested sampling approach the optimization bounding curves as the number of samples in each dimension is increased. Hence, provided the global minimum and maximum can be found (a condition that is virtually impossible to guarantee but appears to be the case for this problem), the optimization should be viewed as more accurate than the nested sampling approach. This conclusion is fortunate because the total number of samples required for exhaustive sampling of the optimization results is only 1.96×10^5 function/gradient evaluations as opposed to the 3.28×10^7 function evaluations required for nested sampling. Given the large savings, any additional cost associated with calculating the gradient appears to be more than off-set by the reduction in total number of evaluations.

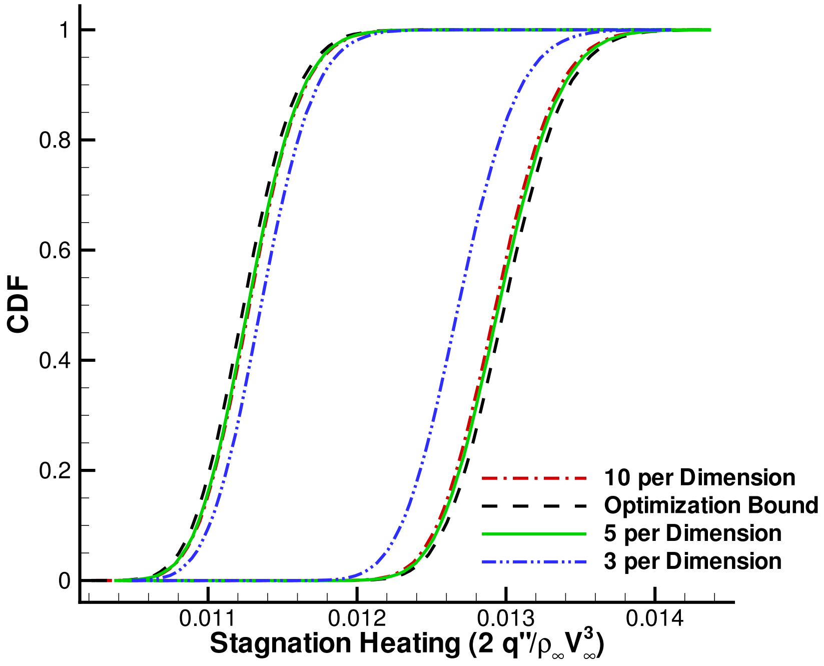


Figure 7.4: Nested sampling bounding CDF curves with increasing numbers of epistemic samples compared with optimization-based bound CDF curves.

7.3.2 Statistics of Intervals-Kriging Results

With the validity of optimization shown for propagating epistemic uncertainty for this problem, sample reduction techniques can be applied over the aleatory variables to reduce the total number of optimizations required to characterize the distribution of the optimization bounds. For this work, two surrogates are created to model the variation of the optimization bounds as a function of the aleatory variables. Although any surrogate could be used, a Kriging model is employed throughout this section (ordinary Kriging unless otherwise specified). Unlike the surrogate used in Chapter 5, this Kriging model is built exclusively from function values. In this context, the use of a gradient-enhanced Kriging model would require the differentiation of the optimal results, a quantity that is difficult to calculate. In order to test the surrogate's ability to represent the distribution of optimization results with a limited number of samples, a Kriging model was created by using the results of four pairs of optimizations run with different values for the aleatory variables. The aleatory variables used for these optimizations were chosen through Latin Hypercube sampling. Once the Kriging model was constructed, aleatory samples were extracted from the surrogate to build up an approximate CDF curve for the optimization results. Figure 7.5 shows the bounding CDF curves for this Kriging-based sampling. For comparison, the CDF curves for exhaustive sampling of the optimization results and the CDF curves from nested sampling are shown.

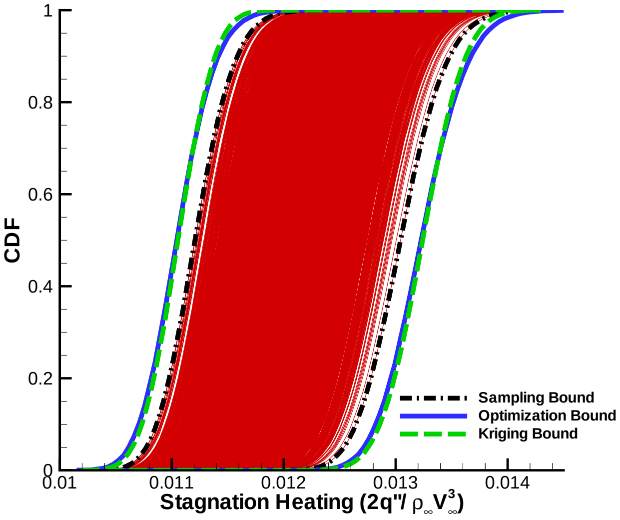


Figure 7.5: CDF of optimization results based on Kriging model using 4 pairs of optimization results.

As the figure demonstrates, with only four pairs of optimizations and 157 function/-gradient evaluations, the Kriging-based results closely approximate the CDF curves for the minimum and maximum surface heating values based on exhaustive sampling of the optimization results. In addition to qualitatively judging the quality of the CDF curves produced by sampling from the Kriging model, specific statistics of the interval bounds can be calculated. By examining a specific statistic, the effect of number of training points (pairs of optimization results in this context) for the model can be characterized, and the total expense in terms of function/gradient evaluations can be shown.

In order to assess the performance of the Kriging model for characterizing the distribution of surface heating due to aleatory variables, the average, standard deviation and $99th$ percentile of the minimum and maximum distributions were calculated with varying numbers of training points. The ordinary Kriging model ($p = 0$ regression) was used for this test. The convergence of the average and variance predictions for the maximum and minimum values as a function of training data size are plotted in Figure 7.6. Table 7.2 shows the $99th$ percentile prediction based on Kriging models with varying training-set sizes. Unlike the uncertain optimization method, quantile predictions for the interval can be made with ease since the statistic of interest need not be differentiable for the SOI method. In order to generate the training data, sets of aleatory variables were generated via Latin Hypercube sampling, and a pair of optimizations (minimization and maximization) was run for each value of the aleatory variables.

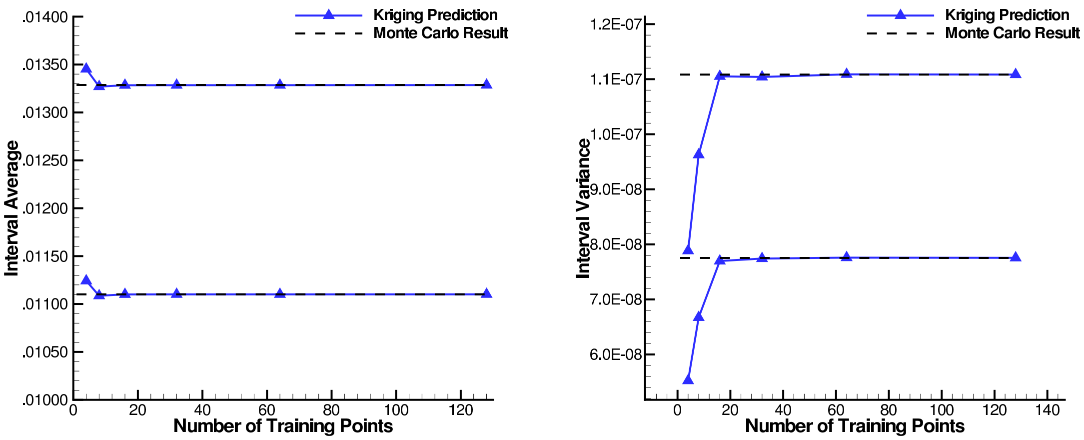


Figure 7.6: Convergence of average (Left) and variance (Right) prediction for minimum and maximum distribution using Kriging models built from increasing numbers of optimization results

As these results demonstrate, the statistic predictions for the ordinary Kriging model closely approximate the exact values with a moderate number of optimization results (matching 4 digits in both the average and variance with 16 pairs of optimizations with a total cost of 603 function/gradient evaluations of the physical model).

Table 7.2: $99th$ percentile predictions for SOI method Using ordinary Kriging model

Training Data Size	F/G Evaluations	$99th$ Percentile of Min	$99th$ Percentile of Max
4	156	1.18169×10^{-2}	1.41405×10^{-2}
8	289	1.17531×10^{-2}	1.40693×10^{-2}
16	603	1.17928×10^{-2}	1.41132×10^{-2}
32	1190	1.17950×10^{-2}	1.41126×10^{-2}
64	2433	1.17951×10^{-2}	1.41133×10^{-2}
128	4862	1.17952×10^{-2}	1.41140×10^{-2}
Exact	196427	1.17951×10^{-2}	1.41151×10^{-2}

7.3.3 Comparison between Statistics of Intervals and Uncertain Optimization

In this section, the performance of the uncertain optimization approach is compared with the SOI approach using the Fay-Riddell heating correlation. For this comparison, the interval on a single statistic was predicted by using both methods. For a fair comparison, a polynomial chaos expansion was used to represent the aleatory variation in both methods while L-BFGS was again used as the optimization strategy.

The statistical metric used in this test was the following mean value reliability metric.

$$J = \mu - \sigma c \tag{7.17}$$

Here, μ is the average and σ is the standard deviation of the distribution due to aleatory variables and c is a defined parameter corresponding to the desired reliability level. For a normal distribution, c equal to -2.33 gives a 99% reliability metric. This statistic was chosen because of the ease with which it and its derivative with respect to epistemic variables are calculated for a polynomial chaos expansion.

The comparison between the two methods is based on the accuracy of the statistical metric as well as the total number of function/gradient evaluations required for each method. For this test, accuracy was assessed based on statistics calculated from exhaustive sampling of the optimization results (5,000 independent pairs of optimizations), and this assessment was based on multiple polynomial orders. Table 7.3 summarizes the results of this test. As these results demonstrate, when the same surrogate is used, the two methods give identical results,

Table 7.3: Method comparison results for 99% reliability metric

Order	UOPT Lower	UOPT Upper	SOI Lower	SOI Upper
P=1	1.174679×10^{-2}	1.405831×10^{-2}	1.174679×10^{-2}	1.405831×10^{-2}
P=2	1.174877×10^{-2}	1.406067×10^{-2}	1.174877×10^{-2}	1.406067×10^{-2}
P=3	1.174878×10^{-2}	1.406068×10^{-2}	1.174878×10^{-2}	1.406068×10^{-2}

Exhaustive Lower	Exhaustive Upper
1.174998×10^{-2}	1.406205×10^{-2}

Table 7.4: Method cost comparison in terms of number of function gradient evaluations for 99% reliability metric

Order	UOPT Cost	SOI Cost
P=1	156	157
P=2	351	355
P=3	624	631

Exhaustive optimization sampling cost = 196427

and these results compare well to those based on exhaustive sampling of the optimization results. For this problem at least, the sampling and the optimization steps of the methods appear to be interchangeable, although this is likely not the case in general.

