

Copyright
by
James Stephen Strand
2012

The Dissertation Committee for James Stephen Strand Certifies that this is the approved version of the following dissertation:

Statistical Methods for the Analysis of DSMC Simulations of Hypersonic Shocks

Committee:

David Goldstein, Supervisor

Robert Moser

Philip Varghese

Ofodike Ezekoye

Ernesto Prudencio

**Statistical Methods for the Analysis of DSMC Simulations of
Hypersonic Shocks**

by

James Stephen Strand, B.S.AS.E.; M.S.E.

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

May 2012

Dedication

This dissertation is dedicated to my parents, Mark and Connie, and to my wife Cori.

Acknowledgements

I would like to gratefully acknowledge the help and support provided by my adviser and my committee, including the late Dr. Graham Carey. I would like to thank the Department of Energy for funding this work through the PSAAP program, and also for providing the necessary computational resources. Finally, I would like to thank Marco Panesi, Kenji Miki, Karl Schulz, Chris Simmons, and Gabriel Terejanu for their assistance along the way.

Statistical Methods for the Analysis of DSMC Simulations of Hypersonic Shocks

James Stephen Strand, Ph.D.

The University of Texas at Austin, 2012

Supervisor: David Goldstein

In this work, statistical techniques were employed to study the modeling of a hypersonic shock with the Direct Simulation Monte Carlo (DSMC) method, and to gain insight into how the model interacts with a set of physical parameters.

Direct Simulation Monte Carlo (DSMC) is a particle based method which is useful for simulating gas dynamics in rarefied and/or highly non-equilibrium flowfields. A DSMC code was written and optimized for use in this research. The code was developed with shock tube simulations in mind, and it includes a number of improvements which allow for the efficient simulation of 1D, hypersonic shocks. Most importantly, a moving sampling region is used to obtain an accurate steady shock profile from an unsteady, moving shock wave. The code is MPI parallel and an adaptive load balancing scheme ensures that the workload is distributed properly between processors over the course of a simulation.

Global, Monte Carlo based sensitivity analyses were performed in order to determine which of the parameters examined in this work most strongly affect the simulation results for two scenarios: a 0D relaxation from an initial high temperature

state and a hypersonic shock. The 0D relaxation scenario was included in order to examine whether, with appropriate initial conditions, it can be viewed in some regards as a substitute for the 1D shock in a statistical sensitivity analysis. In both analyses sensitivities were calculated based on both the square of the Pearson correlation coefficient and the mutual information. The quantity of interest (QoI) chosen for these analyses was the NO density profile. This vector QoI was broken into a set of scalar QoIs, each representing the density of NO at a specific point in time (for the relaxation) or a specific streamwise location (for the shock), and sensitivities were calculated for each scalar QoI based on both measures of sensitivity. The sensitivities were then integrated over the set of scalar QoIs to determine an overall sensitivity for each parameter. A weighting function was used in the integration in order to emphasize sensitivities in the region of greatest thermal and chemical non-equilibrium. The six parameters which most strongly affect the NO density profile were found to be the same for both scenarios, which provides justification for the claim that a 0D relaxation can in some situations be used as a substitute model for a hypersonic shock. These six parameters are the pre-exponential constants in the Arrhenius rate equations for the N_2 dissociation reaction $N_2 + N \rightleftharpoons 3N$, the O_2 dissociation reaction $O_2 + O \rightleftharpoons 3O$, the NO dissociation reactions $NO + N \rightleftharpoons 2N + O$ and $NO + O \rightleftharpoons N + 2O$, and the exchange reactions $N_2 + O \rightleftharpoons NO + N$ and $NO + O \rightleftharpoons O_2 + N$.

After identification of the most sensitive parameters, a synthetic data calibration was performed to demonstrate that the statistical inverse problem could be solved for the 0D relaxation scenario. The calibration was performed using the QUESO code, developed at the PECOS center at UT Austin, which employs the Delayed Rejection Adaptive Metropolis (DRAM) algorithm. The six parameters identified by the sensitivity analysis were calibrated successfully with respect to a group of synthetic datasets.

Table of Contents

List of Tables	xi
List of Figures	xiii
Nomenclature	xxvii
Chapter 1: Introduction	1
Motivation.....	1
Objectives	2
Literature Review.....	3
DSMC	3
Collision Model	4
Chemistry Model	5
Shock Simulations	6
Sensitivity Analysis for DSMC Simulations	7
Calibration of DSMC Parameters	7
Global Sensitivity Analysis.....	7
Sensitivities Based on r^2	8
The Mutual Information.....	10
MCMC	11
DRAM.....	12
Calibration of Parameters Relevant to Hypersonic Flows.....	14
Major Contributions of the Current Work	15
Dissertation Layout.....	17
Chapter 2: DSMC Methodology.....	18
Overview.....	18
Elastic Collisions	18
Inelastic Collisions.....	19
Chemical Reactions	21
1-D Shock Simulation Technique.....	24

Parallel Implementation	26
Verification	29
Proof of Necessity of the TCE Modification	30
Chapter 3: Sensitivity Analysis Methodology	35
Overview	35
Mutual Information	40
Verification	46
Sensitivity Analysis Examples	54
Chapter 4: Solving the Inverse Problem	69
Overview	69
MCMC Algorithm	69
Basic Metropolis-Hastings Algorithm	69
Delayed Rejection Adaptive Metropolis Algorithm	74
Inverse Problem Implementation	78
Example Calibrations	81
Chapter 5: 0D Relaxation Sensitivity Analysis	93
Overview	93
Scenario	93
Parameters	95
Quantity of Interest	102
Sampling the Parameter Space	105
Sensitivities vs. Time	107
Variance-Weighted Sensitivities	118
Convergence	123
Chapter 6: 1D Shock Sensitivity Analysis	129
Overview	129
Scenario	129
Parameters	130
Quantity of Interest	134

Sampling the Parameter Space.....	135
Sensitivities vs. X.....	136
Variance-Weighted Sensitivities.....	139
Convergence	144
Chapter 7: Synthetic Data Calibrations.....	150
Overview	150
Parameters.....	150
Scenario Selection.....	152
Scenario Details	155
Calibration Data.....	157
Likelihood Equation.....	161
Calibration Process	163
Calibration Results.....	165
Chapter 8: Conclusions and Future Work.....	177
Conclusions.....	177
Future Work.....	179
References.....	182

List of Tables

Table 2.1:	Collision parameters for 5-species air. VHS parameters were compiled by Ozawa (2008). C_1 and C_2 are based on those found in Bird (1994), but have been modified to account for the fact that Bird used different values for ω	19
Table 2.2:	Reactions for 5-species air. Nominal Arrhenius forward rate parameters are from Gupta <i>et al.</i> (1989), and backward rate parameters are based on a matching with the equilibrium constant over a range of temperatures.	24
Table 5.1:	The first seventeen parameters for the 0D relaxation sensitivity analysis. These parameters correspond to the reactions which are important for 5-species air. For each reaction, the parameter of interest is $\log_{10}\Lambda$, where Λ is the pre-exponential constant in the Arrhenius rate equation (Eq. 2.2) for the forward reaction. Nominal Arrhenius forward rate parameters are from Gupta <i>et al.</i> (1989), and backward rate parameters are based on a matching with the equilibrium constant over a range of temperatures. Throughout the sensitivity analysis, the ratio of forward to backward rate for a given reaction is kept constant, since these ratios should be fixed by the equilibrium constant.....	100

Table 5.2:	The remaining parameters for the 0D relaxation sensitivity analysis. Parameters 18-38 are collision parameters for 5-species air. Nominal values of d_{ref} were compiled by Ozawa (2008). Nominal values of C_1 (see Eq. 2.1) are based on those found in Bird (1994), but have been modified to account for the fact that Bird used different values for ω . The nominal value of 3 was considered to be a reasonable choice for Z_R for all three diatomic species (for this parameter the nominal value is not at the center of the uniform prior). The final parameter, F_{num} , is the ratio of real to simulated particles, and relates to the level of stochastic noise in the simulations.....	101
Table 6.1:	The parameters for the 1D shock sensitivity analysis. These parameters correspond to the reactions which are important for 5-species air. For each reaction, the parameter of interest is $\log_{10}\Lambda$, where Λ is the pre-exponential constant in the Arrhenius rate equation (Eq. 2.2) for the forward reaction. Nominal Arrhenius forward rate parameters are from Gupta <i>et al.</i> (1989), and backward rate parameters are based on a matching with the equilibrium constant over a range of temperatures. Throughout the sensitivity analysis, the ratio of forward to backward rate for a given reaction is kept constant, since these ratios should be fixed by the equilibrium constant.....	133
Table 7.1:	The six parameters chosen for calibration. The parameter numbers are kept consistent with those in Tables 5.1 and 6.1.....	151

List of Figures

- Figure 2.1: Schematic showing the initial setup of the domain, the boundary conditions, and the early motion of the shock.27
- Figure 2.2: Schematic showing the pressure sampling process, the identification of the shock location, the calculation of the shock speed, and the location and motion of the shock sampling region.28
- Figure 2.3. Instantaneous rates for VHS collisions and for selected reactions over a range of temperatures. Number densities of N_2 , N , O_2 , O , and NO were all set at $2.0 \times 10^{22} \text{ \#/m}^3$. The temperatures shown are total temperatures, and the diatomic species were initialized with their rotational and vibrational modes in equilibrium with the translational mode. Note the excellent agreement at all temperatures.31
- Figure 2.4. Instantaneous rates for selected reactions over a range of temperatures. Initial conditions are the same as in Fig. 2.3, but Λ for every reaction is ten times higher than the nominal value. At higher temperatures, the Arrhenius rates for the reactions shown are higher than the VHS rate for collisions between the reactants. The modified TCE model used in this work matches the Arrhenius rates in spite of this, but the traditional TCE model (which assumes that σ_{VHS} can be used to approximate σ_{Total}) is not able to match the Arrhenius rates in this situation.33
- Figure 2.5. Profiles of ρ_{NO} inside a shock at $\sim 8000 \text{ m/s}$. Note that the profiles differ significantly depending on whether traditional or modified TCE is used. For these runs, Λ for every reaction is ten times higher than the nominal value.34

Figure 3.1. Scatterplot showing values of a hypothetical QoI vs. values of a parameter θ_1 . The dataset is obtained from a Monte Carlo sampling of the full parameter space (for this hypothetical example), and is projected onto the two-dimensional space shown.39

Figure 3.2. Scatterplots and corresponding r^2 values for different hypothetical relationships between a QoI and a parameter θ_1 . Scatterplots which are tightly packed around an obvious trendline lead to high values of r^2 , and indicate a strong relationship between θ_1 and the QoI.....41

Figure 3.3. Scatterplot showing values of a hypothetical QoI vs. values of a parameter θ_1 . The relationship between the parameter and the QoI is clearly non-linear, which results in a near-zero value for r^2 . The mutual information properly captures the non-linear relationship between θ_1 and the QoI.43

Figure 3.4. Estimation of the joint PDF for a hypothetical parameter and QoI, by means of a simple histogram based method.44

Figure 3.5. Schematic for the next step in the calculation of the mutual information. The normalized scatterplot data are used to determine the 2D, joint PDF for the QoI and the parameter θ_1 , and also to determine (by integration along rows and columns) 1D, marginal PDFs of the QoI and the parameter θ_144

Figure 3.6. Schematic for the next step in the process of calculating the mutual information, in which the 1D marginal PDF's calculated in the previous step are multiplied together to obtain a hypothetical 2D PDF corresponding to a case where the QoI is independent of θ_145

Figure 3.7.	Final step in the calculation of the mutual information. The actual 2D joint PDF and the hypothetical 2D PDF are combined as shown, and the term for which contours are shown in the left image is integrated over the 2D space defined by the QoI and θ_1 , as in Eq. 3.2, to obtain the mutual information.....	45
Figure 3.8.	Marginal PDF for the QoI in the verification example.....	47
Figure 3.9.	Joint PDF of θ and the QoI in the verification example.	48
Figure 3.10.	Scatterplot showing values of the QoI vs. values of θ for 10,000 sample points drawn from the joint PDF given by Eq. 3.3.	49
Figure 3.11.	The calculated mutual information as a function of the width of the histogram bins used to estimate the PDFs. Curves are shown for several different numbers of sample points. The true value is shown for comparison.....	51
Figure 3.12.	The calculated mutual information as a function of the width of the Gaussian kernel in the KDE method used to estimate the PDFs. Curves are shown for several different numbers of sample points. The true value is shown for comparison.	52
Figure 3.13.	The calculated mutual information as a function of the kernel width for the KDE method and the equivalent kernel width for the histogram method. Curves are shown for several different numbers of sample points. The true value is shown for comparison.	53
Figure 3.14.	QoI(x) profiles for a variety of values of A, B, and C.....	56
Figure 3.15.	Schematic showing the way in which a vector QoI is broken into individual scalar QoIs during the sensitivity analysis.....	57

Figure 3.16. Sensitivities based on r^2 and the mutual information for each of the three parameters in Eq. 3.5. Sensitivities are plotted as a function of x , and each x location corresponds to a particular scalar QoI (as shown in Fig. 3.15).60

Figure 3.17. Scatterplots for the QoI vs. parameter A at three different x locations. Note that the ratio of the mutual information to r^2 is different for each of the scatterplots.61

Figure 3.18. Sensitivities based on r^2 and the mutual information as a function of x for parameter C for three different numbers of Monte Carlo sample points. The curves for sensitivities based on r^2 are much less affected by the number of sample points.62

Figure 3.19. Sample profiles based on Eq. 3.5 after the addition of white noise with two different amplitudes.64

Figure 3.20. Sensitivities based on r^2 as a function of x for all three parameters for three different levels of white noise which were added to the QoI from Eq. 3.5.65

Figure 3.21. Sensitivities based on the mutual information as a function of x for all three parameters for three different levels of white noise added to the QoI from Eq. 3.5.66

Figure 3.22. Sensitivities as a function of x for all four parameters from Eq. 3.7. The mutual information captures the sensitivity of the QoI to the noise amplitude (parameter D) while r^2 does not.67

Figure 3.23. Scatterplot for the QoI vs. parameter D from Eq. 3.7, which controls the amplitude of the additive white noise. Parameter D clearly affects the distribution of values of the QoI, but the relationship is non-linear and thus r^2 is very nearly zero. The mutual information does capture the relationship.....68

Figure 4.1. Flowchart for the basic Metropolis-Hastings algorithm, starting at the box on the upper left. The loop on the right side is repeated until the Markov chain has reached the desired length or until some specified convergence metrics have been achieved.70

Figure 4.2. Flowchart for the interaction of the various codes involved in the solution of the inverse problem. Yellow and green boxes are codes supplied by PECOS (which in some cases I have modified slightly) and blue boxes are codes which I have written. The arrows indicate the type of information passed between the codes and the direction in which that information is passed.80

Figure 4.3. QoI(x) profile with parameters A, B, C, and D of Eq. 4.5 all set to their nominal value of 0.5. This QoI(x) profile serves as the synthetic data for the first set of example calibrations. Also shown are 2σ error bounds based on the assumption that the uncertainty of the synthetic data points is given by $\sigma^2 = 0.0001$ (the red curves) or $\sigma^2 = 0.001$ (the blue curves).
.....83

Figure 4.4. Posterior PDFs for each of the four parameters from Eq. 4.5 based on calibration with the synthetic data of Fig. 4.3 using 100 calibration data points and the assumption that $\sigma^2 = 0.0001$ for the synthetic data. The calibration is performed with 16 chains of 100,000 positions each, with a burn-in period of 20,000 positions for each chain.84

Figure 4.5. Posterior PDF for parameter D of Eq. 4.5 (zoomed in to show the tails of the distribution) based on the same set of chains as Fig. 4.4. The posterior PDF based on all 100,000 of the chain positions is shown in red. A burn-in period of 20,000 chain positions is used (i.e. the first 20,000 positions of each chain are discarded) when calculating the posterior PDF shown in blue, and a burn-in period of 60,000 chain positions is used when calculating the posterior shown with green symbols.87

Figure 4.6. Posterior PDFs for two of the parameters from Eq. 4.5 based on two different calibrations with the synthetic data of Fig 4.3. The same 100 synthetic data points are used for both calibrations, but the uncertainty assumed for the synthetic data is different in each calibration. The calibrations both use 16 chains of 100,000 positions each, with a burn-in period of 20,000 chain positions.89

Figure 4.7. Posterior PDFs for two of the parameters from Eq. 4.5, based on two different calibrations with the synthetic data of Fig. 4.3. The same uncertainty is assumed for the synthetic data points in both calibrations, but the first calibration uses 100 synthetic data points and the second uses 20. The calibrations both use 16 chains of 100,000 positions each, with a burn-in period of 20,000 chain positions.90

Figure 4.8.	QoI(x) profile with parameters A, B, C, and D of Eq. 4.5 set to 0.3, 0.7, 0.4, and 0.6, respectively. This QoI(x) profile serves as the synthetic data for the final example calibration. Also shown are 2σ error bounds based on the assumption that the uncertainty of the synthetic data points is given by $\sigma^2 = 0.0001$.	91
Figure 4.9.	Posterior PDFs for each of the four parameters from Eq. 4.5 based on calibration with the synthetic data of Fig. 4.8 using 100 calibration data points and the assumption that $\sigma^2 = 0.0001$ for the synthetic data. The calibration is performed with 16 chains of 100,000 positions each, with a burn-in period of 20,000 positions for each chain.	92
Figure 5.1:	Evolution of the density profiles for all five species during a relaxation from an initial state with 79% N ₂ and 21% O ₂ . Initial translational temperature is ~50,000 K while initial rotational and vibrational temperatures are 300 K.	96
Figure 5.2:	Evolution of the translational, rotational, and vibrational temperatures for N ₂ during a relaxation from an initial state with 79% N ₂ and 21% O ₂ . Initial translational temperature is ~50,000 K while initial rotational and vibrational temperatures are 300 K.	97
Figure 5.3.	Schematic showing the way in which the ρ_{NO} vector QoI is broken into individual scalar QoIs during the sensitivity analysis.	103
Figure 5.4:	Evolution of the reaction rates for six reactions which directly affect ρ_{NO} during a relaxation from an initial state with 79% N ₂ and 21% O ₂ . Initial translational temperature is ~50,000 K while initial rotational and vibrational temperatures are 300 K.	105

Figure 5.5: Profiles of ρ_{NO} vs. time for the same 0D relaxation scenario as shown in Fig. 5.1. Each profile comes from a simulation with a set of parameters corresponding to a different point in parameter space.107

Figure 5.6. Sensitivities as a function of time for the six most sensitive parameters, all of which are from among the reaction rate parameters (parameters 1-17 in Table 5.1). Both r^2 and the mutual information are shown on the plot for each parameter.110

Figure 5.7. Scatterplots showing ρ_{NO} (at four discrete points in time given by the labels on the vertical axis in each plot) vs. $\log_{10}\Lambda$ for the reaction $N_2 + N \rightleftharpoons 3N$. The square of the Pearson correlation coefficient and the mutual information are also shown on the plots.111

Figure 5.8. Scatterplots showing ρ_{NO} (at four discrete points in time given by the labels on the vertical axis in each plot) vs. $\log_{10}\Lambda$ for the reaction $O_2 + O \rightleftharpoons 3O$. The square of the Pearson correlation coefficient and the mutual information are also shown on the plots.112

Figure 5.9. Scatterplots showing ρ_{NO} (at four discrete points in time given by the labels on the vertical axis in each plot) vs. $\log_{10}\Lambda$ for the reaction $NO + N \rightleftharpoons 2N + O$. The square of the Pearson correlation coefficient and the mutual information are also shown on the plots.113

Figure 5.10. Scatterplots showing ρ_{NO} (at four discrete points in time given by the labels on the vertical axis in each plot) vs. $\log_{10}\Lambda$ for the reaction $NO + O \rightleftharpoons N + 2O$. The square of the Pearson correlation coefficient and the mutual information are also shown on the plots.115

Figure 5.11. Scatterplots showing ρ_{NO} (at four discrete points in time given by the labels on the vertical axis in each plot) vs. $\log_{10}\Lambda$ for the reaction $\text{N}_2 + \text{O} \rightleftharpoons \text{NO} + \text{N}$. The square of the Pearson correlation coefficient and the mutual information are also shown on the plots.116

Figure 5.12. Scatterplots showing ρ_{NO} (at four discrete points in time given by the labels on the vertical axis in each plot) vs. $\log_{10}\Lambda$ for the reaction $\text{NO} + \text{O} \rightleftharpoons \text{O}_2 + \text{N}$. The square of the Pearson correlation coefficient and the mutual information are also shown on the plots.117

Figure 5.13. Scatterplots showing ρ_{NO} (at two discrete points in time given by the labels on the vertical axis in each plot) vs. $\log_{10}\Lambda$ for the reaction $\text{NO} + \text{N} \rightleftharpoons 2\text{N} + \text{O}$. The vertical scale is the same for both plots in order to show the substantially higher variance of ρ_{NO} in the scatterplot on the left, despite both r^2 and the mutual information being higher for the scatterplot on the right.118

Figure 5.14. Variance-weighted sensitivities as a function of time for the top six most sensitive parameters, all of which are from among the reaction rate parameters (parameters 1-17 in Table 5.1). Sensitivities based on both r^2 and the mutual information are shown on the plot for each parameter, and the variance of ρ_{NO} at each point in time is shown as well. The variance-weighted sensitivities are normalized so that the highest sensitivity at any point in the relaxation is set equal to one. This is done separately for r^2 and the mutual information.121

Figure 5.15. Zoomed-in version of Fig. 5.14, showing in more detail the normalized, variance-weighted sensitivities for early times when the variance of ρ_{NO} is highest.122

Figure 5.16. Overall, variance-weighted sensitivities for each parameter based on each measure of sensitivity. The overall sensitivities are normalized so that the overall sensitivity for the most sensitive parameter is set equal to one. This normalization is done separately for r^2 and the mutual information, so the most sensitive parameter for each measure will have an overall sensitivity of one.123

Figure 5.17. Variance-weighted sensitivities (based on r^2) as a function of time for the six most sensitive parameters. Sensitivities are shown for the main analysis with all 20,000 Monte Carlo sample points in parameter space (solid lines) and also for the analysis with only half of the sample points (symbols).....126

Figure 5.18. Variance-weighted sensitivities (based on the mutual information) as a function of time for the six most sensitive parameters. Sensitivities are shown for the main analysis with all 20,000 Monte Carlo sample points in parameter space (solid lines) and also for the analysis with only half of the sample points (symbols).127

Figure 5.19. Overall sensitivities (based on both measures of sensitivity) for the main analysis with all 20,000 Monte Carlo sample points in parameter space and for the analysis with half as many sample points.....128

Figure 6.1: Profiles of bulk density and individual species densities across a shock at ~ 8000 m/s. Upstream temperature is 300 K, and number densities for N_2 and O_2 are 2.5×10^{21} #/m³ and 6.8×10^{20} #/m³, respectively.130

Figure 6.2: Profiles of O_2 and NO densities from Fig. 6.1, with the vertical axis rescaled.131

Figure 6.3: Profiles of translational, rotational, vibrational, and overall temperatures for N_2 across the shock from Fig. 6.1.	132
Figure 6.4: Schematic showing the way in which the ρ_{NO} vector QoI is broken into individual scalar QoIs during the sensitivity analysis for the 1D shock.	134
Figure 6.5: Sensitivities as a function of streamwise location for five of the most sensitive parameters. Both r^2 and the mutual information are shown on the plot for each parameter.	138
Figure 6.6: Scatterplot showing ρ_{NO} (at $x = 0.066$ m) vs. $\log_{10}\Lambda$ for the reaction $N_2 + O \rightleftharpoons NO + N$. The relationship between $\log_{10}\Lambda$ and ρ_{NO} at this location in the shock is non-linear.	139
Figure 6.7: Scatterplots showing ρ_{NO} at $x = 0.0028$ m vs. $\log_{10}\Lambda$ for the reaction $NO + N \rightleftharpoons 2N + O$ (left image) and ρ_{NO} at $x = 0.1799$ m vs. $\log_{10}\Lambda$ for the reaction $N_2 + N \rightleftharpoons 3N$	142
Figure 6.8: Scatterplot showing the vector QoI ρ_{NO} for all of the shocks simulated in the sensitivity analysis, along with the variance of ρ_{NO} as a function of x	142
Figure 6.9: Variance-weighted sensitivities as a function of streamwise location for the top five most sensitive parameters. Both r^2 and the mutual information are shown on the plot for each reaction.	143
Figure 6.10: Zoomed-in view of the plot in Fig. 6.9, showing the non-equilibrium region where the variance of ρ_{NO} and the variance-weighted sensitivities are highest.	144
Figure 6.11: Overall, variance-weighted sensitivities for each parameter with each measure of sensitivity.	145

Figure 6.12: Variance-weighted sensitivities (based on the mutual information) as a function of streamwise location of the six most sensitive parameters. Sensitivities are shown for the normal resolution and also for the lower resolution analysis. While the curves do not agree exactly, the shapes are very similar.	147
Figure 6.13: Variance-weighted sensitivities (based on the mutual information) as a function of streamwise location of the six most sensitive parameters. Sensitivities are shown for the main analysis (with all 5600 Monte Carlo sample points) and also for the analysis with only half of the sample points.....	148
Figure 6.14: Overall sensitivities (based on the mutual information) for the main analysis presented earlier in the chapter (regular DSMC resolution with all 5600 Monte Carlo sample points), the analysis with lowered DSMC resolution, and the analysis with half as many Monte Carlo sample points.....	149
Figure 7.1: Evolution of the density profiles for all five species during a relaxation from an initial state with 79% N ₂ and 21% O ₂ . Initial translational temperature is 30,000 K while initial rotational and vibrational temperatures are 300 K. This relaxation is the second scenario for our synthetic data calibrations.....	157
Figure 7.2: Synthetic data for the first relaxation scenario, along with 2σ error bars on the data (based on an uncertainty of $\sigma^2 = 2.0 \times 10^{-11}$). A set of 11 evenly spaced points from each curve are used as the synthetic data points for the first scenario.	160

Figure 7.3: Synthetic data for the second relaxation scenario, along with 2σ error bars on the data (based on an uncertainty of $\sigma^2 = 2.0 \times 10^{-11}$). A set of 11 evenly spaced points from each curve are used as the synthetic data points for the second scenario.....161

Figure 7.4: Levels of parallelism used during the calibration process. The blue ovals show the parallel organization for the parts of the process that take place within the QUESO driver code, the orange ovals show the organization for the parts within the MCMC/DSMC interface code, and the green circles show the organization within the DSMC code. See Fig. 4.2 for more information on the codes which participate in the calibration process. The single processor runs are ensemble averaged to provide lower noise results. Any number of chains and single processor runs may be used; the numbers here are just for readability of the schematic. The code is also capable of handling more than two scenarios, but we use only two for the calibration presented in this work.164

Figure 7.5: Post-calibration PDFs for parameters #2 and #9 (based on numbering from Table 7.1).166

Figure 7.6: Post-calibration PDFs for parameters #12 and #14 (based on numbering from Table 7.1).166

Figure 7.7: Post-calibration PDFs for parameters #16 and #17 (based on numbering from Table 7.1).167

Figure 7.8: Post-calibration PDFs for all six of the parameters, with the horizontal axis normalized so the nominal value for all parameters is zero. ...169

Figure 7.9: Same PDFs as in Fig. 7.8, zoomed in to show the PDF peaks more closely.170

Figure 7.10: Post-calibration PDF for parameter #9 from Table 7.1. Also shown are the values of the likelihood for the candidate positions tested during the calibration (both those accepted as chain positions and those which were not).174

Figure 7.11: Post-calibration PDF for parameter #12 from Table 7.1. Also shown are the values of the likelihood for the candidate positions tested during the calibration (both those accepted as chain positions and those which were not).175

Figure 7.12: Post-calibration PDF for parameter #14 from Table 7.1. Also shown are the values of the likelihood for the candidate positions tested during the calibration (both those accepted as chain positions and those which were not).176

Nomenclature

C_1, C_2	=	constants in the equation for Z_V as a function of collision temperature
d_{ref}	=	VHS reference diameter
E_A	=	activation energy
E_C	=	total collision energy
k_b	=	Boltzmann constant
$k(T)$	=	Arrhenius reaction-rate coefficient as a function of temperature
n_T	=	number density of the third body partner for a recombination reaction
N_{MC}	=	number of Monte Carlo samples of the parameter space
N_R	=	number of possible reactions between two species
P_{norm}	=	normalized pressure
P_1	=	upstream (pre-shock) pressure
P_2	=	downstream (post-shock) pressure
r	=	Pearson correlation coefficient
r^2	=	square of the Pearson correlation coefficient
ω	=	temperature-viscosity exponent
σ_{ref}	=	reference cross-section for the VHS model
σ_{VHS}	=	VHS collision cross-section
σ_R	=	reaction cross-section
σ_T	=	total cross-section
A	=	pre-exponential constant in the Arrhenius rate equation
η	=	temperature exponent in the Arrhenius rate equation
λ_R	=	rotational collision probability
λ_V	=	vibrational collision probability
T_{ref}	=	reference temperature for the VHS model
Z_R	=	rotational collision number
Z_V	=	vibrational collision number
m_r	=	reduced mass
$\bar{\zeta}$	=	average number of internal degrees of freedom which contribute to total energy of a collision

Chapter 1: Introduction

MOTIVATION

Direct Simulation Monte Carlo (DSMC) is a particle based method which is often used for the simulation of rarefied and/or highly non-equilibrium gas flows where it is important to account for non-continuum aspects of the gas dynamics. DSMC has been continuously developed and improved by many researchers over the course of several decades, but little or no prior work has been done which integrates DSMC with statistical analysis methods for the purposes of sensitivity analysis or parameter calibration.

The DSMC method includes many parameters related to gas dynamics at the molecular level. Examples include parameters related to elastic collision cross-sections, rotational and vibrational excitation and relaxation, reaction cross-sections, etc. The precise values of some of these parameters are not known and parameter values often cannot be directly measured. Instead, they must be inferred from experimental results, and by necessity parameters must often be used in regimes far from where their values were calibrated. More precisely calibrated values for some of these important parameters could lead to better simulation of the physics, and thus to better predictive capability for DSMC. Furthermore, having post-calibration parameter PDFs which incorporate the uncertainty of the experimental data used in the calibrations would assist greatly in quantifying the uncertainty which is present in predictions made with DSMC. Finally, identifying the set of parameters which most strongly influence the simulation results for a given scenario will allow experimental and other computational efforts to focus on lowering the uncertainty present in those parameters which most strongly affect a particular quantity of interest (QoI).

OBJECTIVES

The current work focuses on sensitivity analysis and synthetic data calibration. It is intended to lay the groundwork for future calibration based on experimental data, which is the long-term goal of the project.

In approaching that goal, the first step is to write and test a DSMC code which is capable of seamless integration with the statistical methods we wish to employ, and which is computationally efficient enough to allow us to perform very large numbers of simulations within a reasonable timeframe. This code must also be able to correctly model the relevant physics for a hypersonic shock, which is the primary scenario we have chosen to examine in this work.

After the DSMC code is in place, the next step is a sensitivity analysis to determine which parameters most strongly affect the results of DSMC simulations of hypersonic shocks. A global sensitivity analysis is needed due to the relatively large number of parameters, the large uncertainties associated with them, and the highly coupled nature of the physical problem. In this work, we will establish a set of codes which allow us to employ global, Monte Carlo based methods to determine the sensitivities for a variety of parameters. We will examine two selected scenarios which are suitable for simulation with DSMC. The first of these is a 0D relaxation from an initial high temperature state and the second is a 1D, hypersonic shock. The 0D relaxation scenario is included in order to test the hypothesis that in some situations this scenario can serve as a substitute (for the purposes of sensitivity analysis) for the much more computationally expensive 1D shock scenario. We will identify a broad set of parameters relevant to modeling each of these scenarios with DSMC, and we will then determine which of these parameters most strongly affect the simulation results for a chosen QoI.

Once the sensitivity analysis is completed, the final step is to perform synthetic data calibrations in order to demonstrate that the statistical inverse problem can be solved, and thus that we can hope to eventually perform calibrations with real data. The machinery put in place in this work will set the stage for subsequent calibrations with experimental data as a part of the ongoing project.

LITERATURE REVIEW

In this literature review we will cover the three main subject areas of this work. The first section will deal with DSMC, the second with global sensitivity analysis methods, and the third with the solution of the inverse problem via the MCMC algorithm.

DSMC

We first address DSMC, the numerical technique used for all of the simulations in this work. DSMC is a stochastic, particle based method for simulating gas dynamics in which simulated particles represent large numbers of real particles. These simulated particles move and interact with one another, and the interactions between particles (such as elastic or inelastic collisions and chemical reactions) are handled statistically. DSMC is more computationally expensive than most CFD codes, but it is often the only realistic option for the simulation of rarefied flows which occur in a diverse set of fields. In this work, our choice to use DSMC is driven by the fact that it is well-suited for the accurate simulation of highly non-equilibrium regions of a flowfield (such as strong shock waves), and it can model thermochemistry on a more detailed level than most CFD codes.

DSMC has been around for decades and has been used for the simulation of a wide variety of physical problems, from space shuttle re-entry to MEMS devices to planetary atmospheres. In this section we will focus on aspects of the method which might pertain to the simulation of hypersonic shocks. The textbook written by Bird (1994), who is considered the inventor of DSMC, provides a valuable source which will be referred to throughout this section.

Collision Model

Bird (1994) provides an overview of the set of elastic collision models which are often employed in DSMC simulations. Prior to the 1980's the hard-sphere collision model was used for most DSMC simulations. While results obtained with the hard-sphere model are not grossly inaccurate, the model does not allow the collision cross-section to depend on the relative velocity between the colliding particles, which is a major drawback. Bird (1981) proposed the variable hard-sphere (VHS) collision model. The VHS model contains a second parameter which allows the collision cross-section to be dependent on the relative velocity, which then allows for a more accurate scaling of viscosity with temperature. A further modification by Koura and Matsumoto (1992), known as the variable soft sphere (VSS) model added a third parameter which affects the scattering angle for the post-collision particle velocities, and adding this third parameter allows DSMC to properly match both the viscosity-temperature relationship and also the relationship between the diffusion coefficients and the temperature. More complex collision models have also been proposed, such as the generalized hard sphere (GHS) model of Hassan and Hash (1993) which allows the parameters of a Lennard-Jones potential to be used in the DSMC collision model.

The models used for inelastic collisions have also changed over the years. The very earliest DSMC simulations could only handle monatomic gases. Bird (1994) described what he refers to as the Larsen-Borgnakke model (Borgnakke and Larsen, 1975), which can be incorporated into DSMC in order to allow the simulation of species with internal energy modes. The model is phenomenological in nature, and allows energy to be redistributed between the translational and internal modes during some portion of the DSMC collisions. Models based on the work of Borgnakke and Larsen were included in DSMC codes by both Bergemann and Boyd (1994) and Haas *et al.* (1994). In recent years some DSMC codes have begun using a state-to-state model for vibrational collisions. In a state-to-state model, such as that of Boyd and Josyula (2011), individual cross-sections are used for each possible vibrational transition which might occur during a collision between a given pair of species. These cross-sections may be a function of the relative velocity between the particles and also of the current internal energy state of both particles.

Chemistry Model

Bird (1994) described several possible chemistry models, including the commonly used total collision energy (TCE) model. The TCE model calculates a reaction-cross section based on the total energy (translational and internal) available in a collision. The parameters for an Arrhenius reaction rate equation are included in the equation for the reaction cross-section, allowing DSMC reaction rates to be set to match Arrhenius rates. Improvements have been incorporated into TCE in order to more properly address the dependence of dissociation reaction rates on the vibrational level of the dissociating

species. The improved model was devised by Haas and Boyd (1993) and is known as vibrationally-favored dissociation (VFD).

More advanced chemistry implementations are also possible, including ones which are based on tabulated reaction cross-sections over a range of collision parameters. An example of this type of reaction model is included in the work of Moore *et al.* (2011).

Shock Simulations

DSMC has been used a number of times in the past to simulate shock waves. In a well-known work by Pham-Van-Diep *et al.* (1989), the molecular velocity distribution function from a DSMC simulation of a hypersonic shock was compared to an experimentally measured molecular velocity distribution function. The bimodal velocity distribution function observed in the shock had been predicted theoretically, but it had never before been observed experimentally. The qualitative and quantitative agreement between the experimental and DSMC simulated distribution functions which was described in Erwin *et al.* (1991) is considered an important validation of the DSMC method.

DSMC has been used to simulate hypersonic shocks in more recent work as well, such as that of Watvisave *et al.* (2011) which examined the effect of a growing boundary layer on the propagation of a shock wave in pure nitrogen in a shock-tube. Another recent example of the use of DSMC for hypersonic shock simulations can be found in the work of Farbar and Boyd (2011), where a hybrid DSMC/PIC code was employed to model a very high temperature shock layer plasma, such as that which would be found in the bow shock region of a re-entry vehicle on a lunar or Mars return trajectory.

Sensitivity Analysis for DSMC Simulations

Sensitivity analyses have been performed with DSMC codes a number of times in the past, but they have been confined to a very small number of parameters (often just one or two) and they have been local in nature. For example, Valentini and Schwartzentruber (2009) examined the sensitivity of an argon shock profile to one of the VHS collision parameters (the temperature-viscosity exponent, ω). Only this one parameter was varied, and it was varied only in a small region near its nominal value.

Calibration of DSMC Parameters

Recent work by Stephani, Goldstein, and Varghese (2012) focused on the calibration of DSMC collision parameters. The calibrations were done with the Nelder-Mead method. The aim in that work was to calibrate the DSMC parameters not against experiments, but rather against a different set of parameters related to the calculation of transport coefficients in a CFD model. This was done so as to enable the transport coefficients to be matched across a hybrid boundary between DSMC and CFD codes.

Global Sensitivity Analysis

The purpose of the sensitivity analyses presented in this work is to determine which parameters most strongly affect the simulation results for a given scenario and QoI. Later, when we perform synthetic data calibrations, we will choose which parameters to calibrate based on the results of our sensitivity analyses.

In this work we use a global sensitivity analysis methodology. In a global sensitivity analysis all of the parameters are varied simultaneously, and our global sensitivity analysis technique requires a Monte Carlo sampling of the parameter space.

Before sampling the parameter space, each parameter is assigned a distribution, called the prior, which reflects the level of uncertainty for that parameter (based on literature values and expert judgment). During the sensitivity analysis the parameters take on values drawn from their priors, rather than in a small region about their nominal values. For each sample from the parameter space, a set of random number draws is performed to determine the values of all of the parameters. A simulation is then run with this set of parameter values, and the results for the QoI are output. Once a reasonable number of sample points have been completed, the results of this sampling can be projected onto a two-dimensional space to generate scatterplots of the values of a given parameter vs. the values of a QoI, and these scatterplots can be used to examine the relationship between each parameter and the QoI. For a much more detailed discussion of the sampling of a parameter space during a global, Monte Carlo sensitivity analysis, see Chapter 3 of this dissertation.

In this section we will cover the two measures of sensitivity which we will use during our sensitivity analyses later in this work. Both of these measures are calculated based on the information contained in the scatterplots described above. The first of these measures is the square of the Pearson correlation coefficient (referred to as r^2) and the second is the mutual information.

Sensitivities Based on r^2

The Pearson correlation coefficient is a very common statistical measure and can be found in any basic statistics textbook, so we will not discuss the measure itself in this section. The theory behind the use of r^2 as a measure of sensitivity is discussed in great detail in Chapter 3 of this dissertation. We will focus here on the use of r^2 as a measure

of sensitivity in past works involving global sensitivity analyses related to hypersonic flows.

Global sensitivity analyses have been performed for problems similar to what we are examining in this work, but they have been done with hypersonic CFD codes rather than with particle based methods. Bose, Wright, and Gökçen (2004) performed a global uncertainty and sensitivity analysis for the thermochemical modeling of a Titan atmospheric entry. Their analysis was based on a Monte Carlo sampling of several hundred parameters, such as reaction rate constants, parameters related to vibrational relaxation and vibration-chemistry coupling, and transport properties. They used the square of the Pearson correlation coefficient as their measure for sensitivity. Their work examined scalar quantities of interest such as the vehicle heating, and they were able to identify the parameters whose uncertainties contributed most to the overall uncertainty in the vehicle heating rate. Similar follow-on work was done by Bose and Wright (2006) for Mars entries. In that work, the QoI was the laminar, convective heating at the vehicle surface, and a total of 130 parameters were examined, including collision integrals and reaction rates. The parameter space was split into four domains, with each domain corresponding to a particular wall-catalysis regime. Four separate analyses were actually conducted, one for each domain, and the parameters which contributed the most to the overall uncertainty of the QoI were identified for each domain.

Miki *et al.* (2010) performed a global, Monte Carlo based sensitivity analysis for a hypersonic CFD simulation of a 1D shock. The parameters included chemical reaction rates, ionization rates, and parameters related to vibrational and electronic excitation. That work also used the square of the Pearson correlation coefficient as the measure for sensitivity. That project is ongoing, and subsequent work done by Miki *et al.* (2011) and Panesi *et al.* (2011) examined sensitivities for additional parameters.

The Mutual Information

The mutual information has not previously been used for a sensitivity analysis related to hypersonic flows, at least in published work. In this section we will primarily discuss the various works which have informed our current technique for calculating the mutual information based on a dataset from a Monte Carlo sampling of a parameter space. An in-depth discussion of the theory behind the use of the mutual information as a measure of sensitivity can be found in Chapter 3 of this dissertation.

The concept of the mutual information was first introduced by Shannon (1948), and the textbook by Cover and Thomas (1991) provides a great deal of background information on the subject. The work of Steuer *et al.* (2002) gives a comprehensive overview of the mutual information and some methods for estimating it based on a sampling of the parameter space. The specific results presented in that work are from the field of bioinformatics and involve the analysis of co-expressed genes, but the description of the methods used to estimate the mutual information is straightforward and useful. The mutual information is first defined and described briefly, and then two methods are given for estimating the mutual information from a dataset. The first of these methods is based on a histogram technique, and the paper demonstrates a key drawback of this method, namely the potential for relatively large, spurious values of the mutual information when the number of histogram bins is large relative to the number of data points in the sample set. The second method for calculating the mutual information is based on kernel density estimation (KDE) and it is shown that KDE is superior to the histogram based method for calculating the mutual information in practical cases where the number of sample points in parameter space is relatively small. Finally, the mutual

information is compared with r^2 as a measure of sensitivity for a set of gene expression datasets. While describing KDE, Steuer *et al.* refer back to the work of Moon *et al.* (1995), where KDE is described in more mathematical detail.

Another technique for estimating the mutual information is known as k-nearest neighbor, and it is described by Kraskov *et al.* (2004), and yet another technique based on B-splines is presented in the work of Daub *et al.* (2004).

MCMC

Markov Chain Monte Carlo (MCMC) is a method which solves the statistical inverse problem in order to calibrate parameters with respect to a set of data. During the calibration process, one or more chains are generated. These chains consist of a series of positions in parameter space, with each position in parameter space corresponding to a set of parameter values. A likelihood for any given chain position (i.e. for any given set of parameter values) is calculated based on the mismatch between the calibration data and the simulation results which are obtained when the model is run with that set of parameters. Each chain begins at some initial position, and a likelihood is calculated for that initial chain position. New chain positions are added by means of proposing candidate positions. Candidate positions are drawn from a multi-dimensional Gaussian proposal distribution centered at the current chain position. The covariance matrix of this Gaussian controls the average distance (in parameter space) that the chain moves in one step. After a candidate position is proposed, the likelihood is calculated for the corresponding set of parameter values, and the candidate position is then accepted or rejected based on a comparison between that likelihood and the likelihood for the current chain position. If the candidate position is accepted, it becomes the current chain

position, and thus the chain grows longer. If the candidate position is rejected, the current position remains unchanged and a new candidate position is proposed. This process is repeated a set number of times and/or until some specified convergence criteria are met. As the chain grows it explores the parameter space, moving via a biased random walk towards regions of higher likelihood. After a sufficient number of positions have been added to the chain, the set of accepted chain positions may be used to estimate post-calibration PDFs for the parameters. A thorough, step-by-step description of the MCMC algorithm is presented in Chapter 4 of this dissertation.

MCMC is used for a vast array of applications in countless fields, and we will not attempt to cover the general MCMC literature here. We will focus instead on two aspects of the literature. We will first discuss the literature relevant to the development and implementation of the delayed-rejection adaptive Metropolis (DRAM) algorithm which is used in this work, and then we will describe recent work in which MCMC is used for the calibration of parameters relevant to hypersonic flows.

DRAM

The Metropolis-Hastings algorithm is the basis for most other MCMC algorithms, including the more advanced DRAM algorithm used in this work. A detailed description of the basic Metropolis-Hastings MCMC algorithm is found in Tierney (1994), and the details of Metropolis-Hastings are also presented in Chapter 4. The DRAM algorithm consists of two primary improvements to Metropolis-Hastings. Both of these improvements are intended to allow chains to converge to the posterior (post-calibration) distribution with the use of fewer chain positions.

The first improvement is called delayed rejection, and it was developed over the course of several years by Tierney and Mira (1999), Green and Mira (2001), Mira (2001), and Mira (2002). When delayed rejection is employed and the initial candidate position is rejected, rather than immediately retaining the current position and drawing a new candidate from the original proposal distribution, instead a candidate position is drawn from a separate, second stage proposal. This second stage proposal may incorporate information about the rejected first stage candidate, or it may simply be a scaled version of the first stage proposal. Delayed rejection can be carried out for an arbitrary number of steps, and the equations are set up so that the chains remain both Markovian and reversible.

The second improvement is the adaptation of the proposal covariance matrix based on the previously accepted chain positions. This adaptation was first employed by Haario *et al.* (1999, 2001), and the algorithm which includes this adaptation is called adaptive Metropolis. When using adaptive Metropolis, the proposal covariance matrix is adapted at fixed intervals in order to incorporate information from the past history of the chain into the proposal distribution. Adaptive Metropolis allows the proposal distribution to scale up or down as necessary in order to explore the parameter space better, and the method can also adjust to account for coupling between the parameters.

When the two improvements described above are put together the resulting algorithm is known as DRAM. This algorithm is described in detail by Haario *et al.* (2006), and also in Chapter 4 of this dissertation. In addition to the description of the algorithm, Haario *et al.* go on to prove the ergodicity of DRAM.

Calibration of Parameters Relevant to Hypersonic Flows

At least three recent publications have centered on the use of MCMC for the calibration of parameters relevant to hypersonic flows.

In Miki *et al.* (2012a), calibrations were performed with MCMC based on both synthetic data and based on data from shock tube experiments. A number of different stochastic models were considered for the likelihood, including models which assume additive error and others which assume multiplicative error. The likelihood models also differed in the way that they address potential dependencies between data points. In two of the four models the data points were assumed to be independent of one another, while in the other two models a non-diagonal covariance matrix was used in the Gaussian likelihood equation in order to account for dependencies between nearby data points. The parameter set being calibrated included parameters related to the stochastic likelihood equation as well as physical parameters which are important for the shock tube simulations.

In Panesi *et al.* (2012), parameter calibrations were performed with MCMC as part of a Bayesian model validation study for data reduction models used for shock tube experiments. Rather than calibrating and/or validating a computational model which would be used to simulate the overall shock tube experiment, the purpose of this work was to clarify the uncertainties which are present in the experimental data as a result of the particular data reduction models which were used to convert the raw data (in the form of photon counts received by an ICCD camera) into radiative intensities. The parameters of interest in the data reduction model were related to the means by which the model accounts for the very short gate widths of the camera.

In Miki *et al.* (2012b), the parameters of a model for atomic nitrogen ionization were calibrated with MCMC based on radiative intensity data from a shock tube

experiment. The model consisted of a one-dimensional plasma flow solver coupled with a radiation solver. The posterior PDFs for the reaction rates were then compared with values from the literature, and the post-calibration value for the reaction rate of the atomic nitrogen ionization was found to be consistent with previous estimates of this rate.

MAJOR CONTRIBUTIONS OF THE CURRENT WORK

The current work makes contributions to a number of areas of scientific interest. Some of these contributions are entirely new, and others involve the combination of techniques which have not previously been integrated with one another.

This work includes several improvements to the modeling of hypersonic shocks with the DSMC method. Most importantly, our technique allows a steady shock profile to be sampled from an unsteady, moving shock without the need for ensemble averaging. When this technique is employed, the post-shock conditions do not need to be known *a priori*, which is very valuable when simulating shocks in a chemically reacting gas where the post-shock conditions will not usually be known initially. The DSMC code is MPI parallel, as are all of the codes used in this work, and substantial effort was put into proper load balancing to allow efficient scaling when simulating shocks. Additionally, we describe and test a modification to the standard TCE chemistry model which allows for the correct simulation of reactions with rates which approach or exceed the elastic collision rate.

Another significant contribution is the application of global, Monte Carlo based sensitivity analysis methods for the case of a stochastic simulation method such as DSMC. A sensitivity analysis of this type does not appear to have been performed in the past for DSMC. In the hypersonics field there has been some past work which included

global sensitivity analyses which are similar to those used here, as discussed in the preceding section. Our work goes into more depth than those previous analyses, however. In addition to making use of the mutual information as a second, more sophisticated measure of sensitivity, we also introduce the concept of variance-weighted sensitivities. Variance weighting allows us to integrate our sensitivity profiles for a vector QoI in an appropriate way in order to determine overall sensitivities for each parameter. Finally, we demonstrate techniques to test whether a given sensitivity analysis has sufficiently sampled the parameter space and also whether the simulations are well enough resolved to provide a converged result for the sensitivity of the parameters. Having a practical way to address convergence is useful for a computationally expensive problem where resolving every simulation to machine precision and using millions of points in the sampling of the parameter space are not feasible options. The stochastic nature of DSMC makes the entire process even more challenging. We describe a method for including DSMC stochastic noise as a parameter in the sensitivity analysis, and we demonstrate that the mutual information can accurately measure the sensitivity of the simulation results to this noise related parameter in a way that r^2 cannot.

Finally, when performing the synthetic data calibrations we employ multiple scenarios and quantities of interest simultaneously in order to obtain well-characterized post-calibration PDFs. The ability to make use of several types of data from multiple experiments simultaneously while solving the inverse problem is useful when performing calibrations for highly coupled sets of parameters, as we show with our synthetic data calibrations in this work.

In addition to the scientific contributions of the current work, it must also be mentioned that a substantial codebase has been generated and tested and will continue to

be used both by myself in my future work and by follow-on graduate students. Prior to this work, there was no existing DSMC code which would have been suitable for the research presented here, and having an in-house code which was developed specifically for the purposes of sensitivity analysis and parameter calibration will be a valuable asset going forward. The sensitivity analysis codes used in this work also provide a variety of capabilities which were not previously available to our group, especially the ability to deal appropriately with a vector QoI.

DISSERTATION LAYOUT

Following this introductory chapter, the dissertation moves on to describe the numerical methods used in this work. Chapter 2 covers the DSMC method including the improvements and modifications used for our shock simulations, Chapter 3 discusses our sensitivity analysis techniques, and Chapter 4 describes the techniques used for the solution of the inverse problem in order to calibrate parameters.

After the numerical methods have been laid out, the main results from this work are presented. Chapters 5 and 6 cover the sensitivity analyses for the 0D relaxation and the 1D shock, respectively, and Chapter 7 addresses the synthetic data calibration. Finally, Chapter 8 sets out the conclusions we have drawn from this work and discusses potential future work which might be of interest.

Chapter 2: DSMC Methodology

OVERVIEW

The DSMC code used in this work is based on the method described by Bird (1994). The primary goal of the current work is to integrate DSMC with Bayesian statistical methods, and therefore we have chosen to employ a well-established, commonly understood DSMC algorithm. In the future, the methods developed here could be applied to more advanced DSMC techniques such as the vibrationally favored dissociation model of Haas and Boyd (1993) or the sophisticated DSMC algorithm of Bird *et al.* (2009).

In order to facilitate integration with various driver codes, the entire DSMC code is written as a subroutine. It is capable of handling multiple species, each with its own molecular properties. Both vibrational and rotational internal energies are included, along with 5-species air chemistry, including dissociation, recombination, and exchange reactions. It is currently set up to simulate 0-D relaxations and 1-D shocks. The code is MPI parallel, and makes use of ensemble averaging (for 0-D relaxations) and adaptive load balancing (for 1-D shocks) to appropriately distribute work across processors.

ELASTIC COLLISIONS

Elastic collisions in the code are performed using the VHS collision model. VHS parameters for the 5 species used in this work are shown in Table 2.1. In this work VHS parameters for cross-species collisions are usually obtained by a simple averaging of the parameters for the two species participating in the collision, but the code is capable of employing specific VHS parameters for cross-species collisions as well, if desired. This feature is utilized during the sensitivity analysis for the 0-D relaxation.

Table 2.1: Collision parameters for 5-species air. VHS parameters were compiled by Ozawa (2008). C_1 and C_2 are based on those found in Bird (1994), but have been modified to account for the fact that Bird used different values for ω .

Species	ω	$d_{\text{ref}} (\times 10^{-10} \text{ m})$	$T_{\text{ref}} \text{ (K)}$	C_1	C_2
N_2	0.65	3.11	1000	5.02	220.0
N	0.68	3.58	1000	-	-
O_2	0.65	2.96	1000	23.17	153.5
O	0.68	3.37	1000	-	-
NO	0.65	3.41	1000	5.02	220.0

INELASTIC COLLISIONS

The Larsen-Borgnakke model (Borgnakke and Larson, 1975) is employed for the modeling of particles with internal degrees of freedom within the DSMC framework. This model is phenomenological in nature. The key aspect of the model is that some fraction of collisions are regarded as inelastic, and in these collisions energy may be redistributed between the translational and internal modes. This redistribution is carried out based on selections of post-collision internal energies from the equilibrium distributions appropriate for the given mode at the collision energy. After the internal energies have been assigned, the remaining energy is assigned to the relative translational kinetic energy of the colliding particles. For a given inelastic collision, the post-collision energies are chosen from a distribution based on the energy of *that particular collision*, rather than a distribution based on the overall cell properties. This allows significant nonequilibrium to be present between the internal and translational modes at a given point in the flowfield.

In this work, rotational modes of the diatomic species are assumed to be fully excited. Each particle has its own value of rotational energy, and this variable is

continuously distributed (rotation is not considered quantized due to the close spacing of rotational levels). Particles have either zero rotational degrees of freedom (monatomic species) or two degrees of freedom (diatomic species). The parameter relevant to rotational excitation and relaxation is Z_R , the rotational collision number. In our code, Z_R is defined as $1/\lambda_R$, where λ_R is the probability of a molecule's rotational energy undergoing redistribution with the translational mode during any given collision. During a collision, a separate random number draw is done for each colliding diatomic particle, and based on this, one, both, or neither of the particles may undergo an exchange of energy between the rotational and translational modes. In the current work, we treat Z_R as a constant (independent of collision partner and temperature) for all species with rotational degrees of freedom.

Unlike rotation, vibration is not assumed to be fully excited, and vibrational levels are quantized. Each particle has its own vibrational level, which is associated with a certain vibrational energy based on the simple harmonic oscillator model. In a given collision there is a separate probability λ_V for each colliding diatom, and a random number draw based on this probability determines whether that particle will exchange energy between the vibrational and translational modes. As with rotation, $\lambda_V = 1/Z_V$, where Z_V is the vibrational collision number. In the work described here, Z_V depends on collision temperature but not on collision partner, based on the expression

$$Z_V = (C_1/T_{coll}^\omega)e^{(C_2T_{coll}^{-1/3})} \quad (2.1)$$

where C_1 and C_2 are constants, T_{coll} is the collision temperature as defined by Bird (1994), and ω is the temperature-viscosity exponent for the given species. C_1 and C_2 are species specific, but they do not depend on the collision partner. The values used for each species for C_1 and C_2 are obtained from Bird (1994), and are adjusted to account for

the fact that Bird's values were calibrated with values of ω which differ from those used here. C_1 and C_2 for the three diatomic species used in this work are listed in Table 2.1.

