

**Surrogate Modelling of Combustion Systems for Robust
Optimized Scramjet Design**

by

Don Steve Gipe

B.A. Political Science, University of Colorado, 2004

A thesis submitted to the
Faculty of the Graduate School of the
University of Colorado in partial fulfillment
of the requirements for the degree of

BS/MS

Department of Aerospace Engineering Sciences

2012

This thesis entitled:
Surrogate Modelling of Combustion Systems for Robust Optimized Scramjet Design
written by Don Steve Gipe
has been approved for the Department of Aerospace Engineering Sciences

Ryan Starkey

Alireza Doostan

John Daily

Date _____

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Don Steve Gipe, (Aerospace Engineering Sciences)

Surrogate Modelling of Combustion Systems for Robust Optimized Scramjet Design

Thesis directed by Prof. Ryan Starkey

Substantial computational time to develop high speed propulsion systems is the number one challenge facing future development of thrust producing scramjets over a wide operational envelope. Current designs times are substantial due to the need to solve the Navier-Stokes equations using computational fluid dynamics. The computational time for one point can lead to only a few design choices being run before a decision is required. In order to develop robust design a level of understanding of off design performance is also required. If a single design point requires significant resources in computational time, solving the same geometry over a number of design parameters creates a cost too great for design. To mitigate this cost, Surrogate surface modelling can be used to generate a statistical function for the design space with a sparse data set to capture the optima points as well as how the system behaves during off design performance. This paper focuses on generating surrogate surfaces using the Kriging method, employing results for both the supersonic flow and the reaction mechanisms required to accurately predict the combustion performance which requires chemical kinetics to accurately capture the low residence time inherent in supersonic combustion.

Acknowledgements

I would like to thank my advisor Dr. Ryan Starkey for guiding me in my research and education. Also, my committee members Dr. John Daily and Dr. Alireza Doostan for their time and help with this thesis. The group from Busemann Advanced Concepts Lab for their support, especially Kevin Basore for his guidance and help when I got stuck. Finally I would like to thank Keith Dalbey from Sandia National Labs for his quick responses to issues I had with Surfpac.

Contents

Chapter

1	Introduction	1
1.1	Motivation	1
1.2	Surrogate Modelling	2
2	Surrogate Modelling Theory	5
2.1	Models	5
2.1.1	Instance Based Learning Methods	5
2.1.2	Machine Learning Methods	7
2.1.3	Statistical Learning Methods	10
2.2	Combustion	20
2.2.1	Comment on Numerical Methods and Stability	20
2.2.2	CHEMKIN	22
2.2.3	Reduced Order Reaction Mechanisms	24
2.3	The Surrogate Model Program	25
2.3.1	Matlab	26
2.3.2	Develop	26
2.3.3	Dakota	26
2.3.4	SurfPack	27

3	Algorithm and Flow Chart for Generating Input Files	29
3.1	Requirements	29
3.2	Generating the files	29
3.3	Generated Results Comparisons	30
3.3.1	One Dimensional	30
3.3.2	Two Dimensional	33
3.3.3	Four Dimensional	35
4	Sampling Procedures for Optimizing Design	38
4.1	One Dimensional Optimization and Methods	39
4.2	Two Dimensional Optimization and Methods	43
4.3	N Dimensional Optimization and Methods	48
5	Surrogate Reaction Systems for Scramjet Design	53
5.1	Supersonic Flow Combustion for Various Upstream Conditions	53
5.2	Supersonic Combustor Geometry Design	63
5.3	Uncertainty Quantification	72
6	Conclusions	76
	Bibliography	78

Tables

Table

2.1	Variable Definition of CHEMKIN Functions Used	23
4.1	Equation 4.1 Variable Definition	49
5.1	Combustor Inlet Design Variables	53
5.2	Combustor Design Variables	54
5.3	Results for 4 Test Locations from Original 14 Point Altitude Parameter Surrogate .	57
5.4	Results for 4 Test Locations from updated 14 Point Altitude Parameter Surrogate with 4 Additional Points	59
5.5	Results for 4 Test Locations from Original 14 Point Engine Parameters Surrogate . .	63
5.6	Results for 4 Test Locations from 14 Point Engine Parameters Surrogate with 4 Additional Points	65
5.7	Results for 45 Additional Test Locations from Original 14 Point Engine Parameters Surrogate with 4 Additional Points	69
5.8	Average, Maximum, and Minimum, Results for 45 Additional Test Locations from Original 14 Point Engine Parameters Surrogate with 4 Additional Points	70
5.9	Results for 4 Test Locations from 14 Point Uncertainty Parameters Surrogate. Vari- ables E_a , A_i , β_i give the Uncertain Change from Initial CHEMKIN Input Files . . .	72

Figures

Figure

2.1	Diagram for Radial Bases Functions from reference [1]	7
2.2	Genetic Algorithm Structure [2]	8
2.3	Logic Flow Diagram for Genetic Algorithm	9
2.4	Example function of the form of equation 2.2	11
2.5	Example function of the form of equation 2.2 with regression line formed from equations 2.6 through 2.11	13
2.6	Correlations with Varying \mathbf{P} [3]	15
2.7	Kriging Interpolation with Noisy Data (Noise Data Generated from Reference [3]) .	17
2.8	MARS with Noisy Data (Noise Data Generated from Reference [3])	17
2.9	Kriging Interpolation fitting a Sine Wave with 5 Points	18
2.10	MARS fitting a Sine Wave with 10 Points	18
2.11	Kriging Interpolation fitting Experimental Combustion Data with 28 Points Data from Reference [4]	19
2.12	MARS fitting Experimental Combustion Data with 28 Points Data from Reference [4]	19
2.13	Example of How Reduced Order Mechanisms Effect Combustion [5]	25
3.1	Matlab Routine Block Diagram for Generating .spk input files	30
3.2	Generated .spk input file for 1-D from Author's code	31
3.3	Comparative Results between Reference [6] and Author's File Generation	32

3.4	Results from Reference [6]	32
3.5	Generated .spk input file for 2-D from Author's code	33
3.6	Matlab's Results for 'Peaks' Generation	34
3.7	Author's Results for Matlab's 'Peaks' Generation	34
3.8	Reference [6] Results for 'Peaks' Generation	35
3.9	Generated .spk input file for 4-D Engineering Analysis Example from Author's code	36
3.10	Original Points Used to Develop Regression Surfaces	36
3.11	Author's Results for the 4-D Engineering Analysis Example	37
3.12	Reference [6] Results for the 4-D Engineering Analysis Example	37
4.1	Three to Seven Point Kriging Solution for $f(x) = (6x - 2)^2 \sin(12x - 4)$	40
4.2	Twelve (Left) and Twenty Three (Right) Uniform Sampling Point Kriging Solution for the Experimental Results from reference [4]	41
4.3	Specific Sampling Point Kriging Solution for the Experimental Results from Refer- ence [4]	42
4.4	Figure 11 from Reference [4], Nondimensional Pressure Distributions Along Tube all for Non Reacting Conditions	42
4.5	Seven to Eleven points sampled using gradient based optimization, Kriging Solution for the Experimental Results from reference [4]	43
4.6	Plot of the function $f(x_1, x_2) = 21.5 + x_1 \sin(4\pi x_1) + x_2 \sin(20\pi x_2)$	45
4.7	Twenty Five Point Optimization of the function $f(x_1, x_2) = 21.5 + x_1 \sin(4\pi x_1) +$ $x_2 \sin(20\pi x_2)$ Point Selection Left, Local Optimization Right	46
4.8	Twenty Five Point Optimization of the function $f(x_1, x_2) = 21.5 + x_1 \sin(4\pi x_1) +$ $x_2 \sin(20\pi x_2)$ Point Selection Left, Local Optimization Right	46
4.9	Twenty Five Point Optimization of the function $f(x_1, x_2) = 21.5 + x_1 \sin(4\pi x_1) +$ $x_2 \sin(20\pi x_2)$ Point Selection Left, Local Optimization Right	47