In addition to assessing the predictions of the two methods, the cost of each method was also compared. Table 7.4 shows the number of function/gradient evaluations required for each method at the different polynomial orders. As the results demonstrate, the costs of the two methods are nearly the same, with the uncertain optimization slightly edging out the statistics-of-intervals approach, but only by an insignificant amount compared with the total cost. Additionally, both methods are significantly less expensive than exhaustive sampling of the optimization results. As the number of aleatory variables increases, the savings experienced would likely decrease as the expense of training the surrogate increases.

For the SOI approach, any surrogate may be used to represent the aleatory dependence of the minimum and maximum values. In the previous section, an ordinary Kriging model was used. In order to compare the polynomial regression approach to Kriging, a universal Kriging model can be used. As previous work has shown, universal Kriging typically provides higher

Table 7.5: Universal Kriging model for 99% reliability metric

Order	UOPT Lower	UOPT Upper	SOI-UK Lower	SOI-UK Upper
P=1	1.174679×10^{-2}	1.405831×10^{-2}	1.17471×10^{-2}	1.40586×10^{-2}
P=2	1.174877×10^{-2}	1.406067×10^{-2}	1.17499×10^{-2}	1.40620×10^{-2}
P=3	1.174878×10^{-2}	1.406068×10^{-2}	1.17500×10^{-2}	1.40621×10^{-2}

Exhaustive Lower	Exhaustive Upper
1.174998×10^{-2}	$1.40641150205 \times 10^{-2}$

accuracy than does standard L_2 regression and under some circumstances can outperform ordinary Kriging [34].

As the results show, although the polynomial chaos results are accurate, the universal Kriging results achieve a higher level of accuracy using the same number of training points and regression order. Although universal Kriging could be used within the UQOPT approach, the calculation of the gradient of statistic predictions is difficult. In contrast, the implementation of universal Kriging within the SOI approach is straight forward, demonstrating another advantage of the SOI approach.

7.4 Real Gas Computational Fluid Dynamics Results

In order to demonstrate the proposed statistics-of-intervals/Kriging approach for a practical computational simulation, the approach was applied for uncertainty quantification within a real gas CFD solver. For these tests, the uncertainty of integrated surface heating was calculated. The uncertain parameters for the simulation are the same as the parameters used for the Fay-Riddell model (Table 7.1).

Because of the expense of the CFD simulation, the exact mixed aleatory/epistemic uncertainty results can not be calculated through either nested sampling or exhaustive sampling of optimization results (which would require approximately 30 million and 300,000 CFD results, respectively). As stated previously, performing nested sampling on this problem would require 3010 cpu-**years** of computing, a number which would increase exponentially as the dimension of the problem expands. Performing exhaustive sampling of optimization results is only slightly more feasible, requiring 29.52 cpu-**years** of computing. Unlike the nested sampling, this number should increase less dramatically as the dimension expands; however, for both methods, the computational cost is clearly beyond the budget for most projects.

Because exhaustive methods are prohibitively expensive for this problem, validation for the SOI-Kriging method applied to the real gas simulation was achieved by validating each element of the method against exhaustive sampling. With each element validated, the mixed aleatory/epistemic uncertainty was calculated by using successively more accurate surrogate models to demonstrate convergence of the statistic predictions.

The optimization portion of this method was demonstrated in the Chapter 6. Based on these results, the optimization produced an output interval encompassing that of sampling. With the optimization portion of the method validated, the ability of a function-only surrogate model to capture the aleatory variation of the integrated surface heating was tested. Additionally, by sampling in the same dimension as the mixed problem, this validation gives insight into the number of optimization pairs required to construct the bounding CDF curves. For this test, the epistemic variables were frozen at their non-perturbed values (1 in the terms of the parameters defined in Table 7.1), and sampling was performed over the aleatory variables. In order to provide validation data, Monte Carlo sampling was performed over the aleatory variables, and the distribution was characterized both by constructing a CDF curve and by calculating specific statistics. In order to acquire accurate statistics, 4,564 samples were used, and a separate simulation was performed for each. With the validation data acquired, ordinary Kriging models with increasing numbers of training points were constructed. Because the epistemic variables for this test were fixed, each training point required only a single CFD simulation.

As a first test, the convergence of the mean, variance, and $99th$ percentile are shown for Kriging models with increasing numbers of training points. The convergence of this metric as a function of training point number is given in Table 7.6. As the results show, predictions of the Kriging model rapidly converge toward the Monte Carlo results. In addition to predicting distribution statistics, a CDF of the output is constructed based on samples extracted from the Kriging model and compared with that of Monte Carlo sampling. Figure 7.7 shows the

Table 7.6: Convergence of Kriging statistic predictions for aleatory uncertainty with fixed epistemic variables with increasing number of training points

Training Points	Average	Variance	$99th$ Percentile
8	1.036110×10^{-2}	6.061055×10^{-8}	1.098518×10^{-2}
16	1.036622×10^{-2}	6.075630×10^{-8}	1.097558×10^{-2}
31	1.034997×10^{-2}	6.145065×10^{-8}	1.098506×10^{-2}
59	1.037171×10^{-2}	6.185576×10^{-8}	1.099184×10^{-2}
121	1.036669×10^{-2}	6.120957×10^{-8}	1.097695×10^{-2}

Monte Carlo Results

Samples	MC Average	MC Variance	MC $99th$ Percentile
4564	1.036082×10^{-2}	6.103365×10^{-8}	1.098385×10^{-2}

predicted CDF curve for a Kriging model with 8 training points and the CDF from Monte Carlo sampling.

Using only 8 samples, the Kriging model produces a CDF curve nearly identical to the curve produced through Monte Carlo sampling, at a fraction of the cost. For this problem, the uncertainty due to only two variables was considered. Obviously, as the dimension of the problem grows, the cost associated with training the Kriging model will increase. Based on these results, the mixed problem likely needs on the order of 8 sample points to accurately predict the bounding CDF curves.

With each element of the SOI-Kriging approach validated independently, the complete mixed aleatory/epistemic uncertainty is predicted by using optimization for the epistemic dependence and an ordinary Kriging model for the aleatory dependence. In order to demonstrate the validity of the full results, the convergence of the minimum and maximum $99th$ percentile predictions are shown as the number of training points for the Kriging model is increased. For the mixed results, a training point now represents a pair of optimizations and has a cost of approximately 60 function/gradient evaluations on average. Table 7.7 shows the convergence of the maximum $99th$ percentile and minimum $99th$ percentile as the number of training points is increased. As the table demonstrates, the statistic predictions quickly converge to asymptotic values. Included in Table 7.7 is the total cost in terms of function/gradient evaluations. While the nested sampling and exhaustive sampling

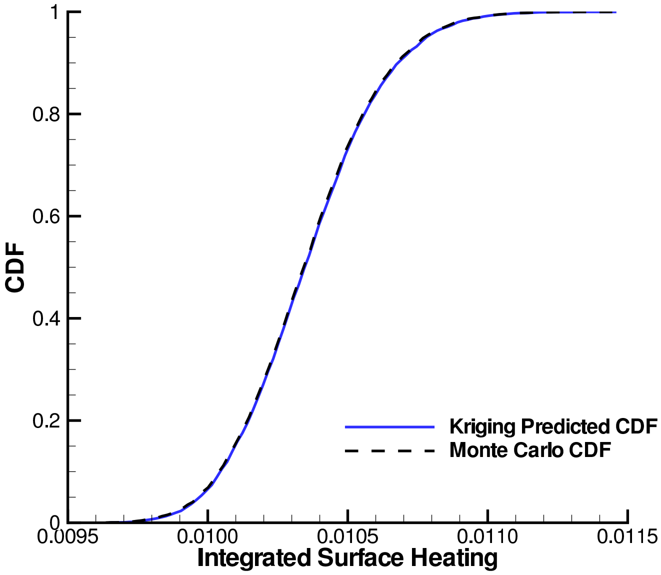


Figure 7.7: CDF based on Kriging model using 8 sample points compared with CDF of Monte Carlo results with fixed epistemic variables.

Table 7.7: $99th$ percentile predictions for SOI method using ordinary Kriging model for real gas CFD simulation

Training Data Size	F/G Evaluations	99^{th} percentile of Min	99^{th} percentile of Max
8	500	1.017556×10^{-2}	1.206949×10^{-2}
15	900	1.016681×10^{-2}	1.207132×10^{-2}
23	1400	1.018928×10^{-2}	1.207939×10^{-2}
52	3000	1.020232×10^{-2}	1.210513×10^{-2}
104	6176	1.020243×10^{-2}	1.210416×10^{-2}

of the optimization were prohibitively expensive for the CFD model, the SOI-Kriging model was able to capture converged statistics with a number of function/gradient evaluations within the computational budget (although still most likely prohibitively high for complex simulations). Nevertheless, by using the Kriging model combined with optimization, the SOI-Kriging method was able to quantify the mixed aleatory/epistemic uncertainty problem where other methods could not be used.

Figure 7.8 shows the convergence of the average and variance prediction based on Kriging models with increasing numbers of training points. As this Figure shows, the convergence of the statistics for the real gas simulation are not as nicely behaved as those from the Fay-Riddell heating correlation; however, it is clear that the Kriging surrogate produces similar results for all numbers of training points. Additionally, it appears that the variability caused by insufficient amount of training data is small compared with the overall interval produced due to the epistemic uncertainty.

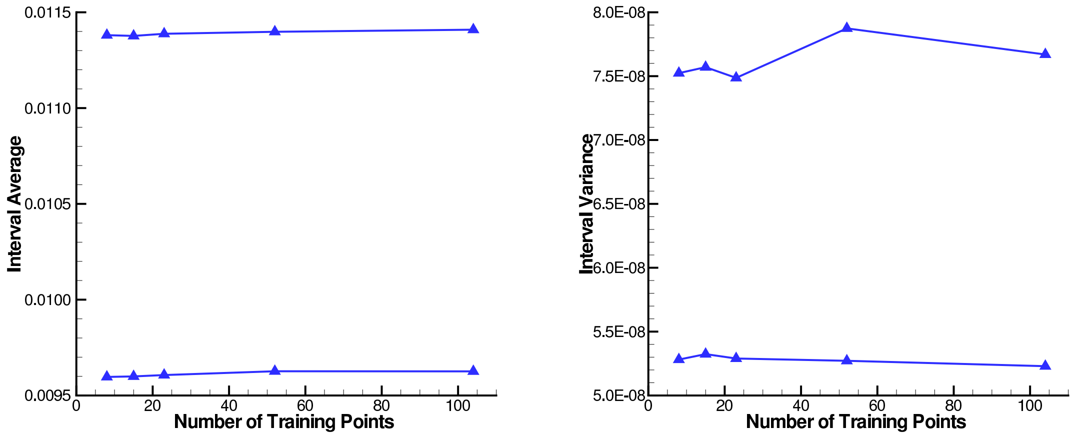


Figure 7.8: Convergence of average (Left) and variance (Right) prediction for minimum and maximum distribution using Kriging models built from increasing numbers of optimization results for real gas CFD simulation.