CHEMICAL REACTIONS

Chemical reactions in our code are handled by means of the TCE model. This model treats the ratio of the cross-section for a given reaction to the total collision cross-section as a function of the total energy (translational and internal) of the two colliding particles. Bird (1994) lays out the process for determining the reaction cross-section as a function of the collision energy, and for determining the parameters of this function based on the parameters of an Arrhenius-type rate equation. An Arrhenius-type rate equation takes the form

$$k(T) = \Lambda T^\eta e^{-E_A/k_b T} \quad (2.2)$$

where Λ and η are reaction-specific constants, E_A is the activation energy for the reaction, k_b is the Boltzmann constant, and T is the temperature of the gas. This form is not directly useful in DSMC because reactions and collisions in DSMC are handled by means of cross-sections, and so the Arrhenius parameters (Λ , η , and E_A) must be used to obtain an equation for the cross-section. Two such equations are derived in Bird (1994), one for exchange and dissociation (two-body) reactions, and one for recombination (three-body) reactions. The equation for two-body reactions is

$$\frac{\sigma_R}{\sigma_T} = \frac{\pi^{1/2} \epsilon \Lambda T_{ref}^\eta}{2\sigma_{ref} (k_b T_{ref})^{\eta-1+\omega_{AB}}} \frac{\Gamma(\bar{\zeta}+5/2-\omega_{AB})}{\Gamma(\bar{\zeta}+\eta+3/2)} \left(\frac{m_r}{2k_b T_{ref}} \right)^{1/2} \frac{(E_c - E_a)^{\eta+\bar{\zeta}+1/2}}{E_c^{\bar{\zeta}+3/2-\omega_{AB}}} \quad (2.3)$$

and the equation for three-body reactions is

$$\frac{\sigma_R}{\sigma_T} = \frac{\pi^{1/2} n_T \epsilon \Lambda T_{ref}^\eta}{2\sigma_{ref}} \frac{\Gamma(5/2-\omega_{AB})}{\Gamma(\eta+3/2)} \left(\frac{m_r}{2k_b T_{ref}} \right)^{1/2} \left(\frac{E_c}{k_b T_{ref}} \right)^{\eta-1+\omega_{AB}} \quad (2.4)$$

where σ_R/σ_T is the ratio of the cross-section for the given reaction to the total cross-section for interactions between the two particles. A and B are the main reacting species

(not including the third-body), ε is a symmetry factor which is equal to one if $A \neq B$ and two if $A = B$, σ_{ref} , T_{ref} , and ω_{AB} are VHS parameters for collisions between species A and B, $\bar{\zeta}$ is the average number of internal degrees of freedom which contribute to the collision energy, n_T is the number density of the third-body, m_r is the reduced mass of species A and B, E_c is the total collision energy (translational plus internal), and $\Gamma()$ is the gamma function. The above equation for three-body reactions assumes that the activation energy for these reactions is zero, which is the case for all recombination reactions which are relevant to five-species air chemistry.

Bird (1994) assumes that the reaction cross-section will be small relative to the VHS collision cross-section, which allows for the VHS collision cross-section to be treated as the total cross-section for use in the above equations. The advantage of this assumption is that reaction cross-sections only need to be calculated after a pair has been accepted for collision, rather than when the pair is initially selected. However, this assumption is sometimes not accurate at high collision temperatures for several of the reactions involved in five-species air chemistry, and it can lead to noticeable error in both the reaction rates and the VHS collision rates in certain cases, as will be discussed in detail later in this chapter. In our work, we require the ability to vary the Arrhenius pre-exponential constants over orders of magnitude while still retaining accurate reaction rates (i.e. the actual reaction rates occurring in the DSMC code must match those predicted by the Arrhenius rate equation for a given set of Arrhenius parameters at a given temperature, after sufficient averaging to minimize statistical noise). To this end, we have modified the model employed by Bird (1994) so that all of the reaction cross-sections are calculated after a pair is selected (in order to compute σ_T), but before the collision is accepted. With this model, σ_R/σ_T in the above equations is actually $\sigma_R/\sigma_{\text{VHS}}$, which is more accurate based on the derivation of the equations. After $\sigma_R/\sigma_{\text{VHS}}$ is

calculated for every potential reaction between species A and B, we can then obtain the total cross-section

$$\sigma_T = \sigma_{VHS} \left(1 + \sum_{i=1}^{N_R} (\sigma_R / \sigma_{VHS})_i \right) \quad (2.5)$$

where N_R is the number of possible reactions between species A and B. This σ_T is then used to determine acceptance or rejection of the given collision pair. If the collision pair is accepted, a random number draw determines whether or not one of the reactions occurs. The probability of a given reaction is simply the ratio σ_R / σ_T for that reaction, and the probability of a VHS collision is σ_{VHS} / σ_T . Finally, note that inelastic (but non-reactive) collisions are included within the VHS collision cross-section. In principle, the above model could easily be applied for inelastic collisions as well, if separate cross-sections were available for them. In that case, σ_{VHS} would instead be $\sigma_{Elastic}$, and σ_T would be expanded to include the inelastic collision cross-sections. This is not done because separate cross-sections are not readily available for vibrational and rotational transitions, so instead the Larsen-Borgnakke model is used as described previously and inelastic collisions are treated as a subset of VHS collisions. This is not a problem since the VHS parameters (d_{ref} and ω) were presumably calibrated with the understanding that the VHS cross-section would represent both elastic and inelastic collisions.

Arrhenius rates for the reactions important for five-species air chemistry have been tabulated, for example by Gupta *et al.* (1989). In this work, we make use of the forward rates provided by Gupta *et al.*, along with backward rates calculated based on a matching with the equilibrium constant over a range of temperatures. Table 2.2 contains the full list of reactions used in this work, along with the nominal Arrhenius parameters for each reaction.

Table 2.2: Reactions for 5-species air. Nominal Arrhenius forward rate parameters are from Gupta *et al.* (1989), and backward rate parameters are based on a matching with the equilibrium constant over a range of temperatures.

#	Reaction	Forward Rate Coefficients			Backward Rate Coefficients			Q_{reaction}
		Λ	η	E_A	Λ	η	E_A	
1	$2\text{N}_2 \rightleftharpoons 2\text{N} + \text{N}_2$	8.0E-13	-0.5	1.6E-18	6.5E-47	0.27	0.0	-1.6E-18
2	$\text{N}_2 + \text{N} \rightleftharpoons 3\text{N}$	6.9E-8	-1.5	1.6E-18	4.8E-46	0.27	0.0	-1.6E-18
3	$\text{N}_2 + \text{O}_2 \rightleftharpoons 2\text{N} + \text{O}_2$	3.2E-13	-0.5	1.6E-18	6.5E-47	0.27	0.0	-1.6E-18
4	$\text{N}_2 + \text{O} \rightleftharpoons 2\text{N} + \text{O}$	3.2E-13	-0.5	1.6E-18	6.5E-47	0.27	0.0	-1.6E-18
5	$\text{N}_2 + \text{NO} \rightleftharpoons 2\text{N} + \text{NO}$	3.2E-13	-0.5	1.6E-18	6.5E-47	0.27	0.0	-1.6E-18
6	$\text{O}_2 + \text{N}_2 \rightleftharpoons 2\text{O} + \text{N}_2$	1.2E-11	-1.0	8.2E-19	1.8E-47	0.27	0.0	-8.2E-19
7	$\text{O}_2 + \text{N} \rightleftharpoons 2\text{O} + \text{N}$	6.0E-12	-1.0	8.2E-19	1.8E-47	0.27	0.0	-8.2E-19
8	$2\text{O}_2 \rightleftharpoons 2\text{O} + \text{O}_2$	5.4E-11	-1.0	8.2E-19	1.8E-47	0.27	0.0	-8.2E-19
9	$\text{O}_2 + \text{O} \rightleftharpoons 3\text{O}$	1.5E-10	-1.0	8.2E-19	1.8E-47	0.27	0.0	-8.2E-19
10	$\text{O}_2 + \text{NO} \rightleftharpoons 2\text{O} + \text{NO}$	6.0E-12	-1.0	8.2E-19	1.8E-47	0.27	0.0	-8.2E-19
11	$\text{NO} + \text{N}_2 \rightleftharpoons \text{N} + \text{O} + \text{N}_2$	6.6E-10	-1.5	1.0E-18	2.1E-46	0.27	0.0	-1.0E-18
12	$\text{NO} + \text{N} \rightleftharpoons 2\text{N} + \text{O}$	1.3E-8	-1.5	1.0E-18	2.1E-46	0.27	0.0	-1.0E-18
13	$\text{NO} + \text{O}_2 \rightleftharpoons \text{N} + \text{O} + \text{O}_2$	6.6E-10	-1.5	1.0E-18	2.1E-46	0.27	0.0	-1.0E-18
14	$\text{NO} + \text{O} \rightleftharpoons \text{N} + 2\text{O}$	1.3E-8	-1.5	1.0E-18	2.1E-46	0.27	0.0	-1.0E-18
15	$2\text{NO} \rightleftharpoons \text{N} + \text{O} + \text{NO}$	1.3E-8	-1.5	1.0E-18	2.1E-46	0.27	0.0	-1.0E-18
16	$\text{N}_2 + \text{O} \rightleftharpoons \text{NO} + \text{N}$	1.1E-16	0.0	5.2E-19	2.5E-17	0.0	0.0	-5.2E-19
17	$\text{NO} + \text{O} \rightleftharpoons \text{O}_2 + \text{N}$	5.3E-21	1.0	2.7E-19	1.6E-18	0.5	5.0E-20	-2.7E-19

1-D SHOCK SIMULATION TECHNIQUE

Techniques for simulating 1-D, steady shocks with DSMC often require that the post-shock conditions be known initially. In the case to be simulated in this work, where real gas effects (internal modes, chemistry, etc.) are present, the post shock conditions will not be exactly known *a priori*. Furthermore, if a steady 1-D shock is desired, some form of artificial stabilization usually must be applied in order to keep the shock steady within the computational domain, since in a nominally-steady 1-D flow the shock may undergo a random walk in space. Other techniques make use of a 2D code and the

assumption that a 1-D profile extracted from a 2D bow shock simulation can be used as a substitute for a 1-D shock, but this technique leads to a good deal of wasted computational effort since most of the 2D flowfield is not used for the desired profile.

Due to these concerns, we have chosen instead to simulate the development of an unsteady 1-D shock. The flow in the domain is initialized with a Maxwellian velocity distribution at the freestream (pre-shock) temperature, with a bulk velocity moving to the right. The right boundary of the domain is set as a specular wall. The left boundary of the domain represents a freestream; it generates molecules entering the domain with a Maxwellian velocity profile (offset by the freestream velocity) at the freestream temperature. We use a uniform spatial grid throughout the 1D domain. With this initial setup, at the beginning of the simulation an unsteady normal shock forms and begins propagating to the left. The initial setup and early shock motion are shown in Figure 2.1.

After a substantial amount of time has passed (50% of the total number of time steps for the run), the shock has moved a significant distance away from the right boundary, and the code begins sampling the upstream (pre-shock) and downstream (post-shock) pressures. The sampling region for the downstream pressure is offset slightly from the edge of the domain on the right side so that the pressure is not altered by the localized effects of the wall boundary. The sampling regions are shown in the first image of Figure 2.2. The shock continues to move while this sampling takes place over the course of a number of time steps (10% of the total length of the run).

After this period of sampling the upstream (pre-shock) and downstream (post-shock) pressures are known very accurately, and at this point the code begins to track the shock location. At each sample interval, the pressure is calculated at every point in the domain. The pressure at any given point is calculated from the gas equation $P = \rho RT$, where R is the gas constant for the mixture, ρ is the mass density of the mixture, and T is

the total temperature of the mixture. After the pressure is calculated, a normalized pressure is obtained for each point based on the equation

$$P_{norm} = \frac{P - P_1}{P_2 - P_1} \quad (2.6)$$

where P is the pressure at a given x -location, P_1 is the pre-shock pressure, and P_2 is the post-shock pressure. This normalized pressure is then boxcar averaged to obtain a smoothed value of P_{norm} in each cell. The shock location is defined as the location at which this boxcar averaged P_{norm} is equal to 0.5, as shown in the second image of Fig. 2.2. Note that this location does not need to correspond to anything of particular physical importance in the shock profile, it is only important that the location be consistently defined so that it moves with a constant speed and stays fixed relative to the shock front. The shock location is tracked over a period of time (10% of the total length of the run) in order to obtain a precise shock propagation speed, as shown in the third image of Fig. 2.2.

Once a shock propagation speed has been obtained a set of sample cells follows the motion of the shock, with the set of sample cells moved each time sampling is performed, based on the previously determined shock propagation speed. The final image of Fig. 2.2 shows the sample region which propagates with the shock. In this sample region, the shock can be viewed as steady.

PARALLEL IMPLEMENTATION

The DSMC code is MPI parallel. When simulating a 0-D relaxation ensemble averaging allows for multiple processors to work on a single simulation; each processor runs its own version of the relaxation with a different random number seed and the results are averaged together at the end. This is the only practical means of running a single cell DSMC calculation in parallel, and it is also very efficient. When a 1-D shock is

simulated each processor is initially assigned a set of contiguous cells, and a given processor handles all movement, indexing, collisions, reactions, and any other required work on the particles within its set of cells. A particle which leaves the domain of one processor is sent to the processor which owns the cell into which the particle has moved. Periodically, domain rebalancing is performed so that the work performed in a given time step is well balanced between all processors.

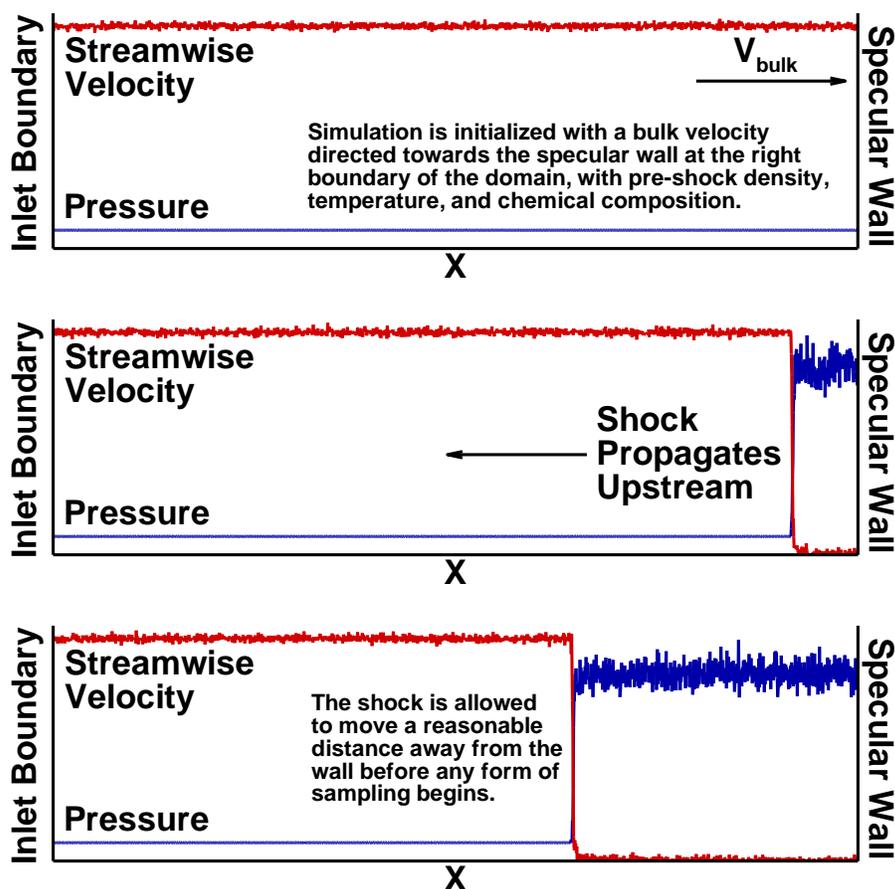


Figure 2.1: Schematic showing the initial setup of the domain, the boundary conditions, and the early motion of the shock.

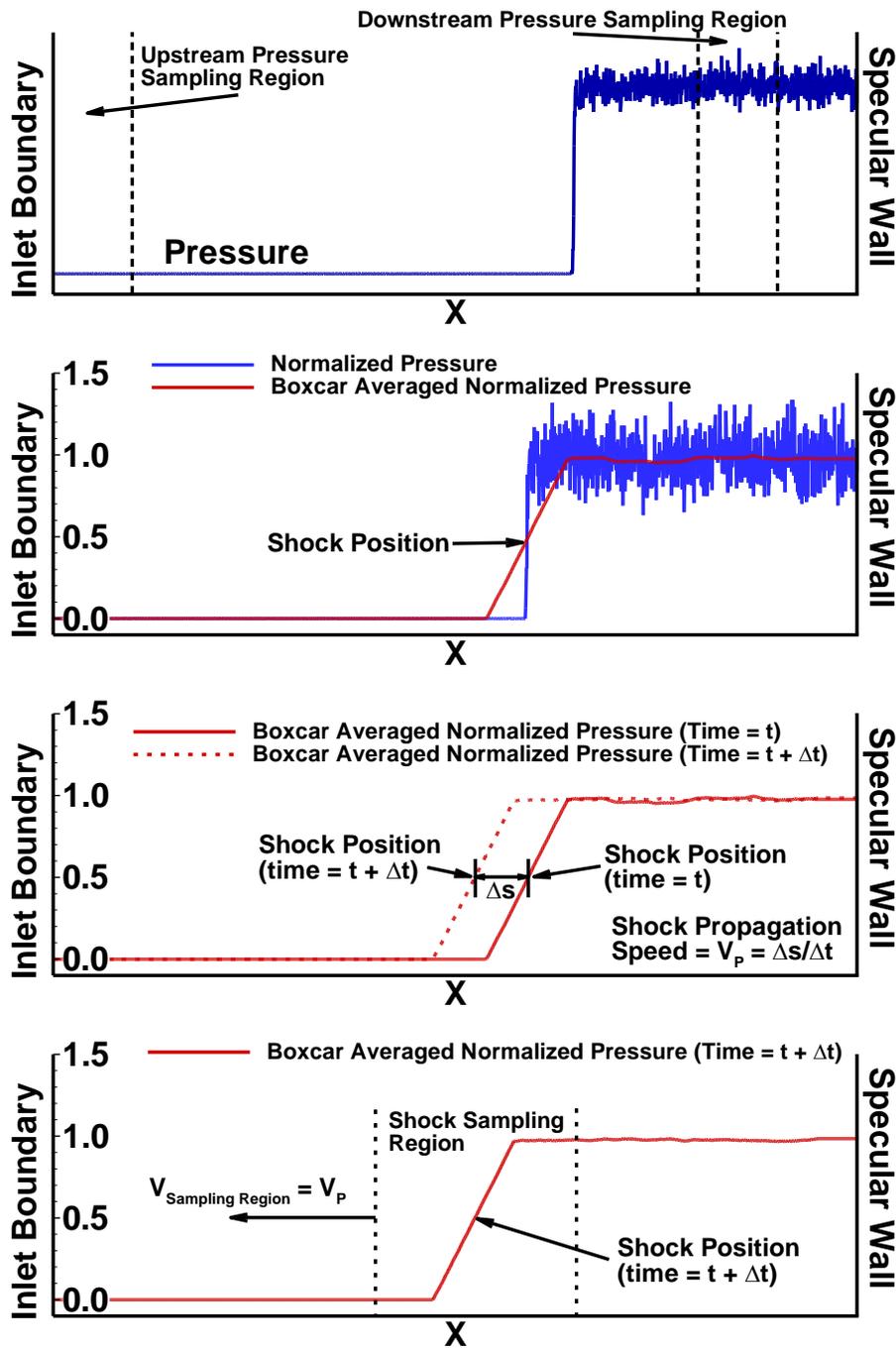


Figure 2.2: Schematic showing the pressure sampling process, the identification of the shock location, the calculation of the shock speed, and the location and motion of the shock sampling region.

VERIFICATION

In order to test the modified TCE model used in this work and examine how well the DSMC code reproduces the VHS collision rates and the Arrhenius reaction rates, we ran a series of single step runs at various temperatures. For each such run, a 0-D box of molecules was initialized with equal number fractions of N_2 , N , O_2 , O , and NO , and a total number density of $1.0 \times 10^{23} \text{ \#/m}^3$. The ratio of real to simulated particles was chosen so that there were $\sim 5,000,000$ simulated particles in the box ($\sim 1,000,000$ of each species). The code was then run for a single time step and the number of VHS collisions and the number of each type of reaction were tabulated. These tabulated values were then ensemble averaged over a total of 320 single-step runs (each with a different random number seed). Since the focus was on instantaneous rates, the reactions and collisions were not actually performed. Instead the code simply identified what type of interaction was chosen and then left the properties of the colliding particle unchanged. This was so that the properties of the gas did not change over the course of the single time step run. To further ensure that the rates we are examining are representative of instantaneous rates at the given conditions, the time step chosen was short enough that the vast majority of particles were not chosen for collisions or reactions at all. Finally, because the recombination rates are so low, it is very difficult to get reaction rates from the DSMC code which are not dominated by statistical noise. In order to examine the recombination rates, the entire process described above was performed a second time, after increasing the pre-exponential constant for the recombination rates by six-orders of magnitude.

The Arrhenius rates are expressed as functions of a single temperature, and they are not very meaningful when local thermal equilibrium does not exist. Therefore, in these test cases all species were initialized with a given overall temperature. For the monatomic species this was simply the translational temperature, but for the diatomic

species the distribution of internal states was initialized to an equilibrium at the given temperature (i.e. $T_{\text{rot}} = T_{\text{vib}} = T_{\text{trans}} = T_{\text{ov}}$). The TCE model does not require thermal equilibrium, it is only necessary in order for comparison with Arrhenius reaction rates to be meaningful. The above process was performed at a total of 64 temperatures between 5000 K and 25000 K, and in Figure 2.3 the results from the DSMC code are compared with VHS collision rates based on kinetic theory and reaction rates based on the Arrhenius rate equations. For clarity of the images, collision rates are only shown for some of the species combinations, and reaction rates are only shown for some of the reactions. The VHS collision rates, dissociation reaction rates, and exchange reaction rates all come from the case with nominal recombination rate parameters. As mentioned above, in order to get results which are not dominated by noise the recombination rate comparison is performed with much larger pre-exponential constants for all the recombination reactions (the predicted Arrhenius recombination rates are also calculated with this higher pre-exponential constant, so the comparison between DSMC and predicted rates is still valid). Note the extremely good agreement between the predicted and actual rates for both VHS collisions and chemical reactions. Those rates not shown have also been examined, and show similarly excellent agreement.

PROOF OF NECESSITY OF THE TCE MODIFICATION

The modified TCE model used in this work does incur some additional computational expense. How large of an additional expense depends on the ratio of selections to VHS collisions, since the modification consists of performing the calculation of $\sigma_{\text{R}}/\sigma_{\text{VHS}}$ whenever a pair of particles is selected, rather than only if the pair is chosen for a VHS collision. In the work presented here, the computational expense of

each simulation was increased by slightly less than a factor of two when using the modified model as compared to when using traditional TCE. Due to this additional computational expense, it is important to justify the use of the modified model.

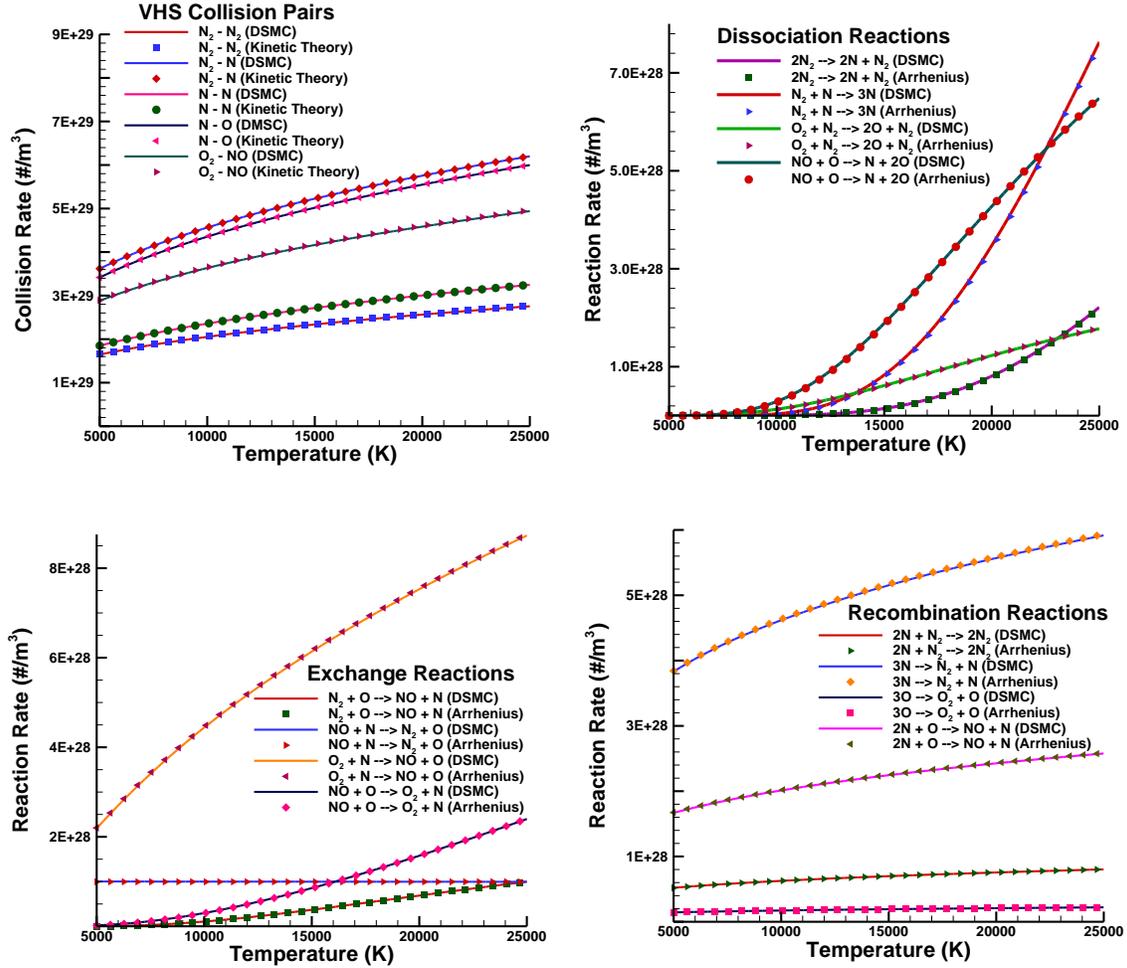


Figure 2.3. Instantaneous rates for VHS collisions and for selected reactions over a range of temperatures. Number densities of N_2 , N , O_2 , O , and NO were all set at 2.0×10^{22} $\#/m^3$. The temperatures shown are total temperatures, and the diatomic species were initialized with their rotational and vibrational modes in equilibrium with the translational mode. Note the excellent agreement at all temperatures.

While the traditional TCE model does fine with the nominal reaction rates, reaction rates in this work are varied over a range of two orders of magnitude and in some regions of the parameter space the majority of the reaction rates will be much higher than the nominal values. Our code must be able to give accurate results even in those cases, which means that the Arrhenius rates must be reproduced even when the reaction rates are substantially higher than the VHS collision rates. In order to test the modified TCE model against traditional TCE, we performed another set of simulations similar to that described above, this time with Λ for every reaction set ten times higher than the nominal value. A set of 64 temperatures was run with the modified TCE model and another set was run with traditional TCE. The results for the two procedures are shown in Figure 2.4. It is very clear from the figure that if we want to accurately model the higher reaction rates we must use the modified model. Errors of greater than a factor of three are present at higher temperatures when using the traditional TCE model. This is not surprising, since the rates for these reactions are higher than the VHS collision rates between the reactant species. It is simply not possible for the traditional model to get the rates right, since with that model reactions are a subset of VHS collisions and there are not enough VHS collisions to go around. Furthermore, in the case of $\text{O}_2 + \text{N} \rightarrow \text{NO} + \text{O}$ this is not even the only reaction between these reactants; a dissociation reaction is also possible and occurs at a high rate. The reaction $\text{N}_2 + \text{N} \rightarrow 3\text{N}$ is the only reaction between these reactants and thus the error for this reaction, while still nearly a factor of two, is less than the error for $\text{O}_2 + \text{N} \rightarrow \text{NO} + \text{O}$.

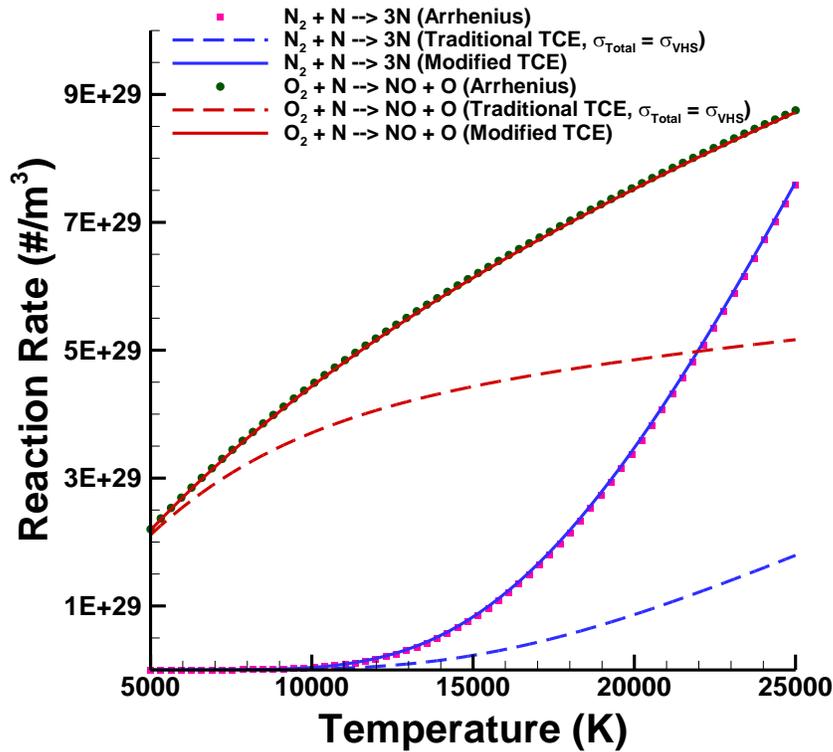


Figure 2.4. Instantaneous rates for selected reactions over a range of temperatures. Initial conditions are the same as in Fig. 2.3, but Λ for every reaction is ten times higher than the nominal value. At higher temperatures, the Arrhenius rates for the reactions shown are higher than the VHS rate for collisions between the reactants. The modified TCE model used in this work matches the Arrhenius rates in spite of this, but the traditional TCE model (which assumes that σ_{VHS} can be used to approximate σ_{Total}) is not able to match the Arrhenius rates in this situation.

In order to further demonstrate the need for the modified TCE model in this work, we also ran two shock simulations (with the same initial conditions as the shocks for the sensitivity analysis which will be described in Chapter 6), and in both of these simulations Λ for every reaction was ten times higher than the nominal value (this represents the extreme corner of the parameter space which will be examined in the sensitivity analysis). We used the modified TCE model in one of these simulations and traditional TCE in the other, and the results for ρ_{NO} (which will be our main quantity of

interest in the sensitivity analysis for the 1-D shock) are shown in Figure 2.5. There are very substantial differences in the two profiles in the non-equilibrium region of interest, and thus we again conclude that we must use the modified TCE model in this work.

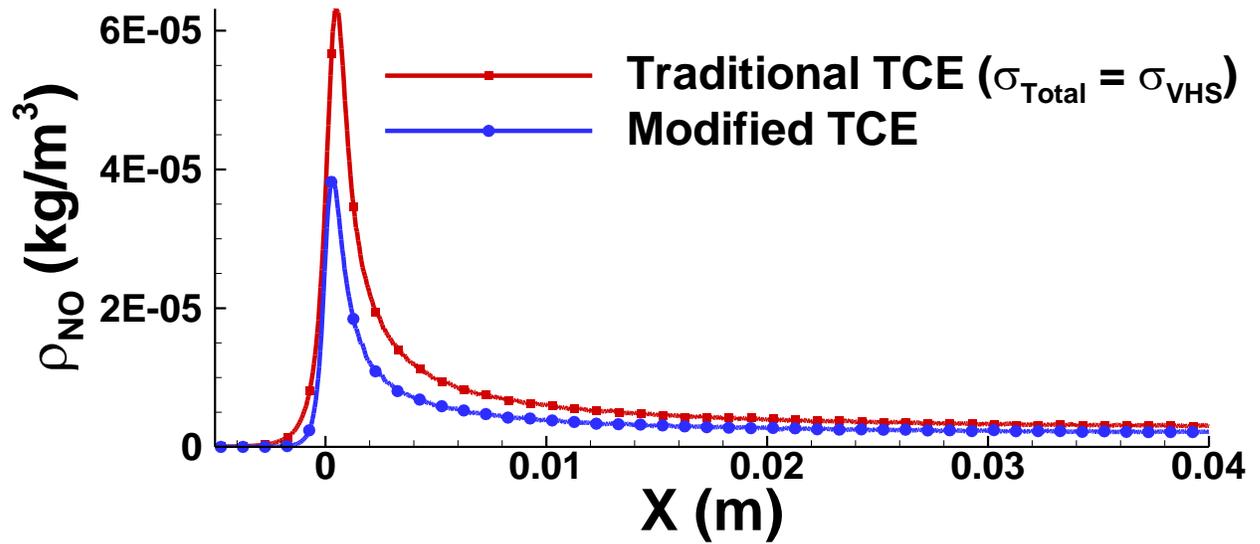


Figure 2.5. Profiles of ρ_{NO} inside a shock at ~ 8000 m/s. Note that the profiles differ significantly depending on whether traditional or modified TCE is used. For these runs, Λ for every reaction is ten times higher than the nominal value.

Chapter 3: Sensitivity Analysis Methodology

OVERVIEW

We will use two sensitivity analysis methods in this work, one based on the square of the Pearson correlation coefficient (r^2) and the second based on the mutual information. Both of these methods are global in nature and require a Monte Carlo sampling of the parameter space. This sampling can be quite time consuming, but fortunately we can use the same data set for both methods.

Once the sampling of the parameter space is complete, we calculate r^2 and the mutual information for each parameter. In cases where the QoI is a vector (i.e. we have values for the QoI at a set of discrete points in space or time) we treat each component of the vector as a distinct, scalar QoI, and we calculate r^2 and the mutual information for each of these scalar QoIs. We can then plot sensitivities as a function of x (or time, as appropriate).

SAMPLING THE PARAMETER SPACE

The type of global sensitivity analysis described here is distinguished from a local sensitivity analysis in several ways. First, in a global analysis all of the parameters are varied simultaneously while in a local analysis the parameters are usually varied one at a time. Second, the global analysis allows the parameters to vary over the entire parameter space, as opposed to only in a small range around their nominal values as is usually the case with a local analysis. Furthermore, local analyses are usually based on calculating partial derivatives of the QoI with respect to each parameter, while no derivatives are calculated in the global analysis presented here. Instead, the global sensitivity analysis is used to estimate the relative contribution of the uncertainty of each parameter to the overall uncertainty of the QoI.

There are a number of methods for performing a global sensitivity analysis. Many of the most advanced methods are intrusive, meaning that they require changes to the algorithm and/or equations used to simulate the physics. Others require the creation of a surrogate model to be used in place of the actual model during the sensitivity analysis. Due to the fact that DSMC simulations do not directly involve the solution of a set of differential equations, common intrusive methods which are based on rewriting differential equations are not practical. Creating a surrogate model is also a challenge and proper creation of a surrogate model is a field of study in and of itself. Due to these limitations, for this work we have chosen a Monte Carlo based approach which is non-intrusive and which does not require the creation of a surrogate model. This choice is also informed by the goal of making the techniques used here suitable for immediate application by the DSMC and rarefied gas dynamics communities, and a somewhat simpler, easier-to-understand method which does not require great knowledge of abstract mathematics is more likely to be used in the near future by those communities.

The Monte Carlo method requires a sampling of the parameter space, and the dataset created by this sampling provides the input for the sensitivity analysis techniques. The sampling of the parameter space proceeds as follows:

- 1.) Boundaries are established for each parameter. In Bayesian terms, this means that we establish prior distributions for all of the parameters. These priors are based on a combination of hard physical limits when those are present for a given parameter within a given model, review of the available literature, and expert judgment. For example in the VHS model a value for ω which is less than 0.5 or greater than 1 would be physically incorrect. The priors are intended to incorporate estimates of the uncertainty in these parameters, so that parameters whose values are considered better

known should be bounded more tightly than parameters whose true values are more uncertain.

In principle, these prior distributions could be uniform, Gaussian, or any other appropriate distribution. Using a Gaussian prior (presumably centered at the nominal value of the parameter from expert judgment or the literature) implies some degree of confidence in that nominal value, with the level of confidence depending on the width of the Gaussian used. In our case, we have very little confidence in the nominal values for many of our parameters, and the nominal value often depends on what particular prior work is referenced. Therefore, in this work we will use the least informative possible prior, a uniform distribution over a broad range for each parameter. The size of the range will be set for any given parameter based on an estimate for the uncertainty in that parameter, with the goal being to err on the side of overestimating the uncertainty rather than underestimating it.

2.) A value for each parameter is selected based on a random number draw from the prior distribution for that parameter. This process is performed individually for each parameter (i.e. an independent random number is drawn for each parameter, so that the values of the various parameters at a sample point are independent of one another).

3.) A simulation is run for the scenario being examined (a 0D relaxation at a given initial temperature, a 1D shock with a given shock velocity, etc.) with the parameter values selected in the previous step. The results of this simulation are stored for later use. Specifically, the value of a scalar QoI or a set of values of a vector QoI are saved. For the time being, it will be assumed that the QoI is scalar (i.e. the value of the QoI can be represented as a single number).

4.) Steps 2 and 3 are repeated a pre-determined number of times. Each sample is independent of all of the others.

The end result of the above process is a dataset containing the model output for the QoI at a set of sample points in parameter space. Thus, for each sample point we have $N_{\text{parameters}} + 1$ numbers, corresponding to the values for each of the parameters and one value for the QoI. This dataset can then be statistically analyzed to determine sensitivities (based on the given QoI) for each parameter relative to the other parameters. In order to calculate a sensitivity for each parameter, the full dataset is projected from its original $N_{\text{parameters}} + 1$ dimensional space onto the two-dimensional space defined by the given parameter and the QoI. This is shown for an example QoI and an example parameter in Figure 3.1. The Pearson correlation coefficient and the mutual information for a given parameter can then be calculated based on this two-dimensional projection of the full dataset.

PEARSON CORRELATION COEFFICIENT

The Pearson correlation coefficient is the first of the two statistical measures used in this work to quantify sensitivities for a set of parameters. The Pearson correlation coefficient is given by the equation

$$r = \frac{\sum_{i=1}^{N_{MC}} (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{N_{MC}} (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^{N_{MC}} (Y_i - \bar{Y})^2}} \quad (3.1)$$

where N_{MC} is the number of Monte Carlo samples of the parameter space in the dataset, X_i is the value of a given parameter at the i^{th} sample point from the dataset, and Y_i is the value of the QoI which is output by the model (for the given scenario and with parameter values corresponding to the i^{th} sample point from the dataset). We are not particularly interested here in whether the correlation is positive or negative, and so we will actually use r^2 (the square of the Pearson correlation coefficient) as our measure of sensitivity.

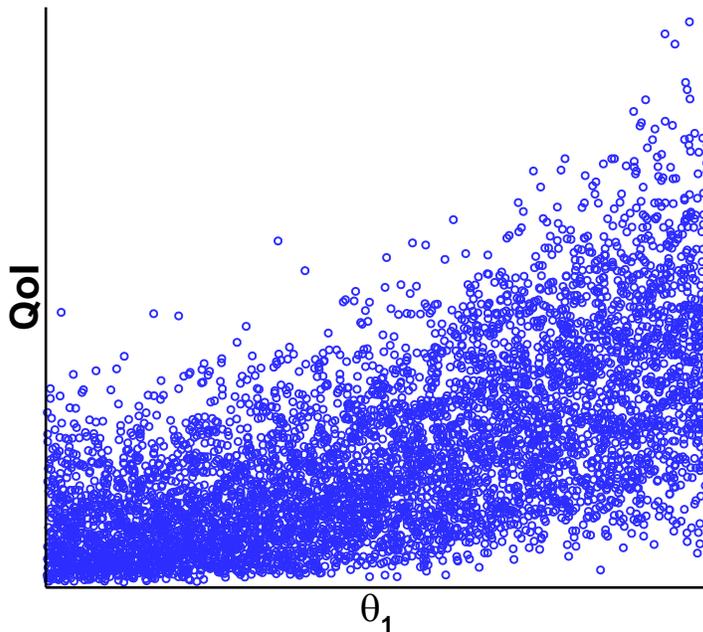


Figure 3.1. Scatterplot showing values of a hypothetical QoI vs. values of a parameter θ_1 . The dataset is obtained from a Monte Carlo sampling of the full parameter space (for this hypothetical example), and is projected onto the two-dimensional space shown.

The fact that scatterplots such as that shown in Figure 3.1 are projected onto a two-dimensional space from the full $N_{\text{parameters}} + 1$ dimensional space means that the variation of the QoI with respect to all of the parameters is inherently incorporated into the scatterplot, and it is for this reason that the r^2 value for the scatterplot is useful as a measure of sensitivity for a given parameter. If a given parameter does not have a strong effect on the QoI, then the r^2 value for a scatterplot of that parameter vs. the QoI will be low, because the variance of the QoI will be mostly explained by other parameters. If a parameter has a strong effect on the QoI, however, then a relationship will be visible in the scatterplot, and the corresponding r^2 value will be relatively high. Figure 3.2 shows a set of scatterplots which correspond to four different relationships between a hypothetical QoI and a hypothetical parameter θ_1 . In the upper left image, there is clearly almost no

relationship between the parameter and the QoI. Virtually all of the variance of the QoI must be explained by other parameters which were varied in the Monte Carlo sampling of the parameter space. This qualitative analysis is also borne out by a very low r^2 value. In the upper right image, it is apparent that the value of the parameter does have some effect on the value of the QoI, but this effect is largely washed out by the variance of the QoI due to all of the other parameters, resulting in a fairly low r^2 value. In the lower left image, a strong relationship between the parameter and the QoI is apparent, to the point that the effect of this parameter on the QoI appears to be stronger than the effect of all of the other parameters combined. This is confirmed by a high r^2 value. Finally, in the lower right image, the variation of the QoI is almost completely explained by this one parameter, and the r^2 value approaches the maximum possible value of 1.

MUTUAL INFORMATION

The square of the Pearson correlation coefficient has the drawback that it cannot accurately measure highly non-linear correlations, and we do not know *a priori* whether the correlations between our parameters and our QoI will be linear. As an example, the scatterplot in Figure 3.3 shows an obvious relationship between a hypothetical parameter and QoI, but the value of r^2 is nearly zero because of the clear non-linearity of the relationship.

A more sophisticated measure of sensitivity can be obtained from the mutual information. In the context of sensitivity analysis, the mutual information represents a measure of the difference between two PDFs. One of these PDFs is the actual joint probability distribution for a given parameter and the QoI, and the other is a hypothetical joint probability distribution for a case where the QoI and the parameter are assumed to

be independent. The calculation of mutual information in this work largely follows the procedure described in the work of Steuer *et al.* (2002). The process is shown in schematic form in Figures 3.4 - 3.7.

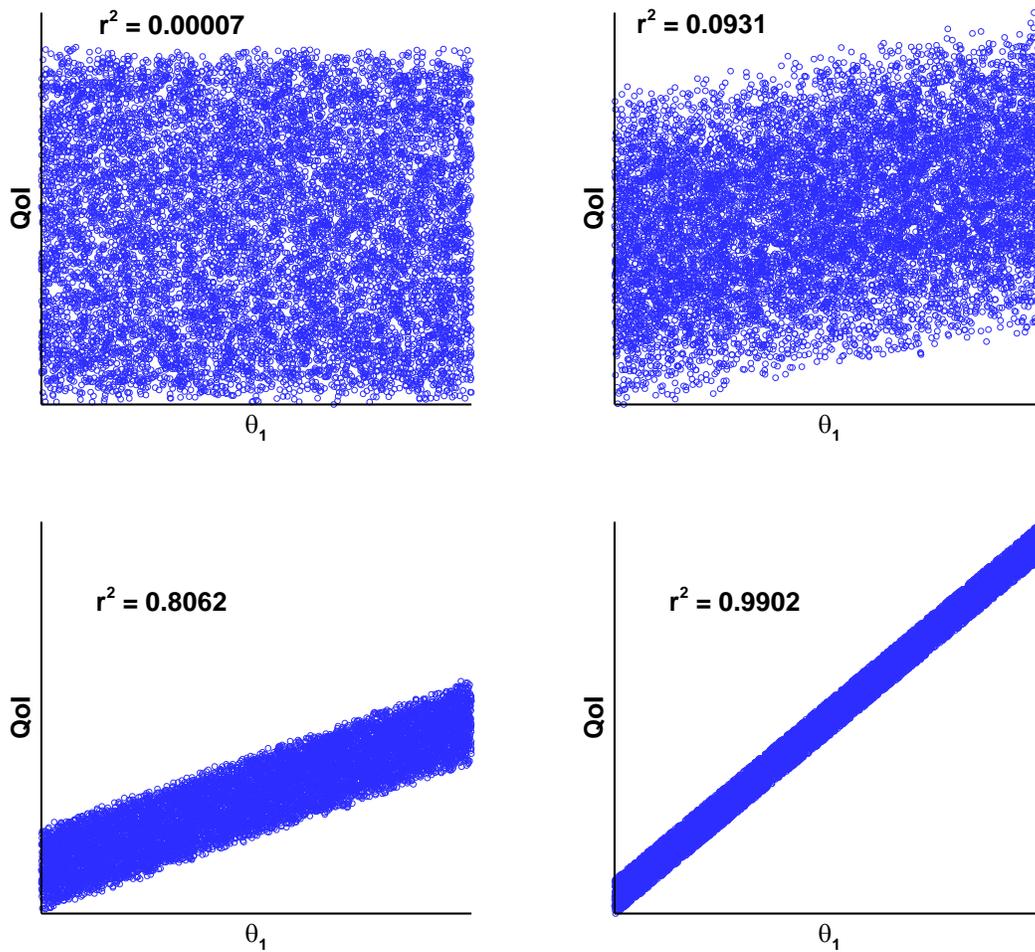


Figure 3.2. Scatterplots and corresponding r^2 values for different hypothetical relationships between a QoI and a parameter θ_1 . Scatterplots which are tightly packed around an obvious trendline lead to high values of r^2 , and indicate a strong relationship between θ_1 and the QoI.

The first step in the process is to normalize the data so that it has a mean of zero and a standard deviation of one. This normalization process preserves any correlation

which is present in the data. The next step is to estimate the true 2D joint PDF (called $p(\theta_1, QoI)$ in the schematic) based on the scatterplot data. There are many ways to estimate the joint PDF, as discussed previously in Chapter 1. In the simple histogram based method which is illustrated in Fig. 3.4, the two-dimensional space is divided into bins, and the estimate of the PDF within any given bin is based on the ratio of the number of sample points which fall into that bin to the total number of sample points. The PDF is appropriately normalized so that $\int_{\theta_1} \int_{QoI} p(\theta_1, QoI) dQoI d\theta_1 = 1$. In this work, kernel density estimation (KDE) is also used to estimate the joint PDF from the scatterplot data. When using KDE, a localized PDF (called a kernel) is placed at each scatterplot point, and the value of the joint PDF at any given location is approximated by the sum of the contribution from the kernels of all of the scatterplot points. In this work, we use only Gaussian kernels with the KDE method. Comparisons between the histogram and KDE methods will be performed in the next section.

We also use the scatterplot data to generate 1D, marginal PDF's of the QoI and θ_1 , as shown in Fig. 3.5 (again making use of either histogram bins or KDE). Then, these 1D PDF's are multiplied together to generate a hypothetical 2D PDF, as shown in Fig. 3.6. This new PDF represents a hypothesis that the QoI is completely independent of the parameter θ_1 . Finally, the mutual information is defined as the Kullback-Leibler divergence between the true 2D joint PDF and the hypothetical 2D PDF, and is given by the equation

$$Mutual\ Information = \int_{\theta_1} \int_{QoI} p(\theta_1, QoI) \left[\ln \left(\frac{p(\theta_1, QoI)}{p(\theta_1)p(QoI)} \right) \right] dQoI d\theta_1 \quad (3.2)$$

where the integrand is understood to be zero when $p(\theta_1, QoI)$ is zero. The calculation of the mutual information based on Eq. 3.2 is shown in schematic form in Fig. 3.7. Since the mutual information can be viewed as a measure of the distance between the true joint PDF and the hypothetical PDF which would exist if the parameter and the QoI were

independent, it thus represents a measure of the sensitivity of the QoI to the parameter θ_1 . A value of zero for the mutual information guarantees independence of the QoI and the parameter (unlike a value of zero for r^2 , which could result from r^2 failing to capture a non-linear relationship between the QoI and the parameter).

The fact that the mutual information can capture non-linear relationships is demonstrated by the scatterplot in Figure 3.3. As discussed earlier, the non-linearity of the relationship between the parameter and the QoI leads to a value of r^2 which is very close to zero. The mutual information for this scatter plot is non-negligible, however, because the mutual information captures the non-linear relationship properly.

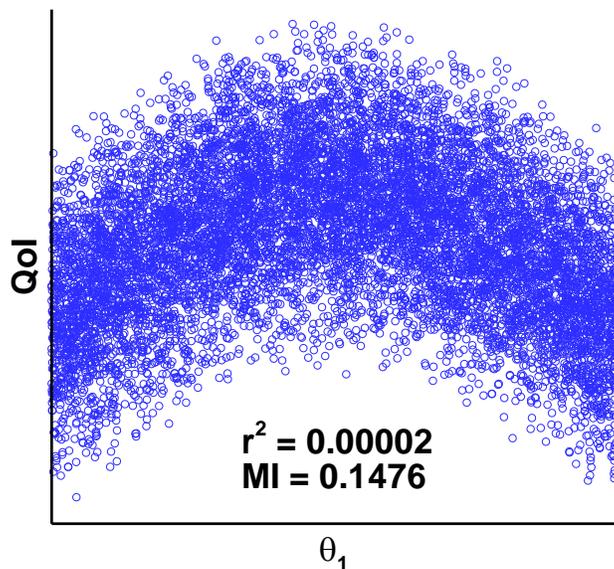


Figure 3.3. Scatterplot showing values of a hypothetical QoI vs. values of a parameter θ_1 . The relationship between the parameter and the QoI is clearly non-linear, which results in a near-zero value for r^2 . The mutual information properly captures the non-linear relationship between θ_1 and the QoI.

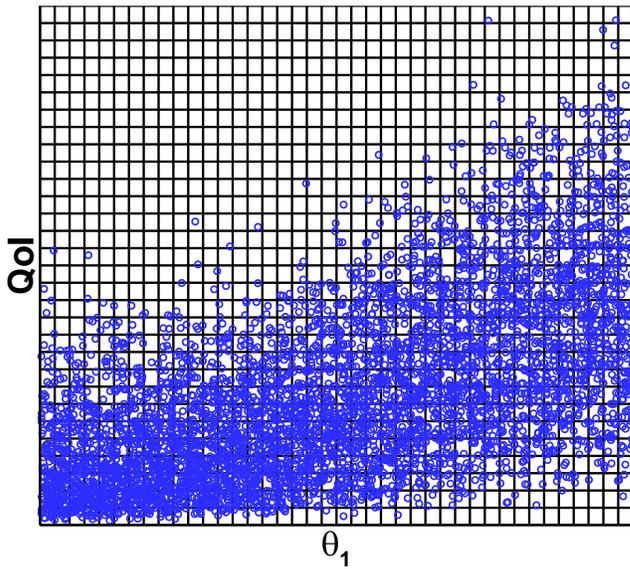


Figure 3.4. Estimation of the joint PDF for a hypothetical parameter and QoI, by means of a simple histogram based method.

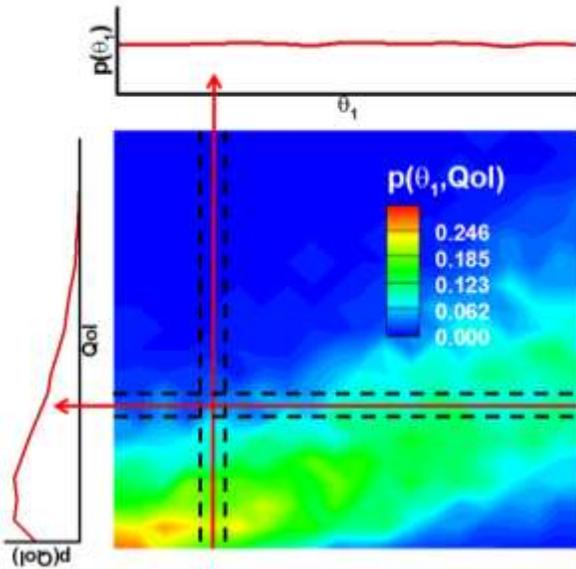


Figure 3.5. Schematic for the next step in the calculation of the mutual information. The normalized scatterplot data are used to determine the 2D, joint PDF for the QoI and the parameter θ_1 , and also to determine (by integration along rows and columns) 1D, marginal PDFs of the QoI and the parameter θ_1 .

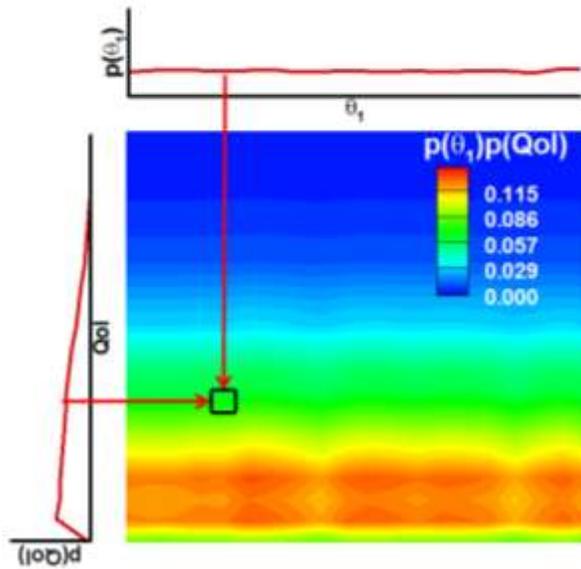


Figure 3.6. Schematic for the next step in the process of calculating the mutual information, in which the 1D marginal PDF's calculated in the previous step are multiplied together to obtain a hypothetical 2D PDF corresponding to a case where the QoI is independent of θ_1 .

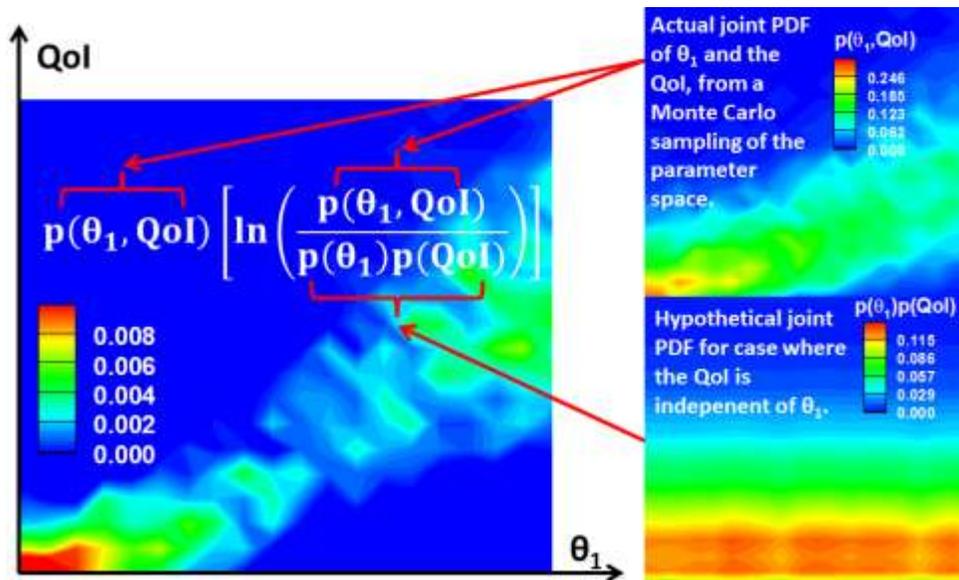


Figure 3.7. Final step in the calculation of the mutual information. The actual 2D joint PDF and the hypothetical 2D PDF are combined as shown, and the term for which contours are shown in the left image is integrated over the 2D space defined by the QoI and θ_1 , as in Eq. 3.2, to obtain the mutual information.

VERIFICATION

Before moving on, it is worthwhile to verify that the methods used here have been implemented correctly and that the codes behave as expected.

In the case of the correlation coefficient, little verification is needed. The correlation coefficient is a very simple statistical measure and the equation is easy to implement. There is a subtle distinction between the correlation coefficient and the mutual information when used as a measure of sensitivity. In the case of the correlation coefficient, the equation itself (Eq. 3.1) is defined in terms of a set of discrete points. Even if the dataset has only a very small number of points, the r^2 value from those points is still exactly the correct value *for that set of points*. It is trivial to check the results from the analysis code used here against any number of common software packages (Excel, MATLAB, etc.). The only question is whether there are enough points in the dataset to allow the r^2 value of the dataset to provide a reasonable estimate of the sensitivity for a given parameter. In the case of the mutual information, the situation is different. The overall equation (Eq. 3.2) is defined in terms of continuous PDFs. If those PDFs can be correctly estimated, the mutual information is guaranteed to provide a useful measure of the sensitivity for a parameter with respect to the given QoI. The challenging part is using the scatterplot dataset to estimate the true joint and marginal PDFs as accurately as possible, which is why a great deal of work has been done in the past on this subject. In this work, two methods were used, and both methods can be shown to properly estimate the true mutual information (for a case where the answer can be determined exactly) provided that enough data points are used.