4.10	144 Point Optimization (combination of previous three figures) of the function $f(x_1, x_2) = 21.5 + x_1 \sin(4\pi x_1) + x_2 \sin(20\pi x_2)$ Point Selection Left, Local Optimization Right .	47
4.11	Tile plot from reference [3] from equation 4.1. Each tile varies two variables while holding the eight other variables at baseline values as per table 4.3	50
4.12	Tile plot of surrogate developed from six independent variables using 14 discrete points from equation 4.1. Each tile varies two variables while holding the eight other variables at baseline values as per table 4.3. Also shown are the chosen discrete points.	51
4.13	Tile plot of surrogate developed from the remaining four independent variables using 14 discrete points from equation 4.1. Each tile varies two variables while holding the eight other variables at baseline values as per table 4.3. Also shown are the chosen discrete points.	52
5.1	Tile plot of exit Mach number surrogate developed from variables in table 5.1 with values from table 5.2 held constant. Also shown are the 14 randomly chosen discrete points.	55
5.2	Tile plot of exit carbon dioxide emission surrogate developed from variables in table 5.1 with values from table 5.2 held constant. Also shown are the 14 randomly chosen discrete points.	55
5.3	Tile plot of exit total temperature surrogate developed from variables in table 5.1 with values from table 5.2 held constant. Also shown are the 14 randomly chosen discrete points.	56
5.4	Tile plot of exit total pressure inlet to exit ratio surrogate developed from variables in table 5.1 with values from table 5.2 held constant. Also shown are the 14 randomly chosen discrete points.	56

5.5	Tile plot of exit Mach number surrogate developed from variables in table 5.1 with values from table 5.2 held constant. Also shown are the 14 randomly chosen discrete points (stars) and the added 4 test locations (circles).	57
5.6	Tile plot of exit carbon dioxide emission surrogate developed from variables in table 5.1 with values from table 5.2 held constant. Also shown are the 14 randomly chosen discrete points (stars) and the added 4 test locations (circles).	58
5.7	Tile plot of exit total temperature surrogate developed from variables in table 5.1 with values from table 5.2 held constant. Also shown are the 14 randomly chosen discrete points (stars) and the added 4 test locations (circles).	59
5.8	Tile plot of exit total pressure inlet to exit ratio surrogate developed from variables in table 5.1 with values from table 5.2 held constant. Also shown are the 14 randomly chosen discrete points (stars) and the added 4 test locations (circles).	60
5.9	Tile plot of exit Mach number surrogate developed from variables in table 5.1 with values from table 5.2 held constant. Also shown are the 14 randomly chosen and 4 evaluated for previous testing discrete points (stars) and the added 4 test locations (circles).	60
5.10	Tile plot of exit carbon dioxide emission surrogate developed from variables in table 5.1 with values from table 5.2 held constant. Also shown are the 14 randomly chosen and 4 evaluated for previous testing discrete points (stars) and the added 4 test locations (circles).	61
5.11	Tile plot of exit total temperature surrogate developed from variables in table 5.1 with values from table 5.2 held constant. Also shown are the 14 randomly chosen and 4 evaluated for previous testing discrete points (stars) and the added 4 test locations (circles).	61

5.12	Tile plot of exit total pressure inlet to exit ratio surrogate developed from variables in table 5.1 with values from table 5.2 held constant. Also shown are the 14 randomly chosen and 4 evaluated for previous testing discrete points (stars) and the added 4 test locations (circles).	62
5.13	Tile plot of exit Mach number surrogate developed from variables in table 5.2 with values from table 5.1 held constant. Also shown are the 14 randomly chosen points (stars) and evaluation points (circles) from table 5.5.	64
5.14	Tile plot of exit carbon dioxide emission surrogate developed from variables in table 5.2 with values from table 5.1 held constant. Also shown are the 14 randomly chosen points (stars) and evaluation points (circles) from table 5.5.	65
5.15	Tile plot of exit total temperature surrogate developed from variables in table 5.2 with values from table 5.1 held constant. Also shown are the 14 randomly chosen points (stars) and evaluation points (circles) from table 5.5.	66
5.16	Tile plot of exit total pressure inlet to exit ratio surrogate developed from variables in table 5.2 with values from table 5.1 held constant. Also shown are the 14 randomly chosen points (stars) and evaluation points (circles) from table 5.5.	66
5.17	Tile plot of exit Mach number surrogate developed from variables in table 5.2 with values from table 5.1 held constant. Also shown are the 14 randomly chosen points and evaluation points (stars) from table 5.6.	67
5.18	Tile plot of exit carbon dioxide emission surrogate developed from variables in table 5.2 with values from table 5.1 held constant. Also shown are the 14 randomly chosen points and evaluation points (stars) from table 5.6.	67
5.19	Tile plot of exit total temperature surrogate developed from variables in table 5.2 with values from table 5.1 held constant. Also shown are the 14 randomly chosen points and evaluation points (stars) from table 5.6.	68

5.20	Tile plot of exit total pressure inlet to exit ratio surrogate developed from variables in table 5.2 with values from table 5.1 held constant. Also shown are the 14 randomly chosen points and evaluation points (stars) from table 5.6.	68
5.21	Logic Flow Diagram for Genetic Algorithm	71
5.22	Tile plot of exit Mach number surrogate developed from variables in table 5.1 with values from table 5.2 held constant along with the uncertainty in the reaction parameters. Also shown are the 14 chosen points(stars) and evaluation points (circles) from table 5.9.	73
5.23	Tile plot of exit carbon dioxide emission surrogate developed from variables in table 5.1 with values from table 5.2 held constant along with the uncertainty in the reaction parameters. Also shown are the 14 chosen points(stars) and evaluation points (circles) from table 5.9.	73
5.24	Tile plot of exit total temperature surrogate developed from variables in table 5.1 with values from table 5.2 held constant along with the uncertainty in the reaction parameters. Also shown are the 14 chosen points(stars) and evaluation points (circles) from table 5.9.	74
5.25	Tile plot of exit total pressure inlet to exit ratio surrogate developed from variables in table 5.1 with values from table 5.2 held constant along with the uncertainty in the reaction parameters. Also shown are the 14 chosen points(stars) and evaluation points (circles) from table 5.9.	75

Chapter 1

Introduction

1.1 Motivation

The future of high speed flight is currently limited by the cost of CFD design, building flight capable models, and pulling quality experimental data. In recent years the budget for scramjets has declined as public belief has become disquieted over failures such as the aerospace plane. As engineers and scientists develop budgets for systems that require seamless integration where small steps outside the design envelope lead to massive budget increases, high speed flight takes to the skies slower and slower. Improving the design process in efficient ways is the first natural step to changing overall perception and gaining the required steady budget to develop the next stage of flight.

In order to develop a robust and reliability based design, "to be insensitive to small changes in uncertain quantities, and some low acceptable probability of failure" engineers must develop efficient ways to accurately characterize optimal designs but also performance outside the design envelope [7]. Hypersonics is defined to be roughly around the region of Mach 5-10 [8]. However, the only way to get scramjets to that speed is with a rocket, and accelerating has only been achieved from 4.73 to just over Mach 5 [9]. Thus the current operational envelop is only at the beginning of the hypersonic region. In order to develop a scramjet that can accelerate, off design performance must be quantified and designed to before another expensive scramjet is designed and built, the x-51 project has costed \$200 million dollars [10].

Hypersonic air breathing flight is fraught with difficulties. The high speeds result in severe

thermal and structural loads along with strong coupling of in aerodynamic, propulsion, structural, and control systems [11]. Difficulties that arise from this is structural fatigue, aero-propulsion integration, aircraft controls, material selection, and thermal management [11]. All these aspects require significant computational analysis including multi-physics due to the inherit difficulties in experimentation at those speeds and altitude [11].

1.2 Surrogate Modelling

In order to predict off design performance and develop a robust design trade studies are required, and are limited in both time and capital [12]. Due to high computational time for fluid dynamic and combustion problems doing these trade studies through CFD can have enormous computational cost. Also, to create a test scramjet requires even more money as seen with the x-51 project. Not knowing how off design performance effects your mission due to high cost of development will ground the program before it ever gets a chance to fly. For example, the national aerospace plane was designed to go from Mach 0-30 in a one stage to orbit flyer that was cancelled when the total required cost moved to 15 billion dollars [13]. A lot of recent research has been in the development of statistical type modelling methods to replace the actual function [3]. Many methods have been developed, but for the need of hypersonic scramjet design a model that only requires a few discrete data points is necessary [14]. Kriging has been shown to be an excellent surrogate function due to not needing a priori understanding of the solution and many studies have shown how surrogates have matched CFD accurately with one thousandth of the computational expense [11]. One study used CFD to measure wing loading due to unsteady aerodynamic pressure that with 100 samples the surrogate matched the solution to the full unsteady Navier-Stokes solutions evaluated at a different 500 points to be within 1 - 20 % accuracy, when the number of samples was increased to 500 the maximum error dropped by a factor of 2 and at 1500 the maximum error was 6 % [11]. Along with using surrogates to help predict CFD results, it can be used to improve uncertainty quantification allowing better understanding of the solution space and a more robust design [15]. Once a surrogate is built the actual design point requires further analysis since the

chosen independent parameters may match within a specific tolerance but often the optimized point is often less conservative than the actual results [16].