In addition to calculating specific statistics of the output interval, the CDF of the minimum and maximum values can be predicted by sampling from the Kriging surface. The bounding CDF curves are plotted in Figure 7.9 for a Kriging model based on 8 and 104 pairs of optimizations. As the figure demonstrates, the CDF curves are nearly identical, suggesting that the Kriging model has reached some level of convergence.

7.5 Summary

In this chapter, a combined optimization-surrogate method for the propagation of mixed aleatory/epistemic uncertainty was outlined. The details of the proposed statistics-of-intervals approach were first given and compared to a similar method, denoted as uncertainty opti-

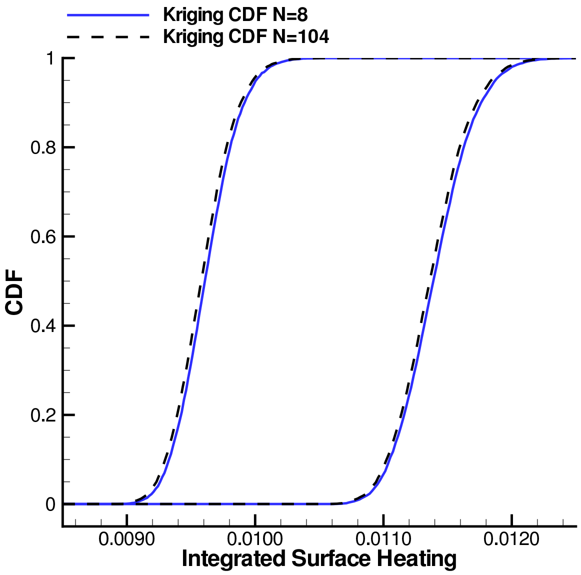


Figure 7.9: Kriging-predicted CDF curves for maximum and minimum values using 8 and 104 optimization pairs.

mization. With the methods detailed, the various properties of the statistics-of-intervals method were demonstrated using the Fay-Riddell heating correlation. Finally, the method was demonstrated for a real gas CFD simulation. From the results of this section, the statistics-of-intervals approach was shown to give results in good agreement with nested sampling approaches and compared favorably with the similar uncertainty optimization method.

In terms of cost, the combined surrogate/optimization approaches examined in this work represented the only practical way of solving the combined aleatory/epistemic uncertainty quantification problem for expensive simulations. In order to apply nested sampling to the real gas CFD test case, 3010 cpu-**years** of computing would be required for the 10 dimension case examined here. In contrast, the SOI-Kriging method was able to produce reasonable approximations for the minimum and maximum statistics using approximately 431 cpu-hours of computing, representing over a factor of $60,000$ savings. This savings was achieved on a problem with 2 aleatory variables and 8 epistemic variables. As the number of epistemic variables increase relative to the number of aleatory variables, this savings should only increase due to the favorable scaling of gradient-based optimization as dimension expands.

Chapter 8

Conclusions and Future Work

For this work, a flow solver capable of simulating real gas hypersonic flows using a five species, two temperature model was developed. Within this solver, the discrete adjoint was implemented and used to efficiently calculate derivatives of the simulation output with respect to model parameters. Using this solver and adjoint implementation, gradient-based methods for both local and global sensitivity analysis were tested for a typical hypersonic flow over a blunt body. While the local sensitivity analysis was based on derivative values directly, a global sensitivity analysis based on derivative-enhanced regression was used as a basis for inexpensive Monte Carlo sampling. In addition to sensitivity analysis, gradient-based methods for rapid uncertainty quantification were developed and implemented for hypersonic flow simulations. For aleatory uncertainty quantification, gradient-based surrogate models were used to inexpensively approximate the output of a simulation and Monte Carlo sampling was performed on this surrogate. For epistemic uncertainty quantification, gradient-based optimization was used to inexpensively determine the interval of the simulation output based on the interval uncertainty of the input parameters. Finally, for mixed aleatory/epistemic uncertainty quantification, a combined surrogate-optimization approach was developed that builds a surrogate based on multiple gradient-based optimization results and Monte Carlo sampling is performed on this surrogate to determine statistics associated with the optimization results. Based on the results presented in the previous chapters, several conclusions can be made and a basis for future work is established.

8.1 Summary of Results

From the results presented in Chapters 4 through 7, a number of outcomes deserve to be highlighted. These outcomes are enumerated below.

1. Despite the complexity of the physical models within the simulation and the highly nonlinear nature of the flow physics, the discrete adjoint can be applied to hypersonic flows and the gradients produced by this approach show good agreement with finite difference. Additionally, because of the nonlinear nature of the flow solution versus the linear nature of the adjoint equations, the adjoint can be solved in a fraction of the time required for the flow solution, a property unique to hypersonic applications. For other flow regimes, the adjoint solution is typically between half to three times the cost of the flow solution [73, 94]. Provided this property holds across a range of test cases, gradient-based approaches show great promise for hypersonic simulations by providing additional information with only a marginal increase in cost.

2. Using the gradient produced by the discrete adjoint, local sensitivity analysis for the physical parameters in the simulation can be performed. Despite being localized in nature, this localized analysis identified a majority of the same important parameters when compared to the globalized sensitivity analysis. This result indicates that the regime where local sensitivity analysis is appropriate may be larger than initially thought in terms of flow complexity and input uncertainty.

3. Although the localized approach identified similar parameters, it was unable to accurately predict the fraction of the output variance contributed by each variable. To accomplish this prediction, global sensitivity analysis was required. Using a gradient-enhanced polynomial regression, this global sensitivity analysis was performed using a fraction of the simulation results required for the Monte Carlo approach. In contrast to similar regression-based global sensitivity [14], this gradient-enhanced method is capable to extending to large numbers of input variables with only linear growth in the cost when the regression order is limited. Although the regression-based results

were not in exact agreement with the Monte Carlo-based GSA, this approach represented a significant improvement over the localized approach in terms of predicting contributions to output variance and ranking of the importance of each variable.

4. For the case of aleatory uncertainty, statistics associated with the simulation output can be predicted inexpensively using gradient-based methods. Although not as accurate as other methods, linear extrapolation methods, such as the moment-method, provided reasonable estimates for the statistics. Through the use of gradient-enhanced surrogate models in conjunction with global sensitivity analysis, improved statistic predictions were possible for large dimensional problems without a dramatic increase in cost. For the 66 dimensional problem analyzed in this work, only 68 function/gradient evaluations were required to accurately predict the statistics associated with a 6000 point Monte Carlo data set. This result indicates that these rapid aleatory uncertainty quantification techniques are indeed applicable to problems involving hypersonic flows and the incorporation of gradient information serves to partially alleviate the curse of dimensionality often associated with these techniques.

5. For epistemic uncertainties, the interval of a simulation result was predicted using gradient-based optimization. In addition to more accurately predicting the output interval, the expense associated with this optimization was significantly less than that of traditional sampling-based approaches and hence, shows promise of scaling to larger dimension more readily than alternative methods. In addition to demonstrating the feasibility of gradient-based optimization for hypersonic problems, the results presented here indicated that local optimization approaches may be sufficient for practical epistemic uncertainty quantification problems in CFD.

6. For mixed aleatory/epistemic uncertainty, a novel optimization/surrogate based approach was developed and applied for a hypersonic simulation. Compared to nested sampling, this novel approach was able to reproduce relevant statistics with a cost several orders of magnitude smaller than the tradition approach for the explicit Fay-Riddell heating correlation. Compared with a similar method [43], the method was

able to produce more valuable information while maintaining the same cost. When applied to a CFD simulation, the method was able to give an estimate for the output uncertainty when the traditional method was prohibitively expensive.

Given these outcomes, the incorporation of gradient information into accelerated methods for uncertainty quantification and sensitivity analysis appears to be a valuable technique for further reduction in the cost associated with these methods and the extension of these methods to higher dimension.

8.2 Conclusions

The goal of this work was the advancement of strategies for rapid uncertainty quantification and sensitivity analysis applicable to complex engineering calculations, namely the simulation of hypersonic fluid flow. Based on the results presented in this dissertation, the gradient-based strategies explored in this work have advanced this goal. By reducing the cost associated with uncertainty quantification, the methods demonstrated in this work enable the use of simulation uncertainty in a wider variety of scenarios. As the role of simulation in the design and certification of new engineering systems expands, computational scientists will be increasing required to supply a confidence level for simulation results. For complex simulations, a single result can require days to weeks of computing, even when performed on massively parallel computers. In these situations, accelerated uncertainty quantification techniques, such as the methods presented in this work, represent the only possible means of providing this confidence level. This ability to assert a confidence level in simulation results is particularly important for situations in which experimental data is difficult or impossible to obtain; however, the utility of uncertainty quantification is not limited to these situations. For situations where experimental data is available, the rapid quantification of uncertainty in the simulation results can aide in the validation of computational models, enabling the uncertainty of both the experimental data and simulation results to be used in assessing the validity of the model. Finally, rapid uncertainty quantification techniques enable the use of simulation uncertainty within the process of numerical optimization. By incorporating uncertainty into the optimization, reliability and realizability constraints can be enforced within the optimization algorithm. Using more realistic constraints and reliability measures, the results of these robust optimizations should provide major gains in terms of safety, cost and performance compared with the traditional design process. Currently, robust optimization is only possible for relatively simple CFD applications [95]; however, if the cost of uncertainty quantification can be sufficiently reduced, it should be possible to apply these methods to more complex systems.

8.3 Future Work

The results of this work lay out a solid foundation for further explorations of adjoint methods and rapid uncertainty quantification and sensitivity analysis techniques in hypersonic flows. As this work only represents initial steps toward applying these techniques to hypersonic CFD applications, the gradient-based methods obviously need to be more extensively validated on a wider variety of flow problems and model parameters, as only a single flow test case and set of variables has been used throughout this work. Assuming the previously enumerated conclusions withstand this validation, future work should likely focus on the following.

Based on the regression models used for the sensitivity analysis and aleatory uncertainty, the design space for the hypersonic simulation appears to be accurately modeled by a quadratic representation. This fact coupled with the extremely low cost associated with solving the adjoint sensitivity equation motivates the use of higher-order derivatives for hypersonic applications. Using a combination of adjoint and forward sensitivity, the Hessian can be computed efficiently by solving $d + 1$ linear equations, where d is the number of parameters in the problem [17]. Based on the results presented here, the cost associated with these linear equations is likely significantly less than a flow solution for hypersonic simulations, allowing for the Hessian to be computed rapidly. Using the Hessian, a number of the methods used in this work can be expanded. For the case of surrogate models, the Hessian can be incorporated into the construction of the model just as the gradient was incorporated for this work [33]. For the case of optimization, the availability of the Hessian enables the use of more sophisticated full-newton optimization methods [96]. Although these methods may not be appropriate for optimization in all CFD applications, the extremely low cost associated with calculating the Hessian for hypersonic flows should justify the use of these optimization strategies.