In order to perform this verification, a simple joint PDF will be used, so that the true mutual information can be determined exactly. The equation for the joint PDF is

$$p(\theta, QoI) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(QoI-\theta)^2}{2\sigma^2}} \quad (3.3)$$

where σ is a constant. We define this PDF over the parameter range $0 \leq \theta \leq 1$. For any given value of θ , the PDF for the QoI is simply a Gaussian with standard deviation σ and mean θ . Integrating for the marginal PDFs, it is easy to see that $p(\theta) = 1$, as intended. The marginal PDF for θ is uniform so that this example is comparable with the actual sensitivity analysis method, since in this work the sampling of the parameter space is always uniform. The integration for the marginal PDF for the QoI is not as simple, but it can be shown (either with a symbolic integrator or a table of integrals) that

$$p(QoI) = \frac{1}{2} \left[\operatorname{erf} \left(\frac{QoI}{\sqrt{2}\sigma} \right) - \operatorname{erf} \left(\frac{QoI-1}{\sqrt{2}\sigma} \right) \right] \quad (3.4)$$

where $\operatorname{erf}()$ is the error function. This marginal PDF is shown in Figure 3.8, and the joint PDF is shown in Figure 3.9, both for a case with $\sigma = 0.1$.

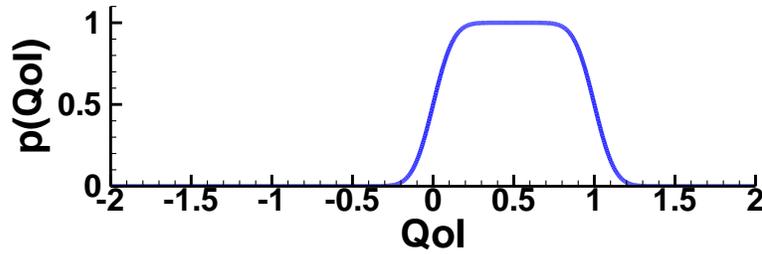


Figure 3.8. Marginal PDF for the QoI in the verification example.

In order to obtain an exact value for the mutual information, we plug $p(\theta, QoI)$ from Eq. 3.3 and $p(QoI)$ from Eq. 3.4 into Eq. 3.2, keeping in mind that $p(\theta) = 1$. The resulting equation cannot be symbolically integrated, but it can be numerically integrated with any number of common software packages in order to obtain the exact value (to any desired accuracy) for the mutual information. For the case with $\sigma = 0.1$, the mutual information is equal to 1.0643.

With this exact value in hand, we can then test our sensitivity analysis methodology. We first perform a uniform sampling of our parameter space (in this case

we have only one parameter). For each sample point, a value is drawn for θ from a uniform distribution between 0 and 1, and then a value is drawn for the QoI from the conditional distribution $p(\text{QoI}|\theta)$, which as discussed before is simply a Gaussian with mean θ and standard deviation $\sigma = 0.1$. Once we have a sample set of the desired size, we can calculate the mutual information as described in the previous section, using either histogram bins or KDE. A scatterplot for a sample set of 10,000 points is shown in Figure 3.10.

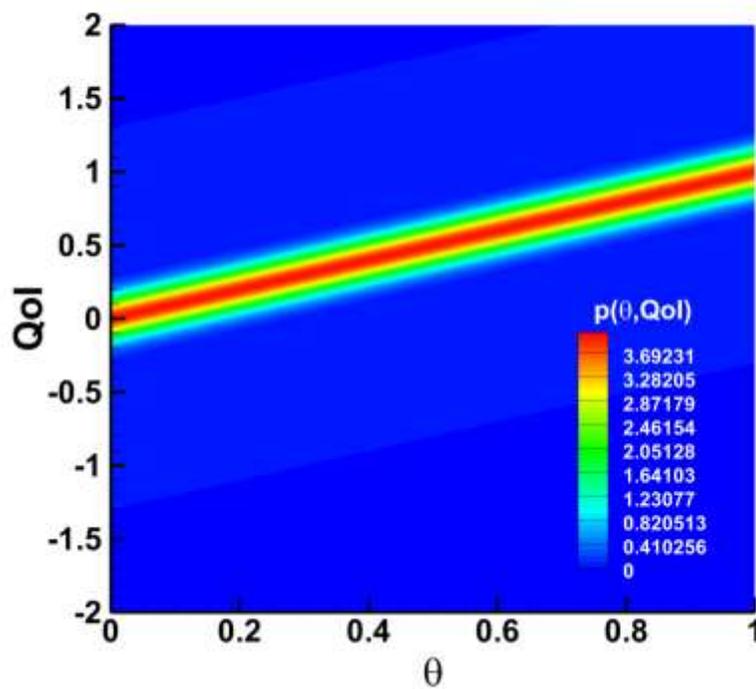


Figure 3.9. Joint PDF of θ and the QoI in the verification example.

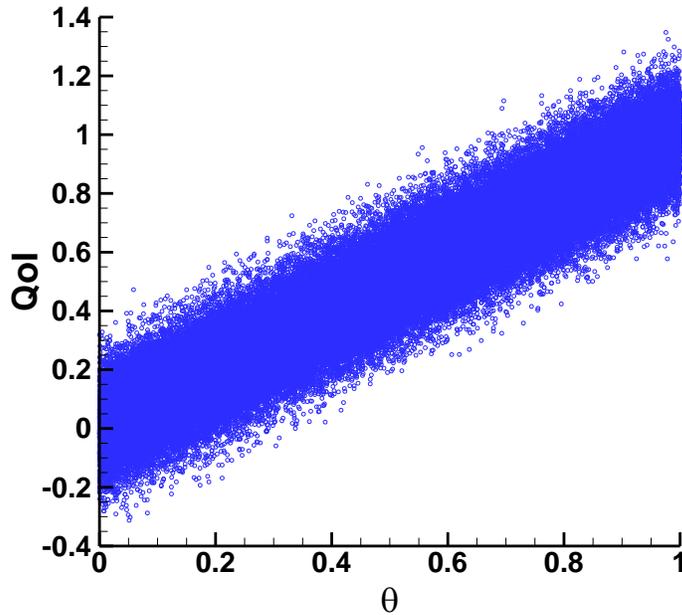


Figure 3.10. Scatterplot showing values of the QoI vs. values of θ for 10,000 sample points drawn from the joint PDF given by Eq. 3.3.

Before we can perform the calculation, we must decide what size bins to use if we are using histogram bins or we must decide the width (i.e. the standard deviation) of the Gaussian kernels if we are using KDE. As discussed by Steuer *et al.* (2002), this choice is important. If we choose bin/kernel widths which are too large, we will wash out the detail of the joint PDF and we will underestimate the mutual information. If, on the other hand, we choose bin/kernel widths which are too small, we will overestimate the complexity of the joint PDF and we will overestimate the mutual information. Unfortunately, choosing the optimal bin/kernel width would require *a priori* knowledge of the joint PDF which we are trying to estimate. For this verification test, though, we know the exact answer, and this allows us to compare the performance of the histogram and KDE methods for various bin/kernel widths.

In Figure 3.11, we plot the mutual information determined from sample sets of various sizes as a function of the width of the histogram bins used to estimate the PDFs. A number of things are apparent from the figure. First, all of the curves intersect the true value. This is to be expected, since as discussed previously, the mutual information is underestimated if the width of the bins is too large and overestimated if the width of the bins is too small. There is a distinction between these two sources of error, however. The underestimation of the mutual information which occurs if the bins are too large is essentially unrelated to the number of sample points, at least for any reasonable sample size. Adding additional sample points does not remove this error because the PDF is only defined on the scale of the histogram bins (i.e. the PDF is defined to be uniform throughout a given histogram bin), and thus adding more sample points only improves the estimation of the PDF on this scale. If large bin sizes wash out the detail of the underlying joint PDF, having additional sample points will not solve the problem.

The error which results in the overestimation of the mutual information when the histogram bins are too small, however, is dependent on sample size. This overestimation is called the spurious mutual information, and it arises because when the sample size is finite, using very small bin sizes leads to a large amount of randomness in the value of the PDF within any given histogram bin. Put another way, when resolved to a small enough scale, any scatterplot based a finite sample set will appear to show fine scale structure, but this fine scale structure will be due to randomness and not due to any genuine complexity of the PDF. Overestimating the complexity of the joint PDF in this way leads to an overestimation of the mutual information. This error source, though, can be mitigated by using larger numbers of sample points. The larger the number of sample points, the smaller the histogram bins may be without resulting in significant overestimation of the mutual information. This is borne out by the results in Fig. 3.11,

where it can be seen that when using 10,000,000 sample points, there is a broad range of histogram bin width for which the calculated mutual information essentially converges to the true value. In this range, the bins are small enough to properly resolve the joint PDF, but due to the large number of sample points spurious small scale structure is not a problem. This range of convergence is present but smaller for the 1,000,000 and 100,000 sample point cases. For the 10,000 sample point case, there is a range where the calculated mutual information remains near the true value but there is not true convergence. Finally, for the 1,000 sample point case there is no range over which the mutual information remains near the true value, the curve simply crosses the true value as it moves from underestimating to overestimating the mutual information.

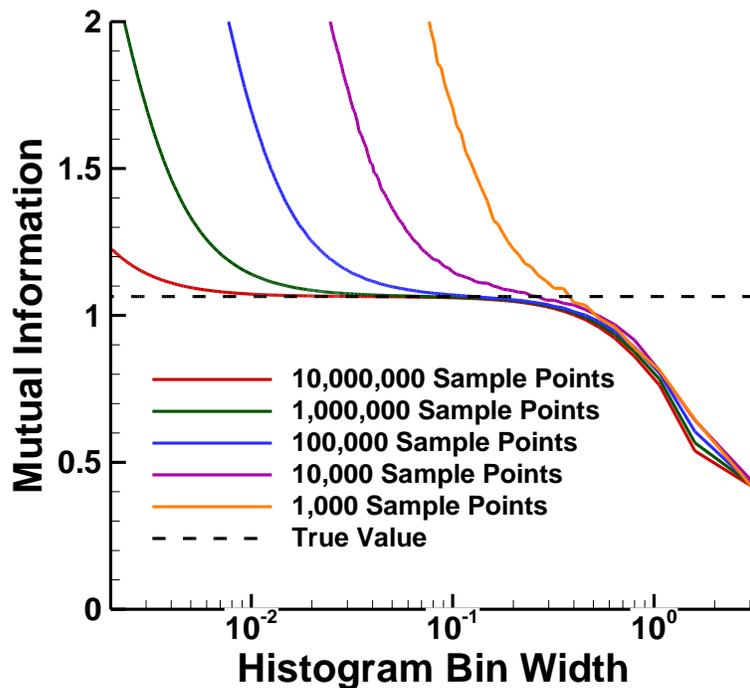


Figure 3.11. The calculated mutual information as a function of the width of the histogram bins used to estimate the PDFs. Curves are shown for several different numbers of sample points. The true value is shown for comparison.

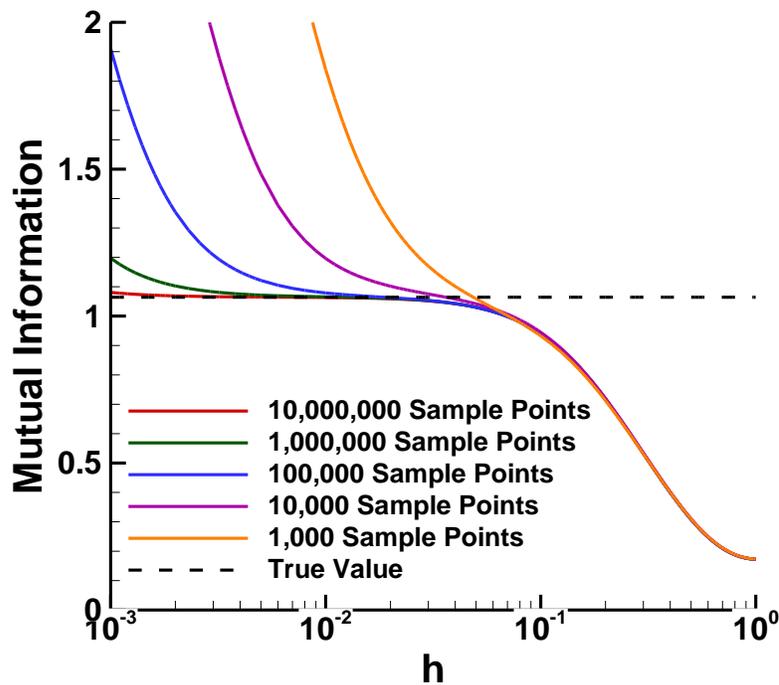


Figure 3.12. The calculated mutual information as a function of the width of the Gaussian kernel in the KDE method used to estimate the PDFs. Curves are shown for several different numbers of sample points. The true value is shown for comparison.

Figure 3.12 is similar to Fig. 3.11, except for this figure KDE is used to estimate the PDFs and the curves show calculated mutual information vs. the standard deviation (i.e. width) of the Gaussian kernels. This width is referred to as h in the figure and discussion, following the notation of Steuer *et al.* (2002). The curves seen in this figure are very similar to those seen in Fig. 3.11, and the same trends are present. If h is too large the mutual information is underestimated, and if h is too small the mutual information is overestimated. When larger sample sets are used, there is a range of h over which the mutual information converges to the true value. The fact that this range of convergence exists (both when using histogram bins and when using KDE to estimate

the PDFs) is important, because it provides some verification that our method for calculating the mutual information from a dataset has been correctly implemented.

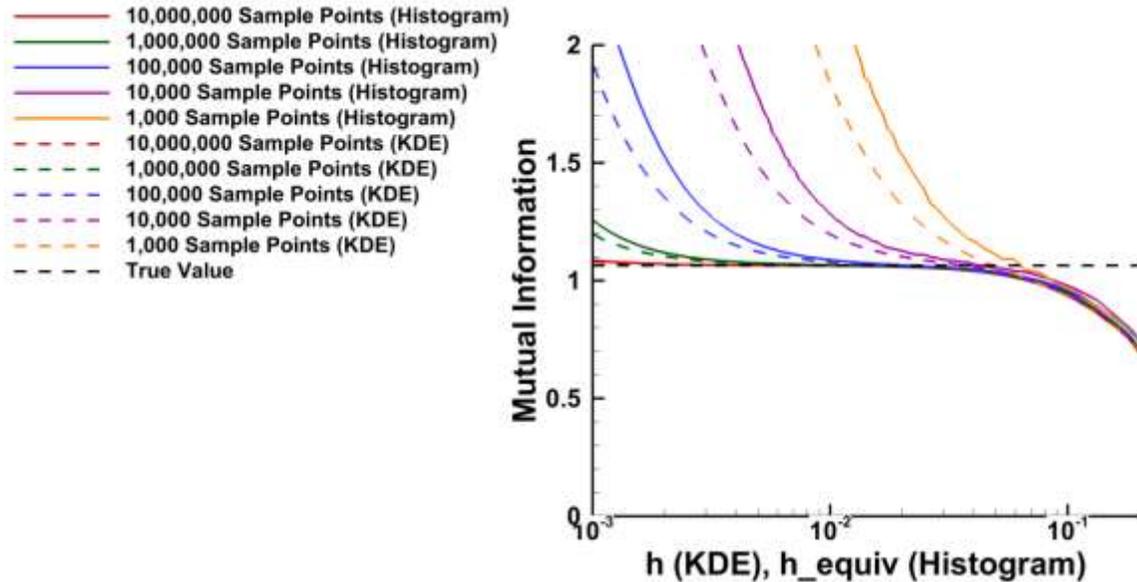


Figure 3.13. The calculated mutual information as a function of the kernel width for the KDE method and the equivalent kernel width for the histogram method. Curves are shown for several different numbers of sample points. The true value is shown for comparison.

We would like to directly compare the histogram and KDE methods, but in order to do this we must decide how to define a relationship between histogram bin width and the width of the Gaussian kernel (i.e. we must have an equation that, for a given histogram bin width, provides an “equivalent” Gaussian kernel width). The trivial solution is to treat the histogram bin width as equivalent to the Gaussian kernel width, but this is not a proper choice. The height of a Gaussian is still more than half its maximum value at a distance one standard deviation from its center. After some empirical examination, it was found that a good equivalent Gaussian kernel width is equal to $1/6^{\text{th}}$ of the histogram bin width. With this Gaussian width, if the Gaussian were centered at the center of the histogram bin then a $\pm 3\sigma$ range of the Gaussian (and thus the vast

majority of the area under the Gaussian curve) would be contained within the histogram bin.

Making use of this equivalent Gaussian kernel width for the histogram cases, in Figure 3.13 we plot curves for calculated mutual information vs. kernel width for both methods of estimating the PDFs. It is apparent from the figure that our equivalent width is well chosen, because the underestimation of the mutual information due to overly large kernel width becomes apparent at very nearly the same point on the x-axis for both the histogram and KDE cases. It is also apparent from this figure that for any given number of sample points, the range of convergence for the KDE method is slightly larger than the range of convergence for the histogram method. For this reason, we elect to use KDE to estimate the PDFs when calculating the mutual information in the remainder of this work.

SENSITIVITY ANALYSIS EXAMPLES

Many of the important features of the sensitivity analysis methodology used in this work can be illustrated with a few simple examples. For these examples, we use a hypothetical QoI which can be expressed as a simple analytical function of a single spatial variable. This analytical function serves as our “model” in this example, and the model equation requires three parameters. The equation is

$$QoI(x) = (A * \text{sqrt}(x) + B * x + C * x^2)e^{-x^2} \quad (3.5)$$

where A, B, and C are the three parameters. This QoI is not intended to represent any physical quantity relevant to hypersonic shocks, it is simply an example for illustrative purposes. In this example, each parameter will be allowed to take on values between 0 and 1, and the spatial variable x falls within the range $0 < x < 2$. In Figure 3.14, curves are shown for QoI(x) for a variety of values of A, B, and C (all between 0 and 1).

Before going any further we must address the fact that our QoI is not a single scalar value, rather it is a function which will take on multiple values as x varies over the range from 0 through 2. This is the type of QoI we will be dealing with when performing sensitivity analyses for both the 0D relaxation (where the spatial variable x is replaced by time) and the 1D shock (where x represents the distance from the shock front). Since we are dealing with numerical simulations, QoIs for those cases will not be continuous; instead they will be vectors where each component of the vector corresponds to a discrete point in time or space. We will treat the QoI for this example in the same way; we will obtain output from our “model” at a series of discrete x locations. However, the sensitivity analysis techniques described in the previous sections are intended to be used for a scalar QoI. For this reason, we treat each component of the vector QoI (i.e. the value of the QoI at each discrete point in x) as an individual, scalar QoI, and we carry out the full analysis for each scalar QoI. The results of these individual analyses are then combined to give us a full picture of the sensitivities for the various parameters as a function of x . This concept is shown in schematic form in Figure 3.15.

With our set of scalar QoIs and our parameters defined, we then carry out a Monte Carlo sampling of the parameter space (using 100,000 sample points), and based on this sampling we calculate sensitivities based on both r^2 and the mutual information for all three parameters for each scalar QoI. We use KDE when estimating the PDFs during the calculation of the mutual information, and thus we need to choose a width, h , for the Gaussian kernel. In the previous section we showed that picking an inappropriate width for the kernel can lead to significant underestimation or overestimation of the mutual information. In this example (and in the sensitivity analyses for the 0D relaxation and the 1D shock in subsequent chapters) we follow the technique of Steuer *et al.* (2002) and use a value for h which can be mathematically shown to be optimal if the PDFs being

estimated are Gaussian. This value of h , which will be referred to as h_{opt} (following the notation of Steuer *et al.*) is given by the equation

$$h_{opt} = \sigma \left(\frac{4}{d+2} \right)^{1/(d+4)} N_{MC}^{-1/(d+4)} \quad (3.6)$$

where σ is the standard deviation of the data (when estimating a 2D PDF, σ is the average marginal standard deviation of the two variables), d is the dimension of the PDF being estimated, and N_{MC} is the number of Monte Carlo sample points. In our case, we have normalized the datasets so that $\sigma = 1$. Thus, when estimating the 1D marginal PDFs the equation reduces to $h_{opt} = \left(\frac{4}{3N_{MC}} \right)^{1/5}$, and when estimating the 2D joint PDF it reduces to $h_{opt} = \left(\frac{1}{N_{MC}} \right)^{1/6}$. We use h_{opt} when calculating the mutual information for the remainder of this work.

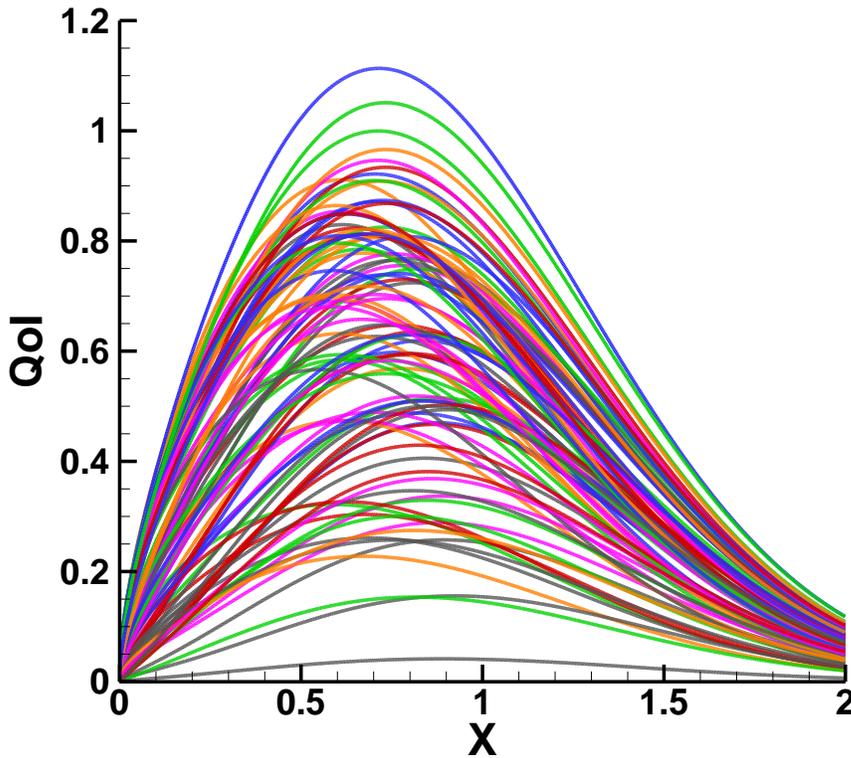


Figure 3.14. QoI(x) profiles for a variety of values of A, B, and C.

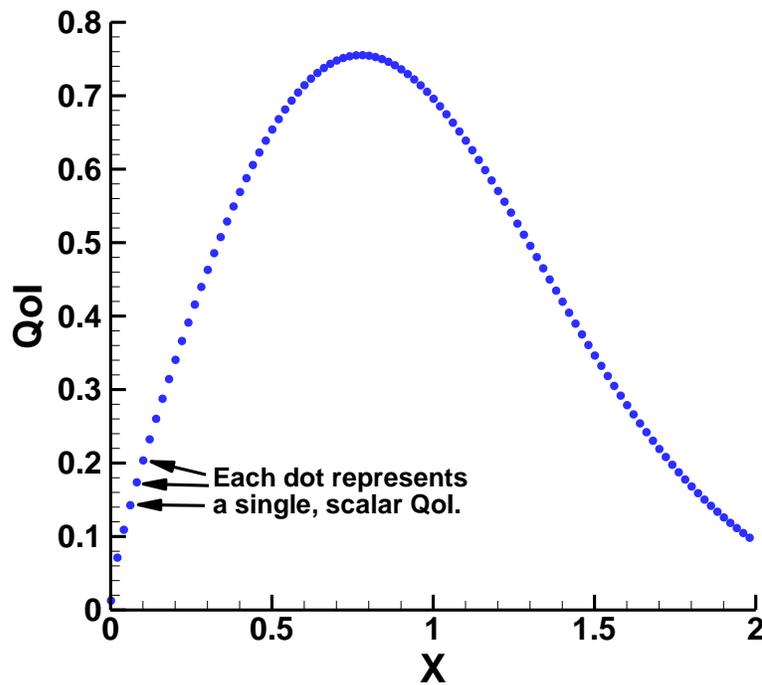


Figure 3.15. Schematic showing the way in which a vector QoI is broken into individual scalar QoIs during the sensitivity analysis.

We also take one other step to mitigate errors in the calculation of the mutual information. In addition to the parameters A, B, and C from Eq. 3.5, we calculate the mutual information for a dummy parameter. This dummy parameter is sampled over the range from 0 to 1 just like the others, but it is not present in the equation for $QoI(x)$, and so by definition the QoI is completely independent of this parameter and the mutual information calculated for this parameter can be used to estimate the spurious mutual information due to randomness. We do not want this spurious mutual information to affect the relative sensitivities of our parameters, so we subtract the mutual information which is calculated for the dummy parameter from the values of mutual information

calculated for the other parameters. We then use this corrected value for the mutual information in all subsequent analysis.

Once we have calculated sensitivities based on r^2 and the mutual information for each scalar QoI, we can then plot these sensitivities as a function of x (since each scalar QoI represents $QoI(x)$ at a particular x location). This plot is shown in Figure 3.16. The sensitivity curves in the plot are actually not extended to show the sensitivities at $x = 0$ because $QoI(0) = 0$ regardless of the values of the parameters, and thus the sensitivities are also all zero there. Looking first at the sensitivities for $x < 1$, by either measure the QoI is most sensitive to parameter A, followed by parameter B, and is least sensitive to parameter C. These results are what we would qualitatively expect based on an examination of Eq. 3.5, since $\sqrt{x} > x > x^2$ for values of x which are between 0 and 1. At $x = 1$, by either measure the QoI is equally sensitive to all three parameters, since $\sqrt{x} = x = x^2$ when $x = 1$. Finally, the order of the sensitivities is flipped on the right side of the plot, where $x > 1$, because $x^2 > x > \sqrt{x}$ when $x > 1$.

The curves in Fig. 3.16 have been shown on separate axes (one for r^2 , one for the mutual information) and the range of these axes is chosen so that the maximum height of any of the three curves based on r^2 is equal to the maximum height of any of the three curves based on the mutual information. This visual normalization of the curves is done because while r^2 is mathematically bounded to be within $0 \leq r^2 \leq 1$, the same is not true of the mutual information. The mutual information will never be less than zero, but it may exceed 1 (and does at certain x locations in this case). The maximum sensitivity based on either measure is that of parameter A as x approaches (but does not actually equal) zero. Comparing the curves over the rest of the range, we see that while both sets pick out the correct order of the parameters, the curves are qualitatively different depending on which measure of sensitivity is used. This is not due to r^2 failing to capture a non-linear

correlation, because all the correlations here are linear by construction. The QoI in Eq. 3.5 depends non-linearly on x , but at any given x -location it depends linearly on the values of A , B , and C . The difference in the curves is due to the fact that r^2 and the mutual information are fundamentally different measures, and they provide different results (even after normalization) depending on the level of scatter in the data. This point is made clearer by Figure 3.17, which shows three scatter plots and their corresponding values of r^2 , the mutual information, and the ratio of the mutual information to r^2 . For the scatterplot in the top image (where the points are all packed tightly about the trendline) MI/r^2 is equal to 1.4, while in the bottom left image (where scatter about the trendline is greater) MI/r^2 is equal to 0.9, and in the bottom right image (where the points are even more scattered) MI/r^2 is equal to 0.6. This cannot be fixed by normalization, of course, since it is the ratios we are discussing. The fact that r^2 and the mutual information measure sensitivity in different ways is actually a good thing, because it allows for the dataset to be examined in more than one way. Even though the mutual information is a more sophisticated method, it is still beneficial to examine sensitivities based on both methods in order to understand the results better.

There is also an area in which r^2 outperforms the mutual information. Because it is a simpler method which does not require the estimation of PDFs, it performs better (in a convergence sense) when using small numbers of sample points. This is illustrated in Figure 3.18, which compares curves for the sensitivity for parameter C for datasets of different sizes. In the left image, the curve for sensitivity based on r^2 with 10,000 sample points is almost indistinguishable from the curve for 100,000 sample points, and even the curve for 1,000 sample points is very close to the curves for the larger datasets. In the right image, though, the mutual information curve for 1,000 sample points is significantly different from the curve for 100,000 sample points, and even the 10,000 point curve is

slightly different when compared to the 100,000 sample point curve. The mutual information behaves in this way because it is necessary to choose a Gaussian kernel width, and the appropriate width depends on the PDF being estimated. We use h_{opt} , as discussed previously, but h_{opt} as we have defined it is only mathematically known to be optimal if the PDFs being estimated are Gaussian, which in practice they are not. Calculating r^2 , on the other hand, does not require that a value be specified for any adjustable numerical parameters; with r^2 there is no parameter like the Gaussian kernel width (or the histogram bin width).

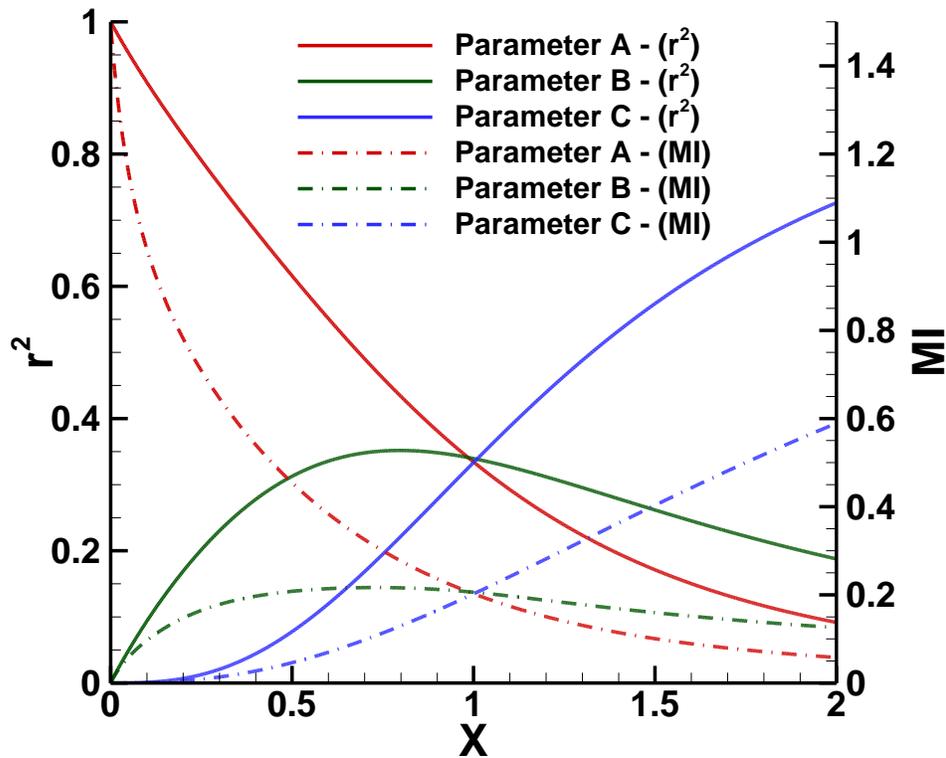


Figure 3.16. Sensitivities based on r^2 and the mutual information for each of the three parameters in Eq. 3.5. Sensitivities are plotted as a function of x , and each x location corresponds to a particular scalar QoI (as shown in Fig. 3.15).

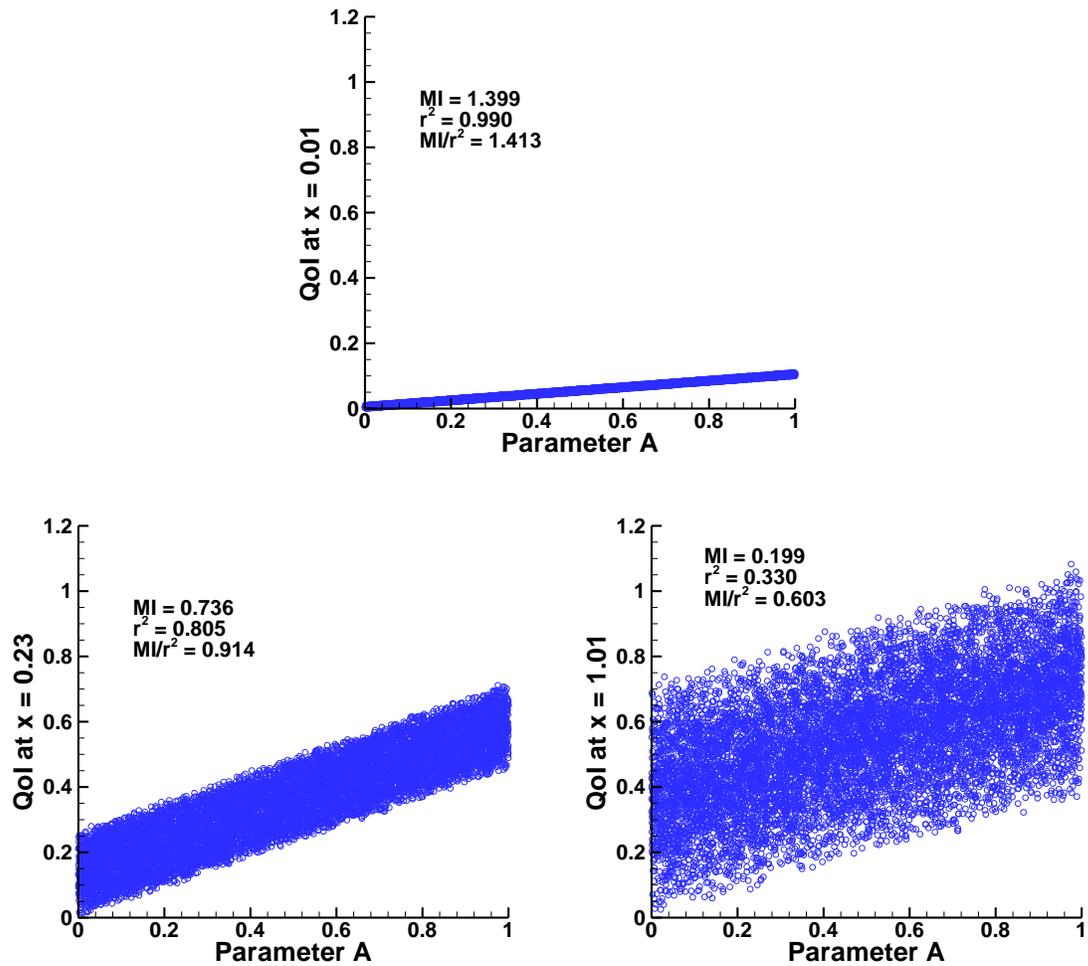


Figure 3.17. Scatterplots for the QoI vs. parameter A at three different x locations. Note that the ratio of the mutual information to r^2 is different for each of the scatterplots.

When we make use of these sensitivity analysis methods in the context of DSMC simulations, we will confront the issue of stochastic noise in the simulation results (and thus in the QoI). To examine how our methods will behave in a situation where the QoI is affected by noise, the analysis above was redone for three additional cases. In the first and second of these cases, white noise with a fixed amplitude (and mean zero) is added to the QoI from Eq. 3.5. The amplitude of the noise added in the second case is four times

the amplitude of the noise added in the first case. In Figure 3.19, profiles are shown for the noiseless case discussed previously and for the two cases with noise added. The values of A, B, and C are the same for each of the three profiles.

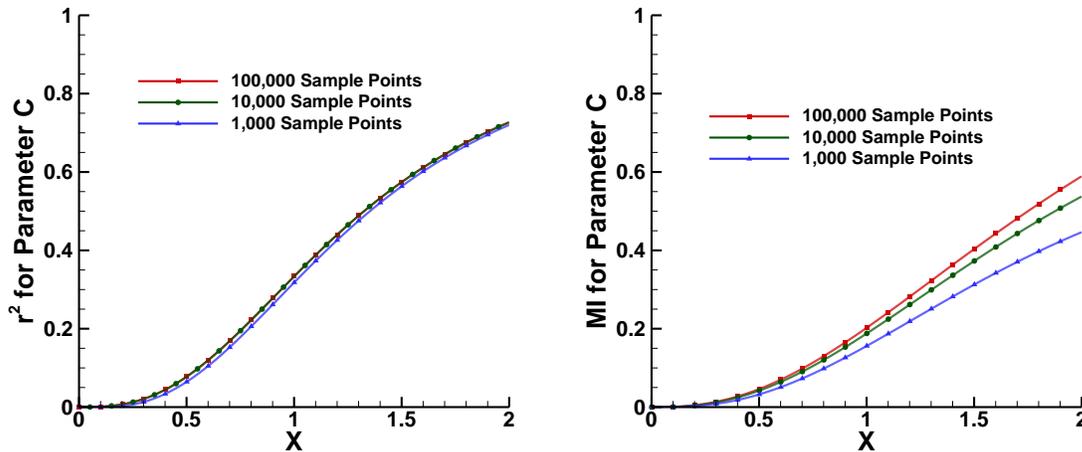


Figure 3.18. Sensitivities based on r^2 and the mutual information as a function of x for parameter C for three different numbers of Monte Carlo sample points. The curves for sensitivities based on r^2 are much less affected by the number of sample points.

The sensitivity analysis was carried out for the low and high noise cases in the same way as it was originally carried out for the noiseless case. The sensitivity vs. x curves based on r^2 for each of parameters are shown for all three cases in Figure 3.20, and Figure 3.21 shows the equivalent curves for sensitivities based on the mutual information. In the middle region near $x = 1$ the sensitivities (based on either measure) are almost the same for the noiseless and low noise amplitude cases, and even the high amplitude noise only lowers the sensitivity slightly. However, in the regions near $x = 0$ and near $x = 2$ the sensitivities are much lower in the cases with noise than in the noiseless case, and the decrease in the calculated sensitivity is substantially larger for the higher amplitude noise

case than for the lower amplitude noise case. This behavior is as expected. In the middle near $x = 1$ the standard deviation of the QoI from the dataset (based on the sampling of the parameter space) is large relative to the noise amplitude (especially for the lower noise amplitude case), and thus the effect on the QoI of changes in the parameters can still be separated from the noise which is superimposed on the QoI. That is, the portion of the variance explained by the uncertainty in the parameters is large relative to the portion of the variance which is caused by the white noise. In the regions near $x = 0$ and near $x = 2$, the standard deviation of the QoI from the dataset is much lower, and therefore the contribution of the noise to the variance of the QoI is large relative to the contributions of the uncertain parameters, and thus the sensitivities calculated for the parameters are low compared to the noiseless case (where all of the variance of the QoI is by definition explained by the parameters). This tells us that, when using DSMC as our model, we will need to make sure that the stochastic noise in the QoI is substantially smaller than the standard deviation of the QoI from the dataset generated by the Monte Carlo sampling of the parameter space.

Before moving on, one more example will help illustrate another aspect of the relationship between stochastic noise and the sensitivities to various parameters. In the previous cases with noise the amplitude of the noise was fixed, but in some practical cases numerical or even physical parameters will control the noise amplitude. To test the methodology for this case, we use a slightly modified version of Eq. 3.5 which includes a fourth parameter (parameter D) that controls the amplitude of the additive white noise. The equation for this new QoI is

$$QoI(x) = (A * \text{sqrt}(x) + B * x + C * x^2)e^{-x^2} + D * (Rnum - 0.5) * 0.2 \quad (3.7)$$

where Rnum is a random number between 0 and 1. A new random number is drawn for each x location (i.e. for each scalar QoI).

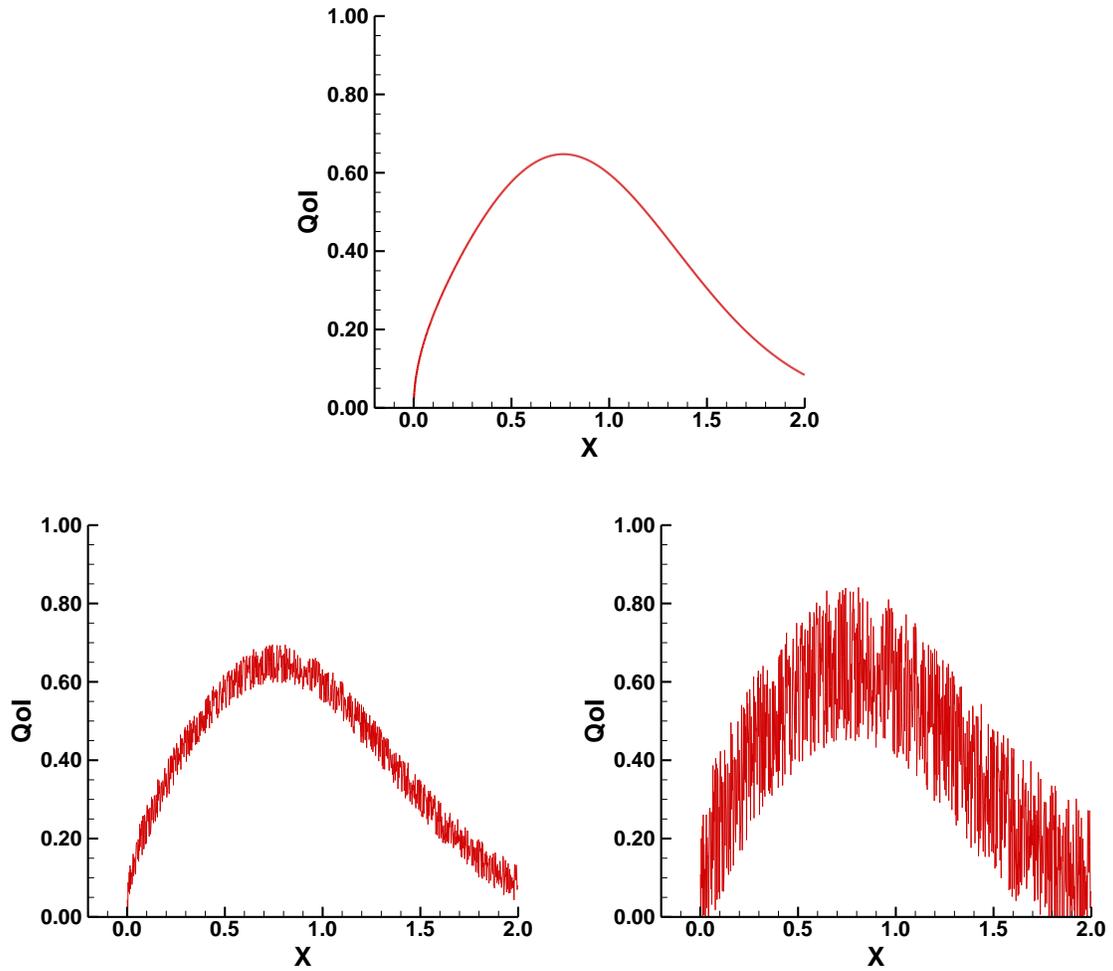


Figure 3.19. Sample profiles based on Eq. 3.5 after the addition of white noise with two different amplitudes.

A sensitivity analysis of the same type performed for the previous examples was carried out with this new QoI and the four parameters. The results are shown in Figure 3.22. As with the previous cases with noise the sensitivities (based on either measure) drop significantly in the region near $x = 0$ and near $x = 2$, when compared with the curves for the noiseless case (see Fig. 3.15). The curves in the region near $x = 0$ are very similar

to the curves in Fig. 3.15 because the effect of the noise is much less significant when the standard deviation of the QoI is large compared to the noise amplitude.

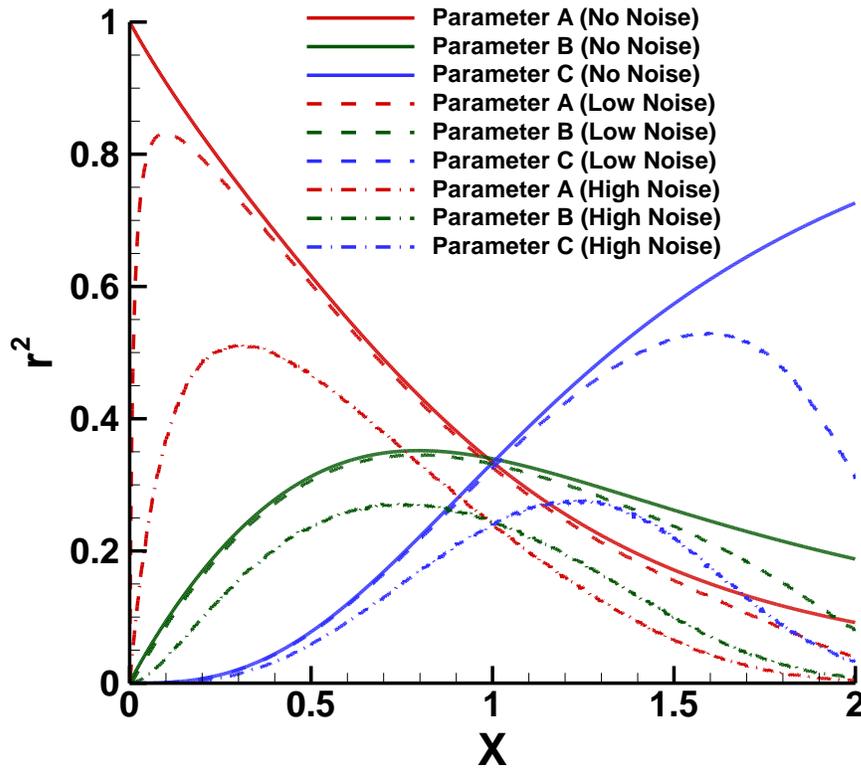


Figure 3.20. Sensitivities based on r^2 as a function of x for all three parameters for three different levels of white noise which were added to the QoI from Eq. 3.5.

The main features of interest in Figure 3.22 are the curves for sensitivity to the parameter which controls the noise amplitude (parameter D). The r^2 curve is almost exactly zero for all values of x , while the curve for the mutual information is near zero in the middle region near $x = 1$ but is significantly above zero in the region near $x = 0$ and the region near $x = 2$. In fact, when using the mutual information as the measure parameter D is the second most sensitive parameter in both of those regions. The fact that the r^2 curve is zero even in regions where r^2 for the other parameters drops due to the

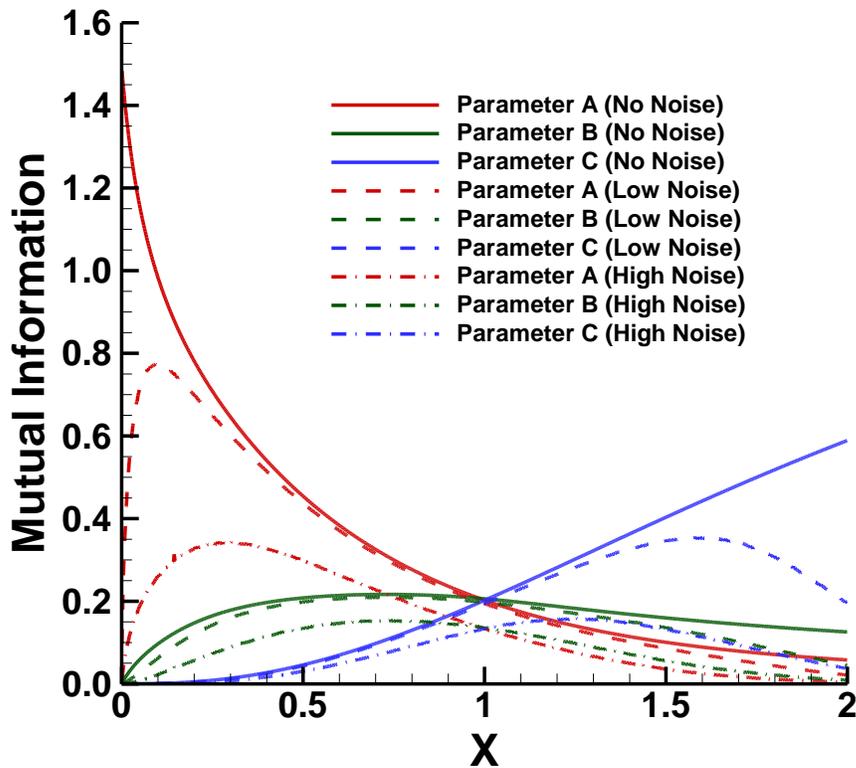


Figure 3.21. Sensitivities based on the mutual information as a function of x for all three parameters for three different levels of white noise added to the QoI from Eq. 3.5.

noise suggests that r^2 is not capturing the relationship properly, and Fig. 3.23 demonstrates the reason for this failure. The figure shows a scatter plot of QoI vs. parameter D at an x location very near the right edge of the domain. Parameter D clearly has a substantial impact on the distribution of the QoI at this x -location, but the relationship is non-linear. For this reason, the mutual information is able to successfully capture the relationship but r^2 is not. This behavior provides a practical example of a situation in which it is important to use the mutual information, since the correlation coefficient alone does not provide a full picture of the sensitivities. Note also that in this

case we had the benefit of being able to compare to a noiseless case, but in a practical problem it would often not be possible to fully eliminate the noise, and thus being able to use the mutual information to characterize the sensitivity of the QoI to noise relative to sensitivities to the other uncertain parameters can provide useful insight about the significance of noise when interpreting sensitivity analysis results for a given case. This will be discussed further in the context of the 0D sensitivity analysis in Chapter 5.

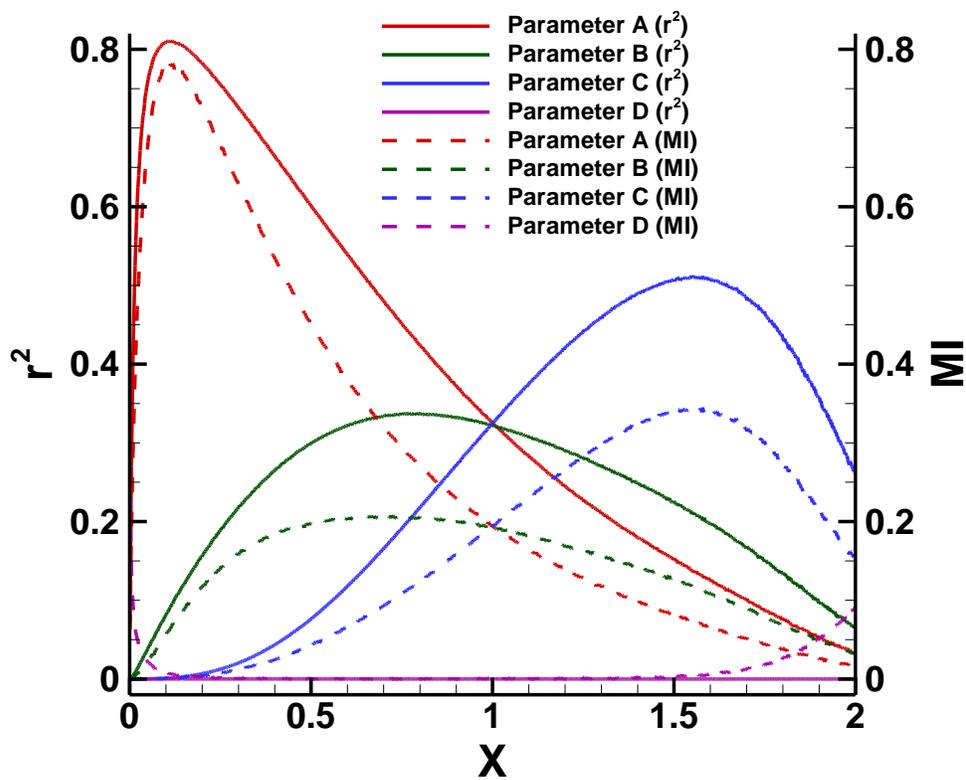


Figure 3.22. Sensitivities as a function of x for all four parameters from Eq. 3.7. The mutual information captures the sensitivity of the QoI to the noise amplitude (parameter D) while r^2 does not.

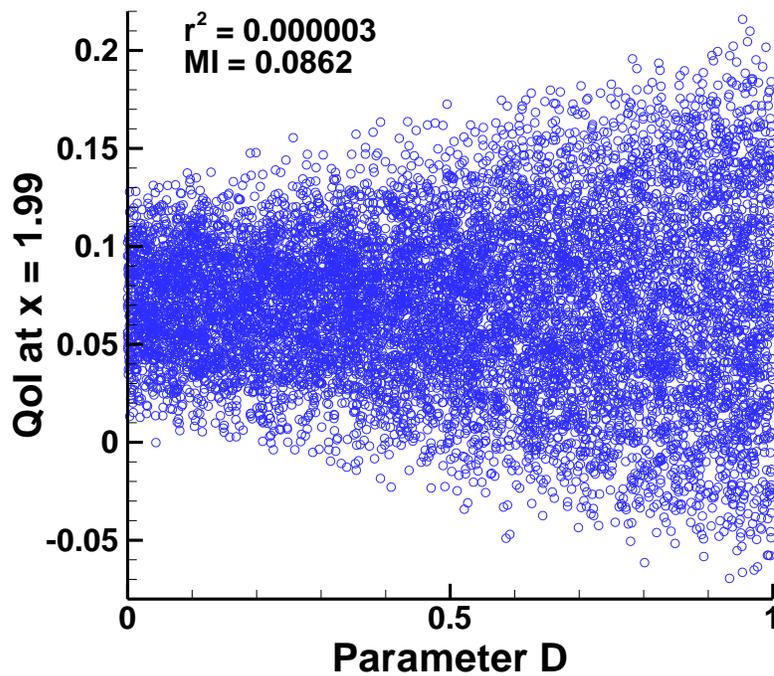


Figure 3.23. Scatterplot for the QoI vs. parameter D from Eq. 3.7, which controls the amplitude of the additive white noise. Parameter D clearly affects the distribution of values of the QoI, but the relationship is non-linear and thus r^2 is very nearly zero. The mutual information does capture the relationship.

Chapter 4: Solving the Inverse Problem

OVERVIEW

In this work we use the Markov Chain Monte Carlo (MCMC) algorithm to solve the statistical inverse problem in order to calibrate parameters. In this chapter we first discuss the basic Metropolis-Hastings MCMC algorithm, and then we move on to address two improvements to the algorithm which can allow the chains to converge to the posterior distributions more quickly (i.e. with the use of fewer chain positions). The first of these improvements is delayed rejection and the second is adaptation of the covariance matrix for the proposal distribution from which candidate positions are selected. The MCMC algorithm which includes both of these improvements is known as Delayed Rejection Adaptive Metropolis (DRAM), and was first implemented by Haario *et al.* (2006).

This chapter will also describe the implementation of these algorithms in the codes used in this work and how the various codes interact during the solution of the statistical inverse problem.

Finally, some example calibrations and their results will be discussed in order to illustrate various aspects of the process.

MCMC ALGORITHM

Basic Metropolis-Hastings Algorithm

The Metropolis-Hastings MCMC algorithm is a statistical method which is used to solve the inverse problem in order to calibrate parameters for a given model with respect to a set or sets of data. Before we make use of MCMC, we assume that we have picked a model, selected a set of parameters from that model for calibration (most likely

based on the results of a sensitivity analysis), established prior distributions for those parameters, and selected a scenario and a set of data on which to base the calibration. All of those aspects of the process were discussed in the previous chapter. Note that in this work we use only uniform prior distributions, which means that the priors cancel out of all the equations in this section and the next. With these elements in place, the algorithm proceeds via the following steps (which can also be seen in flowchart form in Figure 4.1):

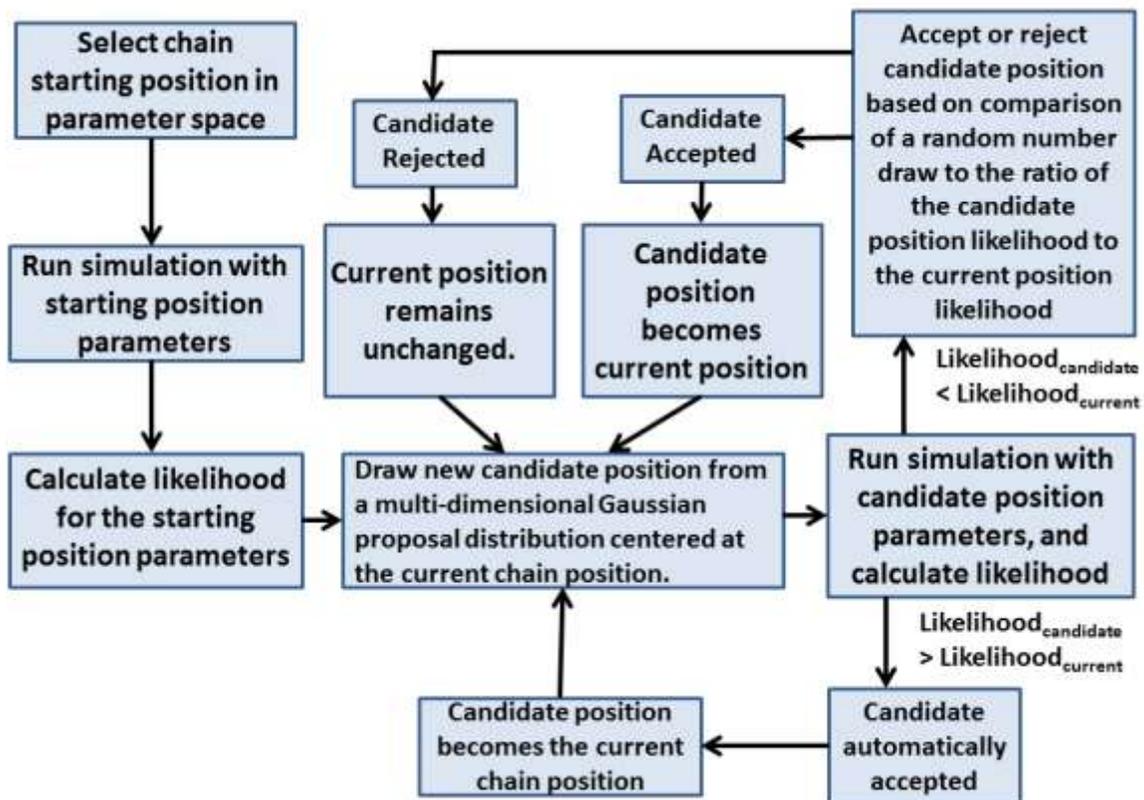


Figure 4.1. Flowchart for the basic Metropolis-Hastings algorithm, starting at the box on the upper left. The loop on the right side is repeated until the Markov chain has reached the desired length or until some specified convergence metrics have been achieved.