Using surrogate methods to accurately characterize uncertainty in low fidelity chemistry models like those used in the Sandia Labs Surpack. Faster and better characterized design processes for high speed flight can be accurately characterized using surrogate models to capture a high speed computational design algorithms. At sufficiently high speeds, the residence times in the combustor are in the fractions of seconds, and at high altitudes and dynamic pressures, disassociation and non-ideal gas effects dominate. With these difficulties, high fidelity chemistry models can take many weeks to develop a solution, which may or may not be accurate. Resolving the incongruities of experiment, multiple models, and actual flight-testing can be extremely costly and time consuming. Understanding the uncertainty in each of these models will benefit the aerospace community in specific, and all engineering in general. By combining sparse high fidelity CFD or experimental data into an inclusive surrogate model will allow understanding of the total design space while still maintaining an overall understanding of the accuracy of the model through independent tests of the surrogate. The designed surrogate function will allow the designer the knowledge to characterize what piece of the model is most likely to lead to designs that are robust and reliable, thus most likely to achieve the optimum design point using as little expensive information as possible significantly decreasing the design time of high cost systems.

This paper goes develops the methods and models to quickly take expensive CFD and experimental data to form a design space over a number of input and output variables through the use of surrogate modelling using kriging interpolation. Chapter two outlines the theory and uses behind surrogate modelling, as well as the mathematical bases for kriging interpolation, combustion, a commonly used combustion tool, CHEMKIN. and the different options of surrogate modelling tools as well as how to compile and run Sandia Labs' Surpack, the tool used in this paper. Chapter three works through one, two and four dimensional examples and compares them with a reference paper from the developers of Surpack to confirm the methods and models outlined in chapter three work. Chapter four develops the methods used for design when using surrogate models and takes

the reader through a series of proofs to show that single to multidimensional surrogate modelling is extremely effective given only a few high cost points. Chapter five combines the knowledge from the previous chapters to develop and evaluate a design space using surrogate models and how off design performance effects the system.

Chapter 2

Surrogate Modelling Theory

2.1 Models

A number of regression models exist and to cover them all would go beyond the scope of this paper. However, a general background is required as well as an introduction into some of the more commonly used methods. Furthermore, an understanding of their general strengths and weaknesses is required before delving more deeply into generating surrogate models. Surrogate models can be broken up into three basic categories: instance-based learning methods, machine learning methods, and statistical learning methods [14]. Each one of these methods will be generalized with some specific methods explained.

2.1.1 Instance Based Learning Methods

Instance based learning methods refer to objective function that learns by example where each instance is a dimension of independent and dependent variables [17]. The independent variables are category attributes and the dependent is the predictor tribute [17]. The category attributes are assumed to be linearly independent allowing instance based learning methods the ability to predict unordered results when using ordered category attributes [17]. Thus these methods allow regression between two known points using a set of training data to make estimations of future predictor values based on new category attributes. For example, Niagara Mohawk Power Company of central New York State uses an instance based learning method for predicting power load to achieve the same level as experts working for two hours in two minutes [17]. A common instance

based learning method is the radial basis function model and will be described in more detail in order to better understand the benefits and costs associated with instance based learning [14].

Radial basis functions are a linear combination of a number of weights that can be applied to the category attributes. Most often this process is explained using a figure nodes and inter-connecting lines. Figure 2.1 is an example of such a figure from reference [1]. The x values are the independent variables that are the example set the method trains on, i.e. the examples. It is assumed that the outputs are related to the inputs through some weighted linear combination. The lines going from each independent variable to each hidden variable and subsequently to each dependent variable shows how each x value has a effect on each y value. The magnitude of the effect is completely dependent on the weight of the hidden variable. These weights are referred to as the hidden units and is a basis function that relates the independent variable and the center of the function from which the distance can be measured. Regular basis functions are normally considered some linear combination of polynomials and can have coordinates in some vector space [18]. The radial describes the idea of the distance from the center. Equation 2.1 gives the numerical formulation of radial basis function. The w values are the unknown or hidden weights that can allow the mapping from the independent variables to the dependent variables. The $\phi_i(||x - c_i||)$ term represents the distance from the center and can be modelled using a number of different linear splines, cubic splines, multi-quadratics, thin-plate splines, and Gaussian kernels [14]. This method is very useful for hand-written digit recognition, medical image screening and fingerprint identification [1]. This method is very similar to a number of machine learning methods such as artificial neural networks and feed forward models. The next section will explain the differences.

$$y(x) = w_0 + \sum_{i=1}^k w_i \phi_i(||x - c_i||) \quad (2.1)$$

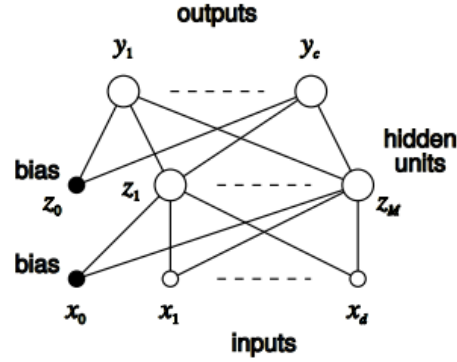


Figure 2.1: Diagram for Radial Bases Functions from reference [1]

Although instance based learning methods are excellent at pattern recognition even with multiple independent variables, they have many weaknesses. In general instance based learning is computationally expensive during training, is intolerant to noise, fails when independent variables show linear dependence on one another, there is no way to deal with missing data points, and they provide little usable information regarding the data structure [17].

2.1.2 Machine Learning Methods

Machine learning methods work by taking a number of input variables and training on, or learning from, the data. The methods seem to fall into two categories, decision trees and neural networks [19]. Decision trees are forward propagating information very similar to instance based learning methods, developing a system of weighted parameters that can be applied to the independent variables resulting in an output [14]. The major variation with the decision trees is that a series of if/then statements also appear in the network resulting in a different weight set from similar parameters depending on the case [20]. Neural networks work the same except the decision trees go through a complex interaction of branches using back propagation [2]. There are too many methods even to make a comprehensive list of machine learning methods so the author chose to only examine genetic algorithms due to their ability to handle very complex data sets.

The following information comes from the book by Michalewicz, reference [2]. Genetic algorithms were discovered by combining the theory of evolution with machine learning. The idea was to have a sample population improve through Darwin's theory of evolution. In a population the animals with improved superlatives survive; faster, stronger, smarter. Where those that are not the best survive through luck. Genetic algorithms conceptually work the same way. Given a sample population, x_i^n where x is a vector space of discrete sample points and i is the number of independent variables and $n = 0$ for initial conditions, the genetic algorithm looks at the neighbours of the original sample points. If the neighbour is 'stronger, faster, or smarter', i.e. is closer to an optimum point, that neighbour becomes the x_i^{n+1} point. This process continues until the solution converges and no change in optima is found. Each sample point finds the local optima using the neighbour or hill climbing method, and since a number of sample points across the domain space exist, the global optima can be found. Michalewicz explains the process best using pseudo code, and is given in figure 2.2. A block diagram is given in figure 2.3 to similarly explain the learning process. To continue with the evolution example; the error associated with those random samples that even though they are not the best but still survives does so through luck [2].

```

1  Genetic Algorithm
2
3  begin
4  Time = initial time
5  P = initial sample space
6  evaluate P
7
8      while (P(t+1) is not equal to P(t) within some bounds) do
9      begin
10         t = t + 1
11         P(t+1) = close neighbours of (p)
12         evaluate P(t+1)
13     end
14 end