In addition to exploring the use of higher order derivatives, the gradient-based methods presented here can be improved in a number of ways. In particular, the surrogate models used in this work should be extended to account for lower fidelity or faulty data. For predicting the behavior of real world systems, numerous sources of data, such as experimental data and various numerical approximations can be leveraged. Each of these sources have an associated quality. Using the appropriate multi-fidelity model, this quality can be accounted for when interpolating between this data to ensure higher quality data takes precedent in the prediction [97]. For CFD, this approach has been demonstrated for aerodynamic simulations using different mesh resolutions [98]. Due to the long history associated with hypersonic flow, numerous approximations exist for the prediction of systems in hypersonic flows and the use of multi-fidelity surrogate models can likely reduce the cost associated with constructing an accurate surrogate for complex design spaces. In addition to intelligently accounting for the quality of data, surrogate models capable of accounting for faulty data should be explored. For this work, the result of the simulation is assumed to be deterministic and exact. Practically, this assumption is not always possible as a CFD solution can be flawed due to incomplete convergence or numerical error. A surrogate capable of accounting for these faulty results will likely produce more accurate predictions of the output by effectively separating trends in the design space from noise in the simulation results [83].

As well as expanding the models used to capture aleatory uncertainty, the optimization methods used for epistemic uncertainty can likely benefit from the incorporation of surrogate models. Although the gradient-based approaches used in this work were capable of capturing what appeared to be global minimum and maximum values, this behavior is likely not the norm for most simulations [42]. The use of surrogate models for efficient global optimization is a topic of active research within CFD. In particular, the Kriging and gradient-enhanced Kriging methods used for aleatory uncertainty in this work have been shown effective for efficient global optimization in CFD [29, 31]. The application of these methods to the quantification of epistemic uncertainty represents a promising avenue toward alleviating the problem of local extrema. Additionally, because a pair of optimization problems are solved for epistemic uncertainty quantification, a surrogate-based approach allows for work to be shared between the two optimization problems, as the same surrogate can be used for both problems [43].

Besides these specific improvements, future work should focus on the goals of increasing the efficiency of simulation by providing additional information through derivative calculations. Additionally, further work is required in the field of uncertainty quantification in general, extending the field to account for all sources of uncertainty in the simulation. In addition to uncertainty quantification, incorporating information about the uncertainty of simulation results into the design and validation process is an area requiring additional research. Moving forward, uncertainty quantification is likely to become a necessary component of future simulation tools as validation of simulation results is moved later in the design process and as modeling is increasingly used to gain insight into the fundamental laws governing nature. For these situations, assessing the quality of simulation results will be critical for drawing relevant conclusions and the limited availability of experimental data will necessitate knowledge of both the uncertainty of the simulation as well as the experiment. The drive toward uncertainty quantification can also be viewed as a practical response to the evolution of computer hardware. Within high performance computing, increases in computing power have focused on increased core count, with Exascale computing likely to accelerate this trend. The scaling of PDE-based simulations, such as CFD, to these large core counts is an open question. Currently, although strong scaling (i.e. reducing time to solution as resources expand) is no longer typical of CFD on large scale simulations, weak-scaling (i.e. improving solution quality without an increase in time as resources expand) is relied on to utilize these resources. For practical engineering systems, diminishing returns in terms of solution quality will eventually occur. At this point, utilizing additional CPU resources by increasing solution accuracy is no longer efficient and this additional computational power will need to be put to better use. Hence in the limit that both strong and weak scaling break down, additional productive uses for parallel resources will need to be determined. One use for these resources is uncertainty quantification, as it is particularly suited for parallelization due to the independent nature of the sampling often accompanying it. Although it remains unclear if and when, these scaling arguments may hold true for real-world CFD applications, the use of uncertainty quantification as a means of improving the parallel efficiency of simulation is worthy of consideration for the future.

Appendix A

Model Parameters for Real Gas Model

This appendix contains the parameters used with the solver. These parameters come from Reference [56]. The only modification to these parameters is to the specific heat parameters. Reference [56] specifies the specific heat at constant pressure while this work uses the specific heat at constant volume. To change from c_p to c_v, 1 is subtracted from the first coefficient A_1. Additionally, the enthalpy of formation $h_{s,o}$ is modified to an internal energy of formation $e_{s,o}$ using the formula:

$$e_{s,o} = h_{s,o} - \frac{\bar{R}}{M_s} T_{ref} \tag{A.1}$$

where $T_{ref} = 298.16K$.

Table A.1: Species Properties for Air including reference energy at $T_{ref} = 298.16K$ and Millikan and White coupling parameter

	Molar Mass (kg/kmol)	$e_{o,s}$ (J/kg)	A_s
O_2	32	-7.7465E+004	129
NO	30	2.9300E+006	168
N	14	3.3579E+007	0
O	16	1.5416E+007	0
N_2	28	-8.8532E+004	220

Table A.2: Curve Fits for Specific Heat at Constant Volume

Species	Range A	A_1	A_2	A_3	A_4	A_5	A_6
O_2	1	2.625598E+000	-1.878218E-003	7.055454E-006	-6.763513E-009	2.155599E-012	-1.047520E+003
	2	2.621953E+000	7.361826E-004	-1.965222E-007	3.620155E-011	-2.894562E-015	-1.201980E+003
	3	2.721E+000	4.254E-004	-2.835E-008	6.050E-013	-5.186E-018	-1.044E+003
	4	2.486660E+000	5.238420E-004	-3.912340E-008	1.009350E-012	-8.871830E-018	-1.044000E+003
	5	2.961980E+000	3.944550E-004	-2.950580E-008	7.397450E-013	-6.420930E-018	-1.044000E+003
NO	1	3.045952E+000	-3.418178E-003	7.981919E-006	-6.113931E-009	1.591907E-012	9.745390E+003
	2	2.189000E+000	1.338228E-003	-5.289932E-007	9.591933E-011	-6.484793E-015	9.828330E+003
	3	2.845E+000	2.521E-004	-2.658E-008	2.162E-012	-6.381E-017	9.764E+003
	4	3.330870E+000	-5.808630E-005	2.805950E-008	-1.569410E-012	2.410390E-017	9.764000E+003
	5	1.350750E+000	5.864300E-004	-3.131650E-008	6.049510E-013	-4.055670E-018	9.764000E+003
N	1	1.503071E+000	-2.180018E-005	5.420528E-008	-5.647560E-011	2.099904E-014	5.609890E+004
	2	1.450268E+000	1.066145E-004	-7.465337E-008	1.879652E-011	-1.025983E-015	5.611600E+004
	3	1.748E+000	-3.909E-004	1.338E-007	-1.191E-011	3.369E-016	5.609E+004
	4	-2.227990E+000	1.926850E-003	-2.437050E-007	1.219300E-011	-1.991840E-016	5.609000E+004
	5	1.452020E+001	-3.885790E-003	3.228840E-007	-9.605270E-012	9.547220E-017	5.609000E+004
O	1	1.946428E+000	-1.638166E-003	2.421031E-006	-1.602843E-009	3.890696E-013	2.914760E+004
	2	1.542059E+000	-2.755061E-005	-3.102803E-009	4.551067E-012	-4.368051E-016	2.923080E+004
	3	1.546E+000	-5.952E-005	2.701E-008	-2.798E-012	9.380E-017	2.915E+004
	4	-1.009787E+000	1.244970E-003	-1.615440E-007	8.037990E-012	-1.262400E-016	2.915000E+004
	5	1.542810E+001	-3.931300E-003	2.983990E-007	-8.161280E-012	7.500430E-017	2.915000E+004
O_2	1	2.674826E+000	-1.208150E-003	2.324010E-006	-6.321755E-010	-2.257725E-013	-1.061160E+003
	2	1.896319E+000	1.515486E-003	-5.723527E-007	9.980739E-011	-6.522355E-015	-9.058620E+002
	3	2.727E+000	4.684E-004	-1.140E-007	1.154E-011	-3.293E-016	-1.043E+003
	4	8.637690E+000	-2.572840E-003	3.301980E-007	-1.431490E-011	2.033260E-016	-1.043000E+003
	5	-6.168080E+000	2.333690E-003	-1.295340E-007	2.787210E-012	-2.135960E-017	-1.043000E+003

Table A.3: Definition of Temperature Ranges for Specific Heat Coefficients

Range	Lower (K)	Upper (K)
1	300	1000
2	1000	6000
3	6000	15000
4	15000	25000
5	25000	35000

Table A.4: Collision Integrals for 5-Species Air Model

Pairs		K=1		K=2	
s	r	2000 K	4000 K	2000 K	4000 K
O_2	O_2	-14.60	-14.64	-14.54	-14.57
O_2	NO	-14.59	-14.63	-14.52	-14.56
O_2	N	-14.58	-14.64	-14.52	-14.56
O_2	O	-14.66	-14.74	-14.59	-14.66
O_2	N_2	-14.66	-14.75	-14.67	-14.66
NO	NO	-14.08	-14.11	-14.74	-14.82
NO	N	-14.69	-14.76	-14.62	-14.69
NO	O	-14.66	-14.74	-14.59	-14.66
NO	N_2	-14.76	-14.86	-14.69	-14.80
N	N	-14.11	-14.14	-14.71	-14.79
N	O	-14.58	-14.63	-14.51	-14.54
N	N_2	-14.57	-14.64	-14.51	-14.56
O	O	-14.67	-14.75	-14.59	-14.66
O	N_2	-14.63	-14.72	-14.55	-14.64
N_2	N_2	-14.56	-14.65	-14.50	-14.58