1.) We must first select a starting position in parameter space from which to begin the Markov chain. This starting position may be specified beforehand or initial values for the parameters may be randomly selected from their prior distributions. In this work, the chain starting positions are always randomly selected.

2.) A simulation is run with this set of parameters, and the model generates output which can be compared with the calibration data.

3.) The model results are compared to the calibration data and based on this comparison a likelihood is calculated for this set of parameters. In this work we use a Gaussian likelihood function. The likelihood for a given set of parameters is given by the equation

$$likelihood = P(D|\theta) = \frac{1}{(2\pi)^{\frac{N_d}{2}} |\Gamma|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\mathbf{D} - \mathbf{X})^T \Gamma^{-1} (\mathbf{D} - \mathbf{X})\right] \quad (4.1)$$

where N_d is the number of calibration data points, $\mathbf{D} \in \mathbb{R}^{N_d}$ is the vector of data points from the calibration dataset, $\mathbf{X} \in \mathbb{R}^{N_d}$ is the model output vector, and $\Gamma \in \mathbb{R}^{N_d \times N_d}$ is a covariance matrix. If we assume that the data points are independent of one another the covariance matrix becomes diagonal, and if we further assume that the uncertainty is the same for all data points, then the above equation reduces to

$$likelihood = P(D|\theta) = \frac{1}{(2\pi\sigma^2)^{\frac{N_d}{2}}} \exp\left[-\frac{1}{2\sigma^2} \sum_{i=1}^{N_d} (D_i - X_i)^2\right] \quad (4.2)$$

where σ^2 is the variance of the value at each data point.

The above equations are worth discussing in a little more detail. When we use these equations we are assuming that, due to uncertainty in the techniques used to measure the data, we do not have complete confidence in the values at each point in the dataset. We assume that, if the experiment which generated the data were repeated a large number of times, we would find that the values for any given data point would follow a Gaussian distribution about some mean value. As mentioned above, if we make

the assumption that all of the data points are independent of one another and also that the uncertainty is the same for all of the data points, we can simplify the Gaussian and represent the uncertainty by only one parameter, σ^2 , the variance of the Gaussian distribution for each data point.

When we use our model to run a simulation for a given point in parameter space we obtain a set of values for the model output at all of the data points. We then hypothesize that our model is correctly formulated and is able to fully and accurately simulate the scenario at hand, and we further hypothesize that the current point in parameter space corresponds to the best possible set of parameter values for the given model and scenario. If this were true, the output of our model for each data point would represent the true mean value for that data point which we would expect to obtain from a large set of experiments. Eq. 4.1 or 4.2 is then used to assess the likelihood that our hypothesis is correct. This is accomplished by calculating $P(\mathbf{D}|\boldsymbol{\theta})$, which is the probability of obtaining the set of data points \mathbf{D} given the assumption that the current set of parameter values $\boldsymbol{\theta}$ is the best possible set of parameter values (i.e. that the model output based on these parameter values represents the true mean values for the data points).

4.) A new candidate chain position is drawn from a multi-dimensional Gaussian proposal distribution centered at the current chain position. The covariance matrix of this Gaussian controls the average distance (in parameter space) that the chain moves in one step. We usually do not know *a priori* exactly how couplings between parameters will manifest themselves, and therefore we often start with a simple, diagonal covariance matrix for the proposal distribution. The proposal distribution can have a very strong effect on the convergence of the Markov chain, and a bad choice for the covariance matrix can dramatically increase the number of chain positions required for convergence.

In some cases the chain will not converge with any practical number of chain positions. The improvements to the basic Metropolis-Hastings algorithm which are incorporated into the more advanced DRAM algorithm help to address this issue, as will be discussed in the next section.

5.) A simulation is run with the candidate set of parameters and model output is generated for comparison with the calibration data. A likelihood is calculated for the candidate set of parameters in the same way as in step 3.

6.) If the likelihood of the candidate position is higher than the likelihood of the current position, the candidate position is always accepted. If the likelihood of the candidate position is lower than the likelihood of the current position, the candidate position is accepted or rejected based on a random number draw, where $\text{Probability}_{\text{acceptance}} = \text{likelihood}_{\text{candidate}} / \text{likelihood}_{\text{current}}$.

7.) Steps 4-6 are repeated until the Markov chain has reached the desired length or until some specified convergence metrics have been achieved. Ideally, by the end the chain will have converged to the posterior (i.e. post-calibration) distribution for the parameters. In this context, we will use convergence to mean that the parameter space has been explored to a sufficient extent, that the starting position chosen for the chain has no impact on the final distribution of chain positions, and that the distribution of chain positions has become stationary. In practice, the early part of the chain may be thrown away in order to help remove the effect of the choice of starting position. Additionally, more than one chain may be used in order to help ensure fuller coverage of the parameter space.

At the end of the above process we have stored a list of chain positions, with each position corresponding to a set of values of the model parameters. If we have performed the process correctly and if the chain has converged, then this list of chain positions can

be used to obtain an accurate estimate of the posterior PDF for each model parameter, based on calibration with the given dataset.

Delayed Rejection Adaptive Metropolis Algorithm

As mentioned in the previous section, a Markov chain must be converged in order for the set of chain positions to be representative of the posterior distributions for the model parameters. The basic Metropolis-Hastings MCMC algorithm sometimes has difficulty achieving any semblance of convergence, especially in cases where an appropriate proposal covariance matrix is not well-known initially. If the proposal distribution is too narrow (leading to very small jumps in parameter space) the Markov chain will have to be extremely long in order to fully explore the parameter space, often to the point where doing enough simulations to achieve a chain of the necessary length becomes impractical. If the proposal distribution is too broad (leading to large jumps in parameter space) the Markov chain will more fully explore the parameter space, but it will have difficulty focusing in to properly explore the regions with high likelihood. It will often find a position where the likelihood is high and then repeatedly select far-off candidates with low-likelihood (which are then rejected), causing it to stay at that position for an excessive amount of time. This can lead to very sharp peaks and valleys in the posterior distribution at some locations. In theory, these spikes would slowly smooth out (provided that the true posterior distribution is itself smooth) as the length of the chain grows very large, but the number of positions required can often be unrealistically large for computational tractability.

Beyond the problem of the initially specified proposal distribution often being either “too narrow” or “too broad”, we also have to deal with the fact that the appropriate

jump size might (and usually will) be different for different parameters. Furthermore, there is also the issue of coupling between the parameters. An appropriate size jump for one parameter might depend on the sizes and directions of the jumps in other parameters. In a case like this, the covariance matrix for the proposal distribution will not be diagonal. It will often be nearly impossible to get a good *a priori* estimate of appropriate values for all of the terms of a full covariance matrix for a problem with several parameters.

As discussed in the Literature Review, this problem has been addressed by others in the past, and a couple of key improvements have been made to the basic Metropolis-Hastings algorithm. One of these improvements is the use of delayed rejection and the other is the adaptation of the proposal covariance matrix. We will discuss each of the improvements, following the description of Haario *et al.* (2006).

The idea behind delayed rejection is simple. If the initially specified proposal covariance matrix for a given calibration is poorly scaled (i.e. it leads to jumps which are in general too large or too small) it can be useful to use a multi-step process for choosing the next chain position. The first candidate is chosen from the proposal distribution as usual. If this candidate is accepted, it becomes the current position and we start the process for a new position (just like we would with basic Metropolis-Hastings). If the candidate is rejected, however, we do not immediately decide to retain the current position. Instead, a second stage candidate is chosen from a scaled version of the proposal distribution. This scaling is done by multiplying the proposal covariance matrix by a single scalar value. In principle, the proposal covariance matrix can be scaled up or down, but in general it is more effective to start with a proposal distribution which is too broad and then use delayed rejection to narrow the distribution for the second attempt. (i.e. we usually scale the proposal covariance matrix by a number which is less than one).

After the second stage candidate position has been selected based on the scaled proposal distribution the likelihood is calculated for this position. If this likelihood is less than the likelihood for the first stage candidate then the second stage candidate is automatically rejected. Otherwise, the second stage candidate is accepted or rejected with a probability equal to

$$Prob_{accept} = \frac{q(X_{SSC}, X_{FSC}) [likelihood_{SSC} - likelihood_{FSC}]}{q(X_{CCP}, X_{FSC}) [likelihood_{CCP} - likelihood_{FSC}]} \quad (4.3)$$

where X_{CCP} is the location in parameter space of the current chain position, X_{FSC} is the location in parameter space of the first stage candidate position, and X_{SSC} is the location in parameter space of the second stage candidate position. The terms $q(X_{SSC}, X_{FSC})$ and $q(X_{CCP}, X_{FSC})$ are, respectively, the probability of proposing a jump from the second stage candidate position to the first stage candidate position (based on the unscaled proposal distribution) and the probability of proposing a jump from the current chain position to the first stage candidate position (again based on the unscaled proposal distribution). The two q terms are present to satisfy the requirement that the Markov chain remains reversible (Haario *et al.*, 2006). Finally, $likelihood_{CCP}$ is the likelihood for the current chain position, $likelihood_{FSC}$ is the likelihood for the first stage candidate, and $likelihood_{SSC}$ is the likelihood for the second stage candidate (each of which would be calculated based on a likelihood equation such as Eq. 4.1 or 4.2). Note that the numerator and denominator will both always be greater than zero, because if $likelihood_{SSC} - likelihood_{FSC} < 0$ (meaning that $likelihood_{SSC} < likelihood_{FSC}$) then the second stage proposal would have been automatically rejected, and if $likelihood_{CCP} - likelihood_{FSC} < 0$ (meaning that $likelihood_{FSC} > likelihood_{CCP}$) then the first stage proposal would have been automatically accepted and there would have been no second stage at all.

If the second stage candidate is rejected, we could propose a third stage candidate based on a further scaling of the proposal distribution, and accept or reject that third stage

candidate based on an equation similar to (but more complicated than) Eq. 4.3. In principle, delayed rejection can be carried out for an arbitrary number of stages. In this work we will stick with only two stages. If the second stage candidate is rejected then the current chain position is retained and the process starts over with a new first stage candidate.

In general delayed rejection can help if the proposal distribution is too narrow or too broad, but since it only scales the proposal covariance matrix it cannot do anything to address couplings between parameters. To deal with these potential couplings we need to adapt the proposal covariance matrix, not just scale it, and this is the purpose of adaptive Metropolis. When using adaptive Metropolis an initial proposal covariance matrix is specified at the beginning of the process. This matrix is left unchanged for a specified number of steps (called the initial non-adaptation period), after which it is adapted for the first time. After this first adaptation, the proposal covariance matrix is adapted after each specified interval of chain positions (the intervals are usually of constant length, but this is not required). When the proposal covariance matrix is adapted, the new version is given by the equation

$$C_n = s_d \text{Cov}(\mathbf{X}_0, \dots, \mathbf{X}_{n-1}) + s_d \epsilon I_d \quad (4.4)$$

where C_n is the adapted proposal covariance matrix for the interval of chain positions starting with the n^{th} position, \mathbf{X}_0 is the starting position of the chain, \mathbf{X}_{n-1} is the chain position just prior to the adaptation, d is the dimensionality of the space being explored by the chain (in the case of a calibration, it is the number of parameters being calibrated), and $I_d \in \mathbb{R}^{d \times d}$ is the d -dimensional identity matrix. The constant s_d is related to the dimensionality of the space, and we use the value $s_d = 2.4^2/d$, as recommended by Gelman *et al.* (1995) and by Haario *et al.* (2006). The term $s_d \epsilon I_d$ is intended to prevent the proposal covariance matrix from ever becoming singular, and it is often not needed.

Thus, ε is usually set to zero or to a small positive value. Finally, $Cov(\mathbf{X}_0, \dots, \mathbf{X}_{n-1})$ is the empirical covariance matrix determined by chain positions $\mathbf{X}_0, \dots, \mathbf{X}_{n-1} \in \mathbb{R}^d$. The adapted covariance matrix is then used for the specified interval of chain positions, after which it is adapted again. This adaptation process allows the appropriate proposal covariance matrix to be discovered as a part of the calibration process, and can dramatically improve convergence of the chain to the posterior distribution.

When both delayed rejection and adaptive Metropolis are implemented together, the resulting algorithm is known as Delayed Rejection Adaptive Metropolis (DRAM). In the work presented here, as in the work of Haario *et al.* (2006), adaptive Metropolis is used to adapt the proposal covariance matrix for the first stage candidate. The covariance matrix for the second stage candidate is a scaled version of the first stage proposal covariance matrix. The two improvements can complement one another very well, as stated by Haario *et al.* (2006):

The key feature is that, as pointed out by Green and Mira (2001), DR works better if the variance of the proposal is too big at first stages and down scaled at higher stages. On the other hand AM recovers well and starts adapting even if the variance of the initial proposal is too small (clearly if the variance is too big no proposals are accepted and adaptation is almost impossible to get started). Thus, a combination of the two, as in DRAM or other variations of it, clearly provides protection against both over and under calibrated proposals.

INVERSE PROBLEM IMPLEMENTATION

A number of codes are involved in the solution of the inverse problem for calibration of model parameters relevant to DSMC simulations of hypersonic shocks. Substantial effort was involved in making sure these codes all work together in a cohesive whole which can be run on large scale computer clusters.

A driver program (written in C++ and supplied by PECOS as part of the QUESO implementation) gets everything started. It reads the QUESO input file and initializes the various vectors and arrays which will be used during the solution of the inverse problem. This driver code has been modified for this work so that chains may be restarted from a previous run (this is necessary because of constraints on the length of runs which may be submitted to the queue on most computer clusters, including the one used for this work). The driver code can read in an adapted proposal covariance matrix which was output from a previous run (this modification was performed by Ernesto Prudencio of PECOS). It can also read in the current positions for a set of chains based the final chain positions output by a previous run. The conversion of the output of one run to the input for the next can easily be automated so that multiple day runs can be set up without the need for intervention along the way, provided that a given cluster allows job scripts to be set up properly for this task.

During the course of a run, the driver code sends the likelihood from the previous candidate position to the main QUESO code (which is also written in C++), which then uses the DRAM algorithm to obtain the parameter values for the next candidate position. This candidate position is then sent back to the driver code, which passes the parameters to a code which interfaces the C++ and Fortran 95 parts of this process. The parameter values are then received by another interface code (written in Fortran 95) which is used to integrate MCMC with the DSMC code. This MCMC/DSMC interface code reads in additional input files which provide information on the scenario, the calibration data, and the specific parameters being calibrated. This information, along with the parameter values, is then passed to the DSMC code. The DSMC code runs the simulation for the chosen scenario with the given parameter values. The results of the simulation are then returned to the MCMC/DSMC interface code, which compares the simulation results

with the calibration data (by means of a likelihood equation) and returns a likelihood to the driver code (via the C++/Fortran 95 interface code). This likelihood is then sent to QUESO and the process starts over. A flowchart for this process is shown in Figure 4.2.

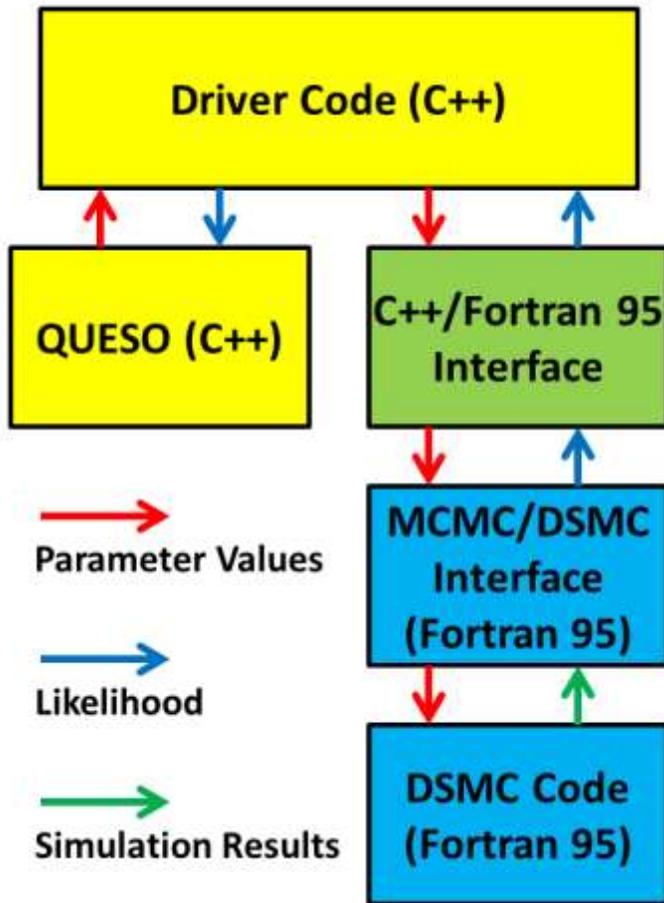


Figure 4.2. Flowchart for the interaction of the various codes involved in the solution of the inverse problem. Yellow and green boxes are codes supplied by PECOS (which in some cases I have modified slightly) and blue boxes are codes which I have written. The arrows indicate the type of information passed between the codes and the direction in which that information is passed.

Upon completion of a calibration with the codes described above, we are left with a list of chain positions from one or more chains. At each chain position we have values

for all of the parameters which we are attempting to calibrate. In order to convert this list of chain positions into a set of posterior PDFs for the parameters we use kernel density estimation. We use KDE in the same way as it was used in Chapter 3 to estimate the 1D marginal PDFs. Now instead of values for a set of scalar QoIs and a set of parameters at each position in parameter space, we have only the positions in parameter space and the corresponding parameter values. We calculate 1D, marginal PDFs only for the parameters, not for any QoI. We once again use h_{opt} , as given by Eq. 3.6, for the width of the Gaussian kernel in our KDE method. When estimating posterior PDFs based on the list of chain positions, it is advisable to discard the early portion of each chain, since this portion of the chain will be heavily influenced by the randomly selected starting position. Deciding how many positions to discard is an empirical process. The goal is to make sure that enough positions are discarded so that the chain has reached a stationary distribution (i.e. it has converged to the posterior distribution for the parameter), while of course keeping enough positions to allow the posterior PDF to be estimated well. The discarded portion of the chain is called a burn-in.

EXAMPLE CALIBRATIONS

In order to illustrate some aspects of the calibration process described above, we will perform a few example calibrations. These calibrations will be based on synthetic data for a hypothetical QoI which depends on four parameters. As with the examples in the previous chapter, this QoI is not intended to represent any physical quantity relevant to hypersonic shocks, it is simply an example for illustrative purposes. The equation for the QoI is

$$QoI(x) = (A * \text{sqrt}(x) + B * x + C * x^2)e^{-Dx^2} \quad (4.5)$$

where the four parameters A , B , C , and D are shown in bold. In this example, each parameter has a uniform prior between 0 and 1, and the spatial variable x falls within the range $0 < x < 2$. Comparing this QoI with the one used for the first example sensitivity analysis (Eq. 3.5), we have simply added an extra parameter (parameter D) which affects the speed of the exponential decline. This was done in order to increase the coupling between the parameters and thus make the calibration slightly more challenging. The value at the center of the uniform prior for each parameter is considered to be the nominal value for that parameter (i.e. all parameters have a nominal value of 0.5).

The QoI(x) profile for the case with all parameters at their nominal values is shown in Figure 4.3. We will use this profile as our synthetic data for most of the example calibrations. When performing a calibration, we must specify an uncertainty for our data. Of course, in this case we are absolutely certain that the data is correct since we generate it from a simple equation, and we are sure that there is no model uncertainty (i.e. we are sure that our model can reproduce the data) because the data came from our model in the first place. However, the MCMC algorithm requires a measure of uncertainty for use in the likelihood equation (Eq. 4.2). This is not only a mathematical constraint, it is also a practical one. If the uncertainty were truly zero, then the likelihood would be exactly zero for any set of parameters which produces a QoI(x) profile which does not exactly match the data. There would be no way (within the context of a likelihood equation) to distinguish sets of parameters which produce results which are close (but not exactly equal) to the synthetic data and those which produce results which are very different from the synthetic data, and thus the whole calibration process would fall apart. With this in mind, since this is a hypothetical example anyway, we simply make up an uncertainty for the synthetic data. In Fig. 4.3 we show 2σ error bounds on the synthetic data based on two different assumed uncertainties.

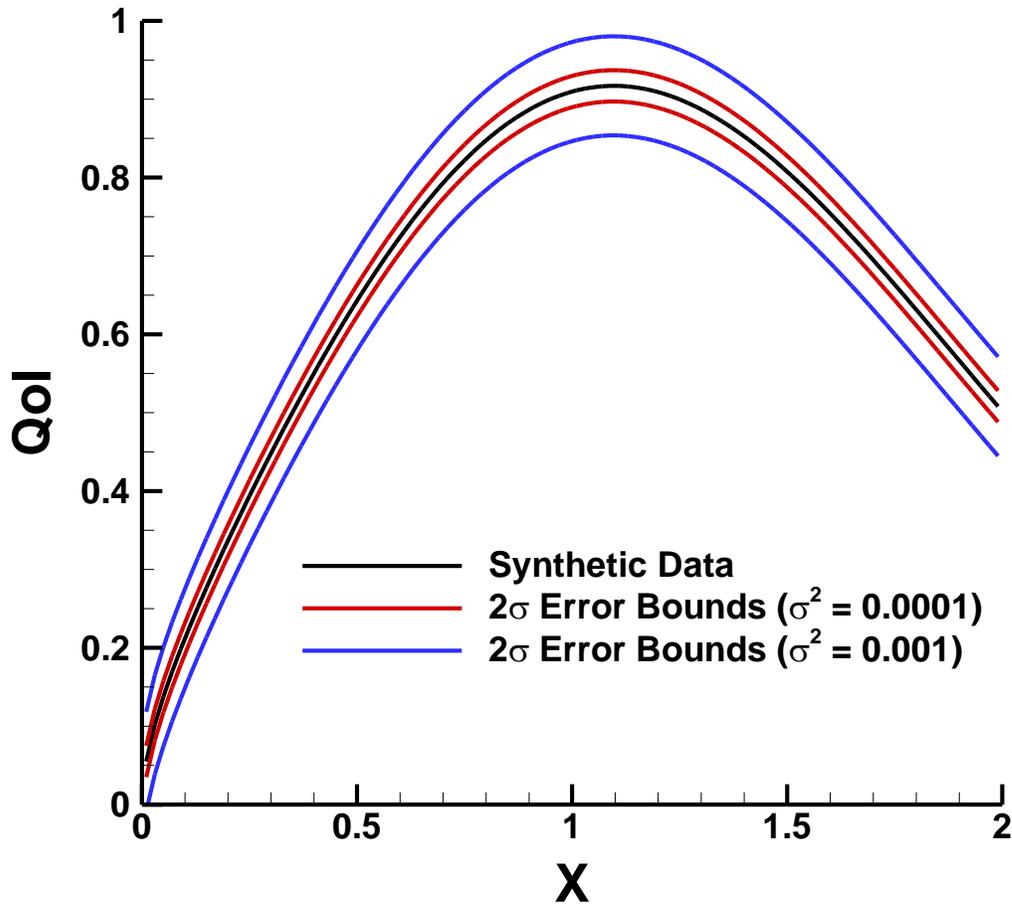


Figure 4.3. QoI(x) profile with parameters A, B, C, and D of Eq. 4.5 all set to their nominal value of 0.5. This QoI(x) profile serves as the synthetic data for the first set of example calibrations. Also shown are 2σ error bounds based on the assumption that the uncertainty of the synthetic data points is given by $\sigma^2 = 0.0001$ (the red curves) or $\sigma^2 = 0.001$ (the blue curves).

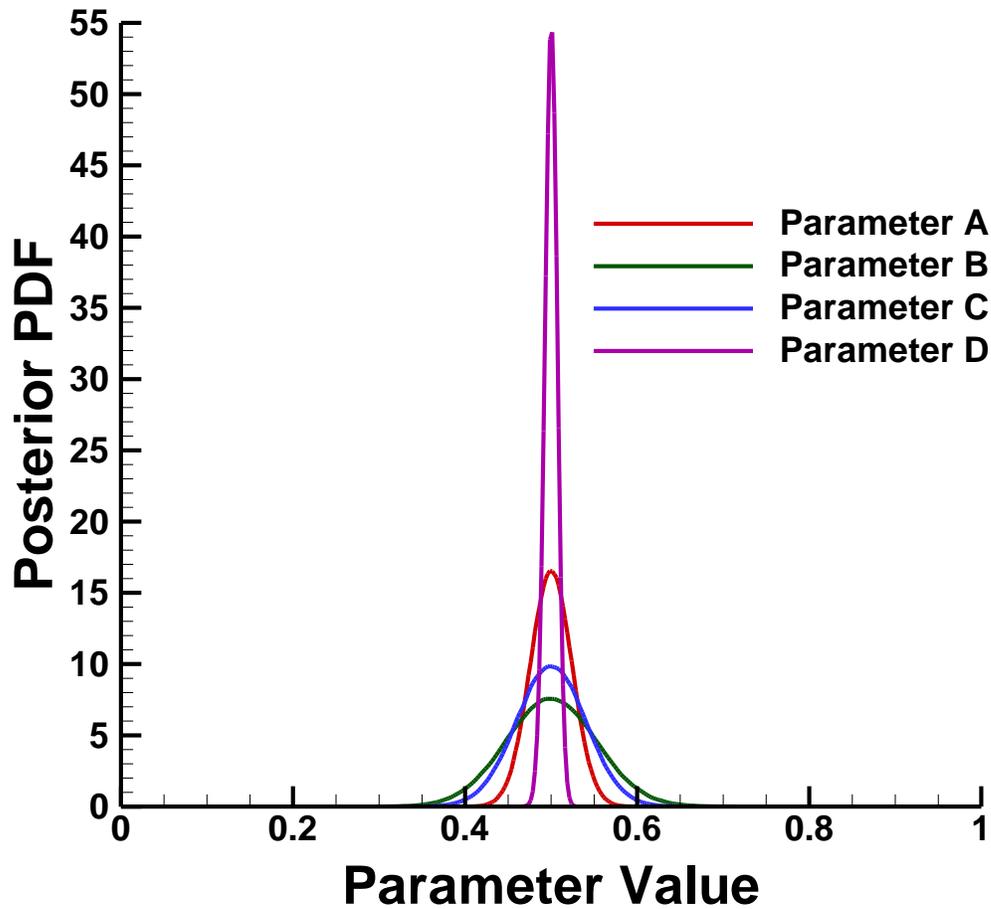


Figure 4.4. Posterior PDFs for each of the four parameters from Eq. 4.5 based on calibration with the synthetic data of Fig. 4.3 using 100 calibration data points and the assumption that $\sigma^2 = 0.0001$ for the synthetic data. The calibration is performed with 16 chains of 100,000 positions each, with a burn-in period of 20,000 positions for each chain.

In addition to specifying an uncertainty for the data, we must also decide how many discrete data points to use. $QoI(x)$ as given by Eq. 4.5 is a continuous curve, but our likelihood equation (Eq. 4.2) is set up to deal with vectors of data points and model results. Also, when using DSMC or another computational method we will always have model results at a set of discrete points, and most forms of experimental data will usually

be available only in discrete form. Therefore, for our synthetic data and our model output we use the value of $QoI(x)$ at a set of uniformly spaced x -positions. We treat the data points as independent of one another, which is of course not true in reality, but this is an assumption which is frequently made when performing calibrations (as discussed in Chapter 1).

With the above issues addressed, we can now perform a calibration. For the first calibration we use 16 chains of 100,000 positions each, with a burn-in of 20,000 positions for each chain. In Figure 4.4 we show the posterior PDFs for the parameters which are obtained from this calibration. We see that all of the posterior distributions are centered at the value of 0.5, which is as expected since the synthetic data is based on the $QoI(x)$ profile with all the parameters set equal to 0.5. The distributions all appear to be Gaussian, which is also not surprising since we are using a Gaussian likelihood function and even with the coupling between parameters this is still a very simple model. The width of the distribution is different for each of the posterior PDFs, again as expected. It is clear from Eq. 4.5 that parameter D will have a very strong influence on the QoI , since it controls the rate of exponential decay. In general, parameters which more strongly affect the QoI will usually have narrower posterior distributions (i.e. they will be better constrained) when compared with parameters which have a less significant effect on the QoI , but this can be altered by couplings between the parameters. The dominant effect of parameter A on $QoI(x)$ in the region where $0 < x < 1$ allows this parameter to be calibrated more precisely than parameters B and C (i.e. the posterior PDF for parameter A is narrower than those for parameters B and C), because the latter two parameters have their greatest effect on the QoI at larger values of x , and thus their influences on the QoI compete with one another. When more than one parameter has a similar effect on the QoI in a given region it is usually more difficult to calibrate them both precisely.

Furthermore, since parameter D also becomes more important at larger values of x , the impact of parameters B and C on the QoI is coupled with the much stronger impact of parameter D, which serves to further broaden the PDFs for B and C (and for D as well, but this is less noticeable because D has such a dominant influence at larger values of x that it can be calibrated very precisely).

In Figure 4.5 we demonstrate the reason for using a burn-in. The posterior PDF for parameter D (based on the same calibration as in Fig. 4.4) is shown with an expanded vertical scale to emphasize the tails of the distribution. The red curve is the posterior PDF calculated based on all 100,000 positions for each chain, the blue curve is the posterior PDF from Fig 4.4 (with a burn-in of 20,000 positions for each chain), and the green symbols are for the posterior calculated with a burn-in of 60,000 positions for each chain. The red curve is clearly not stationary and is influenced by the starting positions of the chains, while the blue curve and the green symbols are stationary and show that the chain is converged, since the posterior based on the final 40,000 positions of each chain is indistinguishable from the posterior based on the final 80,000 positions of each chain. In fact, symbols were only used because the green curve was otherwise completely obscured by the blue curve.

In Figure 4.6 we examine the effect of changing the specified uncertainty of the synthetic data. The figure shows posterior PDFs for two of the parameters. The solid lines are from the original calibration, with $\sigma^2 = 0.0001$. The dashed lines are from a second calibration, once again using 16 chains of 100,000 positions each with a 20,000 position burn-in, and again using 100 synthetic data points, but this time with a specified uncertainty which is ten times larger ($\sigma^2 = 0.001$). As expected, the posterior PDFs are significantly broader when the specified uncertainty on the data is larger.

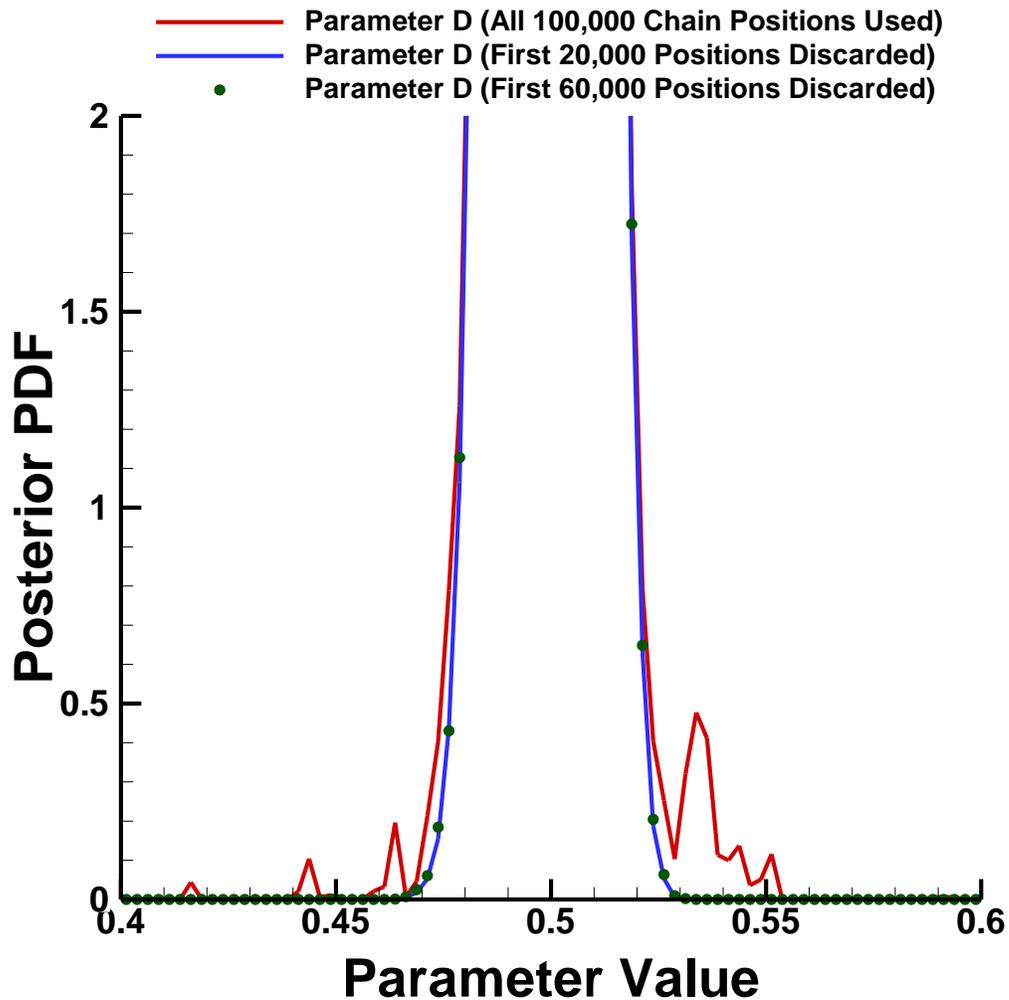


Figure 4.5. Posterior PDF for parameter D of Eq. 4.5 (zoomed in to show the tails of the distribution) based on the same set of chains as Fig. 4.4. The posterior PDF based on all 100,000 of the chain positions is shown in red. A burn-in period of 20,000 chain positions is used (i.e. the first 20,000 positions of each chain are discarded) when calculating the posterior PDF shown in blue, and a burn-in period of 60,000 chain positions is used when calculating the posterior shown with green symbols.

The number of discrete data points used in the likelihood equation can also affect the posterior PDFs. In Figure 4.7 posterior PDFs are shown for two of the parameters, with the solid lines coming from the calibration shown in Fig. 4.4 and the dashed lines

based on a second calibration. This second calibration uses the same number of chains and positions per chain, the same burn-in period, and the same specified uncertainty on the synthetic data, but only 20 discrete data points are used for the calibration. The posterior PDFs are significantly broader when the calibration is performed using only 20 data points. This occurs because by choosing to use Eq. 4.2 for the likelihood we have treated the data points as independent of one another (i.e. they each provide a completely new piece of information), and thus having 100 data points can allow us to calibrate the parameters much more precisely than we could with only 20 data points (since we have 100 pieces of information instead of only 20). In a practical case, if we are going to assume that the data points are independent we should be sure that each data point provides at least some new information.

All of the calibrations thus far have been based on the same synthetic data, and this data was provided by the model with all parameters set to their nominal values. In our simple example all the parameters have the same range and the same nominal value, and we would like to confirm that we could calibrate the parameters even if our nominal values turned out to be incorrect (i.e. the peak of the posterior PDF for one or more of the parameters did not correspond to the nominal value for that parameter). We still need synthetic data generated from our model, but this time the synthetic data points will come from the $QoI(x)$ profile when A, B, C, and D are equal to 0.3, 0.7, 0.4, and 0.6, respectively. This synthetic data is shown in Figure 4.8. We carry out a calibration using 100 data points from this curve with a specified data uncertainty of $\sigma^2 = 0.0001$ (2σ error bars based on this uncertainty are also included in Fig. 4.8). The calibration once again used 16 chains of 100,000 positions each and a 20,000 position burn in, and the posterior PDFs for the parameters are shown in Figure 4.9. The same trends are seen in these PDFs as were seen in the posterior PDFs of Fig. 4.4, except this time the PDFs are not all

centered at the same location (as expected). The posterior PDF for each parameter has its peak at the parameter value which was used to generate the synthetic data, thus indicating that the synthetic data calibration was successful.

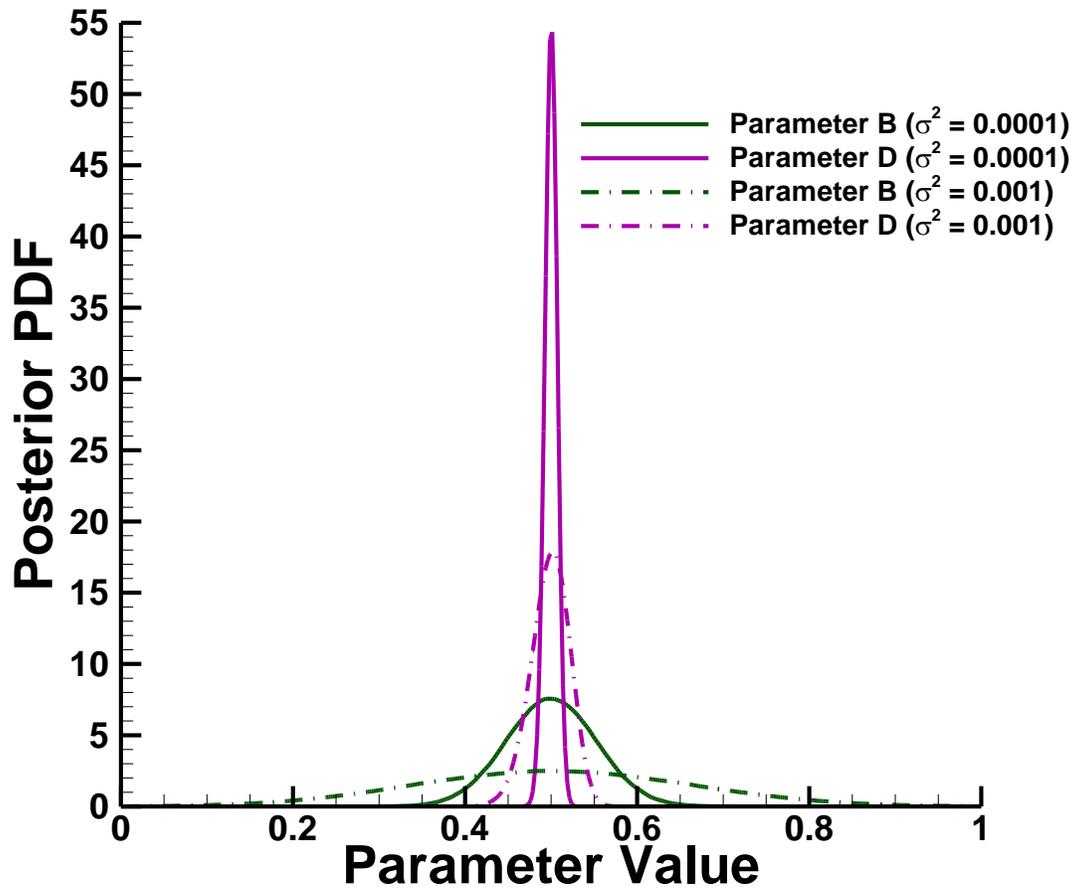


Figure 4.6. Posterior PDFs for two of the parameters from Eq. 4.5 based on two different calibrations with the synthetic data of Fig 4.3. The same 100 synthetic data points are used for both calibrations, but the uncertainty assumed for the synthetic data is different in each calibration. The calibrations both use 16 chains of 100,000 positions each, with a burn-in period of 20,000 chain positions.

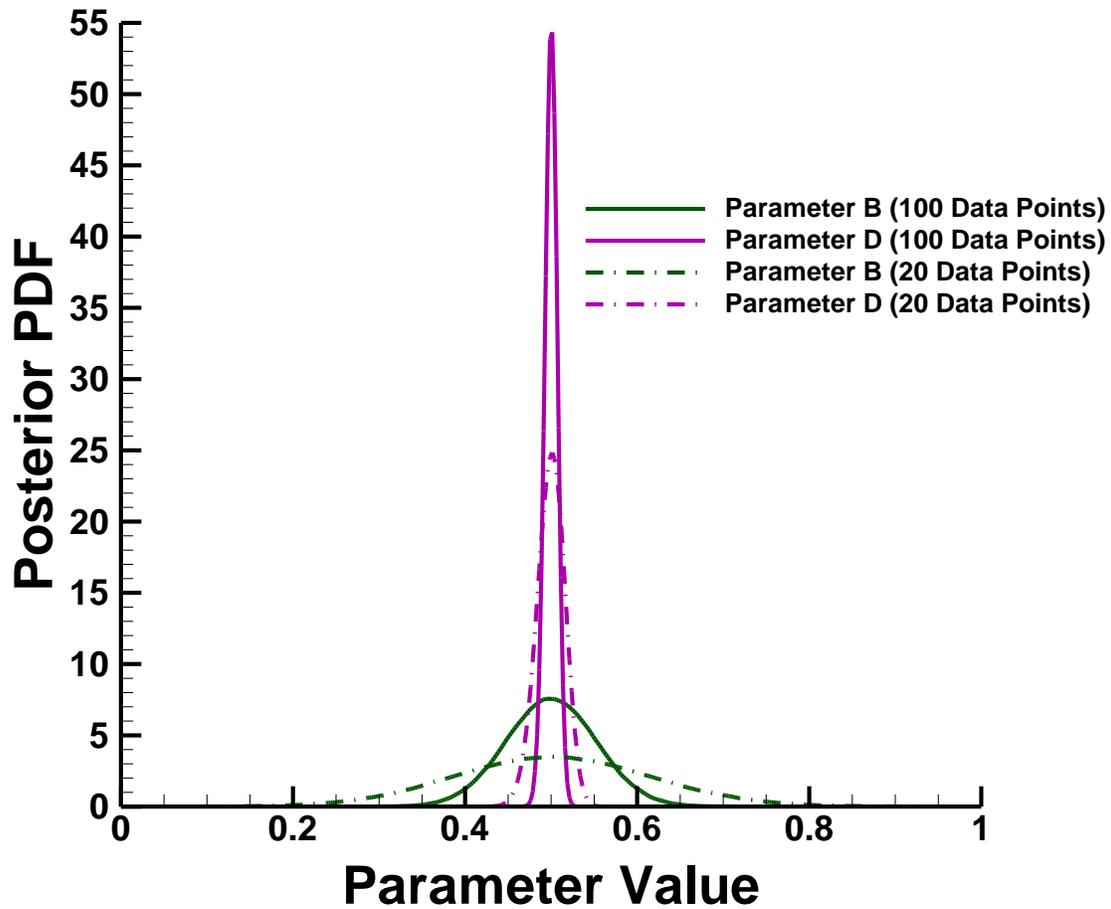


Figure 4.7. Posterior PDFs for two of the parameters from Eq. 4.5, based on two different calibrations with the synthetic data of Fig. 4.3. The same uncertainty is assumed for the synthetic data points in both calibrations, but the first calibration uses 100 synthetic data points and the second uses 20. The calibrations both use 16 chains of 100,000 positions each, with a burn-in period of 20,000 chain positions.

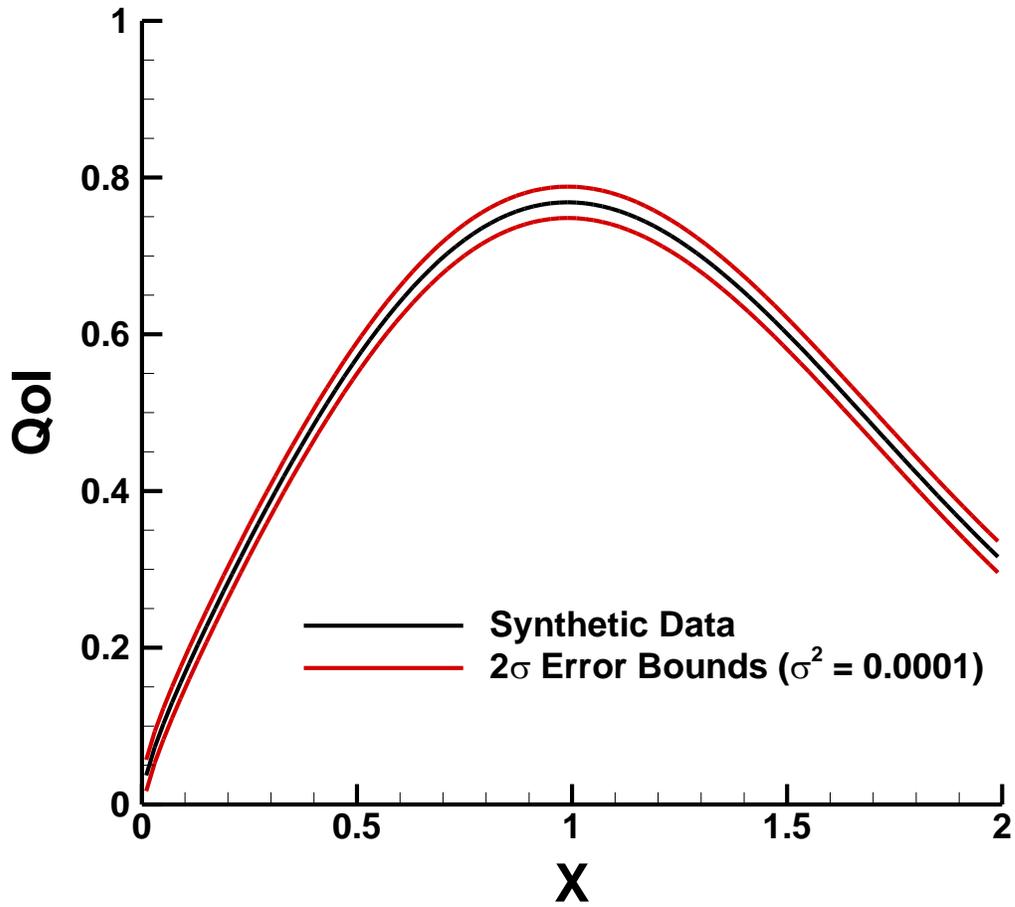


Figure 4.8. QoI(x) profile with parameters A, B, C, and D of Eq. 4.5 set to 0.3, 0.7, 0.4, and 0.6, respectively. This QoI(x) profile serves as the synthetic data for the final example calibration. Also shown are 2σ error bounds based on the assumption that the uncertainty of the synthetic data points is given by $\sigma^2 = 0.0001$.

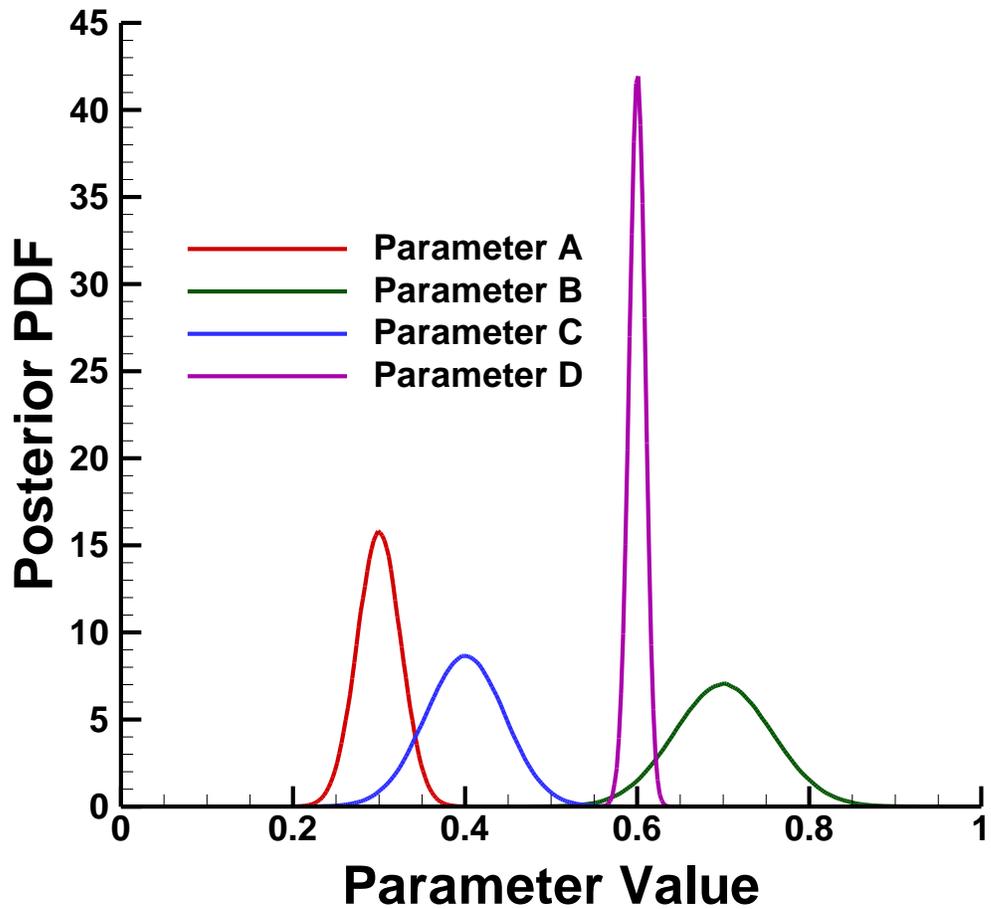


Figure 4.9. Posterior PDFs for each of the four parameters from Eq. 4.5 based on calibration with the synthetic data of Fig. 4.8 using 100 calibration data points and the assumption that $\sigma^2 = 0.0001$ for the synthetic data. The calibration is performed with 16 chains of 100,000 positions each, with a burn-in period of 20,000 positions for each chain.

Chapter 5: 0D Relaxation Sensitivity Analysis

OVERVIEW

With all of the algorithms and codes in place, we can now begin analyzing DSMC simulations relevant to hypersonic shocks. We will examine two types of scenario in this work; the first of these is a 0D relaxation from an initial high temperature state and the second is a 1D hypersonic shock. This chapter will cover the 0D relaxation scenario. The scenario will be described and results will be given for simulations with the nominal values for collision parameters and reaction rates (which were given in Chapter 2). Next, we identify the full set of parameters to be calibrated and establish the boundaries of a uniform prior distribution for each parameter. We will then discuss the Monte Carlo sampling of the parameter space and the sensitivity analyses based on this sampling.

SCENARIO

While it can be an interesting physical problem in its own right, we are interested in this scenario primarily because with the proper initial conditions a 0D relaxation can in some ways serve as a substitute for a 1D shock in the sensitivity analysis and calibration process. A 0D relaxation is less computationally expensive to simulate when compared to a 1D shock (although the relaxation can still take substantial computational time, especially if a low-noise result is needed), and thus it provides a good opportunity to make use of our analysis methods without waiting multiple days or even weeks for runs to be completed.

In the scenario we will use here, we initialize a 0D box with synthetic air (79% N₂, 21% O₂) with a bulk number density of 1.0×10^{23} #/m³. This scenario is intended as a substitute for a hypersonic shock at ~ 8 km/s. Based on the assumption that in a shock the

translational mode equilibrates much faster than the internal modes, the initial translational temperature is set to $\sim 50,000$ K while the rotational and vibrational temperatures start at 300 K. When referring to this scenario as a substitute for a 1D shock we do not mean that the simulation results mirror those of a 1D shock, we only mean that we believe simulation results from this scenario may be sensitive to many of the same parameters which would have a strong effect on the results of a 1D shock simulation.

In order to explore the chosen scenario further, a relaxation was performed with the nominal values of all physical parameters (i.e. with the values listed in Table 5.1 for collision parameters and the values listed in Table 5.2 for reaction rate parameters). Density profiles from this relaxation are shown in Figure 5.1. The bulk density is constant throughout the run (mass is conserved, and no new particles leave or enter the box) and is therefore not shown. N_2 and O_2 density both drop rapidly near the beginning of the relaxation, due primarily to dissociation reactions, although exchange reactions also play a role. Virtually all of the O_2 ends up dissociated less than a fourth of the way through the simulation, but for this set of initial conditions more than half of the initial N_2 density remains even at the end of the simulation. N and O density both increase rapidly early in the relaxation and a substantial amount of NO is also formed by the exchange reactions.

The reactions which dissociate the N_2 and O_2 are endothermic and thus a good deal of translational thermal kinetic energy is converted to chemical potential energy, leading to a rapid drop in the translational temperature. This drop is even more rapid because at the same time as the chemical equilibration, thermal equilibration is also occurring and large amounts of translational thermal kinetic energy are transferred to the internal energy modes (rotation and vibration). This can be seen in Figure 5.2, which

shows profiles for the translational, rotational, and vibrational temperatures of N_2 . The translational temperature of N_2 drops rapidly while the rotational and vibrational temperatures rise quickly from their initial value of 300 K. Thermal equilibrium is reached before chemical equilibrium. In fact, even at the end of the simulation the densities are still changing slightly; we are interested in the strongly non-equilibrium early part of the relaxation, and thus there is no need for us to continue the simulation all the way to equilibrium. As thermal equilibrium is reached, the rotational and vibrational temperatures stop rising, reach a peak, and then begin dropping along with the translational temperature as thermal energy continues to be lost to chemical potential energy due to the ongoing dissociation reactions. As the temperature continues to drop the important dissociation reaction rates drop and thus the temperature drop slows, and the densities and temperatures eventually asymptote toward equilibrium.

PARAMETERS

We will examine sensitivities to a large set of parameters relating to interactions (elastic collisions, inelastic collisions, and chemical reactions) between particles in DSMC simulations. The first group of parameters to be examined is the set of pre-exponential constants in the Arrhenius-type rate equations for the various reactions. We do not include the activation energy (E_A) or the temperature exponent (η) in the sensitivity analysis. In the case of E_A , this is because E_A is considered reasonably well known for diatomic species, at least in comparison to the uncertainties in the other Arrhenius rate parameters. We exclude η because the effects of Λ and η on the reaction rates are very strongly coupled. The reaction rate parameters available in the literature vary greatly and these parameters are considered highly uncertain at high temperatures

such as those in our scenario. Thus, we will allow the parameters to take on values over a range of two orders of magnitude. The lower limit for each pre-exponential constant will be $0.1\Lambda_{\text{nom}}$ and the upper limit will be $10\Lambda_{\text{nom}}$. In order to sample this two-order of magnitude range properly, we will actually sample the parameter $\log_{10}\Lambda$, which will have a uniform prior over the range from $\log_{10}\Lambda_{\text{nom}} - 1$ to $\log_{10}\Lambda_{\text{nom}} + 1$. Table 5.1 contains the full list of reaction parameters used in this work, along with the limits of the uniform prior and the nominal value for each parameter.

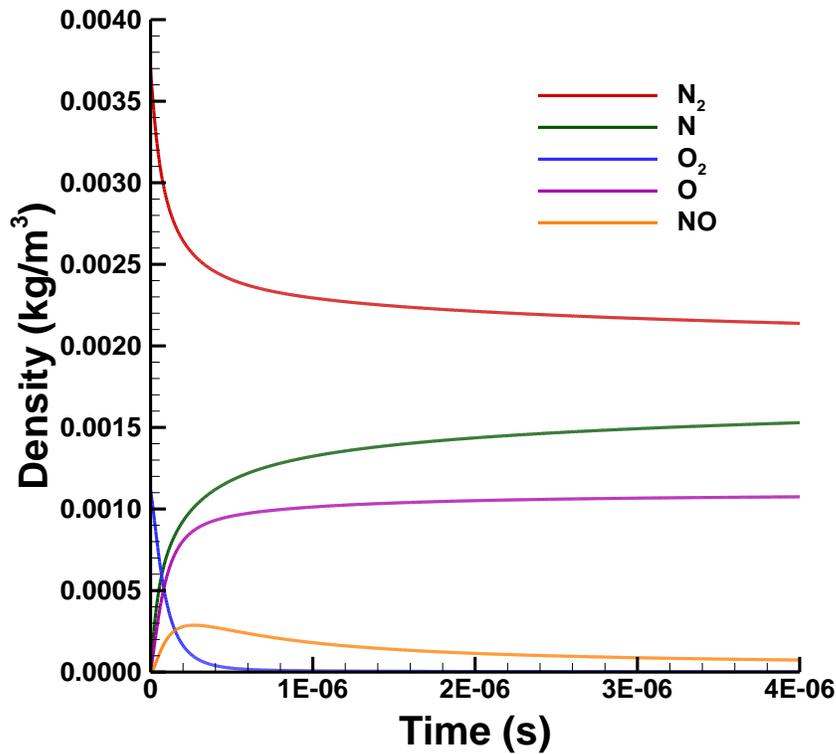


Figure 5.1: Evolution of the density profiles for all five species during a relaxation from an initial state with 79% N₂ and 21% O₂. Initial translational temperature is $\sim 50,000$ K while initial rotational and vibrational temperatures are 300 K.