```

Figure 2.2: Genetic Algorithm Structure [2]

The genetic algorithm is a useful example of machine learning. While it is not an exhaustive explanation it does help to illustrate the concept of how a machine can learn. Combining this with

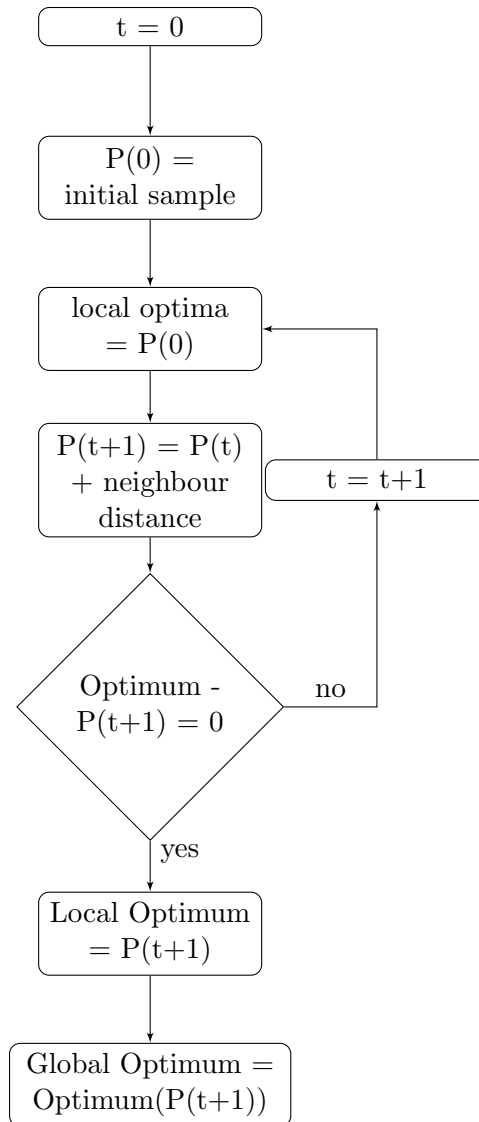


Figure 2.3: Logic Flow Diagram for Genetic Algorithm

decision trees and back propagation it can be seen that many different types exist. It is also a very robust method of choosing an optima with many successful examples [2, 3].

2.1.3 Statistical Learning Methods

Statistical methods are built using basically the regression methods as and approximate the functions that fit the data [14]. The idea of surrogate models comes from the idea that these models can replace the original data or functions as a means of faster operations. For developing surrogate models, polynomial, kriging and support vector machines are the most popular [14].

Regression is well known to engineers and to anyone who works in data analysis in its basic form as line fitting. In order to understand the more complicated nature and different regression schemes an understanding of simple linear regression and multiple variable linear regression is necessary. The goal of this section is to educate a reader who has little to no understanding of the methods behind regression to allow for a more complete understanding of the rest of the thesis.

As previously mentioned, the simplest form of regression is basically developing equations of a line. However many forms of line fitting follow a deterministic model. A deterministic model is one such that matches each point exactly, not allowing for any error, as seen in interpolation [21]. These models work well only when there is no error in the response of the dependent variable. This sort of situation rarely occurs in engineering applications. For one, experimental results are dominated by many sources of systematic and random error [22]. Due to this, even the most linear relationship will have deviations from the line. In order to account for deviations probabilistic models are used. Equation 2.2 [21] gives a linear equation that accounts for the error in the line, ϵ , with β_0 equal to the y-intercept and β_1 equal to the slope of the line. Figure 2.4 gives an example of data points that fit the criteria for equation 2.2 with β_0 , β_1 , and ϵ , equal to seven, three, and a random number between one and ten.

$$y = \beta_0 + \beta_1 x + \epsilon \tag{2.2}$$

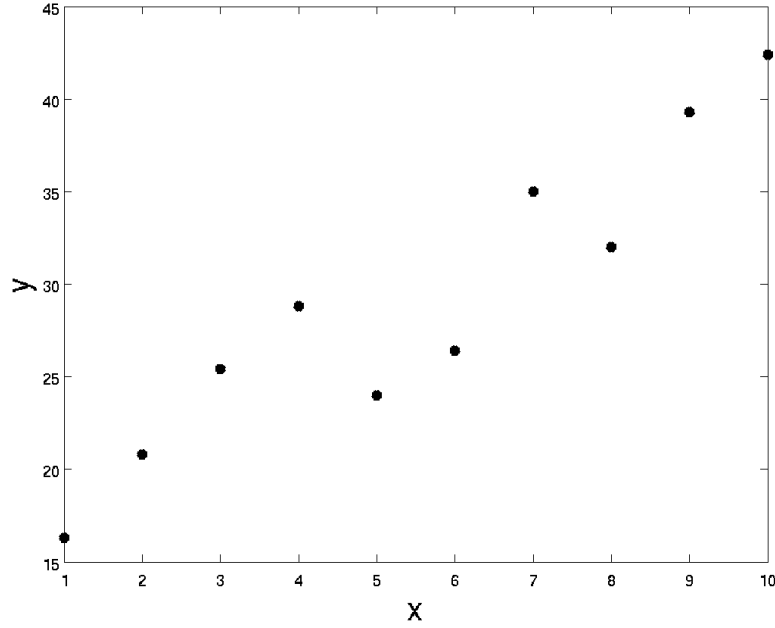


Figure 2.4: Example function of the form of equation 2.2

Any number of lines can be created to fit the the data from figure 2.4. In order to constrain the data set another equation is necessary. There are multiple options including maximum likelihood and method of least squares [21]. Since in the author's opinion the method of least squares seems to be the most common and appears so frequently in regression texts it will be the only method covered in this section.

The idea behind the method of least squares is to minimize the the squared error term. By rearranging equation 2.2 and solving for the error term then squaring and summing gives equation 2.3, where E is the total error.

$$E^2 = \sum (y_i - \beta_0 - \beta_1 x_i)^2 \quad (2.3)$$

Minimizing the error is done by taking the partial derivative with respect to the beta terms and setting them equal to zero as shown in equations 2.4 and 2.5

$$\frac{\partial E^2}{\partial \beta_0} = -2 \sum (y_i - \beta_0 - \beta_1 x_i) = 0 \quad (2.4)$$

$$\frac{\partial E^2}{\partial \beta_1} = -2x_i \sum (y_i - \beta_0 - \beta_1 x_i) = 0 \quad (2.5)$$

Solving equations 2.4 and 2.5 for β_0 and β_1 gives equations 2.6 through 2.11 [21] where n is the sample size.

$$\beta_1 = \frac{SS_{xy}}{SS_{xx}} \quad (2.6)$$

$$\beta_0 = \bar{y} - \beta_1 \bar{x} \quad (2.7)$$

$$SS_{xy} = \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) \quad (2.8)$$

$$SS_{xx} = \sum_{i=1}^n (x_i - \bar{x})^2 \quad (2.9)$$

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (2.10)$$

$$\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i \quad (2.11)$$

Solving these equations for the data given in figure 2.4 gives the regression line in figure 2.5.