Table A.5: Arrhenius Relations for Park Chemical Kinetics Model

Reaction	$C_{f,r}$	$\eta_{f,r}$	$E_{f,r}/k$	B_1^r	B_2^r	B_3^r	B_4^r	B_5^r
$N_2 + O_2 \leftrightarrows 2N + O_2$	3.700E+21	-1.6	1.132E+5	1.858	-1.325	-9.856	-0.174	0.008
$N_2 + NO \leftrightarrows 2N + NO$	4.980E+21	-1.6	1.132E+5	1.858	-1.325	-9.856	-0.174	0.008
$N_2 + N \leftrightarrows 2N + N$	1.600E+22	-1.6	1.132E+5	1.858	-1.325	-9.856	-0.174	0.008
$N_2 + O \leftrightarrows 2N + O$	4.980E+22	-1.6	1.132E+5	1.858	-1.325	-9.856	-0.174	0.008
$N_2 + N_2 \leftrightarrows 2N + N_2$	3.700E+21	-1.6	1.132E+5	1.858	-1.325	-9.856	-0.174	0.008
$O_2 + O_2 \leftrightarrows 2O + O_2$	9.680E+22	-2.0	5.975E+4	2.855	0.988	-6.181	-0.023	-0.001
$O_2 + NO \leftrightarrows 2O + NO$	9.680E+22	-2.0	5.975E+4	2.855	0.988	-6.181	-0.023	-0.001
$O_2 + N \leftrightarrows 2O + N$	2.900E+23	-2.0	5.975E+4	2.855	0.988	-6.181	-0.023	-0.001
$O_2 + O \leftrightarrows 2O + O$	2.900E+23	-2.0	5.975E+4	2.855	0.988	-6.181	-0.023	-0.001
$O_2 + N_2 \leftrightarrows 2O + N_2$	9.680E+22	-2.0	5.975E+4	2.855	0.988	-6.181	-0.023	-0.001
$NO + O_2 \leftrightarrows N + O + O_2$	7.950E+23	-2.0	7.550E+4	0.792	-0.492	-6.761	-0.091	-0.004
$NO + NO \leftrightarrows N + O + NO$	7.950E+23	-2.0	7.550E+4	0.792	-0.492	-6.761	-0.091	-0.004
$NO + N \leftrightarrows N + O + N$	7.950E+23	-2.0	7.550E+4	0.792	-0.492	-6.761	-0.091	-0.004
$NO + O \leftrightarrows N + O + O$	7.950E+23	-2.0	7.550E+4	0.792	-0.492	-6.761	-0.091	-0.004
$NO + N_2 \leftrightarrows N + O + N_2$	7.950E+23	-2.0	7.550E+4	0.792	-0.492	-6.761	-0.091	-0.004
$N_2 + O \leftrightarrows NO + N$	6.440E+17	-1.0	3.837E+4	1.066	-0.833	-3.095	-0.084	0.004
$NO + O \leftrightarrows O_2 + N$	8.370E+12	0.0	1.940E+4	-2.063	-1.48	-0.58	-0.114	0.005

Table A.6: Arrhenius Relations for Dunn-Kang Chemical Kinetics Model

Reaction	$C_{f,r}$	$\eta_{f,r}$	$E_{f,r}/k$	$C_{b,r}$	$\eta_{b,r}$	$E_{b,r}/k$
$N_2 + O_2 \leftrightarrows 2N + O_2$	1.900E+017	-0.5	1.130E+005	1.100E+016	-0.5	0.000E+000
$N_2 + NO \leftrightarrows 2N + NO$	1.900E+017	-0.5	1.130E+005	1.100E+016	-0.5	0.000E+000
$N_2 + N \leftrightarrows 2N + N$	4.085E+022	-1.5	1.130E+005	2.270E+021	-1.5	0.000E+000
$N_2 + O \leftrightarrows 2N + O$	1.900E+017	-0.5	1.130E+005	1.100E+016	-0.5	0.000E+000
$N_2 + N_2 \leftrightarrows 2N + N_2$	4.700E+017	-0.5	1.130E+005	2.720E+016	-0.5	0.000E+000
$O_2 + O_2 \leftrightarrows 2O + O_2$	3.240E+019	-1.0	5.950E+004	2.700E+016	-0.5	0.000E+000
$O_2 + NO \leftrightarrows 2O + NO$	3.600E+018	-1.0	5.950E+004	3.000E+015	-0.5	0.000E+000
$O_2 + N \leftrightarrows 2O + N$	3.600E+018	-1.0	5.950E+004	3.000E+015	-0.5	0.000E+000
$O_2 + O \leftrightarrows 2O + O$	9.000E+019	-1.0	5.950E+004	7.500E+015	-0.5	0.000E+000
$O_2 + N_2 \leftrightarrows 2O + N_2$	7.200E+018	-1.0	5.950E+004	6.000E+014	-0.5	0.000E+000
$NO + O_2 \leftrightarrows N + O + O_2$	3.900E+020	-1.5	7.550E+004	1.000E+020	-1.5	0.000E+000
$NO + NO \leftrightarrows N + O + NO$	7.800E+021	-1.5	7.550E+004	2.000E+021	-1.5	0.000E+000
$NO + N \leftrightarrows N + O + N$	7.800E+021	-1.5	7.550E+004	2.000E+021	-1.5	0.000E+000
$NO + O \leftrightarrows N + O + O$	7.800E+021	-1.5	7.550E+004	2.000E+021	-1.5	0.000E+000
$NO + N_2 \leftrightarrows N + O + N_2$	3.900E+020	-1.5	7.550E+004	1.000E+020	-1.5	0.000E+000
$N_2 + O \leftrightarrows NO + N$	7.000E+013	0.0	3.800E+004	1.560E+013	0.0	0.000E+000
$NO + O \leftrightarrows O_2 + N$	3.200E+009	1.0	1.970E+004	1.300E+010	1.0	3.580E+003

Appendix B

Overloaded Complex Functions

In order to perform complex differentiation, the variables within the code are defined as complex numbers. The real part of the complex variables represents the result of the analysis code while the imaginary part represents the derivative of the code. For the majority of intrinsic functions, the complex versions have the desired behavior. For comparative operators however, the behavior of the operator must be overloaded to give the proper complex behavior for differentiation. The overloaded functions required for complexifying the code used in this work are given in the Fortran 90 module below. In order to use these functions, the module is included in each subroutine of the code and the function names are modified to the new names. For example, all references to *abs* are converted to *myabs*. This process can be performed either by hand or using shell scripting. Although this module does not consist of all the operators that must be overloaded, it does represent the operators encountered in the calculation of the residual used in this work.

```
module complexfuncs
  implicit none
  complex(8), parameter :: null=cmplx(0.D0,0.D0)
contains
```

```
function mymax(x,y)
  complex(8), intent(in) :: X,Y
  complex(8) mymax

  if (real(x)>=real(y)) then
     mymax=X
  else
     mymax=Y
  end if

  return
end function mymax

function mymin(x,y)
  complex(8), intent(in) :: X,Y
  complex(8) mymin

  if (real(x)<=real(y)) then
     mymin=X
  else
     mymin=Y
  end if

  return
end function mymin
```

```
  function mysign(scale,X)
    real(8), intent(in) :: scale
    complex(8), intent(in) :: X
    complex(8) mysign

    if (real(x)>=0.D0) then
       mysign=scale
    else
       mysign=-scale
    end if

    return
  end function mysign

  function myabs(X)
    complex(8), intent(in) :: X
    complex(8) myabs
    real(8) a, b

    a=real(x)
    b=Aimag(x)
    if (a>=0.D0) then
       myabs=X
    else
       myabs=cmplx(-a,-b)
    end if

    return
  end function myabs
end module complexfuncs
```

Appendix C

Comparison of Adjoint to Finite-Difference

In this appendix, the comparison of the derivatives produced by the adjoint approach is compared with those from finite difference. The step size for each variable was set as $\varepsilon = 1\times10^{-3}\bar{x}$ where $\bar{x}$ was the mean value for each parameter. For the case of reaction parameters, because the mean parameter value was zero, the step size was equal to $\varepsilon = 1\times10^{-3}$. Although results are given for a single step size, these results were typical for a range of step sizes. The finite difference results are divided by parameter type in the following tables.

Table C.1: Derivative Values for Freestream variables compared with Finite difference

Number	Variable	Adjoint	Finite Difference
1	ρ_∞	-4.8870323E+000	-4.8868409E+000
2	V_∞	7.1690751E-007	7.6010671E-007

Table C.2: Derivative Values for Collision Integrals compared with Finite difference

Number	Interaction	k	Adjoint	Finite Difference
3	O2-O2	1	-1.0840317E-004	-1.0840420E-004
4	O2-NO	1	-6.9513883E-005	-6.9507882E-005
5	O2-N	1	1.0791157E-004	1.0791818E-004
6	O2-O	1	2.0758635E-004	2.0761970E-004
7	O2-N2	1	-1.1630498E-003	-1.1630843E-003
8	NO-NO	1	-9.3887081E-006	-9.3829684E-006
9	NO-N	1	-5.4009297E-006	-5.4004388E-006
10	NO-O	1	-9.0939378E-005	-9.0936809E-005
11	NO-N2	1	-5.2659642E-004	-5.2659199E-004
12	N-N	1	0.0000000E+000	0.0000000E+000
13	N-O	1	-1.2973699E-004	-1.2974068E-004
14	N-N2	1	2.7172604E-005	2.7169776E-005
15	O-O	1	0.0000000E+000	0.0000000E+000
16	O-N2	1	-1.9250337E-004	-1.9250942E-004
17	N2-N2	1	-9.6708053E-004	-9.6709112E-004
18	O2-O2	2	1.0715441E-005	1.0755055E-005
19	O2-NO	2	-4.9572368E-006	-4.9623971E-006
20	O2-N	2	-7.3993089E-008	-7.4365900E-008
21	O2-O	2	-8.4361474E-005	-8.4372518E-005
22	O2-N2	2	-1.5126972E-004	-1.5108523E-004
23	NO-NO	2	-5.5241359E-007	-5.5394992E-007
24	NO-N	2	-1.9608136E-007	-1.9586574E-007
25	NO-O	2	-1.8534312E-005	-1.8537716E-005
26	NO-N2	2	-4.6173824E-005	-4.6205203E-005
27	N-N	2	-3.4571720E-007	-3.4542269E-007
28	N-O	2	-5.8964621E-006	-5.8934939E-006
29	N-N2	2	-1.1119636E-005	-1.1113398E-005
30	O-O	2	-1.3461675E-004	-1.3461919E-004
31	O-N2	2	-8.6974458E-004	-8.6981023E-004
32	N2-N2	2	-9.4976566E-004	-9.4959133E-004

Table C.3: Derivative Values for Reaction Rate Parameters (Dunn-Kang Model) compared with Finite difference