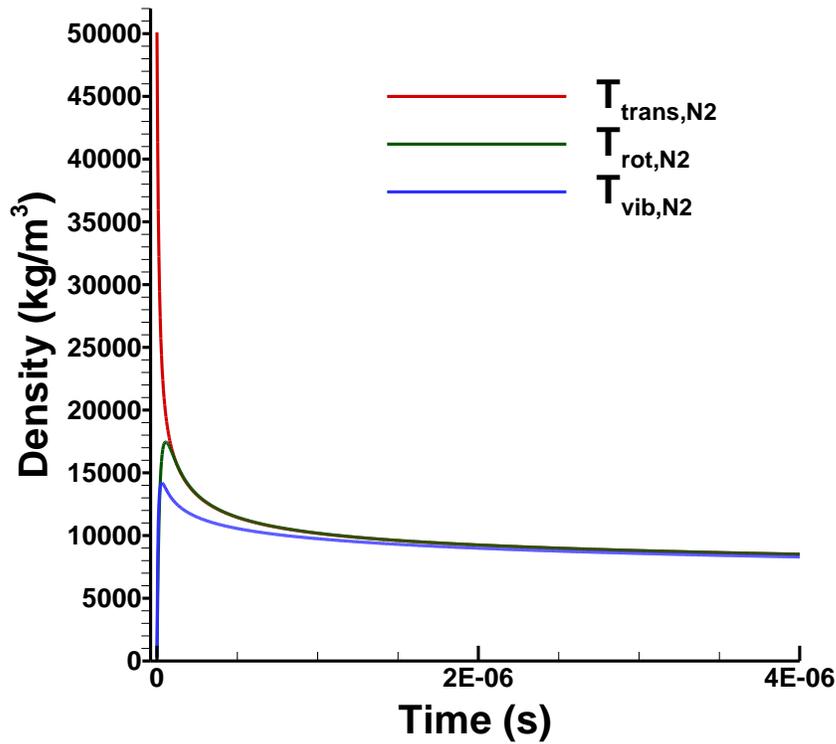


Figure 5.2: Evolution of the translational, rotational, and vibrational temperatures for N_2 during a relaxation from an initial state with 79% N_2 and 21% O_2 . Initial translational temperature is $\sim 50,000$ K while initial rotational and vibrational temperatures are 300 K.

The second group of parameters is the set of reference diameters for use in the VHS collision model. We will examine sensitivities to a total of 15 reference diameters, one for every possible combination of colliding species. We examine only d_{ref} (and not ω) because we found in our past work (Strand and Goldstein, 2010) that d_{ref} and ω are very highly coupled and can usually not be calibrated independently of one another with data from shock profiles. VHS parameters have been calibrated (at moderate temperatures) by comparing the experimentally measured relationship between viscosity and temperature to a set of transport equations derived from the Chapman-Enskog expansion of the Boltzmann equation, and although the validity of these parameter values

for higher temperatures is questionable, they are nonetheless much less uncertain than the reaction rate parameters. Therefore, we do not use order of magnitude uncertainties for these parameters. Instead, the lower limit for the uniform prior for a given reference diameter is taken to be one half of the nominal value for that reference diameter, and the upper limit is taken to be 1.5 times the nominal value. This set of parameters is listed in Table 5.2.

The third group of parameters is related to inelastic collisions. As discussed in Chapter 2, we use a constant rotational collision number (Z_R) for each species. We allow Z_R to vary over a two order of magnitude range, and thus actually check sensitivity to the parameter $\log_{10}Z_R$. This parameter has a uniform prior over the range from 0 to 2 (i.e. from $Z_R = 1$ to $Z_R = 100$). The next three parameters are the values of $\log_{10}Z_R$ for the three diatomic species. We used a nominal value of 3 for Z_R for all three diatomic species when doing the simulations shown in Figs. 5.1 and 5.2, so for $\log_{10}Z_R$ the nominal value of the parameter is not at the center of the uniform prior. This prior was chosen in order to examine a two order of magnitude range of Z_R , and values of $Z_R < 1$ are not physically meaningful. For a species with $Z_R = 1$, every collision involving that species will involve a redistribution between the rotational and translational modes, and it is not possible with our DSMC implementation to transfer energy to or from the rotational mode any more often than this. For most parameters, we use the nominal value to set the center of the prior, but aside from this the nominal value of a parameter plays no direct role in the sensitivity analysis, only the limits of the prior are important. Once those limits have been chosen, whether or not the nominal value is at the center of the prior makes no difference.

We use a collision-temperature dependent vibrational collision number (Z_V) which is calculated based on Eq. 2.1 (repeated here for convenience)

$$Z_V = (C_1/T_{coll}^\omega)e^{(C_2T_{coll}^{-1/3})}. \quad (2.1)$$

We include the values of C_1 for the three diatomic species as parameters 36-38 in our sensitivity analysis, and thus we remain consistent with our choice to check sensitivity to constants which affect the given equation linearly (as we did when choosing Λ and d_{ref} in the equations for the Arrhenius reaction rate and the VHS collision model, respectively) rather than constants which are in exponential terms. Since the uncertainty in C_1 is large for high temperature applications we will once again use a prior which spans two orders of magnitude, and thus we will actually check sensitivity to the parameter $\log_{10}C_1$. The range of the uniform prior will be from $\log_{10}C_{1,nom} - 1$ to $\log_{10}C_{1,nom} + 1$. Parameters 33-38 are listed in Table 5.2.

The final parameter for our sensitivity analysis is the ratio of real to simulated particles, commonly called $Fnum$ in the DSMC community. This is a numerical parameter rather than a physical one. We include this parameter in order to demonstrate both that our sensitivity analysis results are unaffected by DSMC stochastic noise and that we have sufficient simulated particles to accurately simulate the physics. Due to the fact that our scenario is a 0D relaxation there are no numerical parameters related to spatial grid resolution, and thus there are only two numerical parameters to consider, with $Fnum$ being the most important. The other numerical parameter is the time step, but due to the fact that all of the work is done in the DSMC collision/reaction step, the computational time required per time step scales almost linearly with the length of the time step. Therefore, without incurring significant additional computational expense, we can use a time step which is small enough to leave no doubt that our results are unaffected by this parameter. With this in mind, if we can show that $Fnum$ has negligible effect on the simulation results (for values of $Fnum$ within an appropriate range) then we can show that our sensitivity analysis results are independent of numerical parameters.

An appropriate range for this parameter was chosen empirically based on examination of the noise level present in simulations run with various values of F_{num} . In order to be sure that we are numerically converged we allow F_{num} to vary over an order of magnitude, and so we actually check sensitivity to $\log_{10}F_{\text{num}}$. The limits of the prior and the nominal value for F_{num} are given in Table 5.2.

Table 5.1: The first seventeen parameters for the 0D relaxation sensitivity analysis. These parameters correspond to the reactions which are important for 5-species air. For each reaction, the parameter of interest is $\log_{10}\Lambda$, where Λ is the pre-exponential constant in the Arrhenius rate equation (Eq. 2.2) for the forward reaction. Nominal Arrhenius forward rate parameters are from Gupta *et al.* (1989), and backward rate parameters are based on a matching with the equilibrium constant over a range of temperatures. Throughout the sensitivity analysis, the ratio of forward to backward rate for a given reaction is kept constant, since these ratios should be fixed by the equilibrium constant.

#	Parameter	Prior Distribution Limits		Nominal Value
		Minimum	Maximum	
1	$\log_{10}\Lambda (2\text{N}_2 \rightleftharpoons 2\text{N} + \text{N}_2)$	-13.10	-11.10	-12.10
2	$\log_{10}\Lambda (\text{N}_2 + \text{N} \rightleftharpoons 3\text{N})$	-8.16	-6.16	-7.16
3	$\log_{10}\Lambda (\text{N}_2 + \text{O}_2 \rightleftharpoons 2\text{N} + \text{O}_2)$	-13.50	-11.50	-12.50
4	$\log_{10}\Lambda (\text{N}_2 + \text{O} \rightleftharpoons 2\text{N} + \text{O})$	-13.50	-11.50	-12.50
5	$\log_{10}\Lambda (\text{N}_2 + \text{NO} \rightleftharpoons 2\text{N} + \text{NO})$	-13.50	-11.50	-12.50
6	$\log_{10}\Lambda (\text{O}_2 + \text{N}_2 \rightleftharpoons 2\text{O} + \text{N}_2)$	-11.92	-9.92	-10.92
7	$\log_{10}\Lambda (\text{O}_2 + \text{N} \rightleftharpoons 2\text{O} + \text{N})$	-12.22	-10.22	-11.22
8	$\log_{10}\Lambda (2\text{O}_2 \rightleftharpoons 2\text{O} + \text{O}_2)$	-11.27	-9.27	-10.27
9	$\log_{10}\Lambda (\text{O}_2 + \text{O} \rightleftharpoons 3\text{O})$	-10.82	-8.82	-9.82
10	$\log_{10}\Lambda (\text{O}_2 + \text{NO} \rightleftharpoons 2\text{O} + \text{NO})$	-12.22	-10.22	-11.22
11	$\log_{10}\Lambda (\text{NO} + \text{N}_2 \rightleftharpoons \text{N} + \text{O} + \text{N}_2)$	-10.18	-8.18	-9.18
12	$\log_{10}\Lambda (\text{NO} + \text{N} \rightleftharpoons 2\text{N} + \text{O})$	-8.88	-6.88	-7.88
13	$\log_{10}\Lambda (\text{NO} + \text{O}_2 \rightleftharpoons \text{N} + \text{O} + \text{O}_2)$	-10.18	-8.18	-9.18
14	$\log_{10}\Lambda (\text{NO} + \text{O} \rightleftharpoons \text{N} + 2\text{O})$	-8.88	-6.88	-7.88
15	$\log_{10}\Lambda (2\text{NO} \rightleftharpoons \text{N} + \text{O} + \text{NO})$	-8.88	-6.88	-7.88
16	$\log_{10}\Lambda (\text{N}_2 + \text{O} \rightleftharpoons \text{NO} + \text{N})$	-16.95	-14.95	-15.95
17	$\log_{10}\Lambda (\text{NO} + \text{O} \rightleftharpoons \text{O}_2 + \text{N})$	-18.80	-16.80	-17.80

Table 5.2: The remaining parameters for the 0D relaxation sensitivity analysis. Parameters 18-38 are collision parameters for 5-species air. Nominal values of d_{ref} were compiled by Ozawa (2008). Nominal values of C_1 (see Eq. 2.1) are based on those found in Bird (1994), but have been modified to account for the fact that Bird used different values for ω . The nominal value of 3 was considered to be a reasonable choice for Z_R for all three diatomic species (for this parameter the nominal value is not at the center of the uniform prior). The final parameter, F_{num} , is the ratio of real to simulated particles, and relates to the level of stochastic noise in the simulations.

#	Parameter	Uniform Prior Distribution		Nominal Value
		Minimum	Maximum	
18	$d_{\text{ref}}(\text{N}_2\text{-N}_2)$	1.79E-10	5.37E-10	3.58E-10
19	$d_{\text{ref}}(\text{N}_2\text{-N})$	1.67E-10	5.02E-10	3.35E-10
20	$d_{\text{ref}}(\text{N}_2\text{-O}_2)$	1.74E-10	5.21E-10	3.48E-10
21	$d_{\text{ref}}(\text{N}_2\text{-N})$	1.64E-10	4.91E-10	3.27E-10
22	$d_{\text{ref}}(\text{N}_2\text{-NO})$	1.75E-10	5.24E-10	3.50E-10
23	$d_{\text{ref}}(\text{N-N})$	1.56E-10	4.67E-10	3.11E-10
24	$d_{\text{ref}}(\text{N-O}_2)$	1.62E-10	4.86E-10	3.24E-10
25	$d_{\text{ref}}(\text{N-O})$	1.52E-10	4.55E-10	3.04E-10
26	$d_{\text{ref}}(\text{N-NO})$	1.63E-10	4.89E-10	3.26E-10
27	$d_{\text{ref}}(\text{O}_2\text{-O}_2)$	1.69E-10	5.06E-10	3.37E-10
28	$d_{\text{ref}}(\text{O}_2\text{-O})$	1.58E-10	4.75E-10	3.17E-10
29	$d_{\text{ref}}(\text{O}_2\text{-NO})$	1.70E-10	5.09E-10	3.39E-10
30	$d_{\text{ref}}(\text{O-O})$	1.48E-10	4.44E-10	2.96E-10
31	$d_{\text{ref}}(\text{O-NO})$	1.59E-10	4.78E-10	3.19E-10
32	$d_{\text{ref}}(\text{NO-NO})$	1.71E-10	5.12E-10	3.41E-10
33	$\log_{10}Z_R(\text{N}_2)$	0.0	2.0	0.48
34	$\log_{10}Z_R(\text{O}_2)$	0.0	2.0	0.48
35	$\log_{10}Z_R(\text{NO})$	0.0	2.0	0.48
36	$\log_{10}C_1(\text{N}_2)$	-0.3	1.7	0.7
37	$\log_{10}C_1(\text{O}_2)$	0.4	2.4	1.4
38	$\log_{10}C_1(\text{NO})$	-0.3	1.7	0.7
39	$\log_{10}F_{\text{num}}$	14.0	15.0	14.5

QUANTITY OF INTEREST

We must now select a quantity of interest (QoI) for this scenario. In this work, we are attempting to provide a framework for obtaining improved calibrations for parameters which are relevant to DSMC simulations of hypersonic shocks. Therefore, we have two constraints on our QoI. First, if possible, our QoI should be measurable by experiment. At the very least, our QoI should be closely related to some quantity which is experimentally observable. This is desired so that the QoI can be used for future parameter calibrations. The second requirement for our QoI is that calibrations based on it must inform at least some of the parameters we wish to calibrate, and thus we must choose a QoI which is sensitive to those parameters. Of course, we may not be able to find a QoI which is sensitive to all of the parameters we wish to calibrate, and which is also related directly to experimental data, and in that case we will need to pick the one that informs as many of the parameters we most wish to calibrate as possible.

There is a further constraint on our current QoI choices, however, which is imposed by the physical models which have been implemented in our code. We do not yet include modeling of ionization or electronic excitation, and we have not yet coupled our code with a suitable radiation solver. Radiation, ionization, and electronic excitation do not play a major role in the thermochemistry for a 1D shock at ~ 8 km/s (or for the 0D relaxation scenario in this chapter, which serves as a substitute for an ~ 8 km/s shock), and therefore we can correctly simulate the dominant physics of the scenarios we examine in this work, but we cannot generate output which can be compared with experimental radiation data.

With all of the above in mind, and based on the results of a past analysis of various options for the QoI (Strand and Goldstein, 2011), we will use the mass density of NO as the quantity of interest. This QoI will actually be a vector which represents ρ_{NO} at

various discrete points in time (as discussed for a hypothetical QoI in Chapter 3). When shown in figures, these points will be displayed as part of a continuous line, but the actual QoI is a vector composed of values at discrete points. Each discrete point will be viewed as a separate, scalar QoI when calculating r^2 and the mutual information. The sensitivity of ρ_{NO} to the various parameters can then be examined as a function of time over the course of the relaxation, which can provide valuable insight. In order to obtain an overall value of sensitivity for each parameter, we can appropriately integrate the sensitivities over all of the individual scalar QoIs. A schematic showing the definition of the QoI is shown in Figure 5.3.

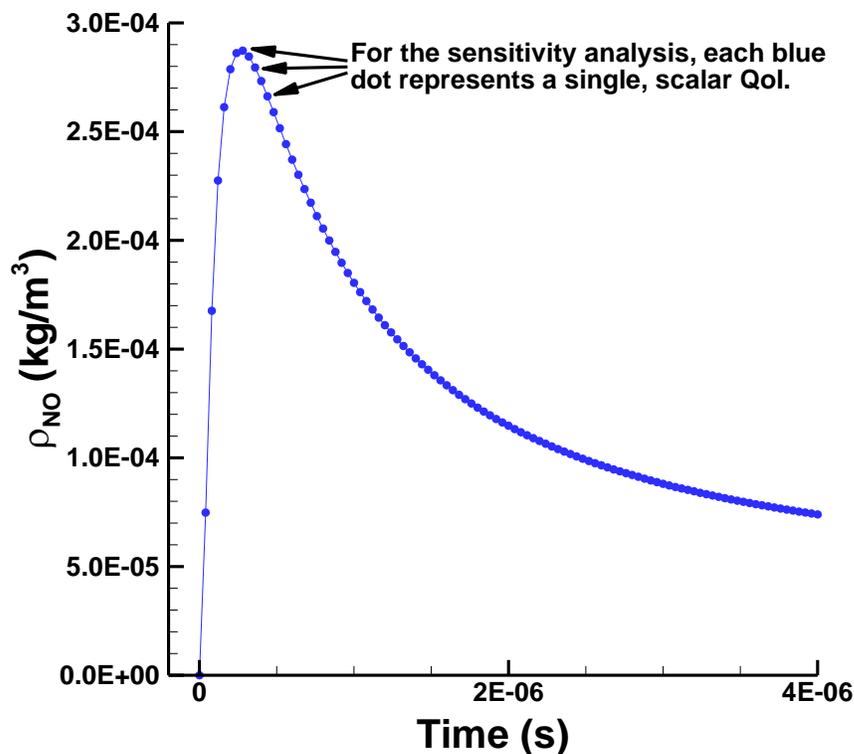


Figure 5.3. Schematic showing the way in which the ρ_{NO} vector QoI is broken into individual scalar QoIs during the sensitivity analysis.

Now that we have identified ρ_{NO} as our QoI, we briefly examine some of the reaction rates which we anticipate will have a strong effect on this QoI. Figure 5.4 shows six reaction rates as a function of time over the course of the relaxation, for a simulation run with the nominal values of all parameters. The first two are dissociation reactions which consume NO. The corresponding recombination reaction rates are very low and are not shown; the recombination rates do not balance the dissociation rates until very late times when the set of particles in the box approaches chemical equilibrium. Both of these dissociation rates peak at early times when the bulk temperature is high. The next two profiles are the forward and backward rates for the exchange reaction $\text{N}_2 + \text{O} \rightleftharpoons \text{NO} + \text{N}$. The forward reaction (which produces NO) peaks very early, and the backward reaction peaks slightly later. They reach equilibrium with one another less than a quarter of the way through the run, and after this point this reaction has no further effect on ρ_{NO} . The final two profiles are the forward and backward rates of the reaction $\text{NO} + \text{O} \rightleftharpoons \text{O}_2 + \text{N}$. It is actually the backward reaction which creates NO (we have defined the forward reaction as the endothermic reaction throughout this work). This reaction is responsible for most of the NO which is present in the early parts of the relaxation, since the backward (NO creating) rate is much higher than the forward (NO destroying) rate. Like the other exchange reaction, the forward and backward rates of this reaction reach equilibrium less than a quarter of the way through the relaxation, and from that point forward NO is slowly removed by the dissociation reactions until those finally come into equilibrium with their corresponding recombination reactions very late in the relaxation. Note that although these reactions are responsible for the majority of the creation and destruction of NO during the relaxation (at least for a simulation run with the nominal values of all parameters), other reactions still have a strong effect on

ρ_{NO} , because those reactions (such as dissociation of N_2 and O_2) will affect the bulk temperature and will thus affect the rates for the reactions shown in Fig. 5.4.

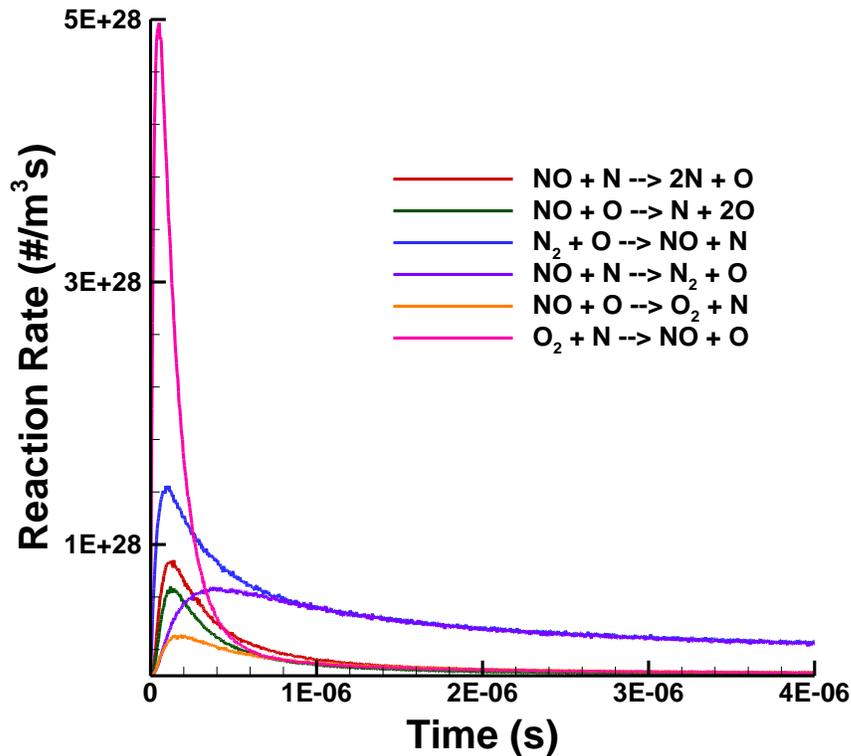


Figure 5.4: Evolution of the reaction rates for six reactions which directly affect ρ_{NO} during a relaxation from an initial state with 79% N_2 and 21% O_2 . Initial translational temperature is $\sim 50,000$ K while initial rotational and vibrational temperatures are 300 K.

SAMPLING THE PARAMETER SPACE

With our scenario and QoI in place, we can now carry out the sensitivity analysis. As discussed in Chapter 3, we will use two sensitivity analysis methods in this work, one based on the square of the Pearson correlation coefficient and the second based on the mutual information. Both of these methods are global in nature, and require a Monte

Carlo sampling of the parameter space. This sampling can be quite time consuming, but fortunately we can use the same data set for both methods.

The dataset for the 0D relaxation sensitivity analysis consists of 20,000 runs of the DSMC code, each at a different point in the 39-dimensional parameter space we are exploring. Before each run, the sensitivity analysis driver code performs a random number draw to determine a value for each of the 39 parameters, and these values are then sent to the DSMC code. When the DSMC code makes use of the first 17 parameters (the pre-exponential constants in the Arrhenius rate equations for the 17 reactions) the ratios between the pre-exponential constants for the forward and backward rate of each reaction are kept constant (so if, based on the random number draw, Λ for the forward rate of a given reaction is twice the nominal value from Table 5.2, then Λ for the backward rate is also double the nominal value). After each run, data for the QoI are output at a set of discrete points in time during the relaxation, and as mentioned previously, each of these points will be treated as an individual, scalar QoI.

The choice to use 20,000 Monte Carlo sample points was made based on past experience which gives us a good idea of how many sample points are needed for a proper sensitivity analysis. We will demonstrate later in this chapter that 20,000 sample points in parameter space is sufficient to accurately characterize the sensitivities we wish to examine.

In Figure 5.5 we show profiles of ρ_{NO} from simulations run at ten random points in the parameter space. The dramatic differences in these profiles demonstrate the large effect the values of the parameters can have on the QoI. Note that if they are followed far enough in time, all of these profiles will eventually reach the same equilibrium state (this has been verified for our code in the past, while implementing and testing the chemistry).

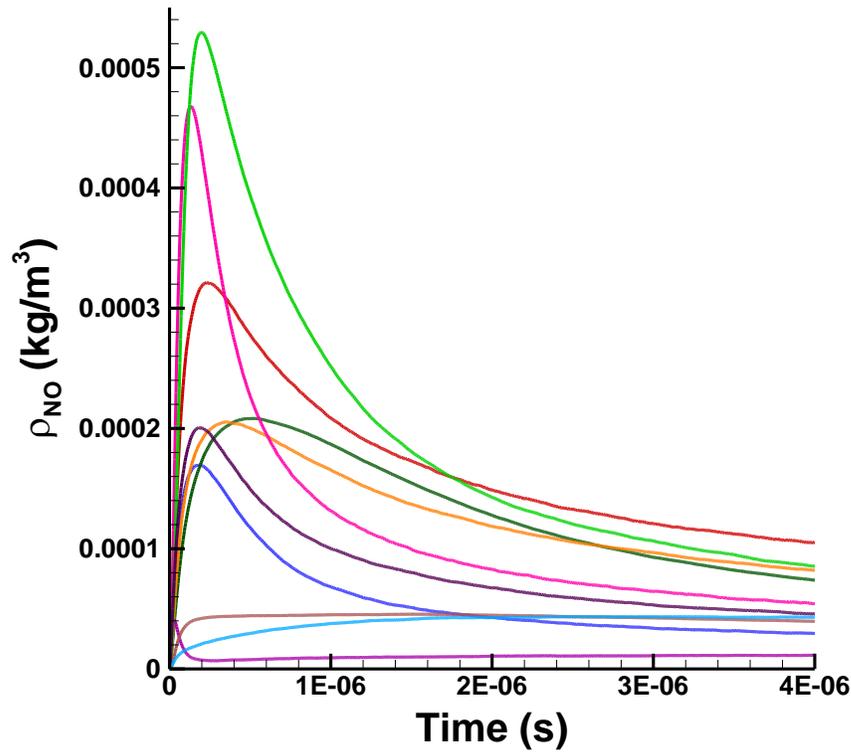


Figure 5.5: Profiles of ρ_{NO} vs. time for the same 0D relaxation scenario as shown in Fig. 5.1. Each profile comes from a simulation with a set of parameters corresponding to a different point in parameter space.

SENSITIVITIES VS. TIME

With the sampling of the parameter space complete, we make use of the resulting dataset and the analysis codes described in Chapter 3 to calculate r^2 and the mutual information for all 39 parameters at each scalar QoI. Once this process is complete, the results can be examined in various ways. First, it is useful to show the sensitivities to various parameters for all of the scalar QoIs on one plot. Since each scalar QoI represents a different discrete point in time, this allows us to see how the sensitivities vary with time during the relaxation. In Figure 5.6 sensitivities (based on r^2 and the

mutual information) are plotted as a function of time for the six most sensitive parameters. These six parameters are the values of $\log_{10}\Lambda$ for the dissociation reactions $\text{N}_2 + \text{N} \rightleftharpoons 3\text{N}$, $\text{O}_2 + \text{O} \rightleftharpoons 3\text{O}$, $\text{NO} + \text{N} \rightleftharpoons 2\text{N} + \text{O}$ and $\text{NO} + \text{O} \rightleftharpoons \text{N} + 2\text{O}$, and the values of $\log_{10}\Lambda$ for the exchange reactions. When comparing sensitivities measured with r^2 to those measured with mutual information note that only the shapes of the curves and qualitative trends can be compared; there is no reason to expect that the magnitudes will agree between the two sensitivity measures. Therefore, the axis scales have been chosen so that the highest peaks are roughly aligned. As discussed in Chapter 3, even the qualitative trends may differ, because the mutual information will capture non-linear correlations that r^2 may miss and also because the two measures respond differently to highly scattered data points. When we make comparisons between the two measures in this section, we are doing so primarily in order to identify potential features of the scatterplots which may be worth considering in more detail.

A number of interesting things are apparent in Fig. 5.6, and they can be more fully examined with the use of scatterplots. The scatterplots in Figures 5.7 – 5.12 are of the same type as the scatterplots used in Chapter 3 to illustrate the sensitivity analysis methods. The values of a parameter are on the x-axis and the values of a particular scalar QoI (in this case ρ_{NO} at a particular point in time) are on the y-axis, and the entire dataset is projected onto this two dimensional space. The values for the given parameter and the given scalar QoI from all of the 20,000 simulations are shown in the scatterplot. We will discuss the sensitivity vs. time profiles and selected scatterplots for each of the six parameters mentioned above.

The reaction $\text{N}_2 + \text{N} \rightleftharpoons 3\text{N}$ does not directly produce or destroy NO, of course, but it strongly affects the bulk temperature of the gas and thus it influences all of the other reaction rates. Two interesting features can be observed in the profiles of

sensitivity vs. time for this reaction. First, the profiles are nearly zero in the early part of the relaxation. The degree of chemical nonequilibrium is greatest in the early part of the shock, and in this region ρ_{NO} is most strongly affected by the NO dissociation reactions $\text{NO} + \text{N} \rightleftharpoons 2\text{N} + \text{O}$ and $\text{NO} + \text{O} \rightleftharpoons \text{N} + 2\text{O}$ and by the exchange reactions, since these are the primary reactions which actually create and destroy NO. After the backward and forward rates of the exchange reactions reach equilibrium and the NO dissociation reactions slow down, however, the bulk temperature begins to have a strong influence on the speed at which the chemical composition approaches equilibrium, and at these later times $\text{N}_2 + \text{N} \rightleftharpoons 3\text{N}$ becomes one of the most sensitive parameters. The second feature of interest concerning this reaction is the fact that the correlation coefficient is very low relative to the mutual information over the entire sensitivity vs. time curve. The reason for this can be seen in the scatterplots of Fig. 5.7. All four scatter plots demonstrate that the effect of this parameter on the QoI is quite non-linear. Moving from left to right in any one of the scatterplots (from low to high values of the parameter), it is apparent that on the whole there is a negative correlation between this reaction rate and ρ_{NO} (as this reaction rate increases, ρ_{NO} at a given point in time decreases for most of the 20,000 runs). However, superimposed on this trend is the fact that the distribution of values of ρ_{NO} at each given time actually broadens at higher values of this reaction rate. This is most likely due to the fact that as the rate of N_2 dissociation increases the bulk temperature generally decreases more quickly, which leads in general to lower rates for the exchange reactions which produce NO. At the same time, a higher rate of N_2 dissociation means that there is more atomic nitrogen available to react with O_2 (before the O_2 has been mostly dissociated) and form NO through the backward portion of the exchange reaction $\text{NO} + \text{O} \rightleftharpoons \text{O}_2 + \text{N}$ (which is a high rate reaction early in the relaxation, as seen in Fig. 5.4). Thus, for some combinations of parameter values a high

rate for $N_2 + N \rightleftharpoons 3N$ can actually lead to the highest values for ρ_{NO} . Regardless of the reason for this nonlinear relationship, the mutual information captures it much better than r^2 , and therefore the mutual information is substantially higher than r^2 at all times during the relaxation.

The profiles for the reaction $O_2 + O \rightleftharpoons 3O$ are simpler. The importance of this reaction peaks at relatively early times, and then drops later once almost all of the O_2 has been dissociated. At higher rates of this reaction, the temperature of the gas drops more quickly and less NO is produced, leading to the negative relationship between this parameter and ρ_{NO} which is seen in the scatterplots of Fig 5.8.

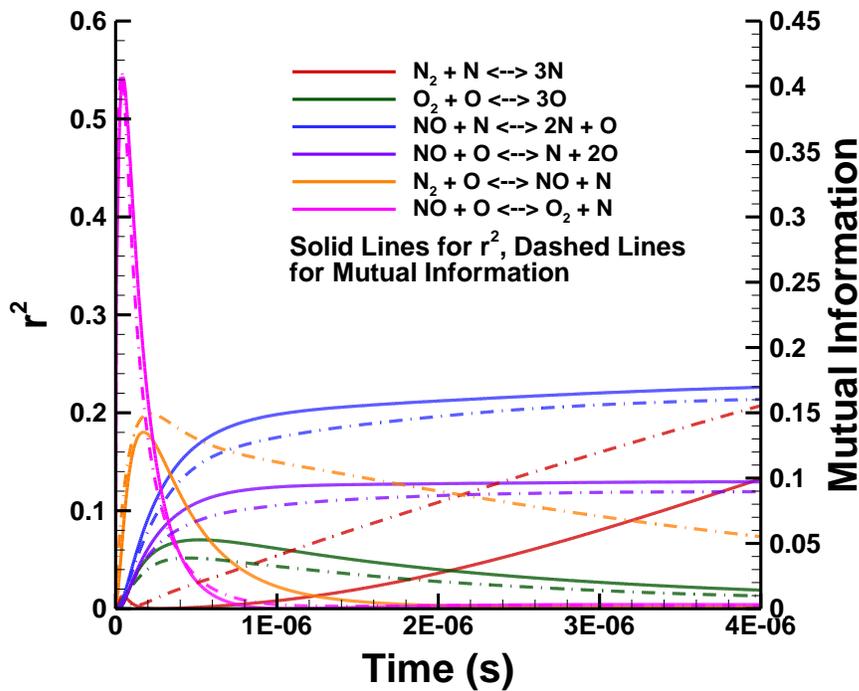


Figure 5.6. Sensitivities as a function of time for the six most sensitive parameters, all of which are from among the reaction rate parameters (parameters 1-17 in Table 5.1). Both r^2 and the mutual information are shown on the plot for each parameter.

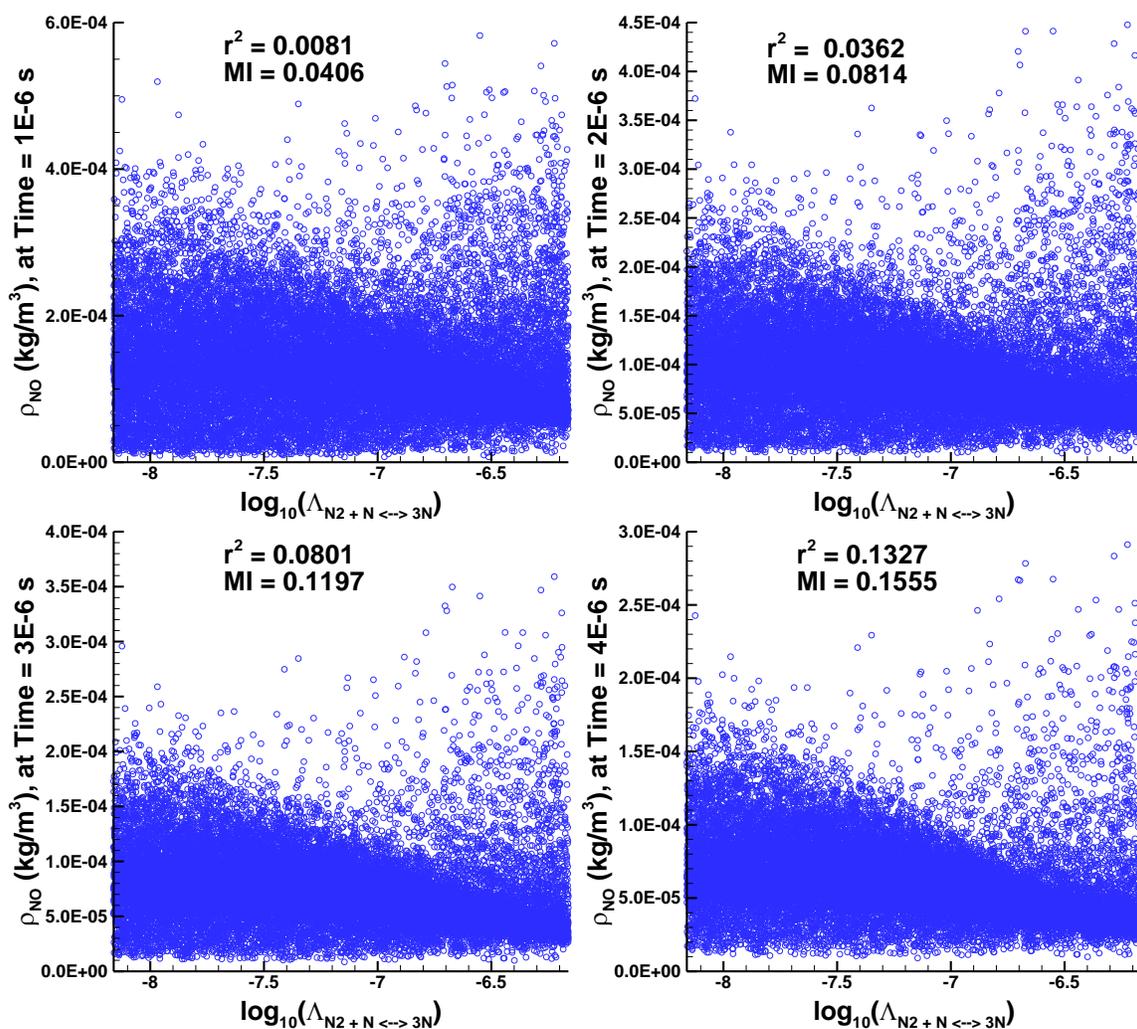


Figure 5.7. Scatterplots showing ρ_{NO} (at four discrete points in time given by the labels on the vertical axis in each plot) vs. $\log_{10}\Lambda$ for the reaction $N_2 + N \rightleftharpoons 3N$. The square of the Pearson correlation coefficient and the mutual information are also shown on the plots.

The profiles for the reactions $NO + N \rightleftharpoons 2N + O$ and $NO + O \rightleftharpoons N + 2O$ are also relatively simple. They start out negligible, since at very early times the reactions which produce NO are dominant. Then, as time passes and more NO is available to dissociate, these two reactions rapidly become more important and they remain important for the rest of the run. It is obvious that an increase in the rate of either of these reactions will lead to

a decrease in ρ_{NO} since the forward reaction destroys NO and occurs much more frequently than the backward (recombination) reaction until very late in the relaxation when the composition approaches equilibrium. This idea is shown to be correct by the scatterplots in Figs. 5.9 and 5.10.

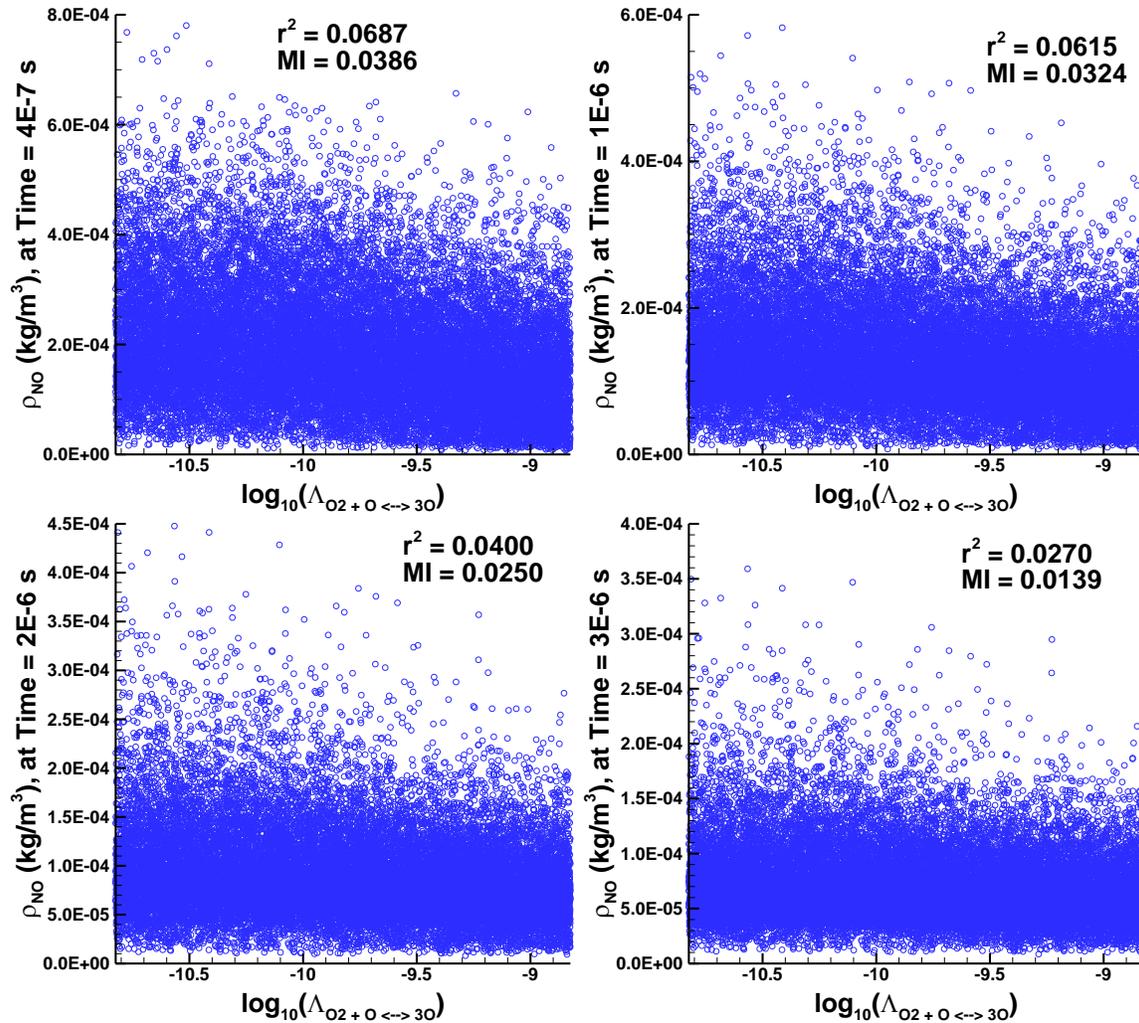


Figure 5.8. Scatterplots showing ρ_{NO} (at four discrete points in time given by the labels on the vertical axis in each plot) vs. $\log_{10}\Lambda$ for the reaction $\text{O}_2 + \text{O} \rightleftharpoons 3\text{O}$. The square of the Pearson correlation coefficient and the mutual information are also shown on the plots.

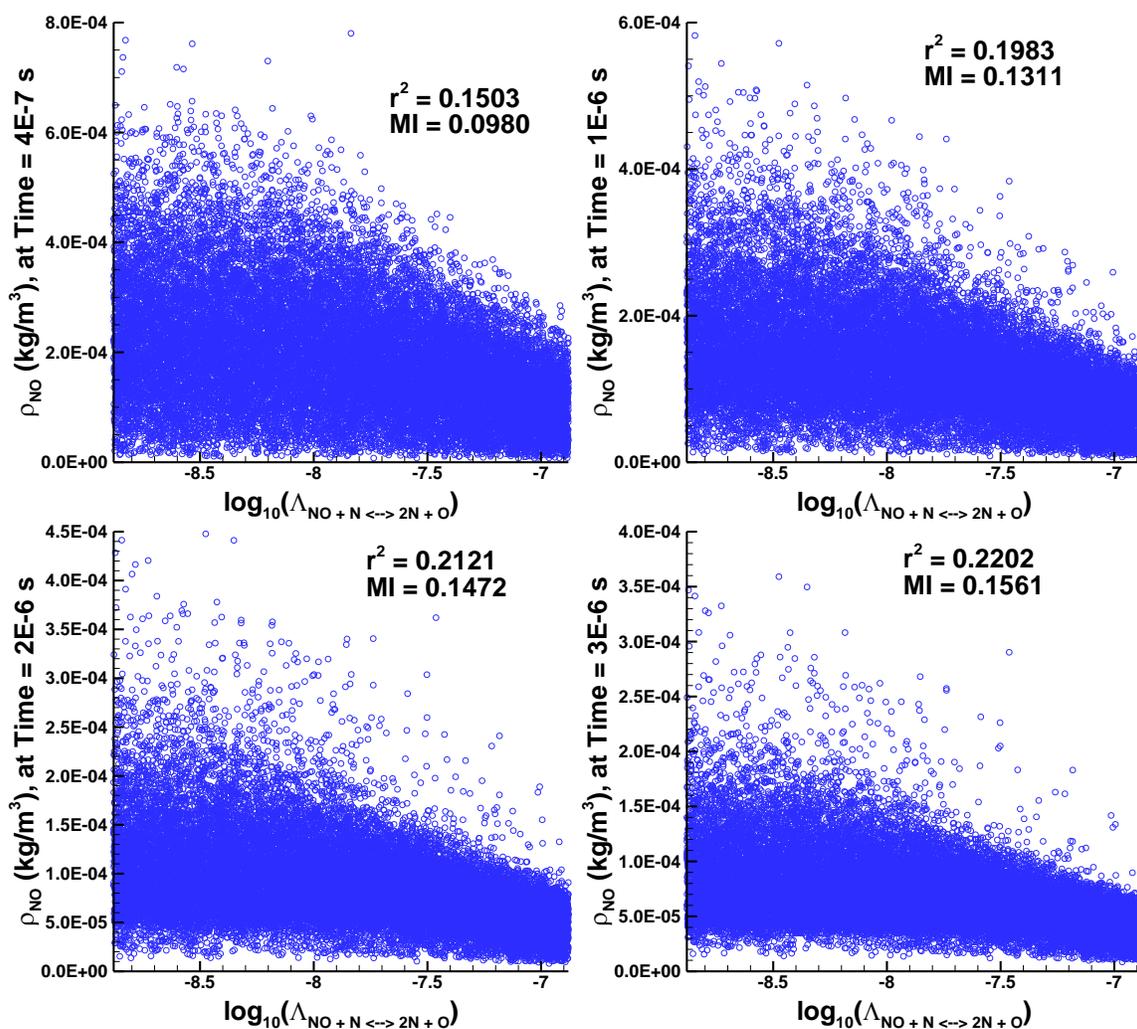


Figure 5.9. Scatterplots showing ρ_{NO} (at four discrete points in time given by the labels on the vertical axis in each plot) vs. $\log_{10}\Lambda$ for the reaction $\text{NO} + \text{N} \rightleftharpoons 2\text{N} + \text{O}$. The square of the Pearson correlation coefficient and the mutual information are also shown on the plots.

The sensitivity vs. time profiles for the exchange reaction $\text{N}_2 + \text{O} \rightleftharpoons \text{NO} + \text{N}$ differ markedly depending on which measure of sensitivity is used. Both measures reach a peak early in the relaxation, but after this the profile for r^2 drops off rapidly and approaches zero while the profile for the mutual information stays non-negligible over the entire relaxation. The reason for this can be discerned from the scatterplots in Fig.

5.11. At an early time there is a clear and obvious positive relationship between this parameter and ρ_{NO} , and this relationship is easily captured by both r^2 and the mutual information. At later times, however, the relationship changes and the distribution of ρ_{NO} is much broader for low values of this parameter than it is for high values. This is most obvious in the lower right scatterplot of Fig. 5.11. Starting with values of this parameter which are slightly below the nominal and moving higher, we see that this parameter no longer has any effect on the distribution of ρ_{NO} . At very low values of this parameter, however, the distribution of ρ_{NO} is clearly broadened. The reason for this was discussed in relation to Fig. 5.4. After a certain point in time the forward and backward rates of this reaction reach equilibrium and at that point this reaction no longer has any effect on the QoI. In general, higher values of the rate coefficient for this parameter (which will result in higher rates of both the forward and backward reaction) lead to this equilibrium being reached at earlier times. Lower values of the rate coefficient for this reaction mean that this reaction takes longer to reach equilibrium and thus affects ρ_{NO} for a longer time during the relaxation. Whether this affect is to decrease or increase ρ_{NO} at a given point in time, however, depends on the values of other parameters, and therefore the distribution of ρ_{NO} is broadened for low values of this parameter. This type of behavior cannot be captured by r^2 . It will be captured by the mutual information because the mutual information compares distributions rather than simply looking for linear correlations. Situations like this are the reason the mutual information was employed in the first place, because it provides a more sophisticated measure of sensitivity.

Finally, we examine the sensitivity vs. time profile for the other exchange reaction, $\text{NO} + \text{O} \rightleftharpoons \text{O}_2 + \text{N}$. The profile is almost identical for the two measures of sensitivity. It has a sharp peak at early times and then drops fairly quickly and soon approaches zero. The scatterplots in Fig. 5.12 show the strong, positive relationship

between this parameter and ρ_{NO} at early times, and they also show how the effect of this parameter on ρ_{NO} becomes negligible at later times. Since the relationships can be reasonably well described by a linear trendline r^2 is able to capture this sensitivity properly.

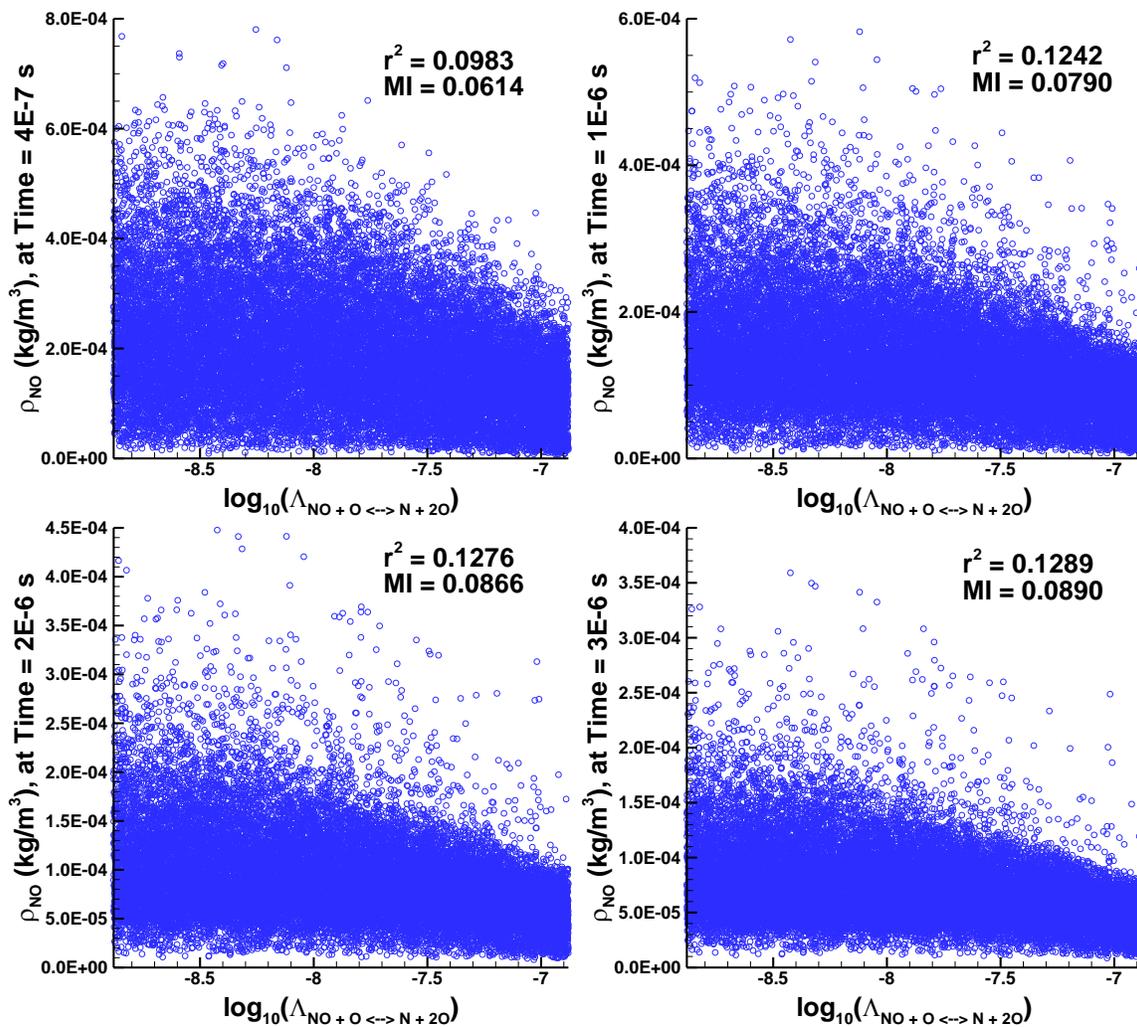


Figure 5.10. Scatterplots showing ρ_{NO} (at four discrete points in time given by the labels on the vertical axis in each plot) vs. $\log_{10}\Lambda$ for the reaction $\text{NO} + \text{O} \rightleftharpoons \text{N} + 2\text{O}$. The square of the Pearson correlation coefficient and the mutual information are also shown on the plots.

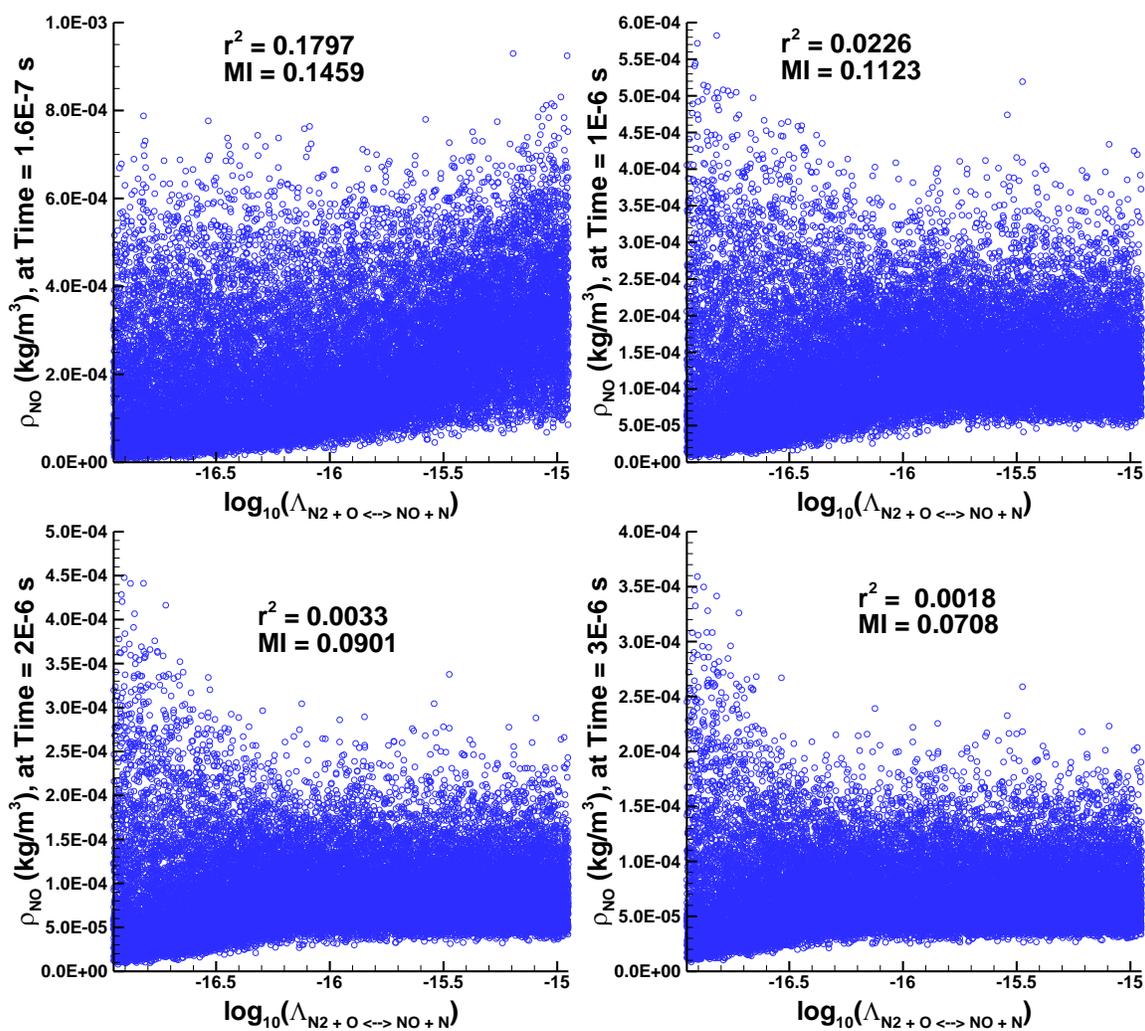


Figure 5.11. Scatterplots showing ρ_{NO} (at four discrete points in time given by the labels on the vertical axis in each plot) vs. $\log_{10}\Lambda$ for the reaction $N_2 + O \rightleftharpoons NO + N$. The square of the Pearson correlation coefficient and the mutual information are also shown on the plots.