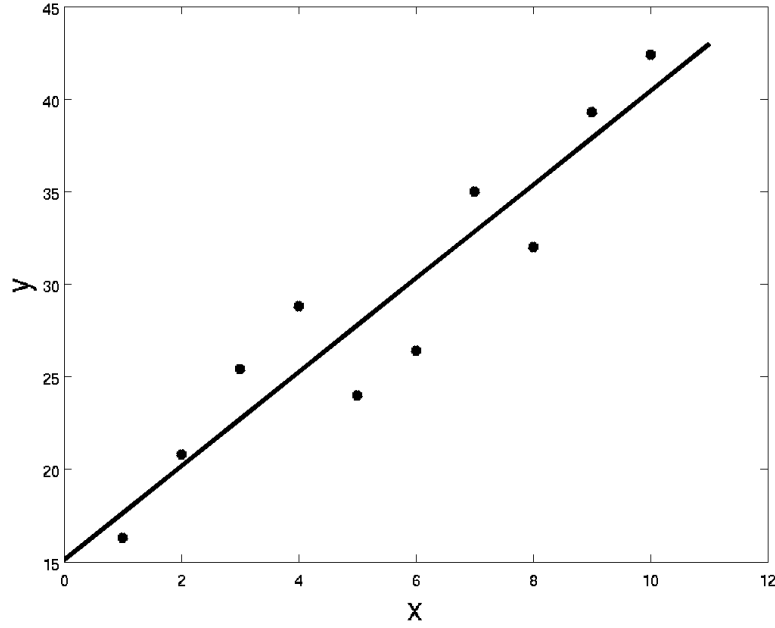


Figure 2.5: Example function of the form of equation 2.2 with regression line formed from equations 2.6 through 2.11

Equation 2.2 can be expanded to form different types of regression surfaces. For example equations 2.12, 2.13, 2.14, and 2.15 give respective solutions for a quadratic model, two dimensional model, second order linear model, and a non linear model respectively [21]. By squaring and summing these equations with the added error term then taking the partial derivatives and setting to zero as done in equations 2.3 to 2.5, will result in a system of equations of the size of the number of β terms that can then be solved using linear algebra. If the equation is of the form $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots \beta_k x_k + \epsilon$ the system can be solved as per equation 2.16 [21]. Bold signifies a matrix, including nx1 matrices.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_1^2 + \epsilon \quad (2.12)$$

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon \quad (2.13)$$

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \beta_4 x_1^2 + \beta_5 x_2^2 + \epsilon \quad (2.14)$$

$$y = \beta_0 e^{-\beta_1 x} + \epsilon \quad (2.15)$$

$$\hat{\beta} = \mathbf{X}^T \mathbf{X}^{-1} \mathbf{X}^T \mathbf{Y} \quad (2.16)$$

Where \mathbf{X} is an $n \times m$ matrix with n being the sample size and m the number of β variables. \mathbf{Y} are the independent variables and is an $n \times 1$ array. $\hat{\beta}$ is of the size $m \times 1$.

There are two major surrogate models that are especially critical to the topics of optimizing designs with reacting flow and high speed aerodynamics. Since each sample point is expensive computationally, when using CFD, or experimentally, a system of sparse discrete points is most likely going to be the initial sample space in which we build a surrogate [3]. Kriging approximates complex surfaces with high accuracy by combining trend mechanisms associated with regression and spatial correlation properties of kernel-based approximation similar to radial basis methods [23]. However, when data becomes much more populated and there is some statistical error which is common in CFD, kriging does a poor job of capturing the overall function [3, 23]. When data is closely populated together more regression is required, in which case kriging regression instead of interpolation can be applied or Multivariate Adaptive Regression Splines (MARS).

Kriging stems from radial bases functions and just modifies the $\phi_i (||x - c_i||)$ term in equation 2.1. w_0 is still a constant and is determined by the equation 2.17 [3]. \mathbf{U} is the Cholesky factorization ψ , equation 2.18, [3, 14]. θ_j can be found most accurately by using a genetic algorithm and most quickly using a concentrated in-likelihood function or generalized least squares [3, 14]. The p value can be most easily explained with a plot which is given in figure 2.6. From looking at figure 2.6, it can be seen that the p parameter effects smoothness of the function thus giving a greater control of the kriging interpolation.

$$w_0 = \frac{\mathbf{1}'\mathbf{U}^{-1}((\mathbf{U}')^{-1}\mathbf{y})}{\mathbf{1}'\mathbf{U}^{-1}((\mathbf{U}')^{-1}\mathbf{1})} \quad (2.17)$$

$$\psi' = \exp\left(-\sum_{j=1}^k \theta_j |x_j^{(i)} - x_j|^{p_j}\right), \quad (2.18)$$

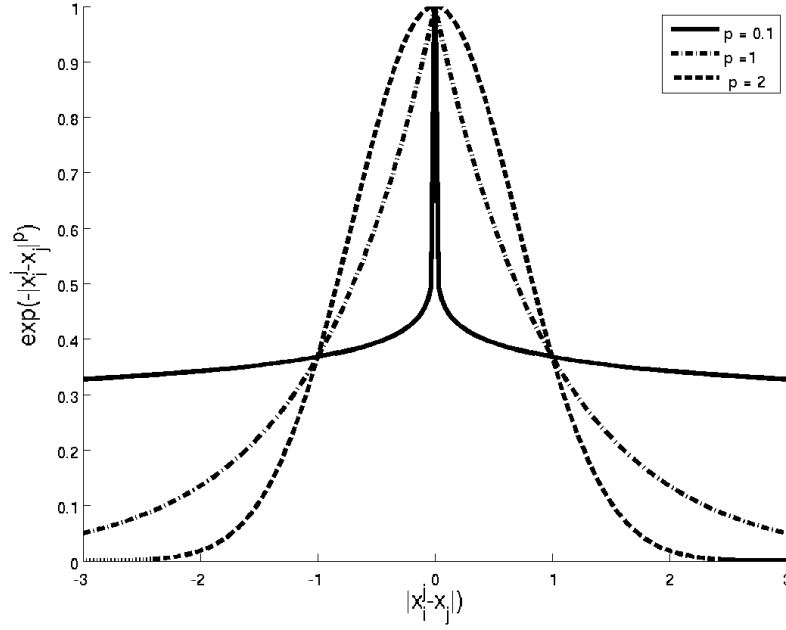


Figure 2.6: Correlations with Varying \mathbf{P} [3]

MARS works similar to the machine learning and statistical methods. MARS generates basis functions that can be added or removed to better approximate the data by creating subregions applying a forward/backward selection process [24]. The general form of the method is given in equation 2.19. The B terms are the basis functions and the a terms are the coefficients and the M_n parameters are for each independent variable [24, 25]. More information on MARS can be found in reference [25].

$$f(x) = a_0 + \sum_{m=1}^{M_1} a_m B_m(x_i) + \sum_{m=1}^{M_2} a_m B_m(x_i, x_j) + \dots \quad (2.19)$$

In order to better understand the differences between MARS and kriging interpolation figures 2.7 and 2.8 are shown for kriging and MARS respectively. It can be seen that the kriging method goes through every point since it is an interpolation technique, where MARS smooths out the function due to the regression. Figures 2.9 and 2.10 give the results for a basic sine wave. From this simple comparison it shows that kriging requires very few points to accurately capture a sine wave, 5 points, to get for an absolute difference from the exact function to be order of 10^{-4} , where with double the points MARS gives a straight line. A final comparison between the benefits of MARS and kriging, or regression vs. interpolation, is given in figures 2.11 and 2.12, for kriging and MARS respectively. The kriging version is orders of magnitude off from the actual results due to a process of over sampling which causes kriging significant errors [23]. This plot can be well approximated using methods outlined in the sampling procedures for optimal design chapter. However, in the case when kriging becomes ill conditioned MARS is able to produce a function that better approximates the data. Kriging captured the maxima within 2.5 % in 11 well placed points where MARS used 28 and missed the maximum by 29 %. The main reason for these differences is that kriging allows for discontinuous derivatives creating difficult functions with only a few points, and MARS is a type of spline which has continuous first derivatives. Both methods have their strengths and weaknesses and depending on the type of data an appropriate method must be chosen.

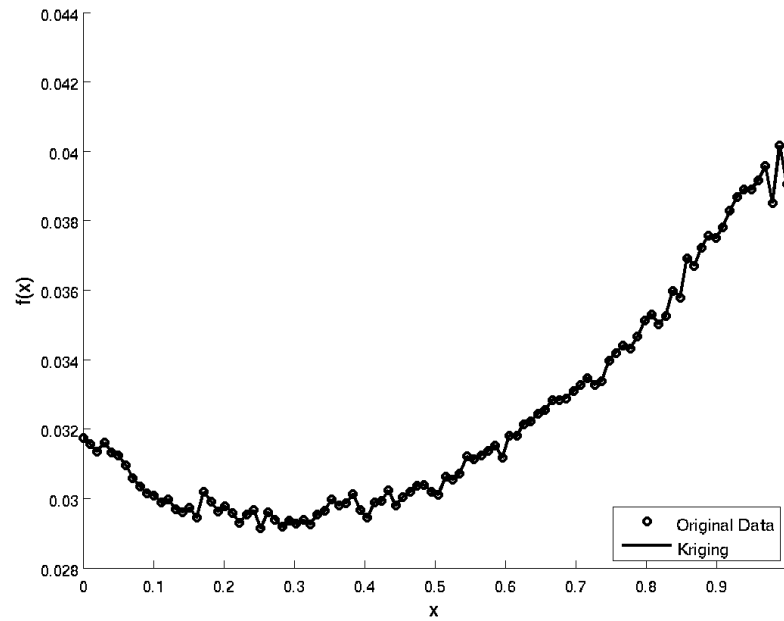


Figure 2.7: Kriging Interpolation with Noisy Data (Noise Data Generated from Reference [3])