Number	Reaction	Direction	Adjoint	Finite Difference
33	$N_2 + 0_2 \leftrightarrows 2N + O_2$	f	-3.1778578E-009	-3.2107095E-009
34	$N_2 + NO \leftrightarrows 2N + NO$	f	-3.0800223E-008	-3.0349421E-008
35	$N_2 + N \leftrightarrows 2N + N$	f	-1.0183336E-005	-1.0122142E-005
36	$N_2 + 0 \leftrightarrows 2N + O$	f	-1.2051486E-006	-1.1960762E-006
37	$N_2 + N_2 \leftrightarrows 2N + N_2$	f	-4.6736098E-006	-4.6388180E-006
38	$O_2 + 0_2 \leftrightarrows 2O + O_2$	f	-1.2969275E-005	-1.3430814E-005
39	$O_2 + NO \leftrightarrows 2O + NO$	f	-2.0709647E-006	-2.0641845E-006
40	$O_2 + N \leftrightarrows 2O + N$	f	-1.8455302E-006	-1.8303218E-006
41	$O_2 + 0 \leftrightarrows 2O + O$	f	-6.2119821E-004	-6.1994923E-004
42	$O_2 + N_2 \leftrightarrows 2O + N_2$	f	-9.8581943E-005	-9.8685557E-005
43	$NO + 0_2 \leftrightarrows N + O + O_2$	f	-1.1568841E-007	-1.1456310E-007
44	$NO + NO \leftrightarrows N + O + NO$	f	-9.5627449E-006	-9.3710407E-006
45	$NO + N \leftrightarrows N + O + N$	f	-3.9708607E-005	-3.9435009E-005
46	$NO + 0 \leftrightarrows N + O + O$	f	-2.3941186E-004	-2.3801602E-004
47	$NO + N_2 \leftrightarrows N + O + N_2$	f	-1.9681584E-005	-1.9518035E-005
48	$N_2 + O \leftrightarrows NO + N$	f	-3.9861457E-004	-3.9666744E-004
49	$NO + O \leftrightarrows O_2 + N$	f	-2.2109266E-004	-2.2023820E-004
50	$N_2 + 0_2 \leftrightarrows 2N + O_2$	b	1.8752425E-009	0.0000000E+000
51	$N_2 + NO \leftrightarrows 2N + NO$	b	3.3659984E-008	3.3543477E-008
52	$N_2 + N \leftrightarrows 2N + N$	b	1.7208950E-005	1.7169968E-005
53	$N_2 + 0 \leftrightarrows 2N + O$	b	2.0205526E-006	2.0160363E-006
54	$N_2 + N_2 \leftrightarrows 2N + N_2$	b	7.7651682E-006	7.7481466E-006
55	$O_2 + 0_2 \leftrightarrows 2O + O_2$	b	1.7013675E-004	1.7013272E-004
56	$O_2 + NO \leftrightarrows 2O + NO$	b	2.7374531E-006	2.7368669E-006
57	$O_2 + N \leftrightarrows 2O + N$	b	2.9400559E-006	2.9351427E-006
58	$O_2 + 0 \leftrightarrows 2O + O$	b	1.1181845E-004	1.1177612E-004
59	$O_2 + N_2 \leftrightarrows 2O + N_2$	b	2.8210339E-005	2.8204919E-005
60	$NO + 0_2 \leftrightarrows N + O + O_2$	b	6.5485560E-008	6.5437740E-008
61	$NO + NO \leftrightarrows N + O + NO$	b	7.9618924E-006	7.9438647E-006
62	$NO + N \leftrightarrows N + O + N$	b	7.4729759E-005	7.4649275E-005
63	$NO + 0 \leftrightarrows N + O + O$	b	4.1446892E-004	4.1451404E-004
64	$NO + N_2 \leftrightarrows N + O + N_2$	b	3.3079241E-005	3.3033146E-005
65	$N_2 + O \leftrightarrows NO + N$	b	3.8755479E-004	3.8538363E-004
66	$NO + O \leftrightarrows O_2 + N$	b	1.9566870E-004	1.9512396E-004

Appendix D

Fay-Riddell Heating Correlation

In this appendix, the process used to solve for the Fay-Riddell stagnation heating correlation is given. This process consists of solving a normal shock equation using equilibrium gas curve fits. Based on the state after the shock, the composition of the gas is then solved for using statistical thermodynamics. With the properties at the edge of the boundary layer determined, the correlation can be calculated directly.

The first step of calculating the correlation is determining the thermodynamic properties after the shock. This state is computed using the Rankine-Hugoniot jump relations and curve fits for the enthalpy of the gas. The Rankine-Hugoniot shock relations are given as:

$$\rho_1 U_1 = \rho_2 U_2 \tag{D.1}$$

$$\rho_1 U_1^2 + P_1 = \rho_2 U_2^2 + P_2 \tag{D.2}$$

$$h_1(\rho_1, P_1) + \frac{1}{2}U_1^2 = h_2(\rho_2, P_2) + \frac{1}{2}U_2^2 \tag{D.3}$$

where the subscript 1 indicate the flow properties upstream of the shock and 2 indicate the properties downstream of the shock. Because the curve fits for enthalpy account for dissociation, the gas is no longer ideal and the enthalpy becomes a function of two thermodynamic variables, namely density and pressure. To determine the state downstream of the shock, the density ρ_2, velocity U_2, and pressure P_2 must be determined. As there are three shock relations and three unknowns, the state can be determined by using a root finder, such as bisection, secant or Newton's method. Because the jump relations can be rearrange into explicit expressions for the density and pressure downstream of the shock, Newton's method only needs to be applied to the energy equation. Parameterizing the problem using the density, the nonlinear equations reduce to those below.

$$U_2(\rho_2) = \frac{\rho_1 U_1}{\rho_2} \tag{D.4}$$

$$P_2(\rho_2) = P_1 + \rho_1 U_1^2 \left(1 - \frac{\rho_1}{\rho_2}\right) \tag{D.5}$$

$$R(\rho_2) = h_2(\rho_2, P_2(\rho_2)) - h_1(\rho_1, P_1) + \frac{1}{2}U_1^2 \left[\left(\frac{\rho_1}{\rho_2}\right)^2 - 1\right] \tag{D.6}$$

With the equations cast in this form, $R(\rho_2)$ is driven to zero by varying ρ_2 using a root finder. With two variables downstream of the shock determined, all other properties can be found using curve fits. These curve fits are found in Reference [52].

With the thermodynamic state after the shock determined, the composition after the shock must also be found. Based on the state after the shock, the equilibrium constants for the dissociation reactions are first determined based on expressions for the partition function. With the equilibrium constants determined, the partial pressure of each species can be determined by solving a system of nonlinear equations.

The partition function for each species is first evaluated using the temperature after the shock. Each energy mode in the atom contributes to the partition function. As equilibrium is assumed in this problem, the partition functions are evaluated at a single temperature. The total partition function is the product of the partition function for each energy mode [53].

$$Q_s = Q_{t,s} Q_{r,s} Q_{v,s} Q_{el,s} \tag{D.7}$$

The partition function for the translational energy, Q_t is given as [53]:

$$Q_{t,s} = \left(\frac{2\pi m_s k_b T}{h^2}\right)^{3/2} \tag{D.8}$$

where m_s is the molecular mass of species s, k_b is Boltzmann's constant and h is Plank's constant. The rotational partition function is 1 for monatomic species and is given by equation D.9 for diatomic species [53].

$$Q_{r,s} = \frac{T}{\theta_{r,s}} \tag{D.9}$$

Here, θ_r is a characteristic rotational temperature for each species given in Table D.1. The vibrational partition function is again 1 for monatomic species and is given below for diatomic species [53].

$$Q_{v,s} = \frac{1}{1 - e^{-\theta_{v,s}/T}} \tag{D.10}$$

Here, θ_v is the characteristic vibrational temperature of each species found in Table D.1. The electronic partition function is approximated by including only one or two modes above the ground state for each species. The partition function for electronic energy is given by [53]:

$$Q_{el,s} = g_{o,s} + g_{1,s} e^{-\theta^1_{el,s}/T} + g_{2,s} e^{-\theta^2_{el,s}/T} + ... \tag{D.11}$$

where $g_{o,s}$, $g_{1,s}$ and $g_{2,s}$ are the degeneracy at the ground state, first mode and second mode(given in Table D.1), and $\theta^1_{el,s}$ and $\theta^2_{el,s}$ are the characteristic temperatures for the first and second electronic energy modes, given in Table D.1.

Using this partition function, the equilibrium constant for each dissociation reaction can be evaluated. Three dissociation reactions are possible for the 5 species model used for these calculations. The expressions for these equilibrium constants are given below.

$$K_{O_2 \leftrightarrows 2O} = k_b T \frac{Q_O^2}{Q_{O_2}} e^{-\theta_{d,O_2}} \tag{D.12}$$

$$K_{N_2 \leftrightarrows 2N} = k_b T \frac{Q_N^2}{Q_{N_2}} e^{-\theta_{d,N_2}} \tag{D.13}$$

$$K_{NO \leftrightarrows N+O} = k_b T \frac{Q_N Q_O}{Q_{NO}} e^{-\theta_{d,NO}} \tag{D.14}$$

The dissociation temperature θ_d is given for each diatomic species in Table D.1. This temperature gives an indication of the temperature at which each species dissociates.

Using the equilibrium constants, the partial pressure of each species can be determined. For each reaction, the equilibrium constant is related to the partial pressure as [48]:

Table D.1: Heat of Formation, Characteristic Temperatures and degeneracy for each species

Species	θ_r (K)	θ_v (K)	θ_d (K)	$\Delta h^o_{f,s}$ (J/kg)
O_2	2.1	2,270	59,500	0
NO	2.5	2,740	75,500	2.996123×10^6
N	—	—	—	3.362161×10^7
O	—	—	—	1.543119×10^7
N_2	2.9	3,390	113,000	0

Species	θ^1_{el} (K)	θ^2_{el} (K)	g_o	g_1	g_2
O_2	11,390	—	3	2	—
NO	174	—	2	2	—
N	—	—	4	—	—
O	228	326	5	3	1
N_2	—	—	1	—	—

$$K_{O_2} = \frac{p_O^2}{p_{O_2}} \tag{D.15}$$

$$K_{N_2} = \frac{p_N^2}{p_{N_2}} \tag{D.16}$$

$$K_{NO} = \frac{p_N p_O}{p_{NO}} \tag{D.17}$$

Using these equations, five partial pressures must be determined. As the system is under-determined, two additional constraints must be determined. The first constraint is that the sum of the partial pressures must sum to the mixture pressure, which is known based on solving the normal shock problem. The second constraint is ensuring that the amount of each element is conserved. As there are only two elements in this model, this constraint is enforced by ensuring the ratio of one element to the other is fixed. These extra constraints are given as:

$$P = \sum_s p_s \tag{D.18}$$

$$2p_{O_2} + p_O + p_{NO} = \chi \left(2p_{N_2} + p_N + p_{NO}\right) \tag{D.19}$$

where χ represents the ratio of elemental oxygen to elemental nitrogen in air. This ratio is set to 0.25 for this work. Using these two constraints, the system is fully determined with the five partial pressures corresponding to five equations. As these equations are nonlinear, the partial pressures are solved using Newton's method. With each partial pressure determined, the mass fraction of each species is determined by inverting the partial pressure equation.

$$c_s = p_s \frac{m_s}{\rho k_b T} \tag{D.20}$$

With the thermodynamic state and composition after the shock determined, the Fay-Riddell heating correlation can be evaluated. The equation for the stagnation heat flux is given as [54]:

$$q" = 0.76(Pr_w)^{-0.6}(\rho_w \mu_w)^{0.1}(\rho_e \mu_e)^{0.4}\sqrt{\left(\frac{dU_e}{dx}\right)}(h_{o,e} - h_w)\left[1 + (Le^{0.52} - 1)\left(\frac{h_D}{h_{o,e}}\right)\right] \tag{D.21}$$

The properties at the edge of the boundary layer are assumed to be the state after the shock. The wall properties are given by the boundary conditions for the problem. In the case of the $5km/s$ cylinder, the temperature and composition at the wall are assumed fixed. The viscosity of the wall μ_w and viscosity at the edge of the boundary layer are evaluated using the transport model specified in Chapter 2. The velocity gradient in the boundary layer is approximated using the pressure at the edge of the boundary layer and the radius of curvature ($R_N = 1m$ for the $5km/s$ cylinder case).