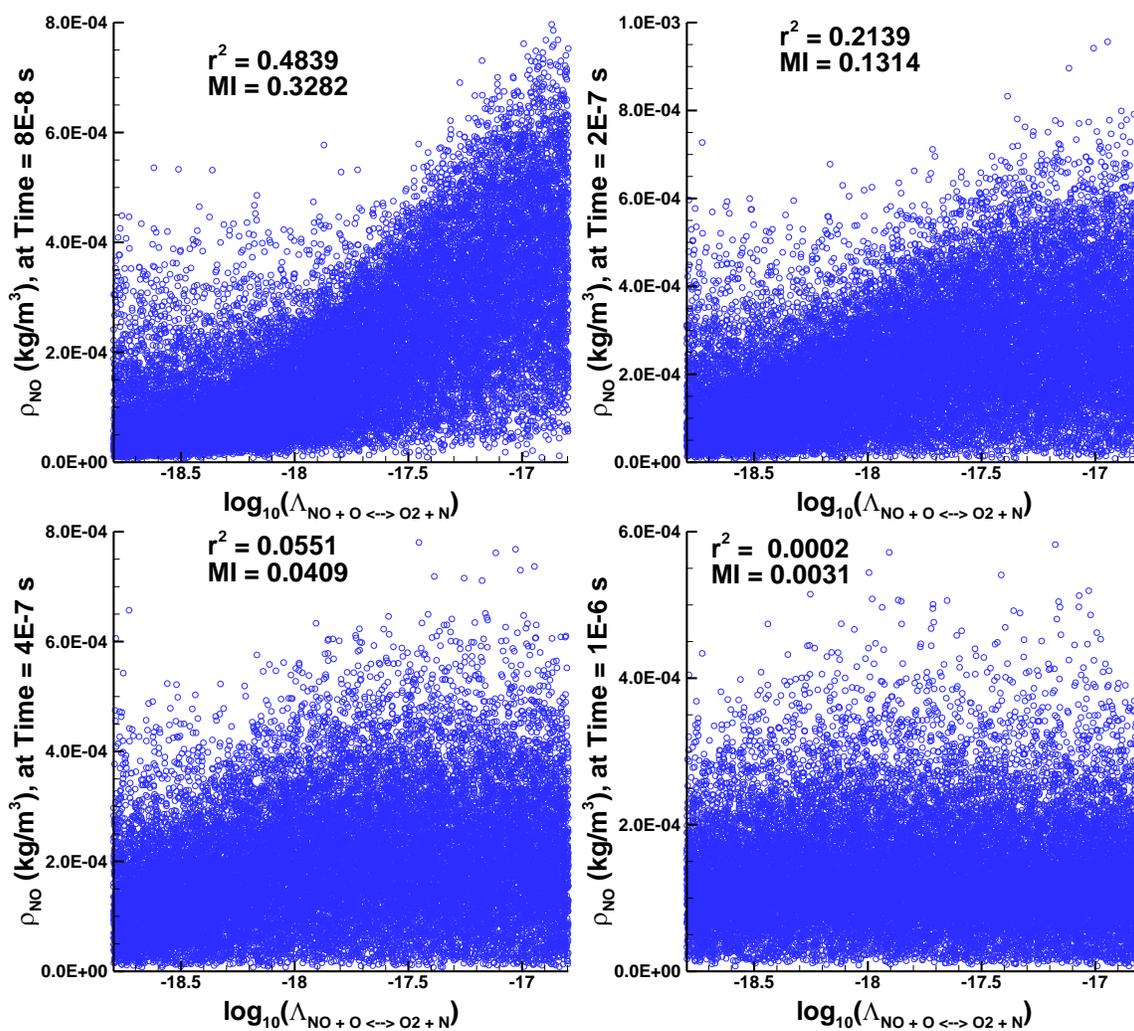


Figure 5.12. Scatterplots showing ρ_{NO} (at four discrete points in time given by the labels on the vertical axis in each plot) vs. $\log_{10}\Lambda$ for the reaction $\text{NO} + \text{O} \rightleftharpoons \text{O}_2 + \text{N}$. The square of the Pearson correlation coefficient and the mutual information are also shown on the plots.

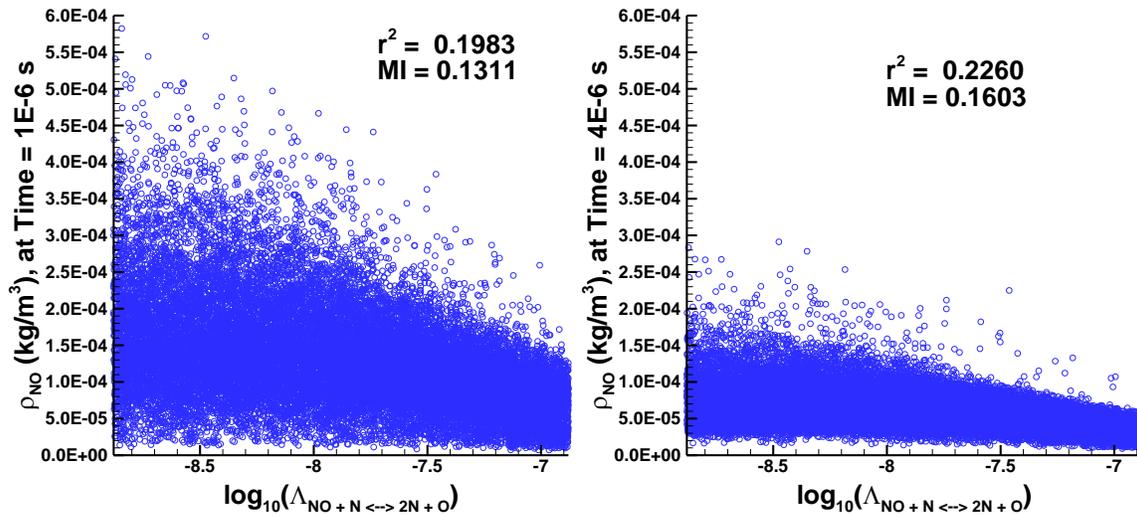


Figure 5.13. Scatterplots showing ρ_{NO} (at two discrete points in time given by the labels on the vertical axis in each plot) vs. $\log_{10}\Lambda$ for the reaction $\text{NO} + \text{N} \rightleftharpoons 2\text{N} + \text{O}$. The vertical scale is the same for both plots in order to show the substantially higher variance of ρ_{NO} in the scatterplot on the left, despite both r^2 and the mutual information being higher for the scatterplot on the right.

VARIANCE-WEIGHTED SENSITIVITIES

We would also like to obtain a single number for the sensitivity of the ρ_{NO} vector QoI to each parameter. We could try to do this by integrating the sensitivity for each parameter with respect to time over the full length of the simulation (that is, obtain the area under each curve shown in Fig. 5.6). This would be a poor measure of overall sensitivity, however. The plot in Fig. 5.6 is useful for comparing sensitivities to various parameters at a given point in time, and also for seeing how those sensitivities vary relative to one another as the relaxation progresses, but this plot is useless for obtaining meaningful overall sensitivities to the parameters. The reason for this is illustrated in Figure 5.13. The values of r^2 and the mutual information for a given parameter, at a given point in time, provide a measure of how much of the total variance of the QoI (at

that point in time) is explained by that parameter. The values of r^2 and the mutual information provide no information at all about the magnitude of the variance of the QoI at that point in time. Therefore, when comparing sensitivities of one or more parameters at two different points in time, it is frequently the case that while the r^2 and mutual information values may be very similar for the two points, how much the parameter actually affects the QoI in an absolute sense may be very different. The r^2 and mutual information values for the scatterplot shown in the left image of Fig. 5.13 are actually lower than the corresponding values for the scatterplot in the right image of Fig. 5.13, but the absolute effect of the parameter on the QoI is clearly larger in the left image. This occurs because of the fact that the relaxation starts from a highly non-equilibrium initial state and approaches equilibrium over time. After a sufficient amount of time ρ_{NO} will reach the same equilibrium regardless of the parameter values chosen for a particular simulation. Therefore, as time passes and the gas approaches equilibrium the parameters have a diminishing effect on the value of the ρ_{NO} .

In order to obtain a meaningful overall sensitivity, we need to make use of a weighting function before integrating the sensitivities over the length of the relaxation. In this work, we have chosen to use the variance of the QoI as the weighting function, so the equation for the overall sensitivity (based on the r^2 measure) to a given parameter is

$$\text{Overall Sensitivity} = \int_t \text{var}_{\text{QoI}}(t)r^2(t)dt \quad (5.1)$$

where $\text{var}_{\text{QoI}}(t)$ is the variance of the QoI at a given point in time (calculated from all the Monte Carlo samples of the parameter space), and the integration is over the entire length of the relaxation. The same equation is used when mutual information is the measure of sensitivity, except in that case $r^2(t)$ is replaced by $\text{MI}(t)$. The variance of ρ_{NO} and the integrand of Eq. 5.1 (which we refer to as the variance-weighted sensitivity) are shown as functions of time in Figure 5.14 for the six most sensitive parameters. Figure 5.15 shows

the same curves, this time zoomed in on the early part of the relaxation where the variance-weighted sensitivities are highest. In both of these figures, the variance-weighted sensitivity curves have been normalized by the peak sensitivity (the highest sensitivity at any point in the relaxation for any parameter). This is done separately for each measure of sensitivity, so the highest r^2 based sensitivity at any point in the relaxation is set equal to one, and similarly for the mutual information.

With the definition of the variance-weighted sensitivity established, Eq. 5.1 is then used to obtain overall sensitivities, based on r^2 and the mutual information, for each parameter. The overall sensitivity to each parameter (based on each measure) is normalized by the largest value of overall sensitivity (based on that measure), so that the most sensitive parameter based on each measure has an overall sensitivity of one.

The results of these calculations are shown in Figure 5.16. The reaction $\text{NO} + \text{N} \rightleftharpoons 2\text{N} + \text{O}$ has the greatest influence on ρ_{NO} when r^2 is used as the measure, and it is a close second based on mutual information. The exchange reaction $\text{N}_2 + \text{O} \rightleftharpoons \text{NO} + \text{N}$ has the highest overall sensitivity based on the mutual information, while it is only the fourth most sensitive based on r^2 . This difference is due to the previously discussed non-linearity of the relationship between this reaction rate and ρ_{NO} . For most of the other parameters the r^2 and mutual information based overall sensitivities are relatively similar, with the exception of the reaction rate for $\text{N}_2 + \text{N} \rightleftharpoons 3\text{N}$, where non-linearity of the relationship again leads to significantly higher values for mutual information than for r^2 . The set of the top six most sensitive reactions is the same for either measure, and these six are all labeled in Fig. 5.16. It is unsurprising that four of the six reactions result directly in the production or destruction of NO (two NO dissociation/recombination reactions and the two exchange reactions). The other two reactions in the top six ($\text{N}_2 + \text{N} \rightleftharpoons 3\text{N}$ and $\text{O}_2 + \text{O} \rightleftharpoons 3\text{O}$) have such a large effect on the

temperature and chemical composition during the relaxation that they still have a strong effect on ρ_{NO} , even though NO is neither a reactant nor a product of either of them.

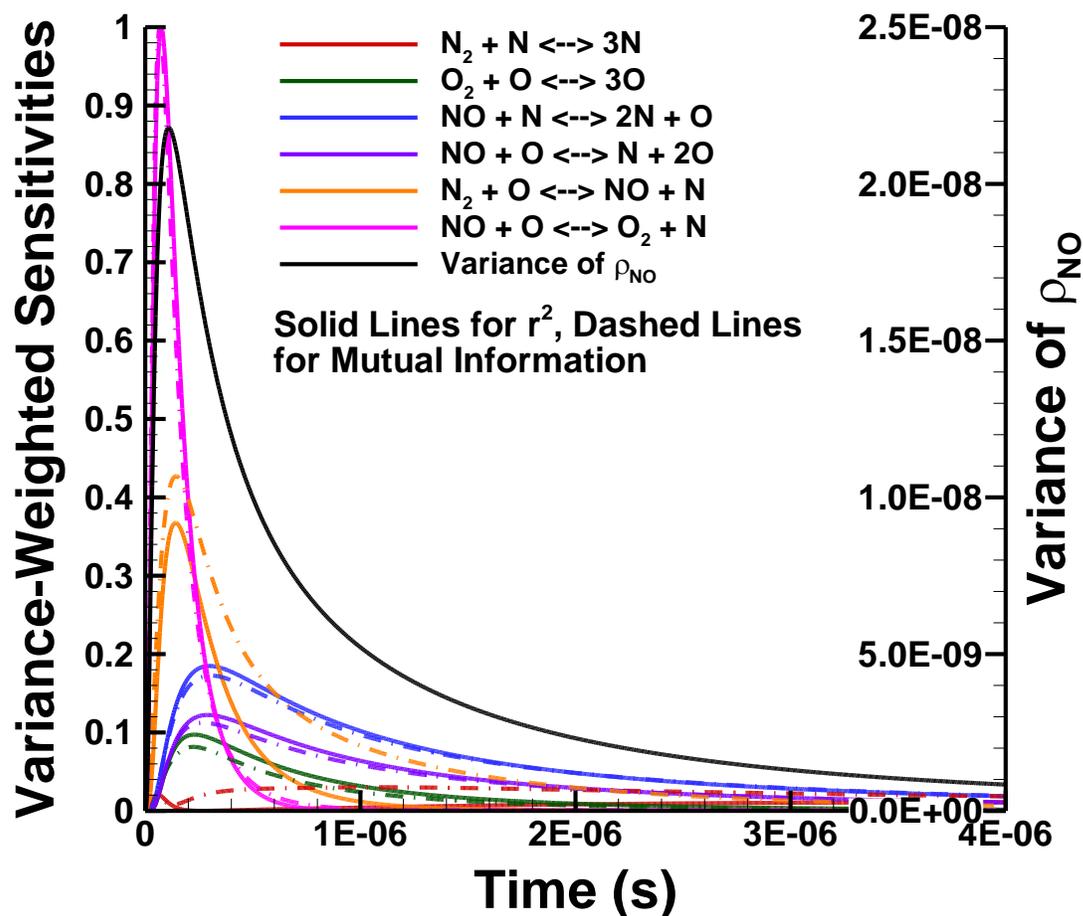


Figure 5.14. Variance-weighted sensitivities as a function of time for the top six most sensitive parameters, all of which are from among the reaction rate parameters (parameters 1-17 in Table 5.1). Sensitivities based on both r^2 and the mutual information are shown on the plot for each parameter, and the variance of ρ_{NO} at each point in time is shown as well. The variance-weighted sensitivities are normalized so that the highest sensitivity at any point in the relaxation is set equal to one. This is done separately for r^2 and the mutual information.

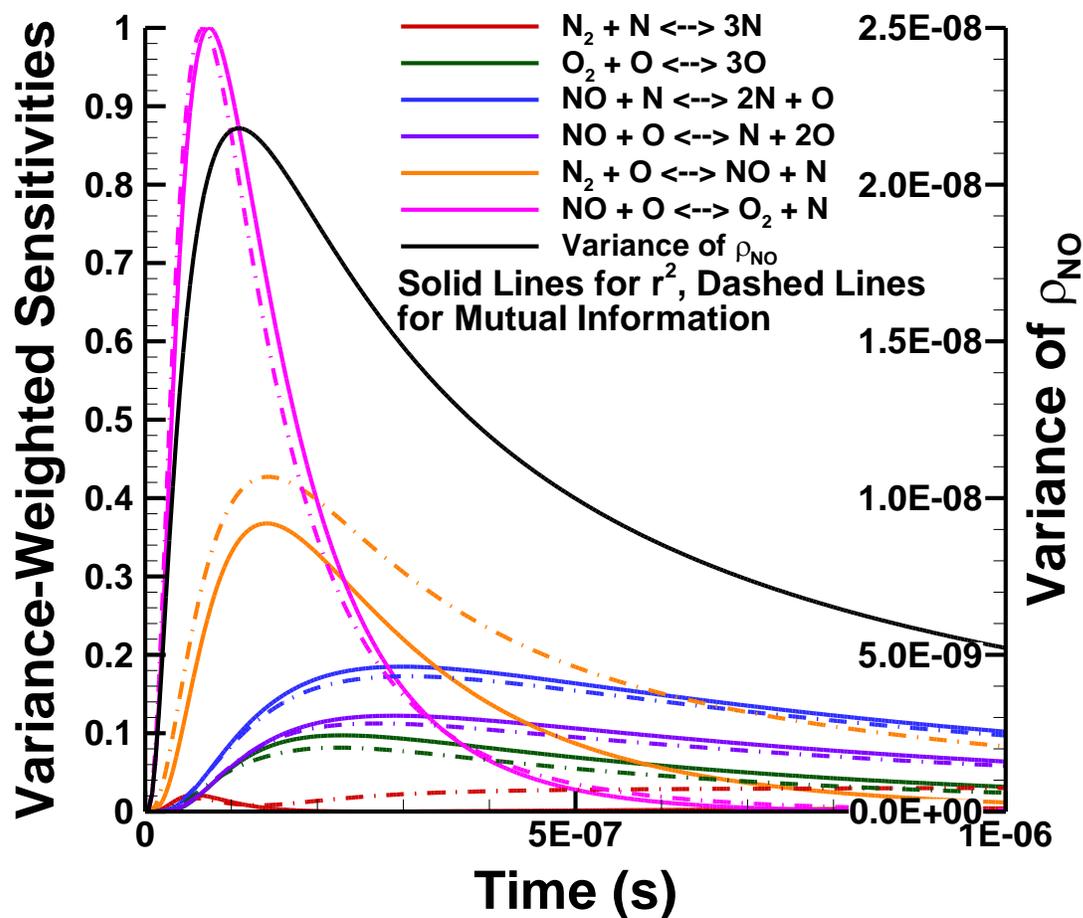


Figure 5.15. Zoomed-in version of Fig. 5.14, showing in more detail the normalized, variance-weighted sensitivities for early times when the variance of ρ_{NO} is highest.

Based on these results, the parameters which are suitable for calibration based on ρ_{NO} as QoI are apparent. If we had sufficient data with reasonable levels of uncertainty, we might hope to calibrate the pre-exponential constants in the Arrhenius rate equations for the six reactions labeled in Fig. 5.16. We would not expect to gain very much meaningful information about the other parameters from a calibration based on our current QoI.

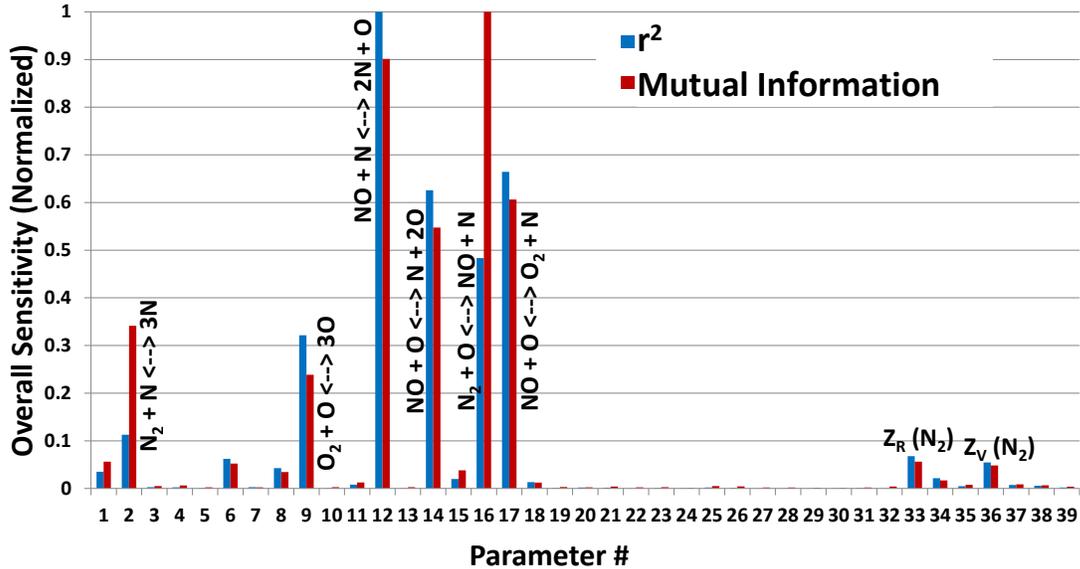


Figure 5.16. Overall, variance-weighted sensitivities for each parameter based on each measure of sensitivity. The overall sensitivities are normalized so that the overall sensitivity for the most sensitive parameter is set equal to one. This normalization is done separately for r^2 and the mutual information, so the most sensitive parameter for each measure will have an overall sensitivity of one.

CONVERGENCE

In any numerical study it is important to demonstrate convergence of the results before drawing conclusions. In our work, there are actually two types of convergence which must be demonstrated. We must show that the results are converged with respect to the DSMC numerical parameters and we must also show that we have sufficient points in our Monte Carlo sensitivity analysis.

Addressing DSMC convergence first, it is important to note that we are not trying to demonstrate that each individual simulation is converged to machine precision; we are only trying to demonstrate that we are sufficiently converged that our sensitivity analysis

results can be considered independent of DSMC numerical parameters. For this to be the case our DSMC simulations will have to be sufficiently well-resolved to allow us to capture the relevant physics, but perfect convergence of the individual relaxations is not required. Fortunately, as discussed earlier in the chapter, there are only two DSMC numerical parameters for the relaxations, the time step and the ratio of real to simulated particles. Furthermore, the time step can easily be chosen to be small enough (relative to the average amount of time between interactions for any given particle) that convergence is assured for this parameter. Thus, we are left with only the ratio of real to simulated particles, and we included this parameter in the sensitivity analysis. In fact, we allowed it to vary over a range of an order of magnitude, and even over this wide range we find that it has a negligible effect on the QoI relative to the majority of the other parameters. Based on r^2 the ratio of real to simulated particles is the 24th most sensitive parameter. It has an overall sensitivity which is 496 times smaller than that of the most sensitive parameter, and 56 times smaller than that of the sixth most sensitive parameter (which is the least sensitive parameter we will attempt to calibrate). Based on the mutual information it is the 25th most sensitive parameter, with an overall sensitivity 272 times smaller than that of the most sensitive parameter and 65 times smaller than the sixth most sensitive parameter. With this analysis we have shown both that we have sufficient particles to resolve the physics and that stochastic noise is not a significant factor. If the physics were not properly resolved for the higher values of the ratio of real to simulated particles (when there are fewer simulated particles) we would expect the distribution of the QoI to change as the ratio of real to simulated particles decreases (when more simulated particles are used), and thus we would see some sensitivity to this parameter. If the number of simulated particles was sufficient to simulate the physics but stochastic noise was still a major factor, then we would expect to find a fan-shaped distribution like

the one discussed in Chapter 3 in the context of an example QoI with a noise related parameter (see Figure 3.23). This type of distribution is not captured properly by r^2 , but it was shown in Chapter 3 that it is captured by the mutual information. Since the sensitivity of the QoI to the ratio of real to simulated particles is very low based on either measure, we can conclude that we have sufficient simulated particles to resolve the physics and that the effect of stochastic noise on the QoI is negligible relative to the effects of other parameters.

We now move on to address the second aspect of convergence. As discussed earlier, our sensitivity analysis method relies on a Monte Carlo sampling of the parameter space. We need to demonstrate that we have sufficient Monte Carlo sample points to consider our analysis converged. In order to demonstrate this, we compare the results of our main analysis (with 20,000 Monte Carlo sample points) with results for an analysis with just half of the total points. This second analysis with only 10,000 points is conducted in exactly the same way as the main analysis with all 20,000 points. The results from the 10,000 point analysis for variance-weighted r^2 vs. time during the relaxation are compared with the 20,000 point analysis results in Figure 5.17. The 20,000 point analysis results are shown with solid lines and the 10,000 point analysis results are shown with symbols. The symbols lie nearly on top of the lines over the entire length of the relaxation, even at early times when nonequilibrium is greatest. Figure 5.18 shows the equivalent figure for sensitivities based on the mutual information, and once again the results of the 10,000 point analysis match those of the 20,000 point analysis closely. Overall sensitivities from the both the 20,000 point analysis and the 10,000 point analysis are shown in Fig. 5.19 for both measures of sensitivity. The results from the 10,000 point analysis agree well with those of the 20,000 point analysis, and the top ten parameters based on each measure of sensitivity (and their ordering) are the same for both analyses.

Based on these results, we conclude that we have sufficient Monte Carlo sample points to consider our results converged.

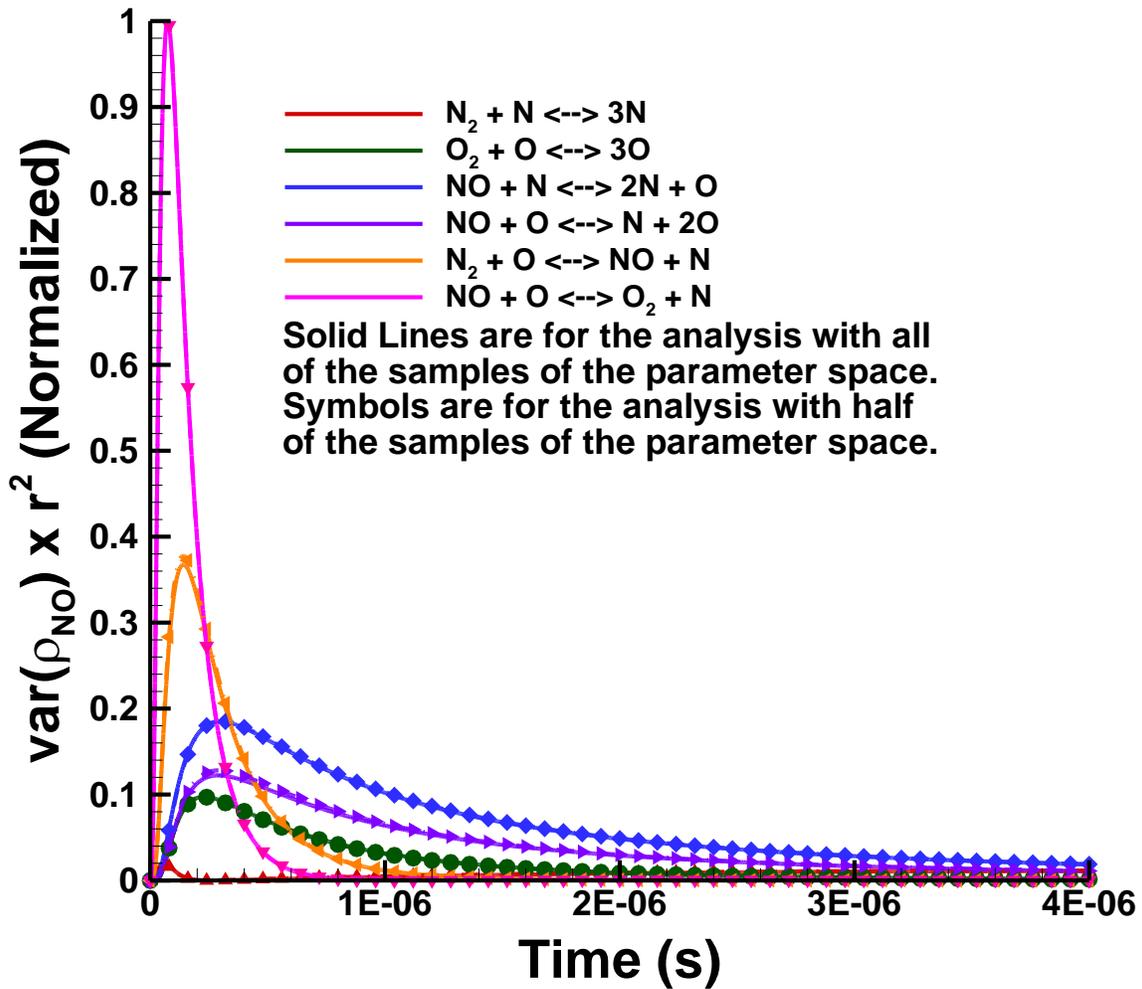


Figure 5.17. Variance-weighted sensitivities (based on r^2) as a function of time for the six most sensitive parameters. Sensitivities are shown for the main analysis with all 20,000 Monte Carlo sample points in parameter space (solid lines) and also for the analysis with only half of the sample points (symbols).

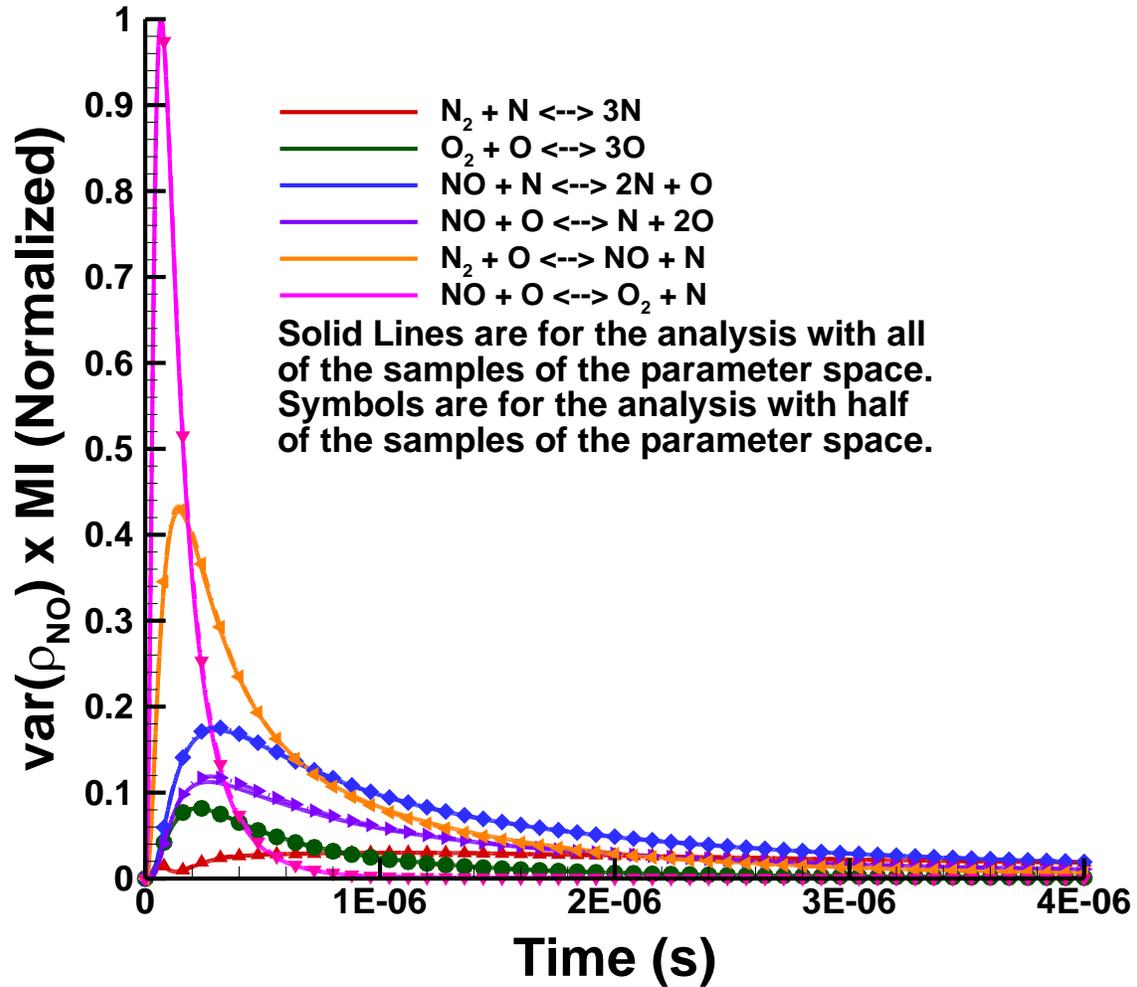


Figure 5.18. Variance-weighted sensitivities (based on the mutual information) as a function of time for the six most sensitive parameters. Sensitivities are shown for the main analysis with all 20,000 Monte Carlo sample points in parameter space (solid lines) and also for the analysis with only half of the sample points (symbols).

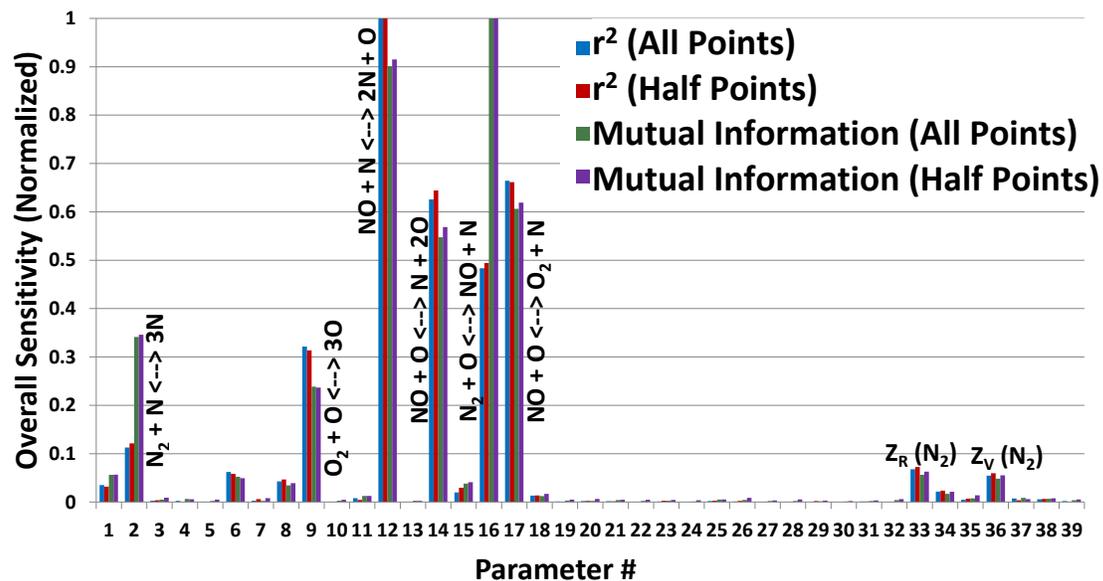


Figure 5.19. Overall sensitivities (based on both measures of sensitivity) for the main analysis with all 20,000 Monte Carlo sample points in parameter space and for the analysis with half as many sample points.

Chapter 6: 1D Shock Sensitivity Analysis

OVERVIEW

After completing our sensitivity analysis for the 0D relaxation and thus demonstrating that our techniques are effective for the analysis of scenarios with complex physics, we are ready to move on to the more realistic scenario of a 1D hypersonic shock. We will first discuss the scenario and show results for a simulation with the nominal values of all parameters. Then, the parameters for this sensitivity analysis will be listed, and we will describe the Monte Carlo sampling of the parameter space. Finally, we will use the techniques discussed in Chapters 3 to conduct our sensitivity analysis and then present the results.

SCENARIO

The scenario we will analyze in this chapter is a 1D shock with a shock speed (relative to the upstream flow) of ~ 8000 m/s. All of the shock simulations in this work are performed with the technique described in Chapter 2, which allows us to obtain a steady shock profile from an unsteady shock moving through the 1D domain, by means of a sample region around the shock which moves at the shock propagation velocity. See Figs. 2.1 and 2.2 for schematics showing the shock sampling technique. A simulation of this scenario was run with the nominal values for all parameters, and some results from this run are shown in Figures 6.1 – 6.3.

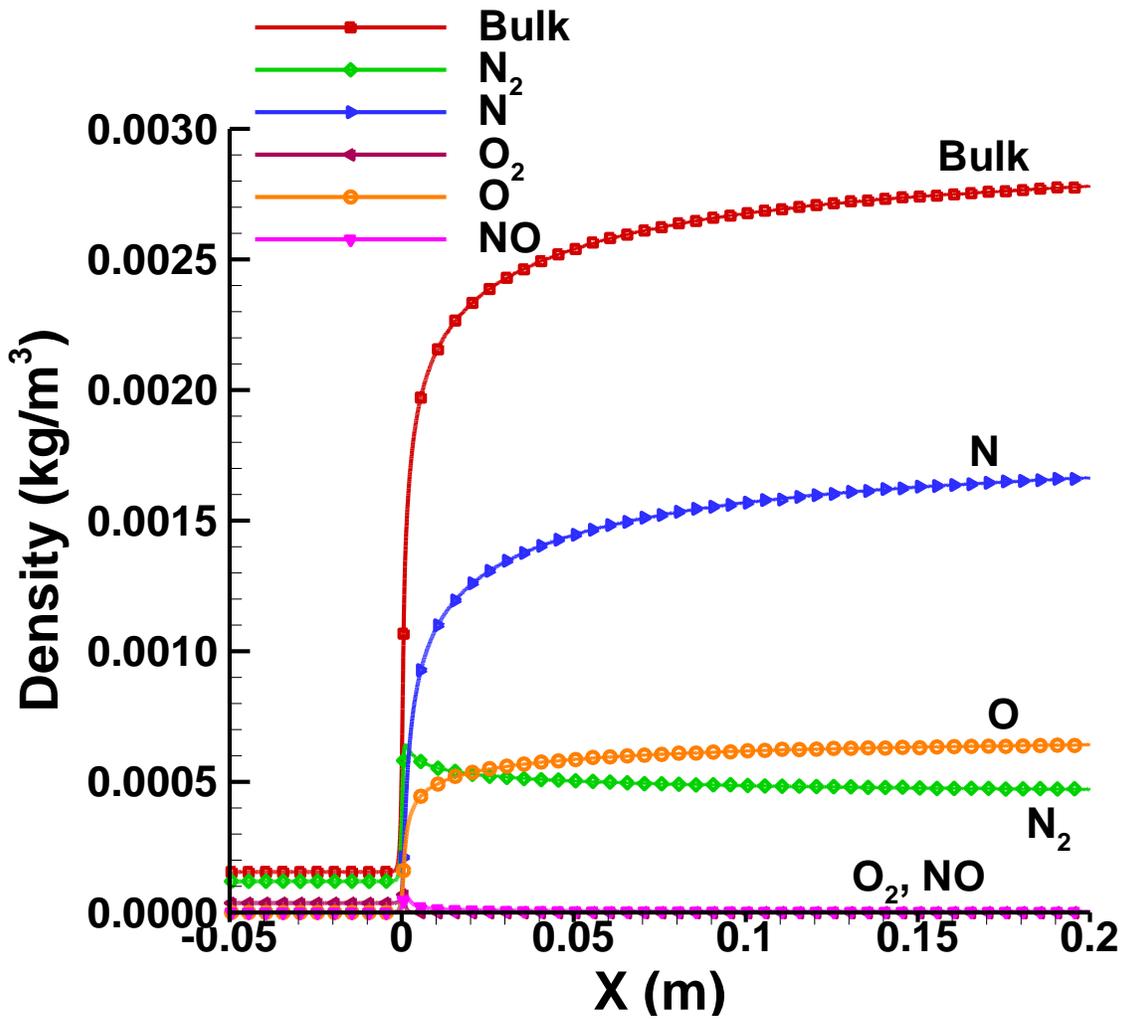


Figure 6.1: Profiles of bulk density and individual species densities across a shock at ~ 8000 m/s. Upstream temperature is 300 K, and number densities for N_2 and O_2 are 2.5×10^{21} $\#/m^3$ and 6.8×10^{20} $\#/m^3$, respectively.

PARAMETERS

The sensitivity analysis results for the 0D relaxation demonstrated that the reaction rate parameters have a much larger effect on the QoI than any of the other physical parameters. With this in mind, and since having fewer parameters makes the analysis more manageable, we have decided that for the 1D shock analysis we will only include the reaction rate parameters.

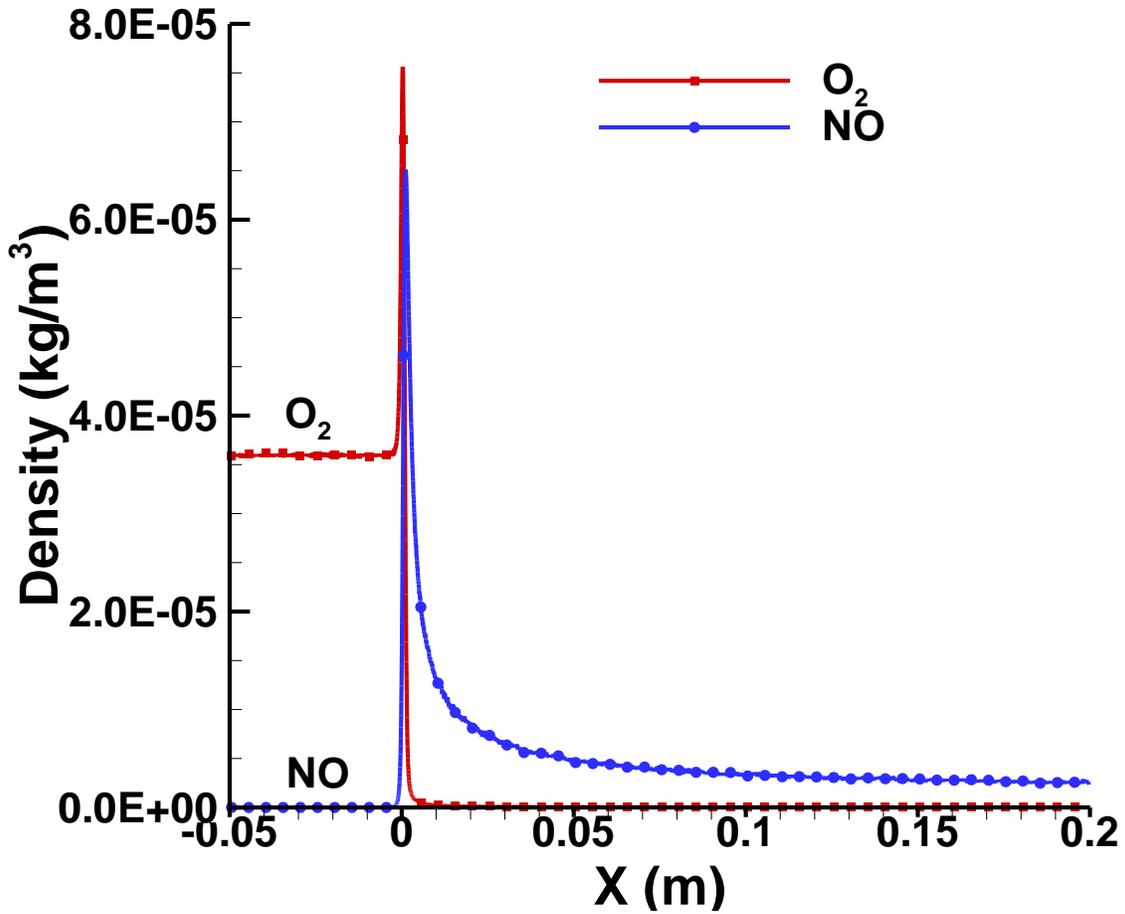


Figure 6.2: Profiles of O_2 and NO densities from Fig. 6.1, with the vertical axis rescaled.

As in the 0D analysis, we will vary one parameter for each of the 17 reactions shown in Table 2.2. As before, the parameters we intend to examine are the pre-exponential constants in the equations for the Arrhenius rate coefficients and we will not vary the forward and backward rates of a given reaction independently, because the ratio of forward and backward rates at a given temperature should be set by the equilibrium constant. If the dissociation rate for the pair is increased, we will correspondingly

increase the recombination rate so that the ratio between the two remains constant. We also keep the ratio of the forward and backward rates for the exchange reactions fixed in this way.

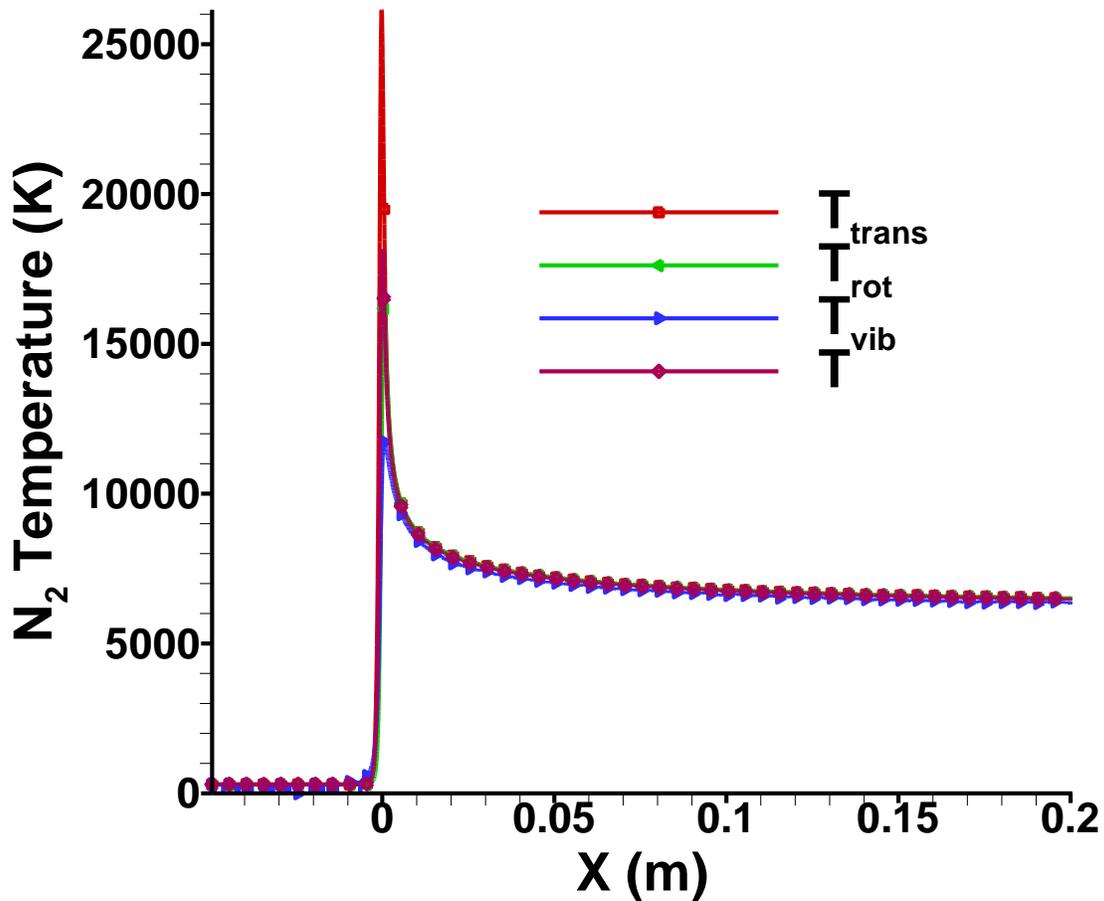


Figure 6.3: Profiles of translational, rotational, vibrational, and overall temperatures for N_2 across the shock from Fig. 6.1.

We will again examine sensitivities to each of these parameters over two orders of magnitude (i.e. we will allow the parameter to vary over a range from 1/10th of the nominal value to ten times the nominal value), and since we are dealing with orders of

magnitude the parameter we will actually examine for each reaction will be $\log_{10}\Lambda$. For a given reaction, $\log_{10}\Lambda$ will vary over the range from $\log_{10}\Lambda_{\text{nom}} - 1$ to $\log_{10}\Lambda_{\text{nom}} + 1$, where Λ_{nom} is the pre-exponential constant for the forward reaction from Table 2.2. The reaction parameters used in the 1D shock sensitivity analysis are listed in Table 6.1 (which is identical to Table 5.1 for the 0D relaxation sensitivity analysis), along with the limits of the uniform prior and the nominal value for each parameter.

Table 6.1: The parameters for the 1D shock sensitivity analysis. These parameters correspond to the reactions which are important for 5-species air. For each reaction, the parameter of interest is $\log_{10}\Lambda$, where Λ is the pre-exponential constant in the Arrhenius rate equation (Eq. 2.2) for the forward reaction. Nominal Arrhenius forward rate parameters are from Gupta *et al.* (1989), and backward rate parameters are based on a matching with the equilibrium constant over a range of temperatures. Throughout the sensitivity analysis, the ratio of forward to backward rate for a given reaction is kept constant, since these ratios should be fixed by the equilibrium constant.

#	Parameter	Prior Distribution Limits		Nominal Value
		Minimum	Maximum	
1	$\log_{10}\Lambda (2\text{N}_2 \rightleftharpoons 2\text{N} + \text{N}_2)$	-13.10	-11.10	-12.10
2	$\log_{10}\Lambda (\text{N}_2 + \text{N} \rightleftharpoons 3\text{N})$	-8.16	-6.16	-7.16
3	$\log_{10}\Lambda (\text{N}_2 + \text{O}_2 \rightleftharpoons 2\text{N} + \text{O}_2)$	-13.50	-11.50	-12.50
4	$\log_{10}\Lambda (\text{N}_2 + \text{O} \rightleftharpoons 2\text{N} + \text{O})$	-13.50	-11.50	-12.50
5	$\log_{10}\Lambda (\text{N}_2 + \text{NO} \rightleftharpoons 2\text{N} + \text{NO})$	-13.50	-11.50	-12.50
6	$\log_{10}\Lambda (\text{O}_2 + \text{N}_2 \rightleftharpoons 2\text{O} + \text{N}_2)$	-11.92	-9.92	-10.92
7	$\log_{10}\Lambda (\text{O}_2 + \text{N} \rightleftharpoons 2\text{O} + \text{N})$	-12.22	-10.22	-11.22
8	$\log_{10}\Lambda (2\text{O}_2 \rightleftharpoons 2\text{O} + \text{O}_2)$	-11.27	-9.27	-10.27
9	$\log_{10}\Lambda (\text{O}_2 + \text{O} \rightleftharpoons 3\text{O})$	-10.82	-8.82	-9.82
10	$\log_{10}\Lambda (\text{O}_2 + \text{NO} \rightleftharpoons 2\text{O} + \text{NO})$	-12.22	-10.22	-11.22
11	$\log_{10}\Lambda (\text{NO} + \text{N}_2 \rightleftharpoons \text{N} + \text{O} + \text{N}_2)$	-10.18	-8.18	-9.18
12	$\log_{10}\Lambda (\text{NO} + \text{N} \rightleftharpoons 2\text{N} + \text{O})$	-8.88	-6.88	-7.88
13	$\log_{10}\Lambda (\text{NO} + \text{O}_2 \rightleftharpoons \text{N} + \text{O} + \text{O}_2)$	-10.18	-8.18	-9.18
14	$\log_{10}\Lambda (\text{NO} + \text{O} \rightleftharpoons \text{N} + 2\text{O})$	-8.88	-6.88	-7.88
15	$\log_{10}\Lambda (2\text{NO} \rightleftharpoons \text{N} + \text{O} + \text{NO})$	-8.88	-6.88	-7.88
16	$\log_{10}\Lambda (\text{N}_2 + \text{O} \rightleftharpoons \text{NO} + \text{N})$	-16.95	-14.95	-15.95
17	$\log_{10}\Lambda (\text{NO} + \text{O} \rightleftharpoons \text{O}_2 + \text{N})$	-18.80	-16.80	-17.80

QUANTITY OF INTEREST

We use ρ_{NO} as the quantity of interest for this scenario, just as we did for the 0D relaxation. As before, this QoI will actually be a vector, except this time the vector represents ρ_{NO} at various discrete points in space rather than in time. Once again, each discrete point will be viewed as a separate, scalar QoI when calculating r^2 and the mutual information. Once the individual analyses for all of the scalar QoIs have been completed, the sensitivity of ρ_{NO} to the various parameters can then be examined as a function of streamwise location in the shock, and we can also appropriately integrate the sensitivities over all of the individual scalar QoIs in order to obtain an overall value of sensitivity for each parameter. A schematic showing the definition of the QoI for this scenario is shown in Figure 6.4.

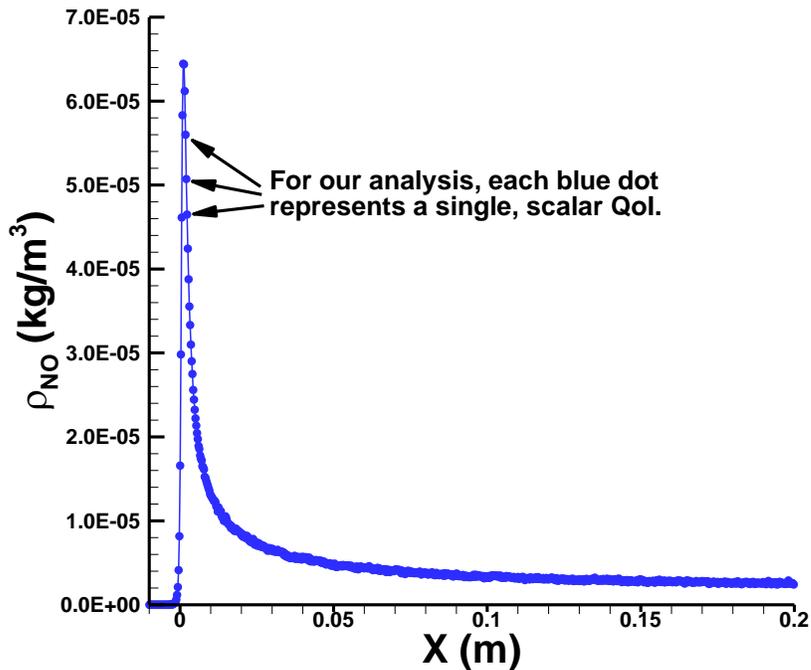


Figure 6.4. Schematic showing the way in which the ρ_{NO} vector QoI is broken into individual scalar QoIs during the sensitivity analysis for the 1D shock.

SAMPLING THE PARAMETER SPACE

When performing the set of simulations which will form our sample of the parameter space, we must work within computational limits. Simulations for the 1D shock scenario take an order of magnitude longer than those for the 0D relaxation scenario. There are two reasons why the 1D shocks take so much longer to simulate. First, the shock must be allowed to propagate for a substantial period of time before the shock profile can be considered self-similar and sampling can begin. This is not the case for the 0D relaxation, where the profiles of density vs. time have their origin at the moment the relaxation begins and therefore no computational effort is wasted on a part of the simulation that will not actually be directly included in the final results. The second reason has to do with the fact that time and space are not equivalent dimensions. We are interested primarily in non-equilibrium chemistry, which for the 0D case occurs mostly at early times near the beginning of the relaxation. We can therefore cut off the simulations well before the relaxation fully reaches equilibrium and save a lot of computational time. This is especially true when we are concerned primarily with variance-weighted sensitivities, since those become negligible (relative to their peak values) well before the relaxation has reached complete equilibrium. We can cut off the later parts of the simulation in this way because the portions of the relaxation which would have come later obviously do not exert any influence backward in time which could have an effect on the profiles at earlier times. The situation is different for a 1D shock. We are once again interested in the non-equilibrium chemistry, which is mostly confined to the region near the shock front, but in this case we cannot cut off the rest of the profile. Even if we had a method for simulating a purely steady shock (rather than our method of sampling from an unsteady shock), we would still have to spend a great deal of computational effort simulating parts of the 1D shock profile that we do not care much about. If we cut

off the profile and place our boundary condition unrealistically close to the shock front the effect will be felt upstream of the boundary and our profile will be altered everywhere downstream of the shock front (everywhere that the flow is subsonic).

With these limitations in mind, we have had to use a smaller number of Monte Carlo samples of the parameter space. The dataset for the 1D shock sensitivity analysis consists of 5600 runs of the DSMC code, each at a different point in the 17-dimensional parameter space we are exploring. Just like for the 0D scenario, before each run the sensitivity analysis driver code performs a random number draw to determine $\log_{10}\Lambda$ for each of the 17 reactions, and then the pre-exponential constants are sent to the DSMC code for each forward reaction. The pre-exponential constants for the backward rates are calculated such that the ratio $\Lambda_{\text{forward}}/\Lambda_{\text{backward}}$ remains constant for a given reaction, regardless of location in parameter space. This set of 5600 runs of the DSMC code took over 112 hours on 4096 processors (each individual DSMC simulation was performed on 128 processors, and the runs were done simultaneously in multiple sets), for a total of ~450,000 CPU hours. After each run data for the QoI were output at a set of discrete points with fixed locations relative to the shock front, and as mentioned previously, each of these points will be treated as an individual, scalar QoI.

Just as we did for the 0D relaxation analysis, we will show later in this chapter that 5600 sample points in parameter space are sufficient to accurately characterize the sensitivities we wish to examine.

SENSITIVITIES VS. X

Once we have performed the set of simulations which constitute our sampling of the parameter space we can then calculate sensitivities (based on r^2 and the mutual

information) of each scalar QoI to each of the parameters, just as we did for the 0D relaxation scenario. As before, it is useful to show the sensitivities to various parameters for all of the scalar QoIs on one plot. This allows us to see how the sensitivities vary with location in the shock. Profiles for sensitivity vs. x are shown in Figure 6.5 for the top five most sensitive parameters.

The degree of chemical nonequilibrium is greatest in the early part of the shock, and in this region ρ_{NO} is most strongly affected by the NO dissociation reactions $\text{NO} + \text{N} \rightleftharpoons 2\text{N} + \text{O}$ and $\text{NO} + \text{O} \rightleftharpoons \text{N} + 2\text{O}$ and by the exchange reactions. Sensitivity to the reaction $\text{N}_2 + \text{N} \rightleftharpoons 3\text{N}$ peaks slightly later, and this reaction is the most sensitive over the remainder of the streamwise domain. As the gas gets closer to equilibrium at large values of x , sensitivity to the NO dissociation reaction $\text{NO} + \text{N} \rightleftharpoons 2\text{N} + \text{O}$ again approaches the sensitivity to the reaction $\text{N}_2 + \text{N} \rightleftharpoons 3\text{N}$.

When comparing sensitivities measured with r^2 to those measured with mutual information, it is clear that the shapes of the curves are similar for four of the five reactions shown, but there is an important difference between the two measures at some streamwise locations for the exchange reaction $\text{N}_2 + \text{O} \rightleftharpoons \text{NO} + \text{N}$. In the region around $x = 0.066$ m, the sensitivity based on the mutual information drops to a low but non-negligible level, while the sensitivity based on r^2 actually drops to zero before increasing again. This occurs because at the point where r^2 crosses zero the correlation switches from positive to negative (i.e. increasing values of the parameter lead to decreasing values of the QoI). Figure 6.6 examines the reason for this in more detail. Fig. 6.6 is a scatterplot similar to those in Figs. 5.7 – 5.12, with the scalar QoI now being ρ_{NO} at the location $x = 0.066$ m and the parameter being $\log_{10}\Lambda$ for the reaction $\text{N}_2 + \text{O} \rightleftharpoons \text{NO} + \text{N}$. This parameter influences the value of the QoI at this location, but the relationship is non-linear and thus is not captured properly by r^2 . The mutual information does capture non-

linear correlations, and for this reason the value of the mutual information at this point is non-negligible (although it is still low relative to the mutual information for some of the other parameters). This is another example of a situation where the mutual information is superior to r^2 as a measure of sensitivity.