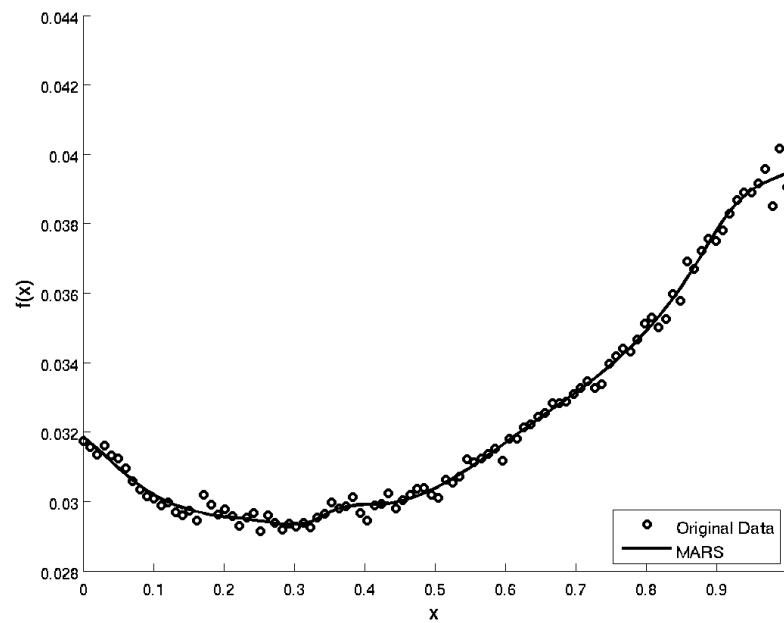


Figure 2.8: MARS with Noisy Data (Noise Data Generated from Reference [3])

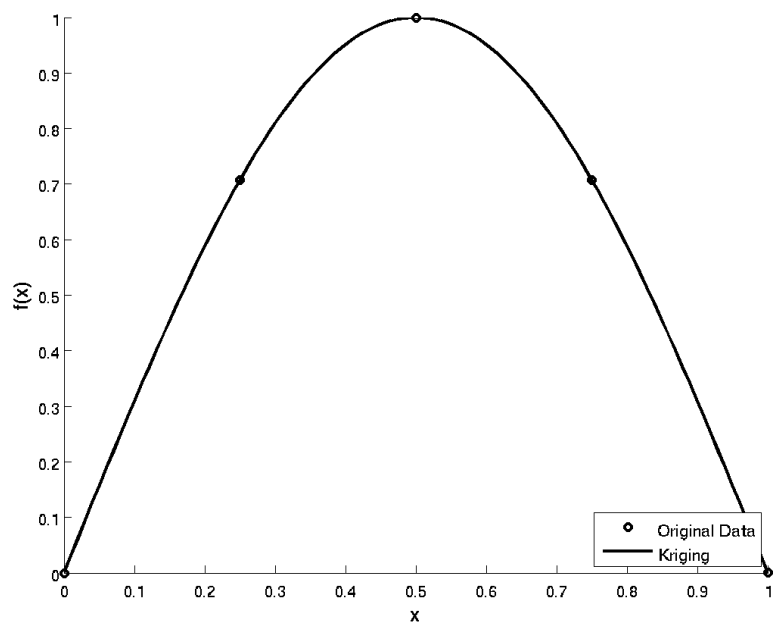


Figure 2.9: Kriging Interpolation fitting a Sine Wave with 5 Points

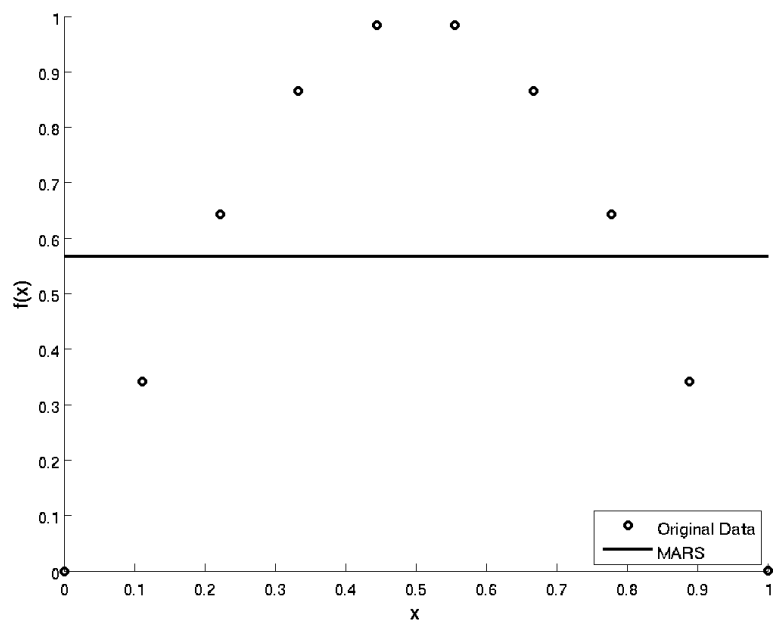


Figure 2.10: MARS fitting a Sine Wave with 10 Points

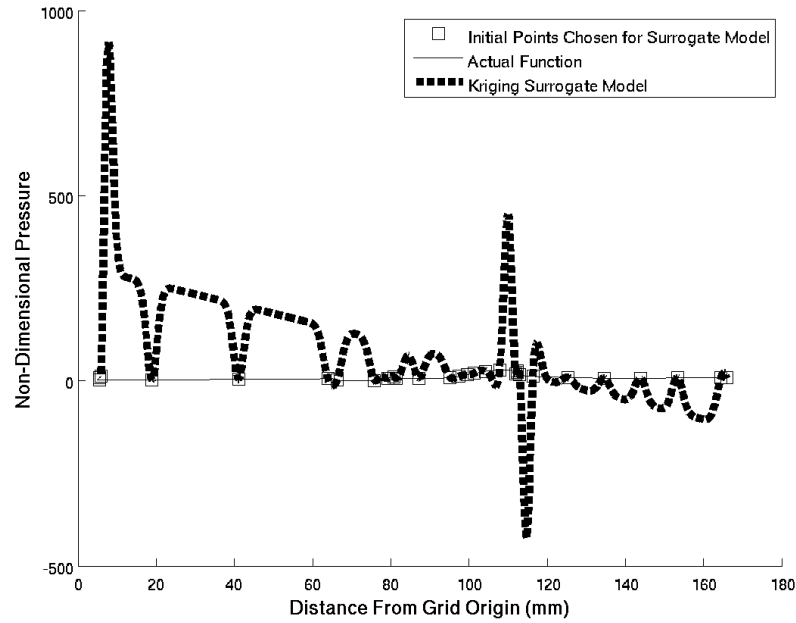


Figure 2.11: Kriging Interpolation fitting Experimental Combustion Data with 28 Points Data from Reference [4]

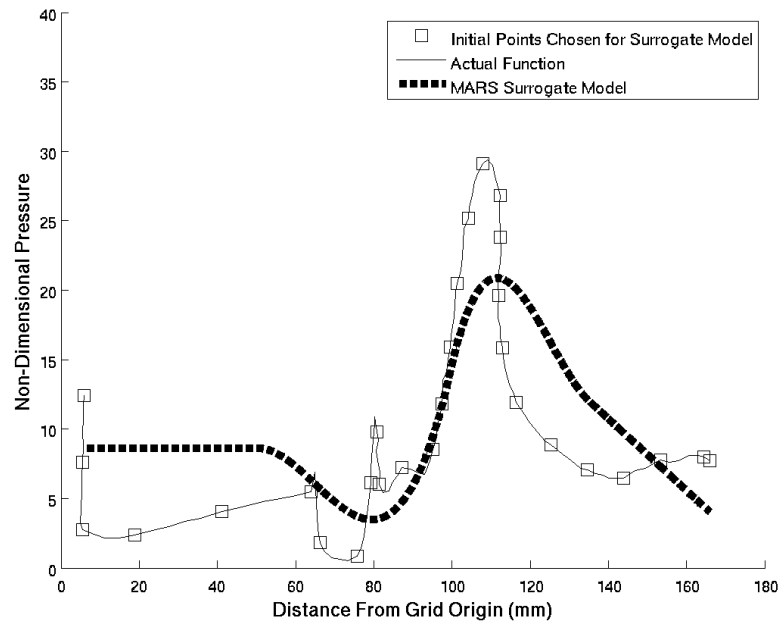


Figure 2.12: MARS fitting Experimental Combustion Data with 28 Points Data from Reference [4]