$$\left(\frac{dU_e}{dx}\right) = \frac{1}{R_N}\sqrt{2\frac{p_e - p_\infty}{\rho_e}} \tag{D.22}$$

The total enthalpy $h_{o,e}$ is calculated based on the post-shock state and h_w is calculated based on the temperature and pressure at the wall. The total enthalpy calculation is straightforward:

$$h_{o,e} = h(\rho_e, P_e) + \frac{1}{2}U_e^2 \tag{D.23}$$

For the enthalpy at the wall, the density at the wall must be approximated. To calculate the density, the pressure at the edge of the boundary layer is used to approximate the pressure at the wall and the density is calculated using the state equation and the prescribed temperature at the wall.

$$P_w \approx P_e \tag{D.24}$$

$$\rho_w = \frac{P_w}{RT_w} \tag{D.25}$$

$$h_w = h(\rho_w, P_w) \tag{D.26}$$

The gas constant for the mixture,R is given as:

$$R = \sum_s c_{s,w} \frac{\bar{R}}{M_s} \tag{D.27}$$

where $c_{s,w}$ is the gas composition at the wall (same as the freestream composition for the super-catalytic wall considered in $5km/s$ cylinder). The heat of formation for the mixture, h_D, at the edge of the boundary layer is given below.

$$h_D = \sum_s c_{s,e} \Delta h^o_{f,s} \tag{D.28}$$

The heat of formation for each species is given in Table D.1. Finally, the Prandtl number and Lewis number are calculated using the flow properties.

$$Pr_w = \frac{c_{p,w}\mu_w}{k_w} \tag{D.29}$$

$$Le_w = \frac{Pr_w}{Sc} \tag{D.30}$$

Because these numbers are evaluated at the wall, the specific heat can be assumed constant as $c_{p,w} = 7/2R$ and the Schmidt number, Sc, is assumed to be 0.5 [51]. With all the terms specified, the stagnation heating can be calculated using the correlation in equation (D.21).

References

[1] S. M. Murman, M. J. Aftosmis, and M. Nemec, “Automated parameter studies using a cartesian method,” Tech. Rep., National Aeronautics and Space Administration, Tech. Rep. NA-04-015, 2004.

[2] M. J. Wright, D. Bose, and Y.-K. Chen, “Probabilistic modeling of aerothermal and thermal protection material response uncertainties,” AIAA *Journal*, vol. 45, no. 2, pp. 399–425, Feb. 2007.

[3] R. Hicks and P. Henne, “Wing Design by Numerical Optimization,” *Journal of Aircraft*, vol. 15 No. 7, pp. 407 – 412, 1978.

[4] O. Pironneau, “On Optimum Design in Fluid Mechanics,” *Journal of Fluid Mechanics*, vol. 64, No. 1, pp. 97–110, 1974.

[5] A. Jameson, “Aerodynamic Design via Control Theorgy,” *Journal of Scientific Computing*, vol. 3, no. 3, pp. 233–260, 1998.

[6] W. K. Anderson and D. L. Bonhaus, “Airfoil Design on Unstructured Grids for Turbulent Flows,” AIAA *Journal*, vol. 37, no. 2, pp. 185–191, 1999.

[7] M. B. Giles and N. A. Pierce, “An Introduction to the Adjoint Approach to Design,” *Flow, Turbulence and Combustion*, vol. 65(3-4), pp. 393–415, 2000.

[8] J. Elliott and J. Peraire, “Practical 3D Aerodynamic Design and Optimization using Unstructured Meshes,” AIAA *Journal*, vol. 35, no. 9, pp. 1479–1485, 1997.

[9] M. Giles, D. Ghate, and M. Duta, “Using automatic differentiation for adjoint CFD code development,” in *Recent Trends in Aerospace Design and Optimization*, B. Uthup, S. Koruthu, R. Sharma, and P. Priyadarshi, Eds. Tata McGraw-Hill, New Delhi, 2006, pp. 426–434.

[10] K. Eriksson, D. Estep, C. Johnson, and P. Hansbo, *Computational Differential Equations.* Lund, Sweden: Studentlitteratur, 1996.

[11] D. G. Cacuci, *Sensitivity and Uncertainty Analysis.* Boca Raton, Florida: CRC Press, 2003, vol. 1.

[12] G. E. Palmer, "Uncertainty analysis of cev leo and lunar return entries," in *39th* AIAA *Thermophysics Conference*, Miami, FL, June 2007, AIAA Paper, 2007-4253.

[13] R. A. T. Bil Kleb and C. O. Johnston, "Blurring the inputs: A natural language approach to sensitivity analysis," in *18th* AIAA *Computational Fluid Dynamics Conference*, Miami, FL, June 2007, AIAA Paper, 2007-4206.

[14] B. R. Bettis and S. Hosder, "Uncertainty quantification in hypersonic reentry flows due to aleatory and epistemic uncertainties," in *49th* AIAA *Aerospace Sciences Meeting and Exhibit*, Orlando, FL, January 2011, AIAA Paper, 2011-252.

[15] C. J. Roy and W. L. Oberkampf, "A complete framework for verification, validation and uncertainty quantification in scientific computing," in *48th* AIAA *Aerospace Sciences Meeting and Exhibit*, Orlando, FL, January 2010, AIAA Paper, 2010-124.

[16] D. Ghate and M. Giles, "Inexpensive Monte Carlo uncertainty analysis," in *Recent Trends in Aerospace Design and Optimization*, B. Uthup, S. Koruthu, R. Sharma, and P. Priyadarshi, Eds. Tata McGraw-Hill, New Delhi, 2006, pp. 203–210.

[17] D. P. Ghate and M. B. Giles, "Efficient hessian calculation using automatic differentiation," in *25th* AIAA *Applied Aerodynamics Conference*, Miami, FL, June 2007, AIAA Paper, 2007-4059.

[18] A. Alexeenko, A. Weaver, R. Greendyke, and J. Camberos, "Flowfield uncertainty analysis for hypersonic cfd simulations," in *48th* AIAA *Aerospace Sciences Meeting and Exhibit*, Orlando, FL, January 2010, AIAA Paper, 2010-1180.

[19] O. Roderick, M. Anitescu, and P. Fischer, "Polynomial Regression Approaches Using Derivative Information for Uncertainty Quantification," *Nuclear Science and Engineering*, vol. 164, no. 2, pp. 122–139, 2010.

[20] S. Hosder and R. W. Walters, "Non-intrusive polynomial chaos methods for uncertainty quantification in fluid dynamics," in *48th* AIAA *Aerospace Sciences Meeting*, Orlando, IL, January 2010, AIAA Paper, 2010-129.

[21] S. R. Gunn, "Support vector machines for classification and regression," Tech. Rep., University of Southampton, Tech. Rep., 1998.

[22] D. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Systems*, vol. 2, pp. 321–355, 1988.

[23] N. Cressie, "The Origins of Kriging," *Mathematical Geology*, vol. 22, no. 3, pp. 239–252, 1990.

[24] J. R. Koehler and A. B. Owen, "Computer Experiments," in *Handbook of Statistics*. pp. 261-308, 1996.

[25] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, vol. 13, pp. 455–492, 1998.

[26] T. W. Simpson, J. J. Korte, T. M. Mauery, and F. Mistree, "Comparison of response surface and kriging models for multidisciplinary design optimization," in *7th* AIAA*/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 1998, AIAA Paper, 98-4758.

[27] H. S. Chung and J. J. Alonso, "Using gradients to construct cokriging approximation models for high-dimensional design optimization problems," in *40th* AIAA *Aerospace Sciences Meeting and Exhibit*, Reno, NV, January 2002, AIAA Paper, 2002-0317.

[28] J. D. Martin and T. W. Simpson, "Use of Kriging Models to Approximate Deterministic Computer Models," AIAA *Journal*, vol. 43, no. 4, pp. 853–863,, 2005.

[29] S. Jeong, M. Murayama, and K. Yamamoto, "Efficient Optimization Design Method Using Kriging Model," *Journal of Aircraft*, vol. 42, no. No. 2, pp. 413–420, 2005.

[30] J. Peter and M. Marcelet, "Comparison of Surrogate Models for Turbomachinery Design," *WSEAS Transactions on Fluid Mechanics*, vol. 3, no. 1, pp. 10–17, 2008.

[31] J. Laurenceau and P. Sagaut, "Building Efficient Response Surfaces of Aerodynamic Functions with Kriging and Cokriging," AIAA *Journal*, vol. 46, no. 2, pp. 498–507, 2008.

[32] J. Laurenceau and M. Meaux, "Comparison of gradient and response surface based optimization frameworks using adjoint method," in *49th* AIAA*/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Schaumburg, IL, April 2008, AIAA Paper, 2008-1889.

[33] W. Yamazaki, M. P. Rumpfkeil, and D. J. Mavriplis, "Design optimization utilizing gradient/hessian enhanced surrogate model," in *40th Fluid Dynamics Conference and Exhibit*, Chicago, IL, June 2010, AIAA Paper, 2010-4363.

[34] B. A. Lockwood and M. Anitescu, "Gradient-enhanced universal kriging for uncertainty propagation in nuclear engineering," *Nuclear Science and Engineering*, vol. 170, no. 2, pp. 168–195, 2012.

[35] A. Jameson, "Optimum Aerodynamic Design Using Control Theory," in *Computational Fluid Dynamics Review*, O. K. Hafez, M., Ed. Wiley: New York, 1995, pp. 495–528.

[36] R. M. Errico, "What is an adjoint model ?" *Bulletin of the American Meteorological Society*, vol. 8, no. 11, pp. 2577–2591, 1997.

[37] W. L. Kleb and C. O. Johnston, "Uncertainty analysis of air radiation for lunar return shock layers," in AIAA *Atmospheric Flight Mechanics Conference and Exhibit*, Honolulu, HI, August 2008, AIAA Paper, 2008-6388.

[38] R. L. Iman, *Latin Hypercube Sampling.* John Wiley & Sons, Ltd, 2008. [Online]. Available: http://dx.doi.org/10.1002/9780470061596.risk0299

[39] J. C. Helton, J. D. Johnson, W. L. Oberkampf, and C. J. Sallaberry, "Representation of Analysis Results Involving Aleatory and Epistemic Uncertainty," Sandia National Laboratories, Tech. Rep. SAND 2008-4379, 2008.

[40] R. R. Yager and L. Liu, *Classic Works of the Dempster-Shafer Theory of Belief Functions. Studies in Fuzziness and Soft Computing Series.* Berlin: Springer, 2008, vol. 219.

[41] V. Kreinovich and S. Ferson, "A new cauchy-based black-box technique for uncertainty in risk analysis," in *Reliability Engineering and Systems Safety*, 2002, pp. 267–279.

[42] M. Pilch, T. G. Trucano, and J. C. Helton, "Ideas Underlying Quantification of Margins and Uncertainties (QMU): A white paper," Sandia National Laboratories, Tech. Rep. SAND2006-5001, 2006.