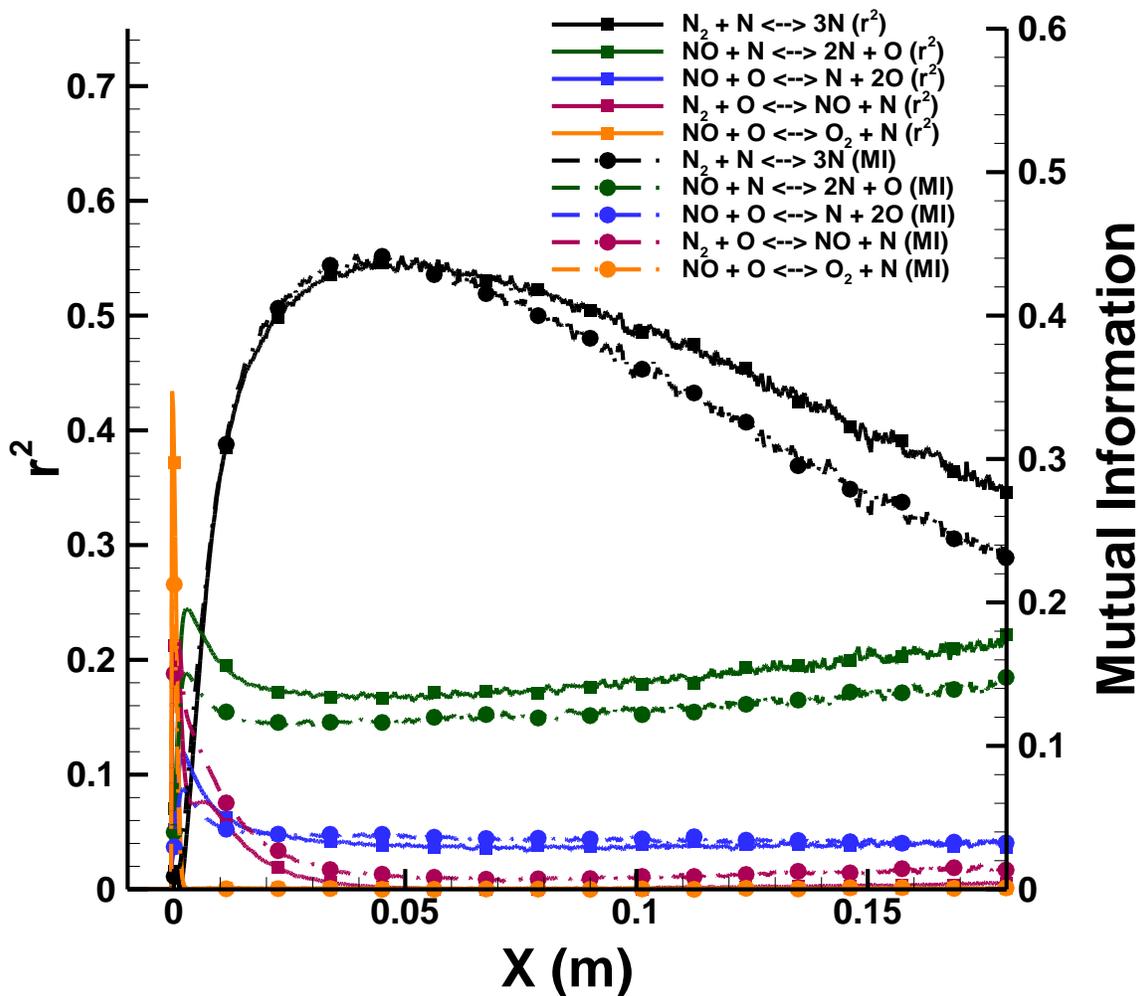


Figure 6.5: Sensitivities as a function of streamwise location for five of the most sensitive parameters. Both r^2 and the mutual information are shown on the plot for each parameter.

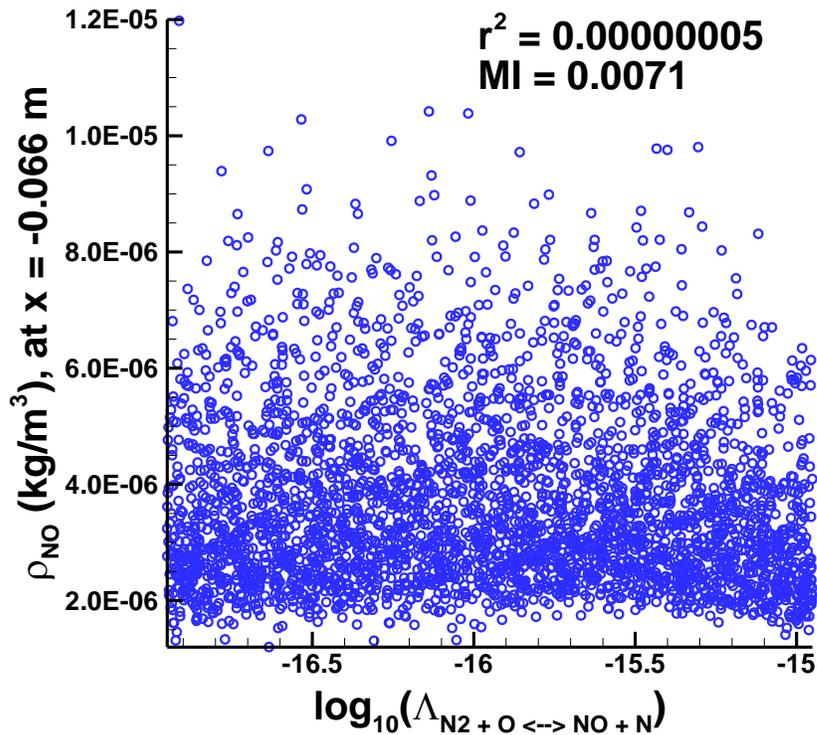


Figure 6.6: Scatterplot showing ρ_{NO} (at $x = 0.066 \text{ m}$) vs. $\log_{10}\Lambda$ for the reaction $\text{N}_2 + \text{O} \rightleftharpoons \text{NO} + \text{N}$. The relationship between $\log_{10}\Lambda$ and ρ_{NO} at this location in the shock is non-linear.

VARIANCE-WEIGHTED SENSITIVITIES

As discussed in the previous chapter, we need to make use of a weighting function before integrating the sensitivities over the scalar QoIs to obtain an overall sensitivity for each parameter. This variance weighting is even more necessary for the 1D shock sensitivity analysis than it was for the 0D relaxation. Figure 6.7 shows two scatterplots, each for a different parameter and a different scalar QoI. The value of r^2 is higher for the scatterplot in the right image of the figure, but the actual effect of the parameter on the QoI is clearly much greater for the scatterplot in the left image. This is

a much more dramatic difference than that shown in Fig. 5.13 (the figure which accompanied the discussion of variance-weighting in Chapter 5), and the reason for this can be seen in Figure 6.8, which is a plot of the variance of ρ_{NO} vs. x . Comparing this figure to the curve for the variance of ρ_{NO} vs. time in Fig. 5.14, we see that the variance of ρ_{NO} drops off much more sharply for the 1D shock scenario than it did for the 0D relaxation. Note that we are not making a direct comparison between profiles of ρ_{NO} as a function of time in a relaxation and ρ_{NO} as a function of x in a shock; we mention that the variance of ρ_{NO} drops off more dramatically in the profiles from the shock scenario only in order to point out the even greater need for variance weighting in this scenario.

We once again make use of the variance of the QoI as the weighting function, and for the 1D shock scenario the equation for the overall sensitivity (based on the r^2 measure) to a given parameter is

$$\text{Overall Sensitivity} = \int_x \text{var}_{QoI}(x)r^2(x)dx \quad (6.1)$$

where $\text{var}_{QoI}(x)$ is the variance of the QoI at a given streamwise location, and the integration is over the entire streamwise domain. When mutual information is the measure of sensitivity, $r^2(x)$ is replaced by $MI(x)$ in Eq. 6.1. The variance-weighted sensitivities are shown as functions of x in Figure 6.9 for the five most sensitive parameters. Figure 6.10 shows the same curves, this time zoomed in on the non-equilibrium region near the shock front where the variance-weighted sensitivities are highest. In both of these figures, the variance-weighted sensitivity curves have been normalized by the peak sensitivity (the highest sensitivity at any point in the shock for any parameter). This is done separately for each measure of sensitivity, so the highest r^2 based sensitivity at any point in the shock is set equal to one, and similarly for the mutual information.

Eq. 6.1 is used to obtain overall sensitivities, based on r^2 and the mutual information, for each parameter. The overall sensitivity to each parameter (based on each measure) is normalized by the largest value of overall sensitivity (based on that measure), so that the most sensitive parameter based on each measure has an overall sensitivity of one.

These overall sensitivities are shown in Figure 6.11 for all of the parameters. The reaction $\text{NO} + \text{N} \rightleftharpoons 2\text{N} + \text{O}$ has the greatest influence on ρ_{NO} , regardless of which measure of sensitivity is used. However, other parameters also have a strong effect on ρ_{NO} . We choose six because there is a clear break in the data and the seventh highest sensitivity is far lower than that for any of the top six. The set of six most sensitive reactions is the same for either measure, and these six are all labeled in Fig. 6.11. Furthermore, these are the same six parameters identified in the 0D relaxation analysis, which provides some justification for our earlier argument that the 0D relaxation could be considered as a substitute scenario for the 1D shock for the purposes of sensitivity analysis and parameter calibration. We can conclude based on these results that these six reactions would be appropriate for calibration with ρ_{NO} as QoI, regardless of whether the 0D relaxation scenario or the 1D shock scenario is used.

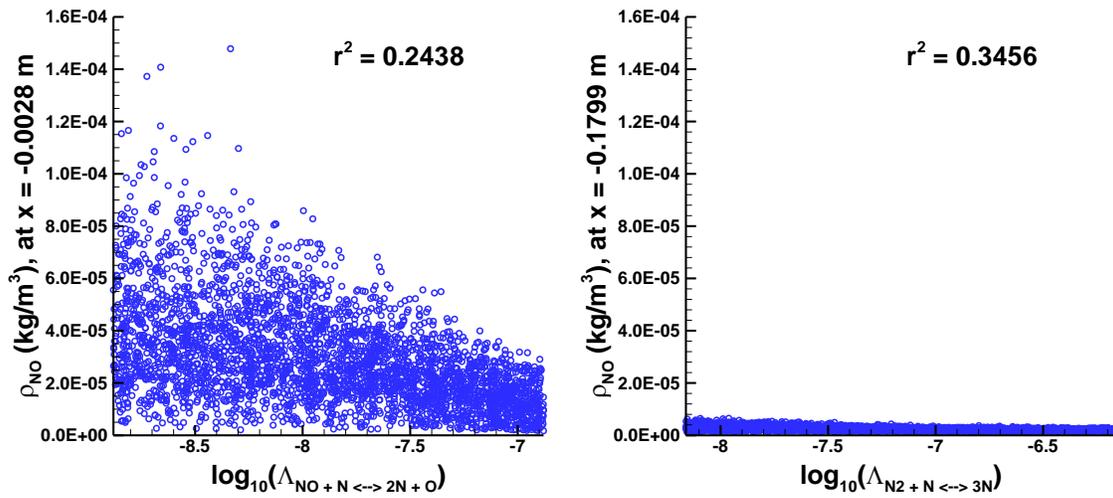


Figure 6.7: Scatterplots showing ρ_{NO} at $x = 0.0028$ m vs. $\log_{10}\Lambda$ for the reaction $NO + N \rightleftharpoons 2N + O$ (left image) and ρ_{NO} at $x = 0.1799$ m vs. $\log_{10}\Lambda$ for the reaction $N_2 + N \rightleftharpoons 3N$.

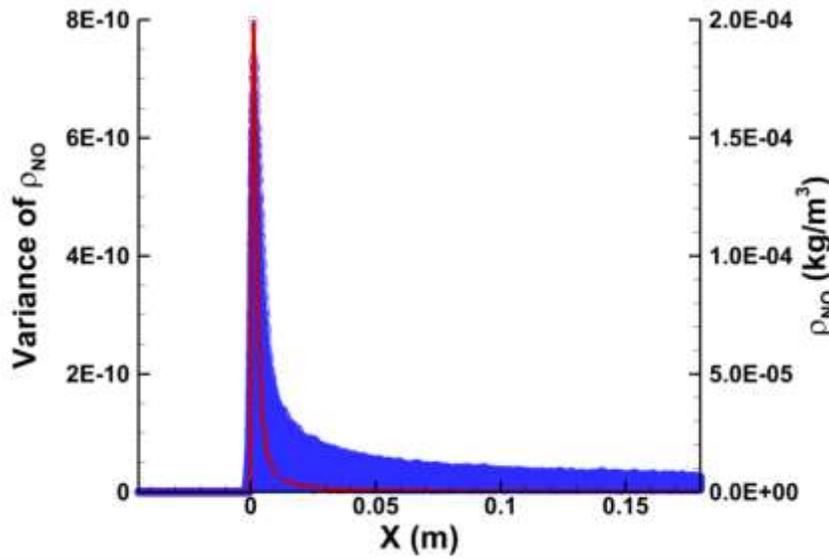


Figure 6.8: Scatterplot showing the vector QoI ρ_{NO} for all of the shocks simulated in the sensitivity analysis, along with the variance of ρ_{NO} as a function of x .

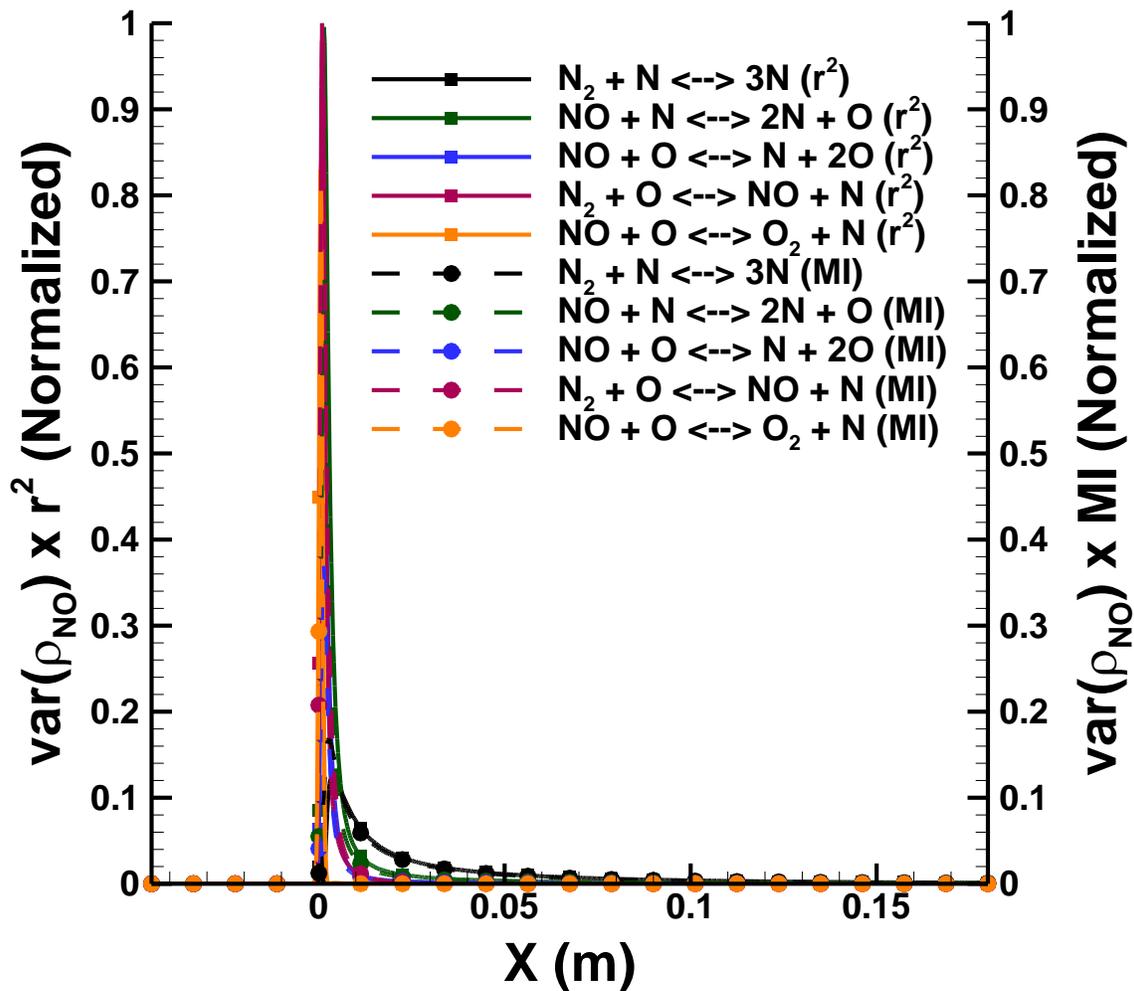


Figure 6.9: Variance-weighted sensitivities as a function of streamwise location for the top five most sensitive parameters. Both r^2 and the mutual information are shown on the plot for each reaction.

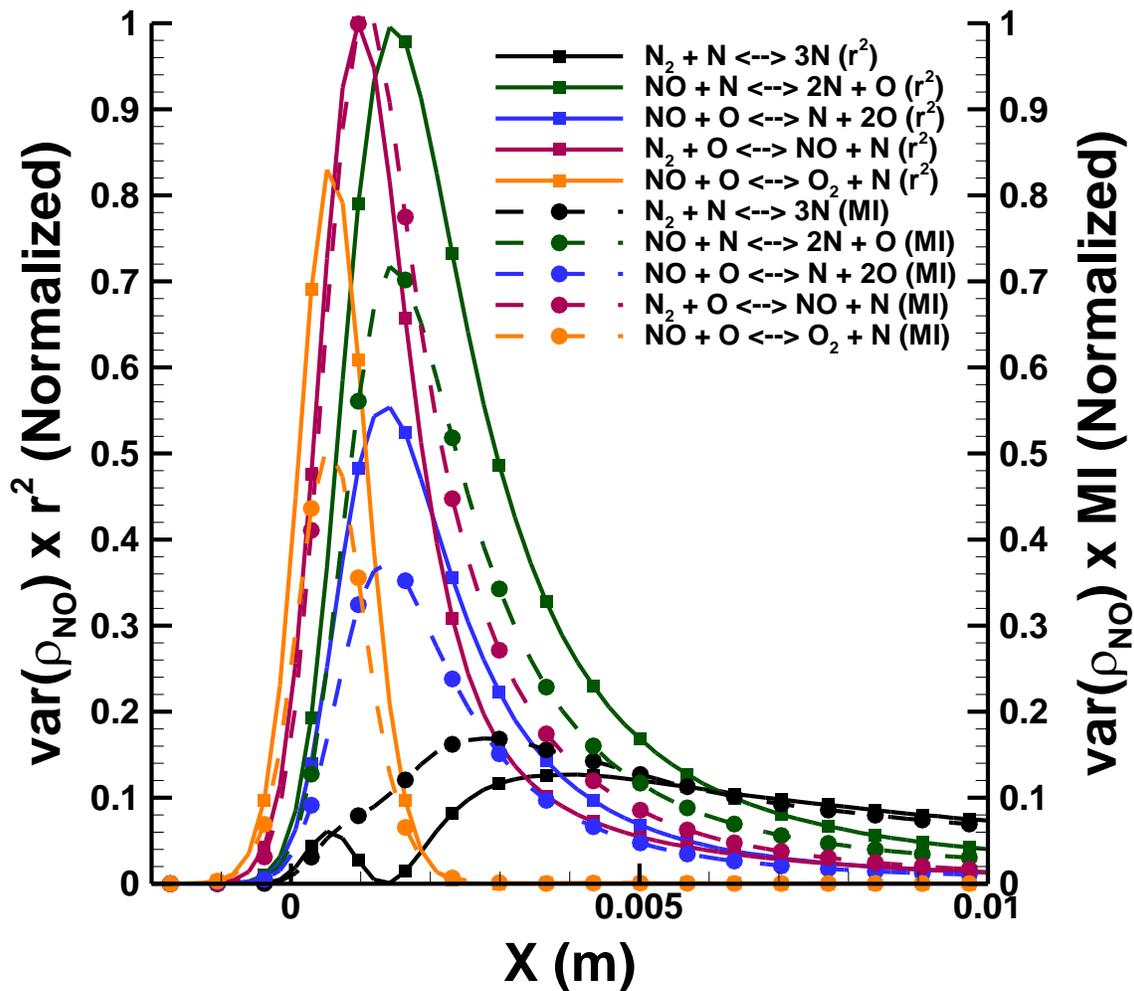


Figure 6.10: Zoomed-in view of the plot in Fig. 6.9, showing the non-equilibrium region where the variance of ρ_{NO} and the variance-weighted sensitivities are highest.

CONVERGENCE

In order to be confident of our results, we must once again show that they are converged with respect to the DSMC numerical parameters (time step, cell size, and the ratio of real to simulated particles), and we must also show that we have sufficient points in our Monte Carlo sampling of the parameter space.

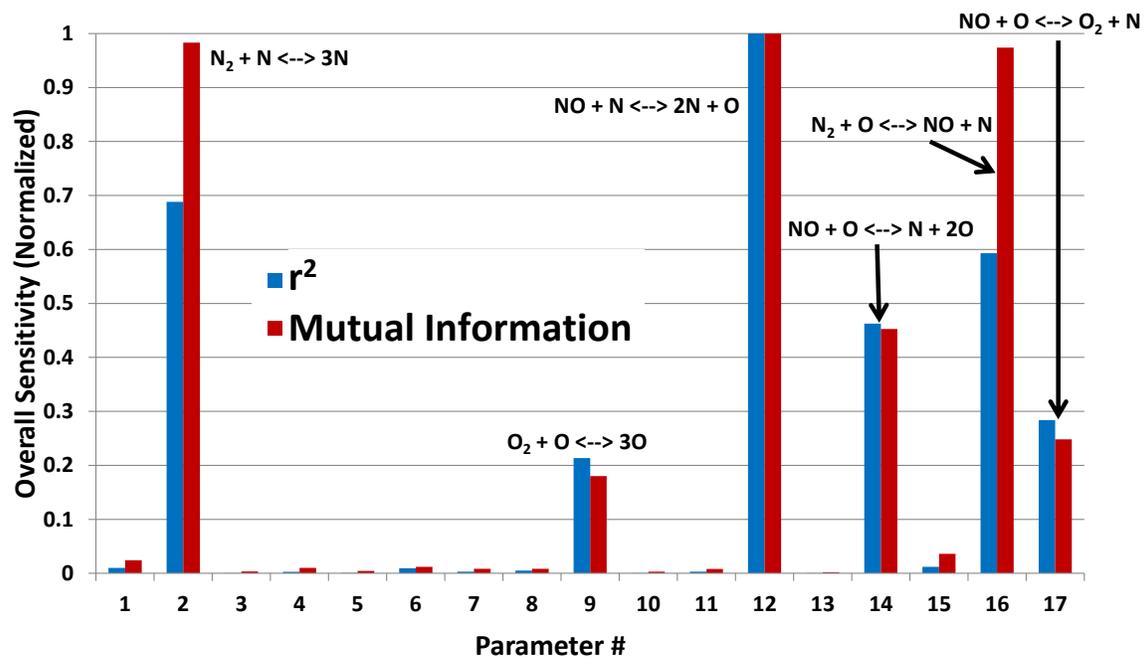


Figure 6.11: Overall, variance-weighted sensitivities for each parameter with each measure of sensitivity.

We first consider the resolution of the DSMC simulations. As before, we are not trying to show that the results for a given simulation are identical at higher or lower resolution, we are only trying to demonstrate that we are sufficiently converged that our sensitivity analysis results can be considered independent of DSMC numerical parameters. Our DSMC simulations must be sufficiently well-resolved to capture the relevant physics, but runs which are sufficiently well resolved to demonstrate perfect convergence of the individual shock profiles are not required (or feasible in this work). There are several DSMC numerical parameters which are relevant for this scenario. The ratio of real to simulated particles is still important, as it was for the 0D relaxation, and for this scenario the spatial extent of the DSMC cells is important as well. Unlike in the 0D relaxation scenario, we cannot make the time step excessively small. Doing so would lead to a great deal of wasted work in the move and index subroutines (the particles did

not have to be moved or indexed to cells in the 0D scenario), and therefore we must consider the time step when examining convergence. Since we are now dealing with three numerical parameters and our runs take substantially longer, it is not feasible to include the numerical parameters in the sensitivity analysis as we did for the 0D relaxation, and so we will show convergence in another way.

In order to demonstrate that our sensitivity analysis results are independent of DSMC numerical parameters, we have performed a second sensitivity analysis, just like the one described previously, with half of the resolution used for the runs shown earlier in this chapter. The physical dimensions and initial conditions for the shock were the same for this second analysis, and the parameter space and Monte Carlo sampling process were the same. However, the number of cells was halved (and thus the cell-size was doubled), the time step was doubled (and thus the number of steps was halved), and the ratio of real to simulated particles was doubled (so the total number of simulated particles was halved). Since the cell-size was doubled at the same time that the number of particles was halved, the average number of particles per cell remained the same (~100 particles per cell downstream of the shock). The noise in the simulation results was nonetheless increased, since with the larger time step the number of samples of the moving shock region over the course of the simulation was halved. Thus, in every aspect (temporal resolution, spatial resolution, and stochastic noise) these simulations were resolved significantly less well than the simulations used for the main analysis.

Results for the variance-weighted mutual information as a function of streamwise location for the two analyses are shown in Figure 6.12. This figure shows only the part of the streamwise domain nearest the shock front, because the curves are nearly identical in the rest of the domain. The curves are very similar in shape in the region shown in Fig. 6.12, despite the fact that, as shown previously in Fig. 6.8, this is the region where

the non-equilibrium is most pronounced. It is this region which dominates the integrated sensitivities, which are shown for both the normal and low resolution analyses in Figure 6.14. While the integrated sensitivities are not identical for the two resolutions, they are similar. The top nine most sensitive parameters are the same for either resolution, and in fact these nine are in the same order from highest to lowest integrated sensitivity.

The primary goal of our sensitivity analysis is to identify the most-sensitive parameters, and to rank them from most-to-least sensitive. We have shown that the most-sensitive parameters, and their order, are unchanged when the resolution of the DSMC simulations used in the analysis is halved, and based on this we conclude that we are sufficiently converged with respect to DSMC numerical parameters.

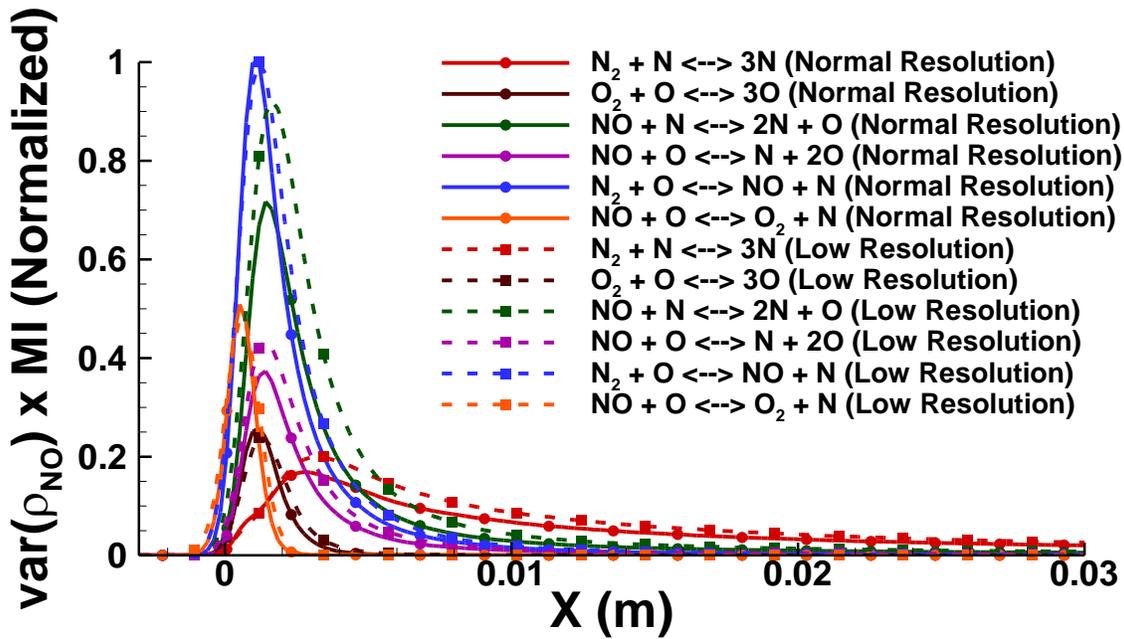


Figure 6.12: Variance-weighted sensitivities (based on the mutual information) as a function of streamwise location of the six most sensitive parameters. Sensitivities are shown for the normal resolution and also for the lower resolution analysis. While the curves do not agree exactly, the shapes are very similar.

We also need to demonstrate that we have sufficient Monte Carlo sample points in parameter space to consider our analysis resolved. To check this we follow the same procedure as we did for the 0D relaxation; we compare the results of our main analysis (which for this scenario used 5600 sample points) with results from just half of our total points. The results from the 2800 point analysis for variance-weighted mutual information vs. streamwise location are compared with the full analysis results in Figure 6.13. As with Fig. 6.12, this figure shows only the part of the streamwise domain nearest the shock front, and the shapes of the curves are nearly identical. Integrated sensitivities from the 2800 point analysis are shown in Fig. 6.14. The results from the 2800 point analysis closely mirror those of the full analysis, and the top eleven parameters (and their ordering) are the same for both analyses. Based on these results, we conclude that we have sufficient Monte Carlo sample points to consider our results converged.

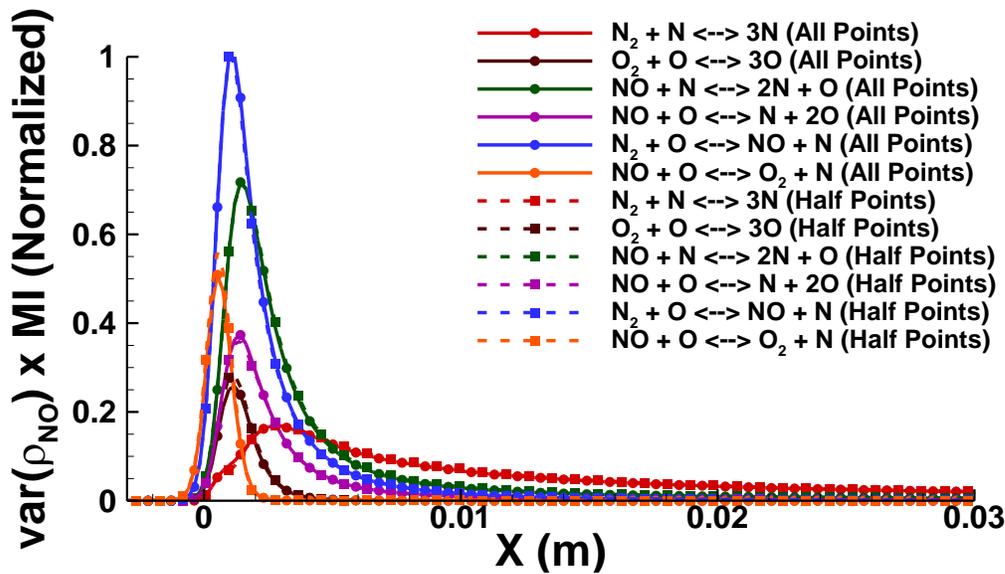


Figure 6.13: Variance-weighted sensitivities (based on the mutual information) as a function of streamwise location of the six most sensitive parameters. Sensitivities are shown for the main analysis (with all 5600 Monte Carlo sample points) and also for the analysis with only half of the sample points.

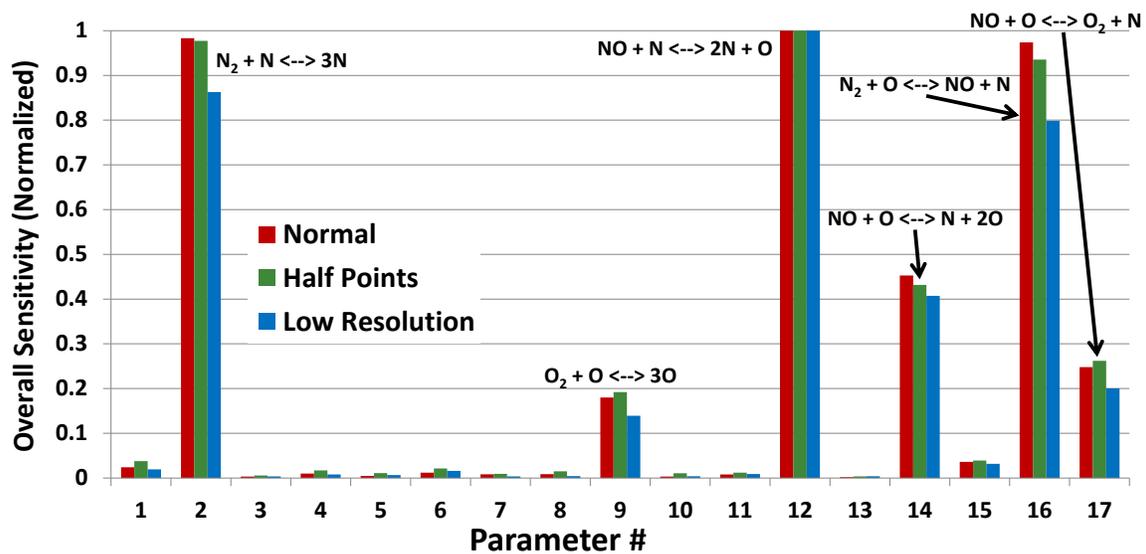


Figure 6.14: Overall sensitivities (based on the mutual information) for the main analysis presented earlier in the chapter (regular DSMC resolution with all 5600 Monte Carlo sample points), the analysis with lowered DSMC resolution, and the analysis with half as many Monte Carlo sample points.

Chapter 7: Synthetic Data Calibrations

OVERVIEW

All of the results presented so far have concentrated on sensitivity analysis, and it is in that area that the majority of our code and algorithm development effort was focused (along with the writing and optimization of the DSMC code for our current needs). Sensitivity analysis can be very useful in its own right, and there are many applications for our methodology even in situations where there is no intention to ever calibrate parameters by solving the statistical inverse problem. However, the long term goal of this project is the calibration of DSMC parameters relevant to the modeling of a hypersonic shock, and our sensitivity analyses were performed primarily with this long-term goal in mind. With sensitivity analyses complete for both the 0D relaxation and the 1D shock scenarios, we would now like to demonstrate that we can calibrate the parameters we identified as suitable during the sensitivity analyses. Due to the fact that we are not yet able to fully simulate the shock tube experiments for which there are available experimental data in a form which is useful to us, we will instead use synthetic data to demonstrate the calibration process for our DSMC parameters.

We first identify the parameters to be calibrated, based on our sensitivity analysis results. Next we describe the scenarios and the calibration data which will be used during the solution of the inverse problem, and then we specify a likelihood equation. Finally we perform the synthetic data calibration and present the results.

PARAMETERS

Both the 0D relaxation and the 1D shock sensitivity analyses identified the same top six most sensitive parameters. These parameters are the pre-exponential constants in

the Arrhenius-type rate equations for six of the reactions which are included in our five-species air chemistry model. We will attempt to calibrate these six parameters based on synthetic data. We will use the same priors which were used in the sensitivity analysis, which allow the parameters to each vary over a two order of magnitude range. Once again, the lower limit for each pre-exponential constant will be $0.1\Lambda_{\text{nom}}$ and the upper limit will be $10\Lambda_{\text{nom}}$. As before, due to this large range, we will actually perform the calibration for $\log_{10}\Lambda$ for each of the six reactions, and $\log_{10}\Lambda$ will have a uniform prior over the range from $\log_{10}\Lambda_{\text{nom}} - 1$ to $\log_{10}\Lambda_{\text{nom}} + 1$. Table 7.1 lists the six reaction parameters to be calibrated, with the parameter numbers kept consistent with those given in Tables 5.1 and 6.1 to avoid confusion when parameter numbers are referred to in later text and figures. The limits of the uniform prior and the nominal value for each parameter are also included in the table. Note that as was the case during the sensitivity analysis, the nominal values do not play a direct role in the solution of the statistical inverse problem and are listed only for informative purposes. They are used initially to specify the centers of the priors and they are also used to generate the synthetic data (as will be discussed momentarily), but once those parts of the process are complete they play no further role in the calibration process.

Table 7.1: The six parameters chosen for calibration. The parameter numbers are kept consistent with those in Tables 5.1 and 6.1.

#	Parameter	Prior Distribution Limits		Nominal Value
		Minimum	Maximum	
2	$\log_{10}\Lambda (\text{N}_2 + \text{N} \rightleftharpoons 3\text{N})$	-8.16	-6.16	-7.16
9	$\log_{10}\Lambda (\text{O}_2 + \text{O} \rightleftharpoons 3\text{O})$	-10.82	-8.82	-9.82
12	$\log_{10}\Lambda (\text{NO} + \text{N} \rightleftharpoons 2\text{N} + \text{O})$	-8.88	-6.88	-7.88
14	$\log_{10}\Lambda (\text{NO} + \text{O} \rightleftharpoons \text{N} + 2\text{O})$	-8.88	-6.88	-7.88
16	$\log_{10}\Lambda (\text{N}_2 + \text{O} \rightleftharpoons \text{NO} + \text{N})$	-16.95	-14.95	-15.95
17	$\log_{10}\Lambda (\text{NO} + \text{O} \rightleftharpoons \text{O}_2 + \text{N})$	-18.80	-16.80	-17.80

SCENARIO SELECTION

Earlier in this work we split our efforts between two types of scenario: a 0D relaxation and a 1D shock. We did this primarily because the 0D relaxation is substantially less expensive computationally. We were able to explore several aspects of our sensitivity analysis methodology more thoroughly with the 0D relaxation than we were with the 1D shock. We also wanted to test the hypothesis that with the right initial conditions, the 0D relaxation could be used as a substitute model for the 1D shock for some purposes. The fact that we identified the same top six parameters for both types of scenario provided some justification for this argument. We do not claim that the 0D relaxation is a substitute model for the 1D shock in the sense that the density (or temperature, pressure, etc.) vs. time profiles for a relaxation are equivalent to the density vs. x profiles for a shock, we have only claimed that the results of DSMC simulations of the two scenarios are sensitive to the reaction rate parameters in similar ways.

The eventual goal of this project is to perform calibrations with experimental data which will most likely come from a shock tube, and therefore the 1D shock scenario is the more realistic of the two scenarios. However, the greatly increased computational expense of the 1D shock becomes much more of an issue when performing calibrations than it was for the sensitivity analysis. This is partly because the calibration process requires more total simulations than the sensitivity analysis did, but the primary issue is due to the nature of the MCMC algorithm. During the sensitivity analysis every sample of the parameter space was completely independent of every other sample, and thus the work could be spread across an arbitrary number of processors in order to maximize parallel efficiency of the DSMC code (a total of seven sixteen-hour runs on 4096 processors were required for the 1D shock sensitivity analysis). When using MCMC to solve the inverse problem, however, simulations must be run in sequence. A new

candidate position cannot be chosen until the simulation for the old candidate position is completed and a value for the likelihood is provided to the MCMC algorithm. This is an inherent aspect of the MCMC process, and there is no way to avoid it. This means that the simulation time required for a given chain will be the number of candidate positions tested in the chain multiplied by the average time required for each simulation. Unfortunately, while our method of simulating 1D shocks in DSMC is reasonably fast, it would still take a long time to run chains of sufficient length. The code scales very well as the size of the problem increases, but a certain number of particles per cell are required to accurately model the physics and therefore even though we use adaptive load balancing to even out the work, for a given problem size the DSMC simulations still do not scale efficiently to arbitrary numbers of processors. This means that if we are already utilizing large numbers of processors (as we did for the sensitivity analysis) we can run multiple chains simultaneously, but we cannot complete a given chain any faster by adding more processors. Running multiple chains is useful and can help greatly in exploring the parameter space more fully, but each chain still needs to reach some reasonable length in order to properly sample the posterior distributions.

The required computation time investment makes calibrations for the 1D shock scenario difficult if not impossible in the timeframe of the current work. As an example some short test chains were run for a 1D shock calibration, using a relatively low resolution for the DSMC simulations. Based on the computational time per chain position, if these chains were to be run for the same number of positions as the chains used later in this chapter for the 0D relaxation calibration, the total computational time per chain would be ~320 hours (~13 days) if the simulations were performed on 128 processors. Switching to 256 processors would cut that time by ~30% (obviously at the cost of lost efficiency), but further increasing the number of processors per simulation

beyond this rapidly becomes pointless. Thus, even if we were to accept the loss of parallel efficiency and use 256 processors per simulation, the chain would still take ~224 hours (~9 days) to complete. While it is a fairly substantial amount of time for a calibration, if this were the total time required we would have performed 1D shock calibrations. The problem is that computational time does not directly translate to actual time when resource availability is considered. The DoE generously allows us the use of Hera (a reasonably large-scale computing cluster), but due to the particulars of the queuing system (jobs have a maximum allowed runtime of 16 hours, the queue system penalizes heavy use by lowering a user's priority for a period of time, and off-site academic users have lower priority than on-site DoE users in the first place) 9 days of computational time on 256 processors would require a month or more of real time. Furthermore, when using relatively short chains it is important to use multiple chains in order to more fully explore the parameter space, and thus we would need to use far more than 256 processors over those 9 days of computation time. While multiple chains can be run in parallel, using large numbers of processors makes the drop in priority (and thus the increase in time spent waiting in queue between each job) substantially worse, to the point that a 1D shock calibration on Hera would take several months, even with relatively low resolution DSMC simulations.

Before moving on, however, note that calibrations for the 1D shock scenario are not prohibitive in an absolute sense; they are only prohibitive with the available computational resources and within a reasonable timeframe for completion of this dissertation research. The limitations described above will become less important over the coming few years as the overall project continues.

Aside from the computational limitations, we will only be using synthetic calibration data regardless of whether we choose the 0D relaxation or the 1D shock for

our scenario, and thus the consideration of which scenario is more experimentally realistic becomes somewhat less important for our purposes. Based on all of the above considerations, and keeping in mind that both scenarios present a complex and challenging physical problem and we have already demonstrated that the 0D relaxation is sensitive to the same parameters as the 1D shock, we elect to use the 0D calibration scenario for our synthetic data calibrations in this work.

SCENARIO DETAILS

The primary scenario we will use for our synthetic data calibrations is the same one we used in Chapter 5 for the 0D relaxation sensitivity analysis. We initialize a 0D box with synthetic air (79% N₂, 21% O₂) with a bulk number density of 1.0×10^{23} #/m³. In order to simulate the thermal equilibrium in the early part of a hypersonic shock, the initial translational temperature is set to $\sim 50,000$ K while the rotational and vibrational temperatures start at 300 K. A relaxation based on this scenario which was performed with the nominal values of all parameters was discussed in some detail in Chapter 5 (see Figs. 5.1, 5.2 and 5.4, along with the accompanying discussion).

When we initially began performing calibrations for the six parameters identified in Table 7.1, we quickly found that we could not properly calibrate them based on the results of only one relaxation scenario. We would get either extremely broad posterior PDFs (which for some parameters would cover the entire range of the prior) or we would get several distinct peaks in the PDF for a given parameter. After further testing, we came to realize that this behavior was not due to the length of the chains or the particular values chosen for various numerical parameters associated with the DRAM algorithm. It was actually due to the fact that, when calibrating six important parameters with such

broad priors, there were actually multiple solutions. Multiple sets of values for the six parameters could produce simulation results which lay within any reasonable set of error bars we might place on our synthetic data.

Therefore, in order to properly calibrate our parameters, we chose to make use of synthetic data from two scenarios. The second scenario is another OD relaxation, with the same initial number densities and the same initial rotational and vibrational temperatures, but this time with an initial translational temperature of 30,000 K. This second relaxation scenario would correspond to a shock with a significantly lower speed than the ~ 8 km/s shock for which the first OD relaxation is intended to be a substitute.

To illustrate this second scenario, a sample relaxation was performed with the nominal values of all parameters, just as was done for the first scenario. Density profiles from this relaxation are shown in Figure 7.1. Due to the lower initial translational temperature most of the dissociation rates are significantly lower, and therefore the timescale for the relaxation to approach equilibrium is longer. We do not need to follow the relaxation all the way to full equilibrium, however. We simply carry out the simulation until a point where the density profiles are changing significantly more slowly, as seen in Fig. 7.1. We also chose the end time for our simulation based on the desire to have a simulation for this scenario take approximately the same amount of computational time as a simulation for the first relaxation scenario.

Examining the curves in Fig. 7.1, O_2 density drops quickly near the beginning of the relaxation, although not as quickly as for the first relaxation scenario. Unlike in the first relaxation, O_2 density does not drop to zero within the timeframe of the simulation; some small amount remains even at the end. The density of N_2 also does not drop as much as it did for the first relaxation scenario. The N density and the O density are lower relative to the first relaxation scenario since N_2 and O_2 are less dissociated and also due to

the fact that less of the NO (which is initially formed by the exchange reactions) is dissociated.

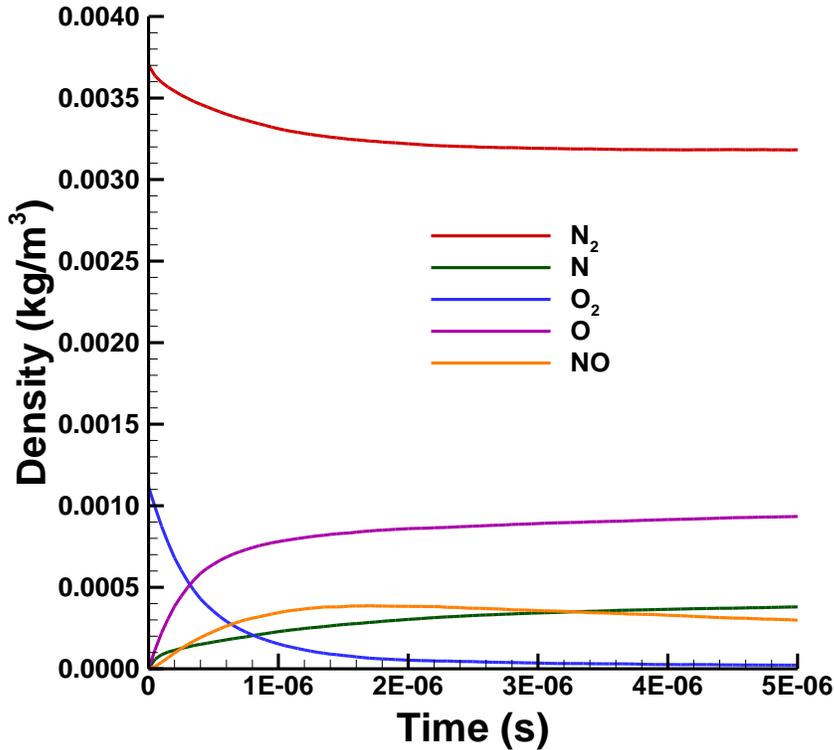


Figure 7.1: Evolution of the density profiles for all five species during a relaxation from an initial state with 79% N₂ and 21% O₂. Initial translational temperature is 30,000 K while initial rotational and vibrational temperatures are 300 K. This relaxation is the second scenario for our synthetic data calibrations.

CALIBRATION DATA

With our scenarios chosen, we must now select a set of calibration data for each scenario. In principle, there is no reason that any of the calibration data has to be directly related to the QoI which was used for the sensitivity analysis, but for this work we choose to make use of ρ_{NO} again in our calibration dataset. We use ρ_{NO} mainly because it is sensitive to the parameters we wish to calibrate (we know this since we used it as the QoI

for the sensitivity analysis and we chose the parameters to calibrate based on that sensitivity analysis). It is also at least plausibly measurable in a shock tube experiment (although we do not actually have any usable experimental data for it) and it is less affected by DSMC stochastic noise than the temperature profiles.

In addition to making it clear that we needed two scenarios in order to properly calibrate our parameters, our initial test calibrations also revealed that we would need to use calibration data based on more than just the NO density profile. Even when using data from two scenarios, calibrations based on only data for ρ_{NO} did not lead to well-characterized post-calibration PDFs for the parameters. We therefore chose to use the densities of N and O in addition to ρ_{NO} in our synthetic dataset. Note that together these three profiles incorporate all possible density information for our five-species air. Since the bulk density is constant and elemental identities are preserved, if we have the density of N, O, and NO at any point in the relaxation then we also know the densities of N_2 and O_2 at that point by conservation of the mass of each element.

Our synthetic data are taken from a run of both scenarios with the nominal values of all parameters. As was discussed in Chapter 4, we must also specify an uncertainty for use in our likelihood equation (the likelihood equation for these calibrations will be discussed in the next section). There is a difference this time, though. In Chapter 4 we were using synthetic data from a hypothetical model which was represented by a single, deterministic equation, and so we were absolutely certain that with the correct parameter values the model results could match the data to arbitrary precision. Therefore, in that case the uncertainty used in the likelihood equation was only a data uncertainty, there was no uncertainty for the model. Since the data were synthetic data from a deterministic model the data uncertainty would in reality be zero, but since some non-zero uncertainty

is required by the MCMC algorithm it was necessary to make up a hypothetical uncertainty in order to do the calibration example in Chapter 4.

The fact that we are now using a stochastic model changes things. We can no longer assume that if we ran our “experiment” many times we would always get the exact same results. If we ran the code to generate the synthetic data multiple times with different random number seeds we would get slightly different output each time, and thus we have actual data uncertainty. Furthermore, we can no longer be sure that our model will exactly match the data, even if we ran it with the exact same parameters which were used to generate the data in the first place. This means that in addition to data uncertainty, we have a form of model uncertainty as well.

The uncertainty we will use for these calibrations is therefore not completely arbitrary. We must use an uncertainty which is large enough to address the fact that our simulations (both those which generated the synthetic data and those run during the calibration process) are stochastic and have finite noise. Based on empirical testing and on experience with the noise levels in our DSMC output when run with the resolutions we will be using for the simulations during the calibration, we choose an uncertainty for our data set. For simplicity, we choose an uncertainty which is high enough that we can use the same uncertainty for all three density profiles for both scenarios (i.e. we choose an uncertainty which is high enough to address the noise level in the noisiest of the density profiles in either scenario). In Figure 7.2 we show the density profiles which we will use as our synthetic data for the first relaxation scenario (the one from Chapter 5) and in Figure 7.3 we show the data for the second relaxation scenario (which we discussed earlier in this chapter). In both figures we also show 2σ error bars on each data set based on the uncertainty we have empirically chosen.

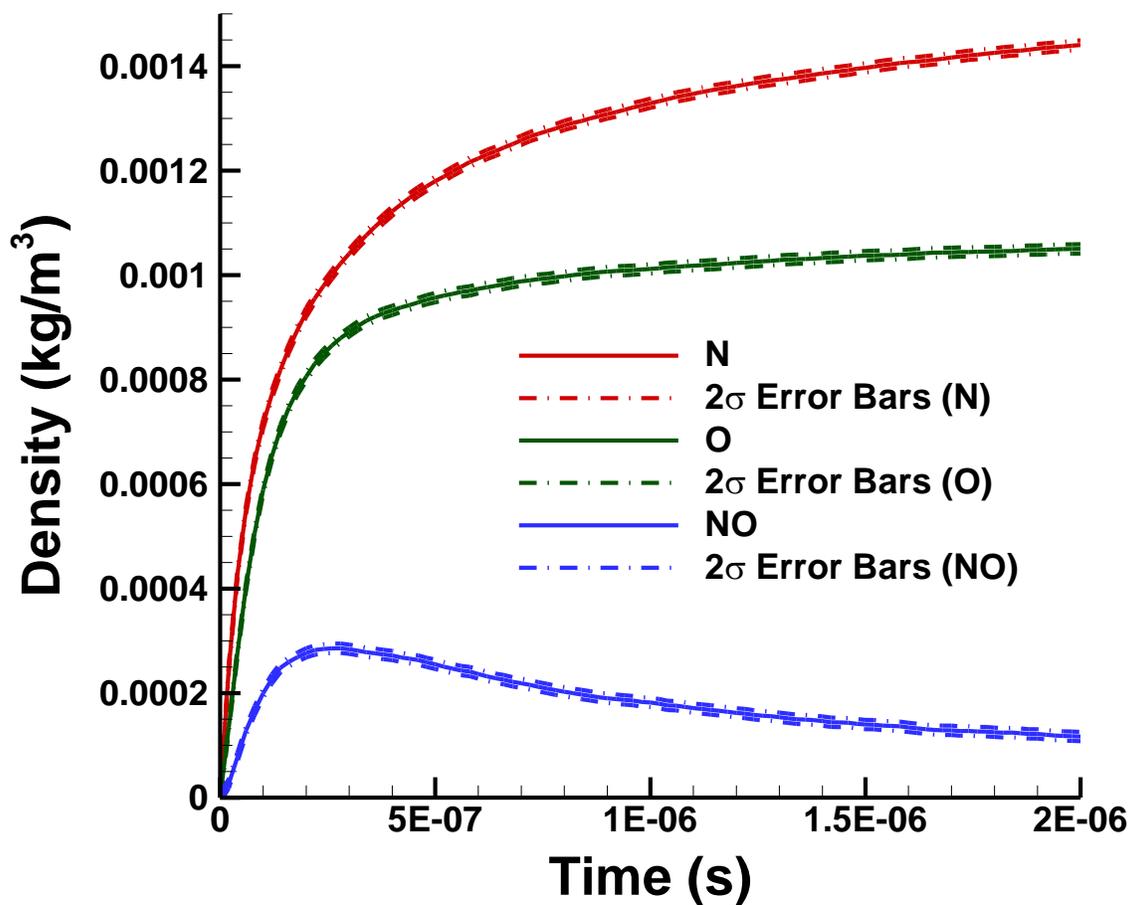


Figure 7.2: Synthetic data for the first relaxation scenario, along with 2σ error bars on the data (based on an uncertainty of $\sigma^2 = 2.0 \times 10^{-11}$). A set of 11 evenly spaced points from each curve are used as the synthetic data points for the first scenario.

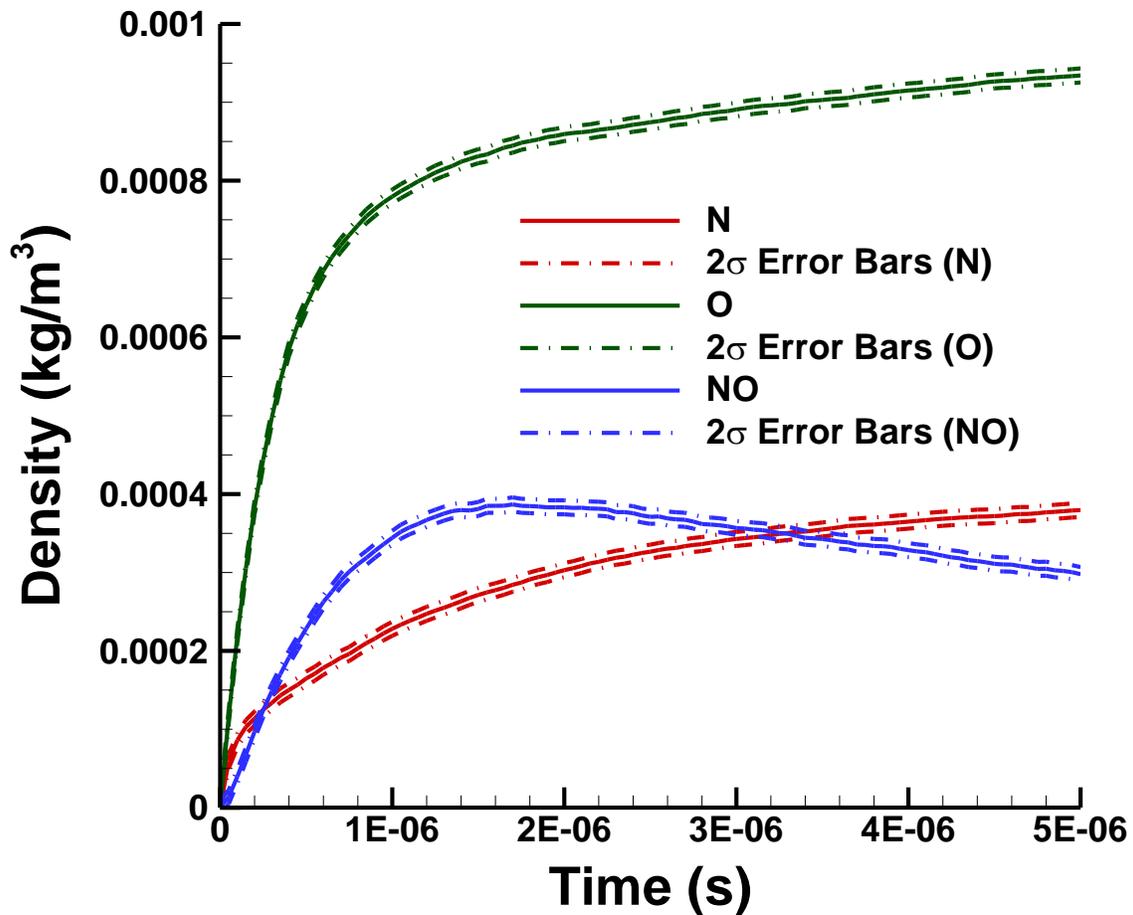


Figure 7.3: Synthetic data for the second relaxation scenario, along with 2σ error bars on the data (based on an uncertainty of $\sigma^2 = 2.0 \times 10^{-11}$). A set of 11 evenly spaced points from each curve are used as the synthetic data points for the second scenario.