2.2 Combustion

Solving for combustion dynamics is a very complicated process due to the large variance in the properties of reacting species. When only the beginning state is known and the end state is the desired outcome chemical thermodynamics can be applied. However, in the case of combustion that steady state solution is not helpful and the rate of combustion and the creation of radicals is of utmost interest [26]. To determine the results of time dependent functions chemical kinetics is required. Solving chemical kinetics equations result in a stiff equation set that is difficult and computationally expensive to solve due to the wide variety of species growing and decaying at substantially different time scales [26, 27]. In order to integrate the ordinary differential equation a time step of sufficiently small values is required to prevent the solution from becoming unstable [27]. Thus the numerical solution time constant is dominated by the species that have the fastest rate [26].

2.2.1 Comment on Numerical Methods and Stability

Euler's method can be applied in order to determine the stability of the chemical kinetics ordinary differential equations, ODE, as seen in equation 2.20, where k is the specific reaction rate constant [26]. Equation 2.20 is a first order decomposition reaction of species A colliding with M to form products [26]. Each species has a time dependent concentration with the total concentration equal to 1. Thus only one ODE needs to be solved for two concentrations. This equation can easily be solved using analytical separation of variables and is given in equation 2.21, which shows that the time constant for the concentration of A to decay is $(kC_M t)^{-1}$ [26]. Thus the concentration of M and the specific reaction rate constant governs the speed of decay of A. Since the equation is of the form as seen in equation 2.22 Euler's method of stability can be easily applied [28]. Which can be discretized using an explicit method to show how stability is a function of the time step, shown in equation 2.23, [28]. Solving for y_{n+1} gives equation 2.24, [28]. In the case of the ODE in equation 2.20, λ is negative leading to a stability requirement that $|1 + \lambda dt| \leq 1$ forces dt to

be less than $2/\lambda$. For the case of the reaction shown in equation 2.25 with k being represented by the equation 2.26 the resulting value makes $\lambda = 1.24 \times 10^{27}$. which means for stability the change in each time step must be 1.6×10^{-27} [26]. The computational time to solve for 1 second is enormous. Equation 2.26 uses A the Arrhenius factor, E_a the activation energy, b the exponent for temperature dependence between 0 and 1, T the temperature, and R_u the universal gas constant [26]. A number of programs have been developed to solve the various chemical kinetics that appear in different combustion situations, CHEMKIN was the software package used in this thesis and will be described in detail in the following section.

$$\frac{dC_A}{dt} = -kC_M C_A \quad (2.20)$$

$$C_A(t) = C_A(0) \exp(-kC_M t) \quad (2.21)$$

$$y' = \lambda y \quad (2.22)$$

$$y_{n+1} = y_n + \lambda dt y_n \quad (2.23)$$

$$y_{n+1} = y_n(1 + \lambda dt) \quad (2.24)$$



$$k = AT^b \exp\left(\frac{E_a}{R_u T}\right) \quad (2.26)$$

2.2.2 CHEMKIN

CHEMKIN is a set of 100 modular subroutines written in Fortran developed by Sandia National Laboratories in the departments of Thermal and Plasma Processes and Combustion Chemistry [29]. The goal behind the development of CHEMKIN was to develop a system of elementary gas phase chemical kinetics that could be used in simulations of fluid dynamics [29]. The expensive computational cost associated with modelling CFD that includes fuel vaporization, fuel-air mixing, chemical reactions, and complicated boundary geometries necessitates the use of elementary chemical kinetics [30]. Using CHEMKIN allows the use of accurate and efficient means of describing pressure dependent reactions [29], the Landan-Teller vibrational energy transfer rate, multiple species reactions and rates, and Arrhenius reverse reaction rates [29]. The solution of the stiff equation set is done by using a finite difference discretization scheme to solve the Jacobian matrices [29]. Information regarding finite difference can be found in reference [27]. The format of the input data is in the NASA chemical equilibrium code for ease of integration [29]. Equations 2.27 through 2.41 give the equations used in the CHEMKIN subroutines for solving the Scramjet design code found in chapter 5. Table 2.2.2 gives the descriptions of the variables used. Equation 2.27 is the equation of state relating pressure, temperature, mean molecular weight, and density together. The equations between equation 2.27 and 2.38 relate the different properties of each species together, and the final equation, equation 2.38, is the rate of progress variable given by the difference in the forward and reverse reaction rates used for the Arrhenius model [29, 26]. The Landau-Teller formulation of the vibrational energy transfer invalidates the Arrhenius model, and thus the routines employed in the Scramjet code do not consider this [26]. These CHEMKIN routines make the cumbersome and slow reactions mechanisms much easier to calculate than if they were implemented alone. More information regarding the CHEMKIN models can be found in reference [29] or [26].

$$\rho = \frac{P\bar{W}}{R_u T} \quad (2.27)$$

$$\bar{W} = \frac{1}{\sum_{k=1}^K Y_k / W_k} \quad (2.28)$$

Symbol	Description
Y_K	total mass fraction
X_K	total mole fraction
y_k	species k mass fraction
x_k	species mole fraction
$[X_k]$	total molar concentration
\bar{W}	mean molecular weight
W_k	total molecular weight
w_k	species molecular weight
k_{fi}	forward reaction rate constant
k_{bi}	backward reaction rate constant
v'_{ki}	stoichiometric coefficient for species k appearing as a reactant
v''_{ki}	stoichiometric coefficient for species k appearing as a product
E_{ai}	activation energy
K_{Ci}	equilibrium constant
S_i^0	standard-state entropy
H_i^0	standard-state enthalpy
\dot{w}_k	production rate
\bar{C}_P	total mixture-average specific heat at constant pressure
\bar{C}_V	total mixture-average specific heat at constant volume
\bar{c}_P	species mixture-average specific heat at constant pressure
\bar{c}_V	species mixture-average specific heat at constant volume
\bar{h}	total specific enthalpy
H_k	species enthalpy
ρ	density
P	pressure
T	temperature
R_u	universal gas constant
a_i	polynomial fit constants for NASA chemical equilibrium code
K	number of species
I	number of elementary reversible reactions
k	individual species

Table 2.1: Variable Definition of CHEMKIN Functions Used

$$X_k = \frac{Y_k \bar{W}}{W_k} \quad (2.29)$$

$$Y_k = \frac{X_k W_k}{\bar{W}} \quad (2.30)$$

$$\frac{C_{pk}^0}{R} = a_{1k} + a_{2k}T + a_{3k}T^2 + a_{4k}T^3 + a_{5k}T^4 \quad (2.31)$$

$$\bar{c}_V = \sum_{k=1}^K c_{V_k} Y_k \quad (2.32)$$

$$\bar{c}_P = \sum_{k=1}^K c_{P_k} Y_k \quad (2.33)$$

$$h_k = \frac{H_k}{W_k} \quad (2.34)$$

$$[X_k] = \left(\frac{P \bar{W}}{RT} \right) \frac{Y_k}{\bar{W}_k} \quad (2.35)$$

$$\bar{h} = \sum_{k=1}^K h_k Y_k \quad (2.36)$$

$$\dot{w}_k = \sum_{i=1}^I (v''_{ki} - v_{ki}') q_i \quad (2.37)$$

$$q_i = \left(\sum_{k=1}^K a_{ki} [X_k] \right) \left(k_{fi} \Pi_{k=1}^K [X_k]^{v'_{ki}} - \Pi_{k=1}^K [X_k]^{v''_{bi}} \right) \quad (2.38)$$

$$k_{fi} = A_i T^{\beta_i} \exp \left(-\frac{E_{ai}}{R_u T} \right) \quad (2.39)$$

$$k_{bi} = \frac{k_{fi}}{K_{Ci}} \quad (2.40)$$

$$K_{Ci} = \left[\frac{P_{atm}}{R_u T} \right]^{\sum_{k=1}^K (v''_{ki} - v_{ki}')} \times \exp \left[\frac{\Delta S_i^0}{R_u} - \frac{\Delta H_i^0}{R_u T} \right] \quad (2.41)$$