[43] M. S. Eldred and L. P. Swiler, "Efficient algorithms for mixed aleatory-epistemic uncertainty quantification with application to radiation-hardened electronics," Tech. Report., Sandia National Laboratories, Tech. Rep. SAND2009-5805, 2009.

[44] B. A. Lockwood, M. Anitescu, and D. J. Mavriplis, "Mixed aleatory/epistemic uncertainty quantification for hypersonic flows via gradient-based optimization and surrogate models," in *50th* AIAA *Aerospace Sciences Meeting and Exhibit*, Nashville, TN, January 2012, AIAA Paper, 2012-1254.

[45] *FUN3D: Fully Unstructured Navier-Stokes Manual*, NASA, May 2009, http://fun3d.larc.nasa.gov/index.html.

[46] S. Prudhomme, "Overview of the PECOS center," PSAAP Meeting - V & V UQ Workshop, Caltech, July 2009.

[47] M. Eldred, L. Swiler, and G. Tang, "Mixed aleatory-epistemic uncertainty quantification with stochastic expansions and optimization-based interval estimation," *Reliability Engineering & System Safety*, vol. 96, no. 9, pp. 1092–1113, 2011.

[48] J. D. Anderson, *Hypersonic and High-Temperature Gas Dynamics*, 2nd ed. Reston, VA: American Institute of Aeronautics and Astronautics, Inc., 2006.

[49] P. A. Thompson, *Compressible-Fluid Dynamics*, 3rd ed. Troy, N.Y.: Rensselaer Polytechnic Institute, 1988.

[50] D. R. Olynick, "A new lu-sgs flow solver for calculating reentry flows," Ph.D. dissertation, North Carolina State University, 1992.

[51] B. Hassan, "Thermo-chemical nonequilibrium effects on the aerothermodynamics of hypersonic vehicles," Ph.D. dissertation, North Carolina State University, Albuquerque, NM, December 1993.

[52] J. Tannehill and P. Mugge, “Improved curve fits for the thermodynamics properties of equilibrium air suitable for numerical computation using time-dependent or shock-capturing methods.” Tech. Rep., National Aeronautics and Space Administration, Tech. Rep. NASA CR-2470, 1974.

[53] W. G. Vincenti and C. H. Kruger, *Introduction to Physical Gas Dynamics.* Malabar, FL: Krieger Publishing Company, 1965.

[54] J. Fay and F. Riddell, “Theory of stagnation point heat transfer in dissociated air,” *Journal of Aeronautical Sciences*, vol. 25, no. 2, pp. 73–85, 1958.

[55] G. V. Candler, “The computation of weakly ionized hypersonic flow in thermo-chemical nonequilibrium,” Ph.D. dissertation, Stanford University, Albuquerque, NM, June 1993.

[56] P. A. Gnoffo, R. N. Gupta, and J. L. Shinn, “Conservation equations and physical models for hypersonic air flows in thermal and chemical nonequilibrium,” NASA, Tech. Rep., February 1989.

[57] J. H. Lee, “Basic governing equations for the flight regimes of aeroassisted orbital transfer vehicles,” in *Progress in Aeronautics and Astronautics*, ser. Progress in Aeronautics and Astronautics, H. Nelson, Ed. American Institute of Aeronautics and Astronautics, vol. 96, pp. 3–53.

[58] C. Park, *Non-equilibrium hypersonic aerothermodynamics.* John Wiley and Sons, New York, 1990.

[59] M. G. Dunn and S.-W. Kang, “Theoretical and experimental studies of reentry plasmas,” Tech. Rep., National Aeronautics and Space Administration, Tech. Rep. NASA CR-2232, 1973.

[60] R. Millikan and D. White, “Systematics of Vibrational Relaxation,” *Journal of Chemical Physics*, vol. 39, no. 12.

[61] B. R. Bettis, “Quantification of uncertainty in aerodynamics heating of a reentry vehicle due to uncertain wall and freestream conditions,” Master’s thesis, Missouri University of Science and Technology, 2010.

[62] M.-S. Liou, “A sequel to ausm, part ii: Ausm+-up for all speeds,” *Journal of Computational Physics*, vol. 214, pp. 137–170, 2006.

[63] J. R. Edwards, “A low-diffusion flux-splitting scheme for navier-stokes calculations,” *Computers & Fluids*, vol. 26, no. 6, pp. 635–659, 1997.

[64] D. J. Mavriplis, “Solution of the two-dimensional euler equations on unstructured triangular meshes,” Ph.D. dissertation, Princeton University, 1987.

[65] V. Venkatakrishnan, “Convergence to steady state solutions of the euler equations on unstructured grids with limiters,” *J. Comput. Phys.*, vol. 118, no. 1, pp. 120–130, 1995.

[66] J. P. M. Jonathan M. Weiss and W. A. Smith, "Implicit solution of preconditioned navier-stokes equations using algebraic multigrid," AIAA *Journal*, vol. 37, no. 1, pp. 29–36, January 1999.

[67] L. Wang and D. J. Mavriplis, "Implicit solution of the unsteady euler equations for high-order accurate discontinuous galerkin discretizations," *Journal of Computational Physics*, vol. 225, pp. 1994–2015, 2007.

[68] D. Hänel, R. Schwane, and G. Seider, "On the accuracy of upwind schemes for the solution of the navier-stokes equations," 1987, AIAA Paper, 87-1105.

[69] *The TAPENADE Tutorial*, INRIA Sophia-Antipolis, March 2009. [Online]. Available: http://www-sop.inria.fr/tropics/tapenade.html

[70] F. M. Cheatwood and P. A. Gnoffo, "User's manual for the langley aerothermodynamic upwind relaxation algorithm (laura)," NASA, Tech. Rep., April 1996.

[71] Y. Saad, *Iterative Methods for Sparse Linear Systems.* International Thomson Publishing Company, Boston, 1996.

[72] D. A. Knoll and D. E. Keyes, "Jacobian-free Newton-Krylov methods: a survey of approaches and applications," *Journal of Computational Physics*, vol. 193, pp. 357–397, 2004.

[73] A. Griewank, *Evaluating derivatives: principles and techniques of algorithmic differentiation.* Philadelphia: SIAM, 2000.

[74] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*, 3rd ed. Cambridge University Press, Sep. 2007.

[75] G. Candler, D. Mavriplis, and L. Trevino, "Current status and future prospects for the numerical simulation of hypersonic flows," AIAA Paper, 2009-0153, 2009.

[76] P. Gnoffo and J. A. White, "Computational aerothermodynamic simulation issues on unstructured grids," AIAA Paper, 2004-2371, 2004.

[77] P. Roe, "Approximate riemann solvers, parameter vectors, and difference schemes," *Journal of Computational Physics*, vol. 43, no. 2, pp. 357 – 372, 1981. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0021999181901285

[78] D. J. Mavriplis, "A discrete adjoint-based approach for optimization problems on three dimensional unstructured meshes," AIAA *Journal*, vol. 45, no. 4, pp. 741–750, Apr. 2007.

[79] L. Hascoët, "TAPENADE: A Tool for Automatic Differentiation of Programs," in *Proceedings of 4^{th} European Congress on Computational Methods, ECCOMAS'2004, Jyvaskyla, Finland*, 2004.

[80] M. B. Giles, M. B. Giles, J.-D. Mller, M. C. Duta, and M. C. Duta, “Adjoint code developments using the exact discrete approach,” *AIAA Paper*, vol. 2001, pp. 2001–2596, 2001.

[81] E. J. Nielsen, J. Lu, M. A. Park, and D. L. Darmofal, “An exact dual adjoint solution method for turbulent flows on unstructured grids,” *Computers & Fluids*, vol. 33, pp. 1131–1155, 2003.

[82] “Covariance,” Wikipedia, March 2012, http://en.wikipedia.org/wiki/Covariance, Last modified 30 March 2012.

[83] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning.* The MIT Press, 2006.

[84] Z.-H. Han, S. Görtz, and R. Zimmermann, “On improving efficiency and accuracy of variable-fidelity surrogate modeling in aero-data for loads context,” in *CEAS 2009 European Air and Space Conference*, Manchester, UK, October 2009.

[85] J. Nocedal and S. J. Wright, *Numerical Optimization.* New York: Springer-Verlag, 1999.

[86] E. Solak, R. Murray-Smith, W. Leithead, C. Rasmussen, and D. Leith, “Derivative observations in gaussian process models of dynamic systems,” 2003, http://books.nips.cc/papers/files/nips15/AA70.pdf.

[87] E. Lewis and W. Miller, *Computaional Methods of Neutron Transport.* American Nuclear Society, 1993.

[88] “Variance,” Wikipedia, March 2012, http://en.wikipedia.org/wiki/Variance, Last modified 30 March 2012.

[89] “Kurtosis,” Wikipedia, March 2012, http://en.wikipedia.org/wiki/Kurtosis, Last modified 30 March 2012.

[90] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, “L-BFGS-B: A Limited Memory FORTRAN Code for Solving Bound Constrained Optimization Problems,” Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, Illinois, USA, Tech. Rep. NAM-11, 1994.

[91] “Observable universe,” Wikipedia, March 2012, http://en.wikipedia.org/wiki/Observable _ Univserse, Last modified 30 March 2012.

[92] “Orders of magnitude (numbers),” Wikipedia, March 2012, http://en.wikipedia.org/wiki/Orders _ of _ magnitude _ (numbers), Last modified 20 March 2012.

[93] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, “A Limited Memory Algorithm for Bound Constrained Optimization,” *SIAM Journal on Scientific Computing*, vol. 16(5), pp. 1190–1208, 1995.

[94] R. Hartmann, J. Held, and T. Leicht, “Adjoint-based error estimation and adaptive mesh refinement for the RANS and k-ω turbulence model equations,” *J. Comput. Phys.*, vol. 230, no. 11, pp. 4268–4284, 2011.

[95] M. Rumpfkeil, “Optimization under uncertainty using gradients, hessians, and surrogate models,” in *50th* AIAA *Aerospace Sciences Meeting and Exhibit*, Nashville, TN, January 2012, AIAA Paper, 2012-149.

[96] M. P. Rumpfkeil and D. J. Mavriplis, “Efficient Hessian Calculations using Automatic Differentiation and the Adjoint Method with Applications,” AIAA *Journal*, vol. 48, no. 10, pp. 2406–2417, 2010.

[97] M. C. Kennedy and A. O’Hagan, “Predicting the output from a complex computer code when fast approximations are available,” *Biometrika*, vol. 87, pp. 1–13, 1998.

[98] W. Yamazaki and D. J. Mavriplis, “Derivative-enhanced variable fidelity surrogate modeling for aerodynamics functions,” in *49th* AIAA *Aerospace Sciences Meeting and Exhibit*, Orlando, FL, January 2011, AIAA Paper, 2011-1172.

[99] J. M. Blottner, F.G. and M. Ellis, “Chemically reacting gas viscous flow program for multi-component gas mixtures,” Sandia National Laboratories, Albuquerque, NM, Tech. Rep. Sandia Release No. SC-RR-70-754, December 1971.