LIKELIHOOD EQUATION

The role of the likelihood equation in the calibration process was described in detail in Chapter 4. In short, a likelihood equation tells us the degree to which we believe that a given set of parameter values is reasonable in light of the comparison of the output of the model (when run with that set of parameter values) to the calibration data. For the

calibrations in this work we will use a Gaussian likelihood equation and we will assume that the data points are all independent of one another. While not always realistic, this assumption is not uncommon. For example, both Miki *et al.* (2012b) and Panesi *et al.* (2012) chose to treat their data points as independent when calibrating parameters related to shock tube simulations. We also use a constant uncertainty for all of the data points, as discussed in the previous section. With those assumptions in place, the likelihood for a given set of parameters (based on a given dataset) is obtained from Eq. 4.2, which is repeated here for convenience.

$$likelihood = P(D|\theta) = \frac{1}{(2\pi\sigma^2)^{\frac{N_d}{2}}} \exp\left[-\frac{1}{2\sigma^2} \sum_{i=1}^{N_d} (D_i - X_i)^2\right] \quad (4.2)$$

where N_d is the number of calibration data points, $\mathbf{D} \in \mathbb{R}^{N_d}$ is the vector of data points from the calibration dataset, $\mathbf{X} \in \mathbb{R}^{N_d}$ is the model output vector, and σ^2 is the variance which is expected at each data point due to the uncertainty described earlier. When we employ Eq. 4.2 in the code we actually calculate and return the natural log of the likelihood, and we make use of basic properties of the natural logarithm in order to calculate the ratio of the candidate and current position likelihoods without encountering numerical problems related to storing very large or small numbers.

When dealing with multiple types of data from multiple scenarios, we calculate a separate likelihood based on each dataset, and the overall likelihood is simply

$$likelihood_{overall} = \prod_{i=1}^{N_{datasets}} likelihood_{i^{th}dataset} \quad (7.1)$$

where $N_{datasets}$ is the total number of data sets being used in the calibration and the likelihood based on the i^{th} dataset is calculated from Eq. 4.2. By using Eq. 7.1, we have implicitly given equal weight to all of our datasets.

CALIBRATION PROCESS

We use the PECOS-developed software package QUESO (Prudencio and Schulz, 2012) to solve the statistical inverse problem in order to calibrate our chosen parameters with respect to the synthetic datasets previously described. The calibration methodology was discussed at length in Chapter 4, so in this section we will discuss only the parallel organization employed for our current synthetic data calibration, which was not discussed previously since the calibrations in Chapter 4 were run in serial due to their negligible computational cost.

There are several levels of parallelism which are employed during the calibration process, as illustrated by the schematic in Figure 7.4. QUESO is capable of running multiple chains in parallel, and we make use of this feature. A subset of the overall group of processors is given to each chain and those processors work on that chain exclusively until it reaches the specified length, at which point they remain idle until all other chains have also been completed. When the overall C++ driver code calls the MCMC/DSMC interface subroutine (via the C++/Fortran 95 interface subroutine, see the flowchart in Fig. 4.2 which illustrates the interactions of the various codes during the calibration process), it passes along the communicator for the group of processors assigned to each chain. The MCMC/DSMC interface subroutine then further subdivides each chain group, and assigns these new subgroups to each simulate one of the two scenarios used for the calibration. The communicator for each scenario group of processors is then sent to the DSMC code. The DSMC code then assigns a unique random number seed to all of the processors in the scenario group, and each of them simulates a 0D relaxation with the appropriate conditions for the scenario and using the parameter values corresponding to the candidate chain position. The results from these individual relaxations are ensemble averaged to provide relatively low-noise density profiles, and these profiles are sent back

to the MCMC/DSMC interface code, which calculates the likelihood for the candidate position parameter values based on the dataset for each scenario. Finally, the likelihoods based on the two scenarios are multiplied together to obtain an overall likelihood for the candidate chain position.

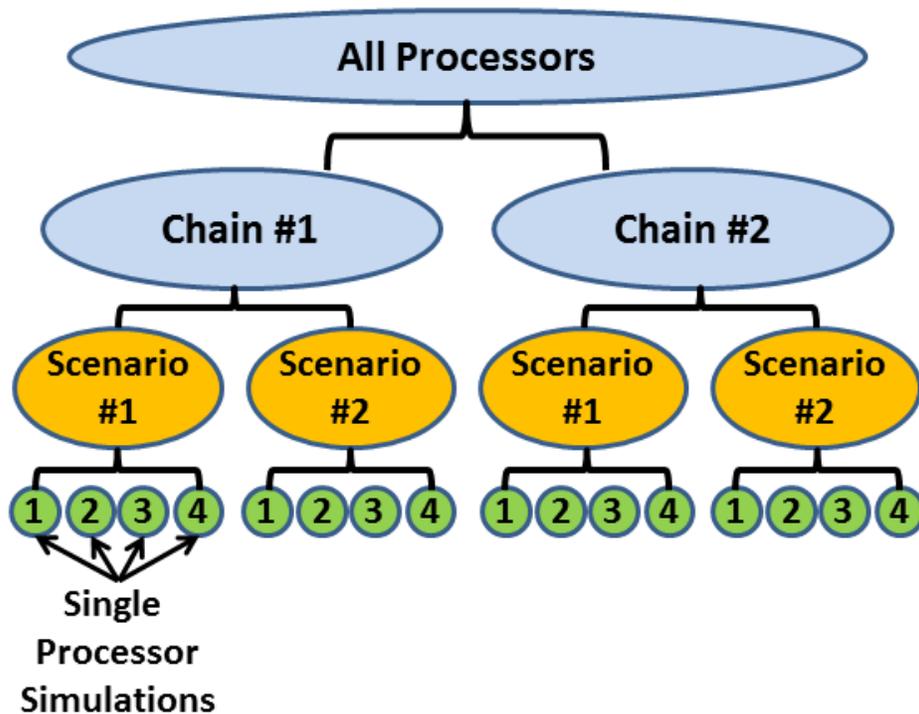


Figure 7.4: Levels of parallelism used during the calibration process. The blue ovals show the parallel organization for the parts of the process that take place within the QUESO driver code, the orange ovals show the organization for the parts within the MCMC/DSMC interface code, and the green circles show the organization within the DSMC code. See Fig. 4.2 for more information on the codes which participate in the calibration process. The single processor runs are ensemble averaged to provide lower noise results. Any number of chains and single processor runs may be used; the numbers here are just for readability of the schematic. The code is also capable of handling more than two scenarios, but we use only two for the calibration presented in this work.

CALIBRATION RESULTS

With all of the necessary pieces now in place, we are finally ready to perform our synthetic data calibration. Thirty-two chains were used in the calibration, each with a random starting position in parameter space. Each chain contained a total of 8,000 positions, and both delayed rejection and adaptation of the proposal covariance matrix were used to improve the convergence of the chains (see Chapter 4 for a discussion of these two improvements to the MCMC algorithm). Running the simulations for this calibration required four 16-hour, 4096 processor runs, for a total of more than 250,000 CPU hours.

Once the chains have reached their specified number of positions, we are left with a set of $32 \times 8,000 = 256,000$ positions in parameter space. The chains take some time to explore the parameter space before they begin to approach the posterior distribution, and therefore we use a burn-in of 4,000 positions per chain when we estimate the post-calibration PDFs for the parameters. This leaves us with 128,000 positions. Making use of KDE (as discussed in Chapter 4), we estimate the posterior distribution for each of the six parameters based on these 128,000 positions.

The posterior PDFs for all six parameters are shown in Figures 7.5 – 7.7. It is immediately apparent from these figures that the posterior PDF for each of the six parameters has a dominant peak. Several of the posterior PDFs also have one or more smaller, secondary peaks. We will investigate these secondary peaks later in this section, but first we would like to examine all of the dominant peaks in more detail.

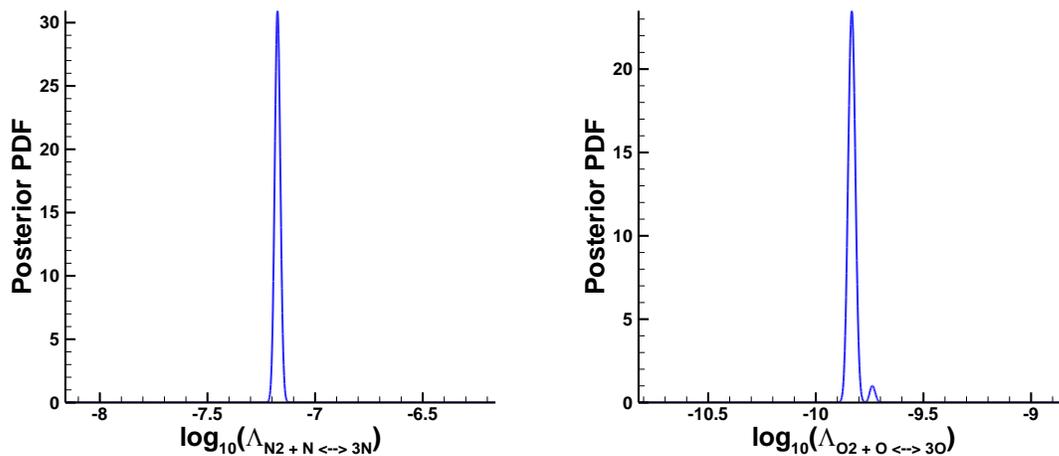


Figure 7.5: Post-calibration PDFs for parameters #2 and #9 (based on numbering from Table 7.1).

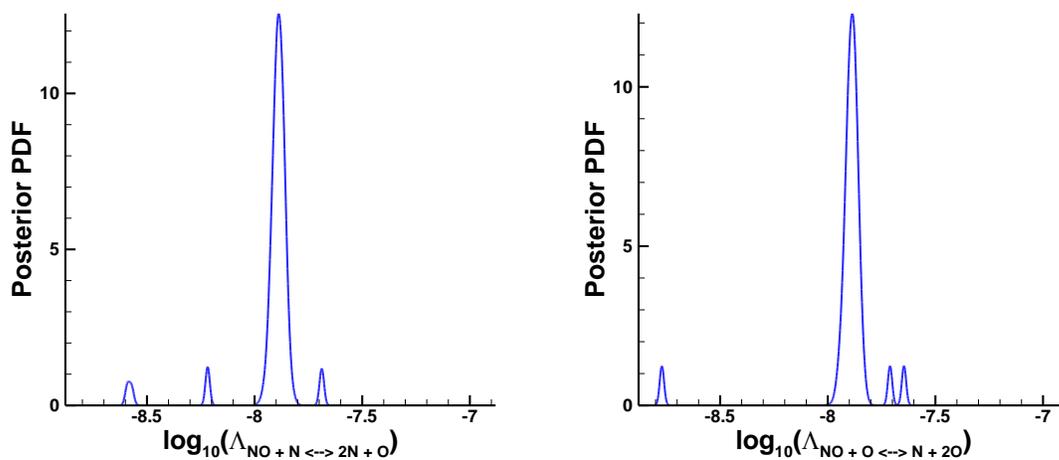


Figure 7.6: Post-calibration PDFs for parameters #12 and #14 (based on numbering from Table 7.1).

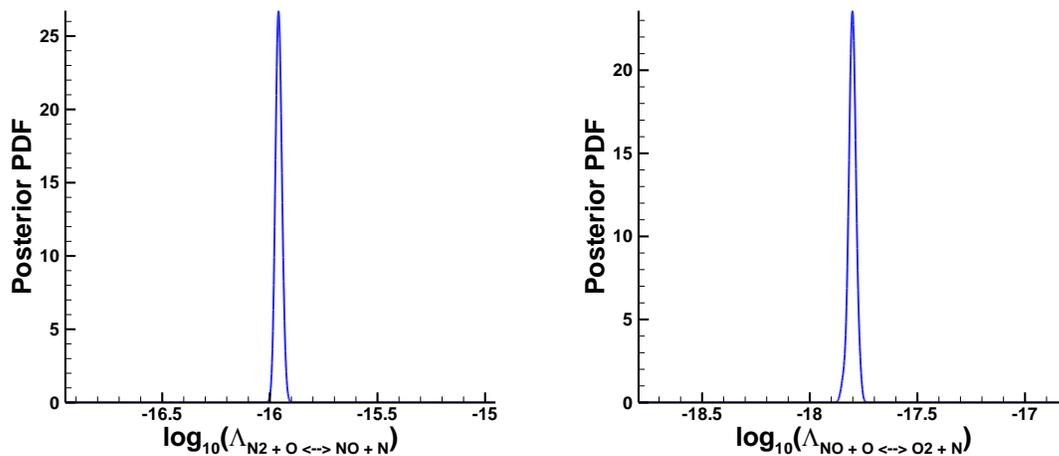


Figure 7.7: Post-calibration PDFs for parameters #16 and #17 (based on numbering from Table 7.1).

To this end, we plot all of the PDFs vs. a normalized parameter value which is given by $\log_{10}\Lambda - \log_{10}\Lambda_{\text{nom}}$ (so that this normalized parameter value will vary between -1 and 1 , and a normalized parameter value of zero corresponds to the nominal value of each parameter). Posterior PDFs for all six parameters are plotted vs. this normalized parameter value in Figure 7.8. It is clear from this figure that the dominant peak for each parameter is located at or near the nominal value for the given parameter (since the dominant peaks are all near a normalized parameter value of zero). These same PDFs are shown again in Figure 7.9, this time zoomed-in closely to see the dominant peaks. While the peaks of the posterior PDFs for the six parameters are not located at a normalized parameter value of exactly zero, they are all close (keeping in mind that the PDFs in Fig. 7.9 are greatly expanded in the region near zero). The peak for $\log_{10}\Lambda$ for the reaction $\text{N}_2 + \text{N} \rightleftharpoons 3\text{N}$ (which is parameter #2 based on the numbering in Tables 5.1, 6.1, and 7.1) is the furthest from its nominal value. Its peak is at the normalized parameter value of -0.013 , which means that the value of Λ at the location of the peak of the posterior PDF

differs from the nominal value of this parameter by $\sim 3\%$. Keeping in mind that the chains only have 8,000 positions (4,000 of which were discarded as the burn-in), that some amount of stochastic noise is present in both the synthetic data and the simulations for the candidate chain positions, and that the parameters were allowed to vary over a two-order of magnitude range, we consider a maximum error of 3% in the locations of the peaks of the posterior PDFs to be an indication that the synthetic data calibration was successful. However, in the future it would be worthwhile to run more and longer chains with better DSMC resolution in order to confirm that this 3% error is due only to stochastic noise and not due to a bias error of some kind. Moreover, it might also be worth exploring how many chain positions are required to achieve a desired level of accuracy in the posterior PDFs. This desired level of accuracy could be based on either the full PDF or just the location of the peak. This is a difficult problem because there are so many numerical parameters to consider. It would be necessary to consider how convergence to the posterior distributions is affected by the number of delayed rejection steps, the scaling factors used for the later stage proposals, the number of positions between each adaptation of the proposal covariance matrix, the number of positions discarded for the burn-in, the number of chains employed, and the appropriate computational time trade-off between increasing the DSMC resolution and obtaining more chain positions.

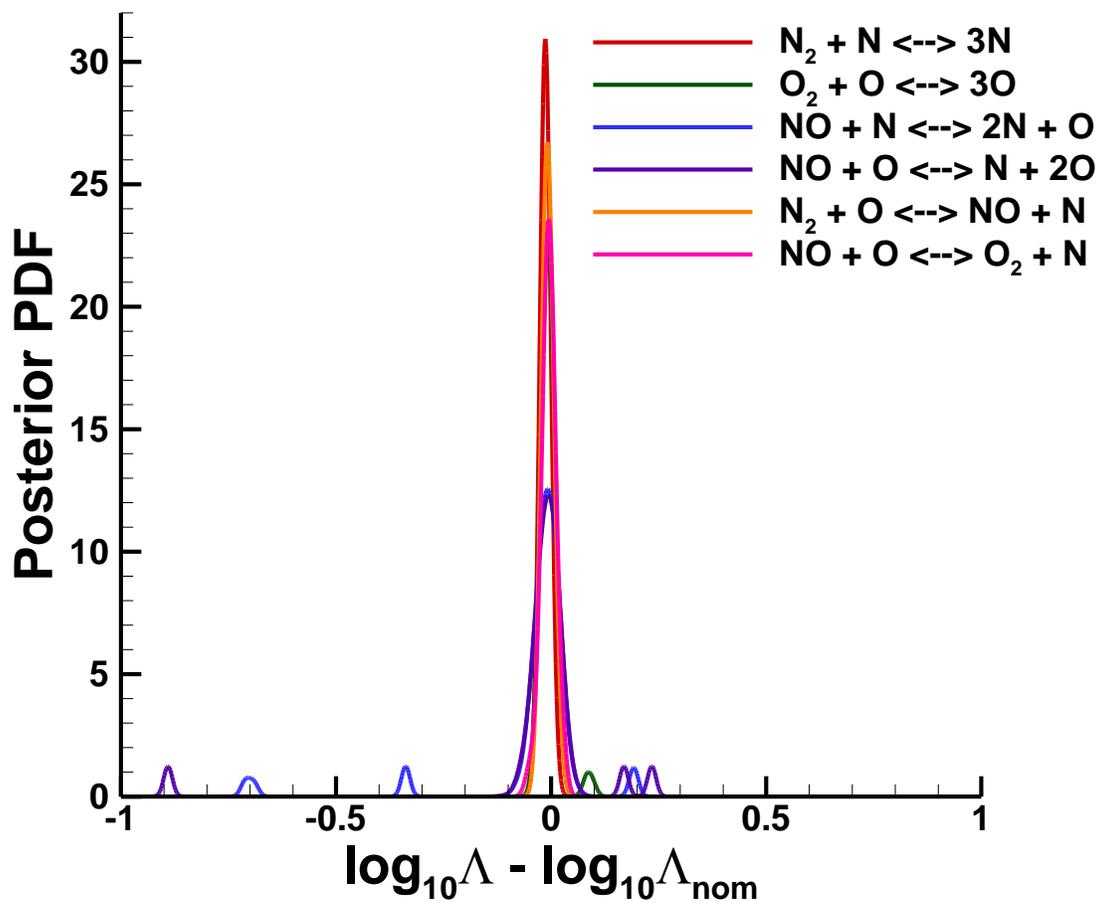


Figure 7.8: Post-calibration PDFs for all six of the parameters, with the horizontal axis normalized so the nominal value for all parameters is zero.

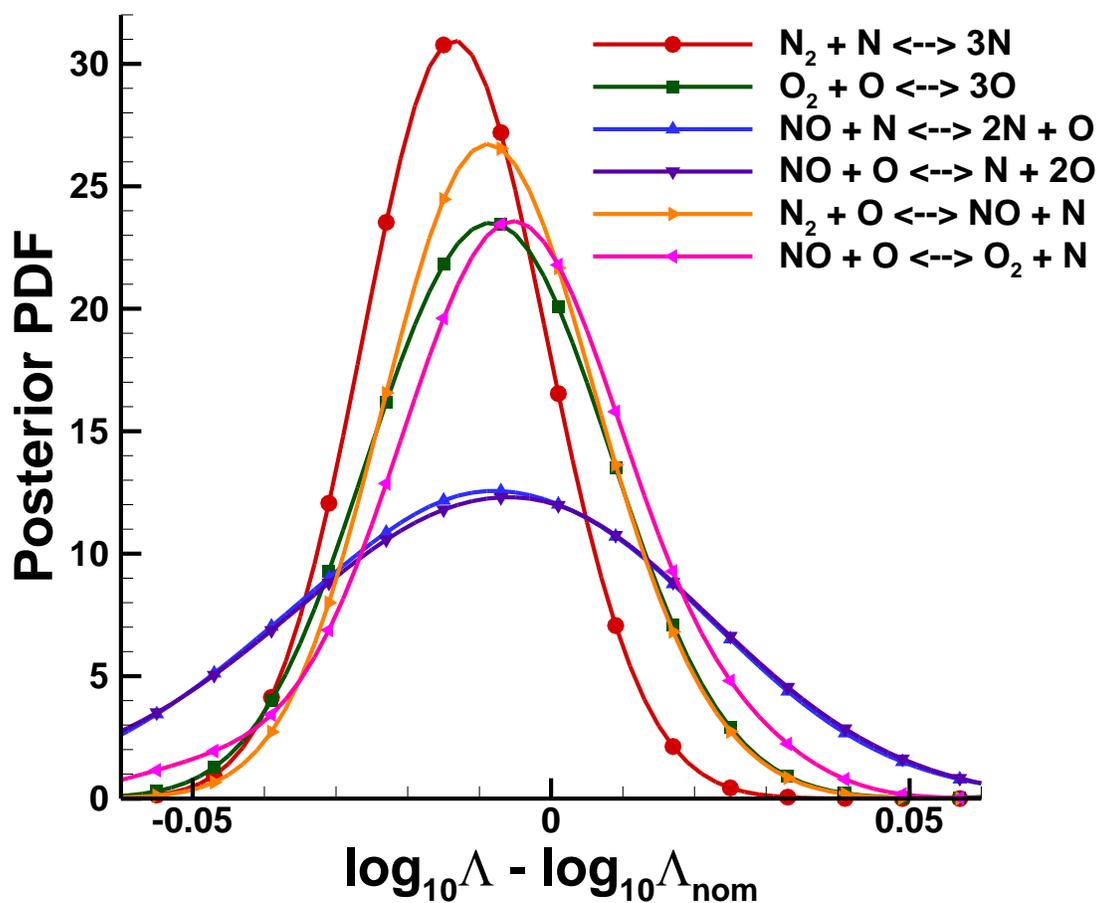


Figure 7.9: Same PDFs as in Fig. 7.8, zoomed in to show the PDF peaks more closely.

Before we are fully ready to call the synthetic data calibration a success, however, we need to understand the secondary peaks seen in the posterior PDFs for $\log_{10}\Lambda$ for the reactions $\text{O}_2 + \text{O} \rightleftharpoons 3\text{O}$, $\text{NO} + \text{N} \rightleftharpoons 2\text{N} + \text{O}$, and $\text{NO} + \text{O} \rightleftharpoons \text{N} + 2\text{O}$ (parameters 9, 12, and 14). The posterior PDFs for these three parameters are plotted in Figures 7.10 - 7.12, which show the dominant peak and all of the secondary peaks for each parameter.

What we would like to determine is whether these secondary peaks exist because there are truly multiple solutions (as there were when we were using only one scenario

and/or only one density profile for our dataset) or whether these secondary PDFs are numerical artifacts which result from our relatively short chains. In order to make this determination, we consider the fact that we actually have values of the likelihood for well over 450,000 distinct positions in parameter space. These include the likelihoods for the 256,000 first stage candidates (we do not need to discard the burn-in if we are only concerned with the likelihoods for specific positions in parameter space), and also more than 200,000 second stage candidates which were tried by the delayed rejection algorithm after the first stage candidate for a given chain position was rejected.

In order to make use of these positions, we must remember that having the likelihood at a given position in parameter space gives us the value of the posterior PDF at that location (aside from a normalization constant), because according to Bayes' theorem the posterior distribution is equal to the prior multiplied by the likelihood and divided by a constant. The equation for Bayes' theorem is

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{\int_{\theta} P(D|\theta')P(\theta')d\theta'} \quad (7.2)$$

where $P(\theta|D)$ is the posterior distribution for the parameter after calibration based on the dataset D , $P(D|\theta)$ is the likelihood, $P(\theta)$ is the prior (which is always uniform in our work, and therefore can be neglected), and the integral in the denominator is carried out over the entire range of the parameter θ . The integral in the denominator is not a function of θ , and therefore the denominator is a constant for a calibration based on any given dataset.

The purpose of our calibration is not to provide the posterior distribution at any given location in parameter space; based on the equation above we can determine the posterior PDF (aside from the normalization constant) for any point in parameter space from the results of a single simulation with our model. Instead, the purpose of the calibration is to sample the posterior distribution in an intelligent way, in order to explore the regions of parameter space where the likelihood is relatively high. Sometimes,

however, a chain will find a location in parameter space where the likelihood (and thus the posterior distribution, since we use a uniform prior) is higher than in all of the surrounding regions, while still being much lower than at some other far off location. The likelihood around this local maxima may be vastly lower than the likelihood at a true peak of the posterior PDF which is located elsewhere in parameter space, but in spite of this the chain may stay in the region around this local maxima for an extremely long period of time, such that chains of a practical length may never find the true peak. Haario *et al.* (2006) proved the ergodicity of the DRAM algorithm, and therefore all of the chains should eventually converge to the posterior distribution, but there is no way to know how many chain positions would be required to achieve any given convergence metric, and in practice it could be a vast number. This situation may be exacerbated by the fact that we adapt the proposal covariance matrix relatively early (after only 1,000 chain positions). We adapt after only 1,000 positions because it allows the chains which have actually found the true peak to start exploring that region in more detail (rather than wasting computational time by continuing to select far flung first stage candidate positions which are almost always rejected). Unfortunately, the same chain behavior that we want to occur in the region of the true peak also occurs in the region of false peaks (local maxima), because those chains also have their proposal distributions adapted (resulting in a narrower proposal distribution if a substantial number of positions have already been accepted in a small region around the false peak), which then makes it even less likely for that chain to leave the false peak during a calibration with a practical number of chain positions.

Fortunately, however, we can address this problem by simply examining the likelihoods for the positions of each chain. In addition to showing the posterior PDFs, Figs. 7.10 – 7.12 also display the likelihoods for the candidate positions tested during the

calibration. The likelihood values shown in the plot neglect the constant in front of the exponent in Eq. 4.2 because this constant does not change depending on the location in parameter space, and we are only interested in the value of the likelihood at a given position relative to other positions. If the secondary peaks in the posterior PDFs for these parameters occurred because multiple peaks were actually present in the true posterior PDF (i.e. sets of parameter values from multiple regions in parameter space would lead to simulation results which closely match the synthetic datasets), we would expect to find values of the likelihood in the regions near the secondary peaks which are comparable to the values of the likelihood near the dominant peak. These figures show that for all three of these parameters the values of the likelihood in the region near the dominant peak are orders of magnitude greater than the likelihood for any position in the regions around the secondary peaks. This confirms for us that these secondary peaks are false peaks and can safely be ignored as numerical artifacts.

Having confirmed that the dominant peaks in the posterior PDFs for all of the parameters are within 3% of the nominal values of those parameters (which were used to generate the synthetic data), and having demonstrated that the secondary peaks are false peaks which result from our use of a relatively small number of chain positions, we are now willing to consider our synthetic data calibration successful.

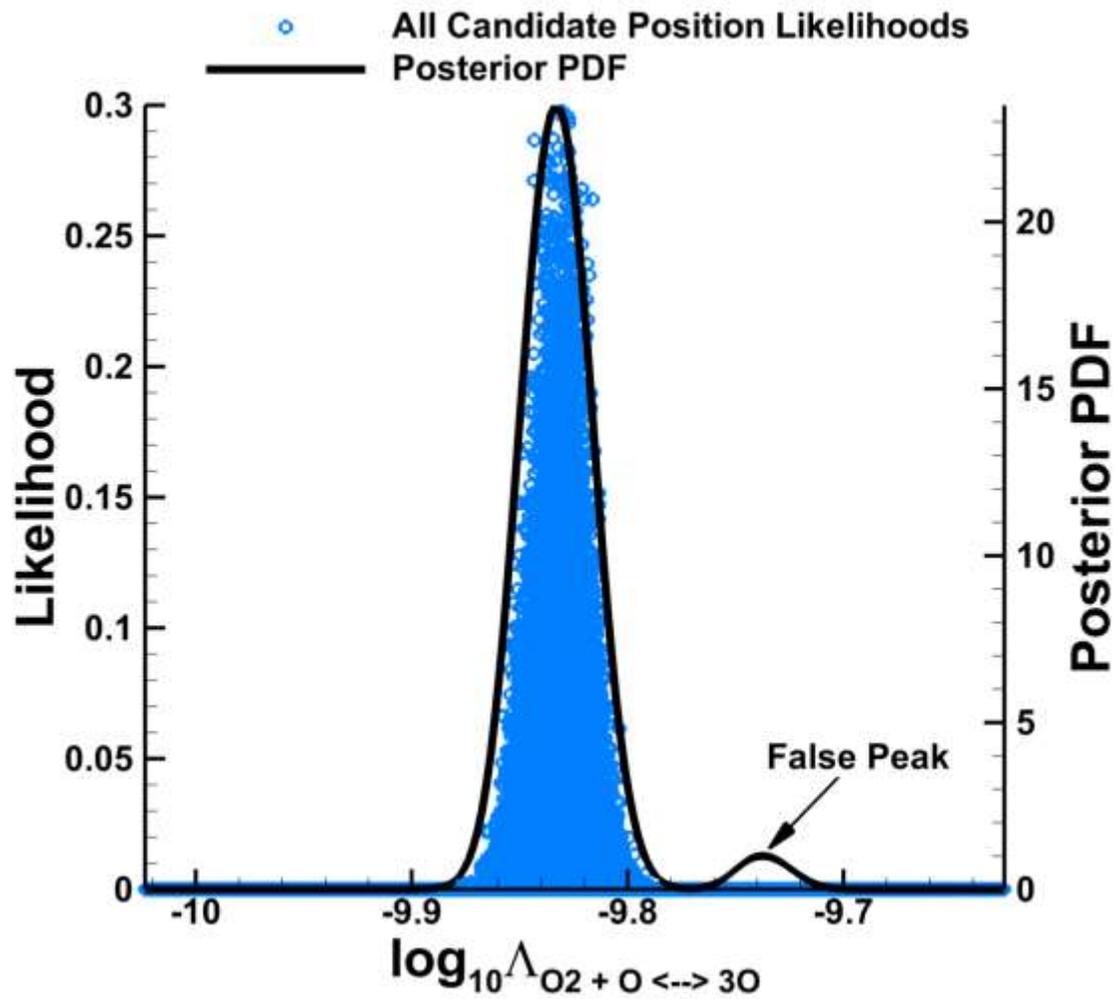


Figure 7.10: Post-calibration PDF for parameter #9 from Table 7.1. Also shown are the values of the likelihood for the candidate positions tested during the calibration (both those accepted as chain positions and those which were not).

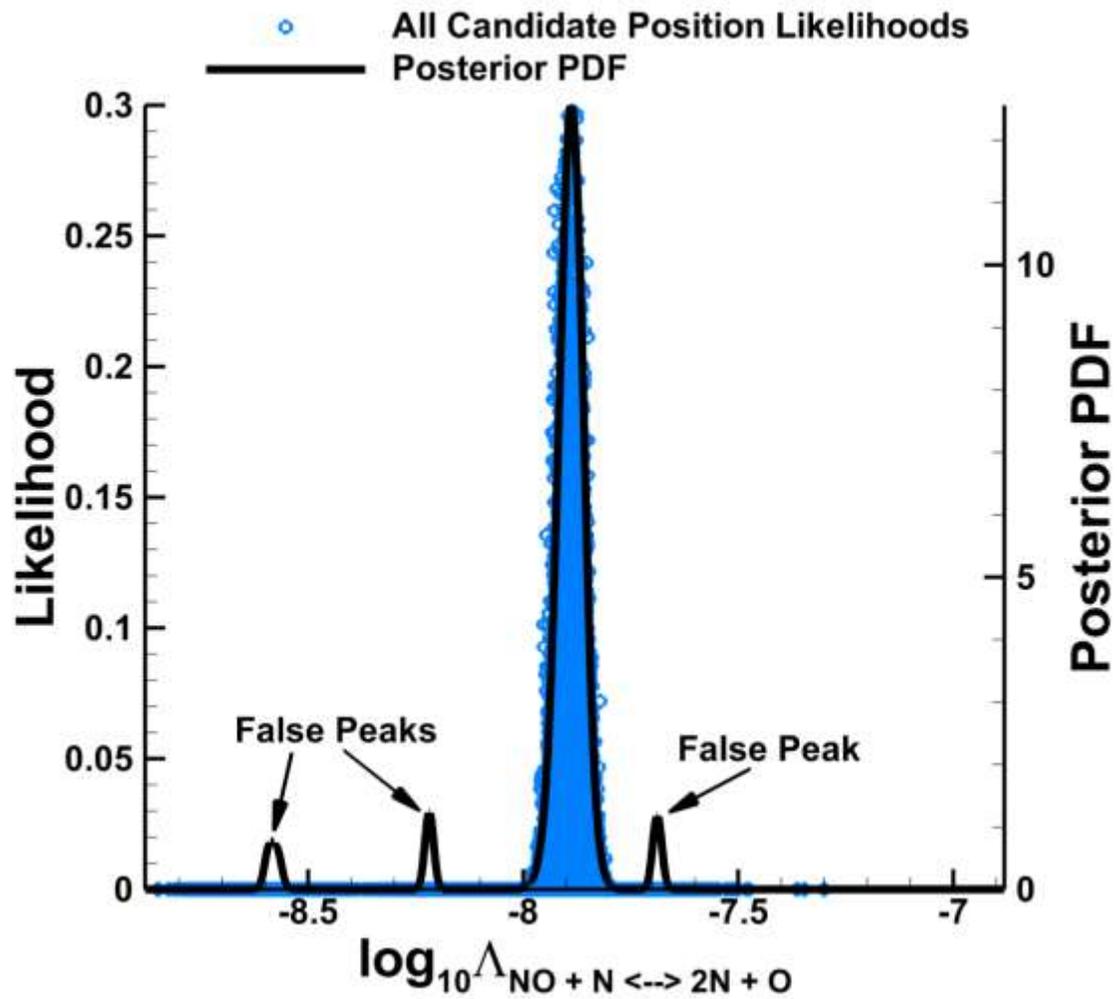


Figure 7.11: Post-calibration PDF for parameter #12 from Table 7.1. Also shown are the values of the likelihood for the candidate positions tested during the calibration (both those accepted as chain positions and those which were not).

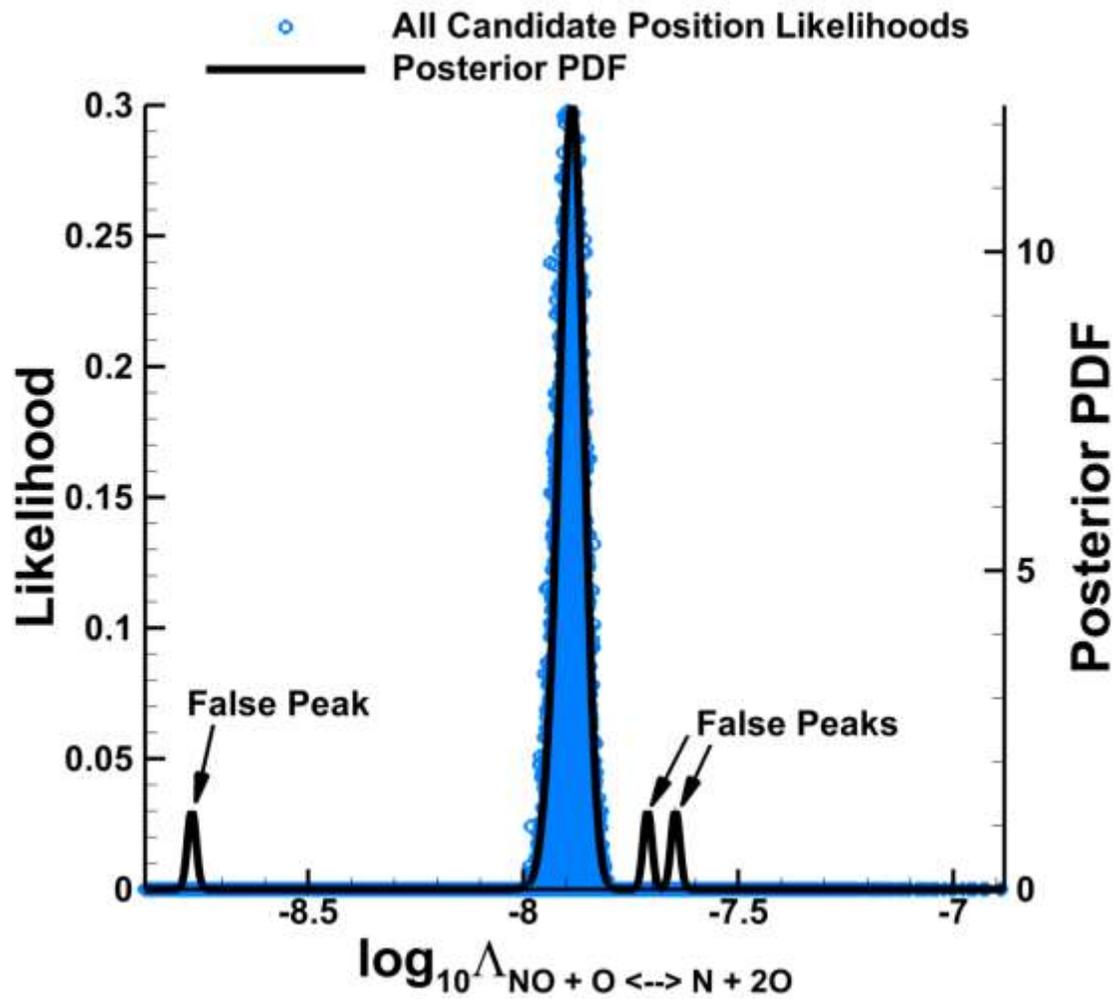


Figure 7.12: Post-calibration PDF for parameter #14 from Table 7.1. Also shown are the values of the likelihood for the candidate positions tested during the calibration (both those accepted as chain positions and those which were not).

Chapter 8: Conclusions and Future Work

CONCLUSIONS

At this point, we recall the objectives we set out for this work in the first chapter and assess whether these objectives have been achieved.

Our first objective was to write and test a DSMC code which would be suitable for use in this work. We now have a DSMC code which was written, tested, and optimized for the simulation of 1D hypersonic shocks, the primary scenario of interest in this project. We have used this code to simulate thousands of shocks. Our shock simulation technique is well-suited for situations where the post-shock conditions are not known *a priori*, as is the case in almost any scenario where real gas effects are important. We have also modified the commonly used TCE model to allow for the accurate simulation of reaction rates which are large relative to the elastic collision rate.

Our second objective was to perform sensitivity analyses for both a 0D relaxation scenario and a 1D shock scenario, and to test the hypothesis that the 0D relaxation can in some ways serve as a substitute model for the 1D shock for the purpose of sensitivity analysis. A complete sensitivity analysis methodology was generated. We integrated our DSMC code with a sensitivity analysis driver code in order to sample a high-dimensional parameter space, including parameters for elastic collision cross-sections, rotational and vibrational energy transfer, and reaction rates. We successfully completed sensitivity analyses for both the 0D relaxation and the 1D shock. In each of these analyses we presented results for sensitivities based on both r^2 and the mutual information. This appears to be the first global, Monte Carlo based sensitivity analysis for DSMC simulations. Furthermore, it appears to be the first sensitivity analysis in the hypersonics field to use the mutual information as a measure of sensitivity. Perhaps most significantly, we based our sensitivity analyses on vector quantities of interest. We

described a technique for obtaining overall sensitivities which incorporate the sensitivities for all of the components of a vector QoI while emphasizing the regions of the flow where non-equilibrium is most significant. Making use of these overall sensitivities, we determined that the six most sensitive parameters were the same for both the 0D relaxation and the 1D shock scenarios, regardless of whether r^2 or the mutual information was used as the measure of sensitivity. We also investigated the differences which do exist between the sensitivities calculated with the two different measures, and found that they were due to nonlinearities in the relationships between certain parameters and certain components of the vector QoI. These nonlinear relationships were captured by the mutual information but not by r^2 . For this reason, if in a future sensitivity analysis there were substantial differences between sensitivities based on the two measures, we would believe the results based on mutual information over those based on r^2 . This provided some justification for our hypothesis that the 0D relaxation can in some cases be used as a substitute model for the 1D shock for the purpose of sensitivity analysis.

Our final objective was to perform a synthetic data calibration for the parameters selected in our sensitivity analysis. In order to perform this calibration, we made use of the QUESO software package along with a set of interface codes which link the DSMC code to QUESO. These interface codes can calculate the likelihood at any given candidate position based on multiple types of data which come from simulations of many different scenarios. This feature should prove useful in the future when calibrating based on experimental data. A calibration was performed based on a set of synthetic data which consisted of three density profiles from each of two different 0D relaxation scenarios. This calibration was successful in generating posterior PDFs with peaks near the nominal values of the parameters which were used to generate the synthetic data. We also

demonstrated that the small secondary peaks found in the posterior PDFs for several of the parameters are numerical artifacts which may be ignored.

FUTURE WORK

There are several possibilities for future work as the overall project moves forward. These possibilities mostly fall into two broad categories: improvements to the DSMC code and the use of more advanced statistical analysis techniques.

The current DSMC code is optimized for the simulation of 1D shocks, and it was written with the intent of eventually simulating experiments performed at a shock tube facility such as NASA EAST (Grinstead *et al.*, 2008). To properly simulate the EAST shock tube the DSMC code would need to be upgraded to include modeling of ionization and electronic excitation. Ionization has been modeled in DSMC by Bird (1989), Bartel and Justiz (1993), Boyd (2007), and Ozawa *et al.* (2008), among others. Electronic excitation has also been incorporated into DSMC codes in various past works, including Carlson and Hassan (1992), Gallis and Harvey (1994), and Li *et al.* (2011).

In order to compare DSMC simulation results with data from NASA EAST it would be necessary for the simulation to provide results for the radiation (either spectrally integrated, or if possible, spectrally resolved) from the shock-heated gas at a series of locations downstream of the shock front. The radiation could be calculated directly within the DSMC code as was done by Gallis and Harvey (1995), Berghausen *et al.* (1996), and Moore *et al.* (2011), or it could be obtained from the DSMC results via post-processing with a line-by-line radiation solver such as NEQAIR (Whiting *et al.*, 1996). Finally, although it would not be required in order to simulate shocks from NASA EAST, the DSMC code could be upgraded to include some or all of the improvements

(such as nearest neighbor collisions) incorporated in the sophisticated DSMC algorithm of Bird *et al.* (2009).

Future work could also focus on the use of more advanced statistical techniques. Kernel density estimation was used in this work to obtain PDFs from scatterplot data during the process of calculating the mutual information, but we could also have used k-nearest neighbor (Kraskov *et al.*, 2004) or B-splines (Daub *et al.*, 2004). Non-uniform (most likely Gaussian) priors could be employed during both the sensitivity analysis and the calibrations in order to better handle parameters for which we have a nominal value in which we are reasonably confident. We could also make use of even more state of the art MCMC algorithms, such as that of Prudencio and Cheung (2012). That algorithm incorporates multi-level modeling, which means that multiple calibrations are run sequentially and each new calibration makes use of refined prior distributions for the parameters which are based on the posterior distributions from the previously completed calibrations. Another area for additional work is the likelihood equation employed during the calibration process. In this work we have assumed in our likelihood equation that all data points are independent of one another, but this is not required. Future work could address dependencies between the data points by making use of a non-diagonal covariance matrix in the Gaussian likelihood equation. Non-diagonal covariance matrices which were designed to account for dependencies between the data points have been utilized, for example, by Miki *et al.* (2012a). The current work also uses a likelihood equation which assumes that the discrepancy between simulation results and calibration data is best explained by an additive error structure. Future work could examine the results of assuming that the error structure is multiplicative instead.

Finally, due to the computational expense of DSMC simulations it might be worthwhile to consider the use of Gaussian process emulation. A Gaussian process

emulator is a statistical technique which treats the output of a computational model as a Gaussian process over a space which is defined by the inputs to the computational model. A set of simulations is run with the real computational model in order to “train” the emulator, and afterwards the emulator may be used in place of the computational model for certain applications where computational constraints require that some amount of accuracy be sacrificed in order to obtain estimated model output more quickly. Rasmussen and Williams (2006) provide a detailed description of the algorithms involved in Gaussian process emulation, and an example of the use of this technique for the emulation of a complex physical model can be found in Higdon *et al.* (2008).

References

- T. J. Bartel and C. R. Justiz. 1993. "DSMC Simulation of Ionized Rarefied Flows," 24th AIAA Fluid Dynamics Conference, AIAA Paper 93-3095.
- F. Bergemann and I. D. Boyd. 1994. "New discrete vibrational energy model for direct simulation Monte Carlo method," in *Rarefied Gas Dynamics: Experimental Techniques and Physical Systems*, edited by B. D. Shizgal and D. P. Weaver, AIAA, Washington, D.C., pp. 174–183.
- A. K. Berghausen, J. C. Taylor, and H. A. Hassan. 1996. "Direct simulation of shock front radiation in air," *Journal of Thermophysics and Heat Transfer*. Vol. 10, No. 3, pp. 413-418.
- G. A. Bird. 1981. "Monte Carlo simulation in an engineering context," in *Rarefied Gas Dynamics*, edited by S. S. Fisher, AIAA, New York, Part 1, p. 239.
- G. A. Bird. 1989. "Computation of Electron Density in High Altitude Re-entry Flows," 20th AIAA Plasma Dynamics and Lasers Conference, AIAA Paper 89-1882.
- G. A. Bird. 1994. *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*, Oxford Univ. Press. Oxford.
- G. A. Bird, M. A. Gallis, J. R. Torczynski, and D. J. Rader. 2009. "Accuracy and Efficiency of the Sophisticated Direct Simulation Monte Carlo Algorithm," *Physics of Fluids*, Vol. 21.
- C. Borgnakke and P. S. Larsen. 1975. "Statistical collision model for Monte Carlo simulation of polyatomic gas mixture", *Journal of Computational Physics*, Vol. 18, Issue 4, pp. 402-420.
- D. Bose, M. Wright, and Tahir Gökçen. 2004. "Uncertainty and Sensitivity Analysis of Thermochemical Modeling for Titan Atmospheric Entry", 37th AIAA Thermophysics Conference, AIAA Paper 2004-2455.

- D. Bose and M. Wright. 2006. "Uncertainty Analysis of Laminar Aeroheating Prediction for Mars Entries," *Journal of Thermophysics and Heat Transfer*, Vol. 20, pp. 652-662.
- I. Boyd. 2007. "Modeling of associative ionization reactions in hypersonic rarefied flows," *Physics of Fluids*, Vol. 19.
- I. Boyd and E. Josyula. 2011. "State resolved vibrational relaxation modeling for strongly nonequilibrium flows," *Physics of Fluids*, Vol. 23.
- A. B. Carlson and H. A. Hassan. 1992. "Radiation modeling with direct simulation Monte Carlo," *Journal of Thermophysics and Heat Transfer*, Vol. 6, pp. 631-636.
- T. M. Cover and J. A. Thomas, 1991. *Elements of Information Theory*, John Wiley & Sons, New York.
- C. O. Daub, R. Steuer, J. Selbig, and S. Kloska. 2004. "Estimating mutual information using B-spline functions – an improved similarity measure for analysing gene expression data," *BMC Bioinformatics*, Vol. 5.
- D. A. Erwin, G. C. Pham-Van-Diep, and E. P. Muntz. 1991. "Nonequilibrium gas flow I: A detailed validation of Monte Carlo direct simulation for monatomic gases," *Physics of Fluids A*, Vol. 3, pp. 697.
- E. D. Farbar and I. D. Boyd. 2011. "Direct Simulation of Shock Layer Plasmas," in *Proceedings of the 27th International Symposium on Rarefied Gas Dynamics*, Pacific Grove, CA, edited by D. A. Levin, I. J. Wysong, and A. L. Garcia, AIP, Melville, NY.
- P. J. Green and A. Mira. 2001. "Delayed rejection in reversible jump Metropolis-Hastings," *Biometrika*, Vol. 88, pp. 1035–1053.
- M. A. Gallis and J. K. Harvey. 1994. "Nonequilibrium Thermal Radiation from Air Shock Layers Modeled with Direct Simulation Monte Carlo," *Journal of Thermophysics and Heat Transfer*, Vol. 8, No. 4, pp. 765-772.

- M. A. Gallis and J. K. Harvey. 1995. "Atomic Species Radiation from Air Modeled with Direct Simulation Monte Carlo Method," *Journal of Thermophysics and Heat Transfer*, Vol. 9, No. 3, pp. 456-463.
- J. Grinstead, M. Wilder, J. Olejniczak, D. Bogdanoff, G. Allen, and K. Danf. 2008. "Shock-Heated Air Radiation Measurements at Lunar Return Conditions", 46th AIAA Aerospace Sciences Meeting, AIAA Paper 2008-1244.
- R. Gupta, J. Yos, and R. Thompson. 1989. "A review of reaction rates and thermodynamic and transport properties for the 11-species air model for chemical and thermal nonequilibrium calculations to 30000 K", NASA Technical Memorandum 101528.
- H. Haario, E. Saksman, and J. Tamminen. 1999. "Adaptive proposal distribution for random walk Metropolis algorithm," *Computational Statistics*, Vol. 14, pp. 375-395.
- H. Haario, E. Saksman, and J. Tamminen. 2001. "An adaptive Metropolis algorithm," *Bernoulli*, Vol. 7, pp. 223-242.
- B. L. Haas and I. D. Boyd. 1993. "Models for direct Monte Carlo simulation of coupled vibration-dissociation," *Physics of Fluids A*, Vol. 5, No. 2, pp. 478.
- H. A. Hassan and D. Hash. 1993. "A generalized hard-sphere model for Monte Carlo simulation," *Physics of Fluids A*, Vol. 5, pp. 738.
- B. L. Haas, D. B. Hash, G. A. Bird, F. E. Lumpkin, and H. A. Hassan. 1994. "Rates of thermal relaxation in direct simulation Monte Carlo methods," *Physics of Fluids*, Vol. 6, pp. 2191.
- D. Higdon, J. Gattiker, B. Williams, and M. Rightley. 2008. "Computer Model Calibration Using High-Dimensional Output," *Journal of the American Statistical Association*, Vol. 103, No. 482, pp. 570-583.

- K. Koura and H. Matsumoto. 1992. "Variable soft sphere molecular model for air species," *Physics of Fluids A*, Vol. 5, pp. 1083.
- A. Kraskov, H. Stogbauer, P. Grassberger. 2004. "Estimating mutual information," *Physical Review E*, Vol. 69.
- Z. Li, T. Ozawa, I. Sohn, and D. A. Levin. 2011. "Modeling of electronic excitation and radiation in non-continuum hypersonic reentry flows," *Physics of Fluids*, Vol. 23.
- K. Miki, M. Panesi, E. Prudencio, A. Maurente, S. H. Cheung, J. Jagodzinski, D. Goldstein, S. Prudhomme, K. Schulz, C. Simmons, J. Strand, and P. Varghese. 2010. "On the (In)Validation of a Thermochemical Model with EAST Shock Tube Radiation Measurements," 48th AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper 2010-1557.
- K. Miki, M. Panesi, E. E. Prudencio, and S. Prudhomme. 2011. "Probabilistic Models and Uncertainty Quantification for the Ionization Reaction Rate of Atomic Nitrogen", 49th AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper 2011-0624.
- K. Miki, M. Panesi, E.E. Prudencio, and S. Prudhomme. 2012a. "Probabilistic models and uncertainty quantification for the ionization reaction rate of atomic Nitrogen," *Journal of Computational Physics*, Vol. 231, pp. 3871–3886.
- K. Miki, M. Panesi, E. E. Prudencio, and S. Prudhomme. 2012b. "Estimation of the nitrogen ionization reaction rate using electric arc shock tube data and Bayesian model analysis," *Physics of Plasmas*, Vol. 19.
- A. Mira. 2001. "On Metropolis-Hastings algorithms with delayed rejection," *Metron*, Vol. LIX, pp. 231–241.
- A. Mira. 2002. "Ordering and improving the performance of Monte Carlo Markov Chains," *Statistical Science*, Vol. 16, pp. 340–350.
- Y. Moon, B. Rajagopalan, and U. Lall. 1995. "Estimation of mutual information using kernel density estimators," *Physical Review E*, Vol. 52, No. 3, pp. 2318–2321.

- C. H. Moore, H. Deng, D. B. Goldstein, D. A. Levin, P. L. Varghese, L. M. Trafton, B. D. Stewart, and A. C. Walker. 2011. "Simulation of Plasma Interaction with Io's Atmosphere," in *Proceedings of the 27th International Symposium on Rarefied Gas Dynamics*, Pacific Grove, CA, edited by D. A. Levin, I. J. Wysong, and A. L. Garcia, AIP, Melville, NY.
- T. Ozawa, J. Zhong, and D. A. Levin. 2008. "Development of Kinetic-Based Energy Exchange Models for Noncontinuum, Ionized Hypersonic Flows," *Physics of Fluids*, Vol. 20.
- M. Panesi, K. Miki, S. Prudhomme, and A. Brandis. 2012. "On the assessment of a Bayesian validation methodology for data reduction models relevant to shock tube experiments," *Computer Methods in Applied Mechanics and Engineering*, pp. 383–398.
- M. Panesi, K. Miki, K. Schulz, E. Prudencio, and S. Prudhomme. 2011. "Calibration of Rates Parameters for Multi-Temperature Models using Bayesian Formulation," 42nd AIAA Thermophysics Conference, AIAA Paper 2011-3138.
- G. Pham-Van-Diep, D. Erwin, and E. P. Muntz. 1989. "Nonequilibrium Molecular Motion in a Hypersonic Shock Wave," *Science* 2, Vol. 245, No. 4918, pp. 624-626.
- E. E. Prudencio and S. H. Cheung. "Parallel adaptive multilevel sampling algorithms for the Bayesian analysis of mathematical models," *International Journal for Uncertainty Quantification*, Vol. 2, No. 3, pp. 215-237, 2012.
- E. E. Prudencio and K. W. Schulz. 2012. "The parallel C++ statistical library QUESO: Quantification of Uncertainty for Estimation, Simulation and Optimization," In Euro-Par 2011 Workshops, Part I, Volume 7155 of Lecture Notes in Computer Science, pp. 398-407, edited by M. Alexander *et al.*, Springer-Verlag, Berlin, Heidelberg.
- C. E. Rasmussen and C. K. I. Williams, 2006. *Gaussian Processes for Machine Learning*, The MIT Press.

- C. E. Shannon. 1948. "A mathematical theory of communication," *The Bell System Technical Journal*, Vol. 27, pp. 379–423, 623–656.
- K. A. Stephani, D. B. Goldstein, and P. L. Varghese. 2012. "Consistent Treatment of Transport Properties for Five-Species Air DSMC/Navier-Stokes Applications," accepted for publication in *Physics of Fluids*.
- R. Steuer, J. Kurths, C. O. Daub, J. Weise, and J. Selbig. 2002. "The mutual information: Detecting and evaluating dependencies between variables," *Bioinformatics*, Vol. 18, Suppl. 2, S231–S240.
- J. Strand and D. Goldstein. 2011. "Application of the MCMC Method for the Calibration of DSMC Parameters", in *Proceedings of the 27th International Symposium on Rarefied Gas Dynamics*, Pacific Grove, CA, edited by D. A. Levin, I. J. Wysong, and A. L. Garcia, AIP, Melville, NY.
- J. Strand and D. Goldstein. 2011a. "Sensitivity Analysis for DSMC Simulations of High-Temperature Air Chemistry," 49th AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper 2011-0535.
- J. Strand and D. Goldstein. 2011b. "Application of Bayesian Statistical Methods for the Analysis of DSMC Simulations of Hypersonic Shocks," 41st AIAA Fluid Dynamics Conference and Exhibit, AIAA Paper 2011-3705.
- L. Tierney. 1994. "Markov chains for exploring posterior distributions," *Annals of Statistics*, Vol. 22, pp. 1701–1762.
- L. Tierney and A. Mira. 1999. "Some adaptive Monte Carlo methods for Bayesian inference," *Statistics in Medicine*, Vol. 18, pp. 2507–2515.
- P. Valentini and T. E. Schwartzentruber. 2009. "Large-scale molecular dynamics simulations of normal shock waves in dilute argon," *Physics of Fluids*, Vol. 21.

- A. C. Walker, S. L. Gratiy, D. A. Levin, D. B. Goldstein, P. L. Varghese, L. M. Trafton, C. H. Moore, and B. Stewart. 2008. "Modeling Io's Sublimation-Driven Atmosphere: Gas Dynamics and Radiation Emission," in *Proceedings of the 26th International Symposium on Rarefied Gas Dynamics*, Kyoto, Japan, edited by T. Abe, AIP, Melville, NY.
- D. S. Watvisave, U. V. Bhandarkar, and B. P. Puranik. 2011. "Numerical Investigation of Shock Tube Flow under Rarefied Conditions," in *Proceedings of the 27th International Symposium on Rarefied Gas Dynamics*, Pacific Grove, CA, edited by D. A. Levin, I. J. Wysong, and A. L. Garcia, AIP, Melville, NY.
- E. E. Whiting, C. Park, Y. Liu, J. O. Arnold, and J. A. Paterson. 1996. "NEQAIR96, Nonequilibrium and Equilibrium Radiative Transport and Spectra Program: User's Manual", NASA Reference Publication 1389.