2.2.3 Reduced Order Reaction Mechanisms

Combustion of fuel can be an extremely complicated process due to many fuels containing as much as 300 components, solving the equations often require a massive amount of reaction steps since the equations are stiff a very small time constant is required [30]. These reactions can be characterized by as few as five steps to capture heat release rates, flame temperature, and emissions which are the qualities most needed to know [30]. Figure 2.13 shows how accuracy can be lost in some situations and maintained in others. For example, the temperature profile fails

to capture the flame holding region accurately in the 12 species compared to the 16 species, as well as under predicts the NO_X emissions. However, the loss in accuracy comes at the benefit of decreased computational time [5]. Applying these mechanisms is a common way of simplifying the computational time necessary to compute the chemical kinetics of a reaction.

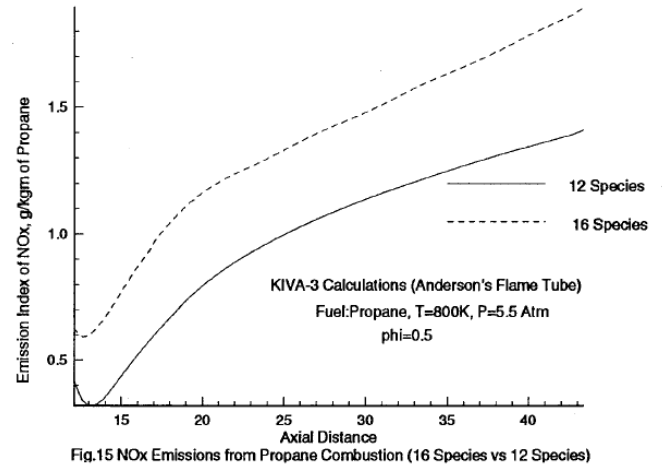
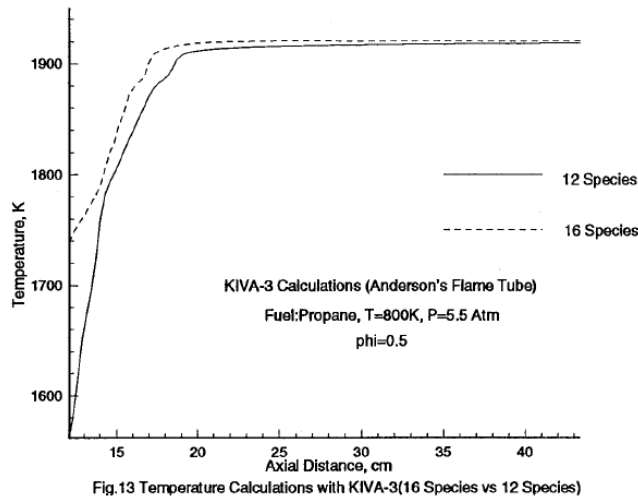


Figure 2.13: Example of How Reduced Order Mechanisms Effect Combustion [5]

2.3 The Surrogate Model Program

A number of surrogate modelling programs exist that allow the development of a number of regression surfaces including Kriging, Artificial Neural Networks, Multivariate Adaptive Regression Splines, etc. In choosing a program, a number of factors were considered: usability, clear methods, access, and robustness.

Usability is defined by the author as the ease at which the surrogate model can be developed using already compiled data or actively generate using a dynamic data set. In order for the surrogate modelling program to be useful, it needs to easily formulate models after each CFD run and allow the author to test for local extrema. Clear methods are defined by how well the program is documented. While a "black box" is acceptable in the terms that the author does not need to understand how to implement the regression model used, however the author still needs to understand what methods

are being used and be able to test the surrogate model. Access is defined as how easy it is to actually obtain the program.

2.3.1 Matlab

Matlab has a number of routines that develop regression surfaces. The embedded one in simulink failed the final criteria as the author would have to purchase an extension in order to use the program. There is a free access package for Matlab known as the Surrogate Modelling Toolbox, or SUMO. This program fails the first two criteria. Running the program was too black box oriented. A surrogate model could be generated but evaluating the model did not prove to be easy, and understanding what was done to generate the model was not very clear in the SUMO documentation. For these reasons the Matlab surrogate modelling methods were not used.

2.3.2 Develop

The book "Engineering Design via Surrogate Modelling A Practical Guide" [3] explains many of the surrogate modelling methods very well and includes access to their surrogate programs. Using these programs would force the author to fully generate and debug the surrogate modelling process, and complications would likely arise during development of multi-variable surrogate models. While this would be beneficial for improving understanding of the theory behind surrogate modelling, and would allow more flexibility in the methods employed, developing a robust code would be outside the scope of the thesis.

2.3.3 Dakota

Dakota was determined to be the most likely for their surrogate modelling and optimization routines. However, after learning to implement Dakota and generating input files, the author found that all the evaluation of the surrogate model was all done inside the Dakota optimization routines and accessing the models for the author's own goals was impossible. Thus Dakota failed the ease of access requirement.

2.3.4 SurfPack

After learning Dakota and understanding how the code develops surrogate models, the author contacted Sandia Labs and was put in touch with the developers of the surrogate program that Dakota uses. The author found that the program Surfpack that was used by Dakota fit all the requirements for a surrogate modelling tool. It runs a number of regression surface methods, and is easy to test after the model is built. It is also fast enough to generate models orders of magnitude faster than it takes to generate points for Surfpack to use. The rest of this chapter is dedicated to the explanation of the author's methods for compiling and speeding up the generation of regression surfaces to be easily implemented in any CFD or combustion routine.

2.3.4.1 Compiling and Running

In order to compile and run Surfpack C, C++, Fortran 77, BLAS, and LAPACK, libraries are required [24]. Most of the routines are written in C++, except for the MARS and kriging algorithms which are written in Fortran [24]. BLAS is for the data fitting routines, and LAPACK is required for different drivers [24]. Since Surfpack was used as a standalone, lex and yacc were also required for build. Once all the libraries and drivers are installed on the machine, the Surfpack files downloaded and extracted, the enable standalone configure option can be run by make and make install or :

```
./configure --enable-standalone --prefix=<path to surfpack directory>/SURFPACK_INSTALL
make
make install
```

Replace < *path...* > with the actual path and be sure there are no spaces in any of the names in the directory. In order to run Surfpack with a given input file the following line of code is required inside the linux terminal:

```
./surfpack InputFileName.spk
```


The .spk, and .spd files are required along with the Surfpack executable inside the directory that the ./surfpack call is made. This will generate a new .spd file with all the data stipulated for by the .spk file.

Chapter 3

Algorithm and Flow Chart for Generating Input Files

3.1 Requirements

In order to efficiently generate surrogate model surfaces for the wide range of applications required in scramjet design, a program that allows for n-dimensional regression surfaces was required. As mentioned in the previous chapter, Surfpac can create n-dimensional regression surfaces given data and input files in a specific form. Creating files in these forms can be relatively tedious, especially when trying to generate files for many dimensions, as each step requires manually typing in information required for running Surfpac. Also, Surfpac does not accept common input files such as comma delimited.

3.2 Generating the files

The first task to make Surfpac functional for the applications suitable to the author was to write routines to quickly generate the Surfpac required data sets for any type of data. Figure 3.1 gives the short routine used to generate Surfpac input files given any M by N data set. By using key words, fprintf, and for loops, the Surfpac input files could be generated quickly and efficiently with this one function and convert the common type .csv files to the less common format .spd. Thus any type of data can easily be converted into Surfpac readable files with minimal effort on the side of the developer. With these tools in place generating useful regression surfaces for nth dimensional data is relatively simple. The following section shows that these routines including the Surfpac program matches results from the Surfpac developers at Sandia National Labs.

```

1 Determine size of mxn matrix, where m is the number of data sets
2     and n the number of variables
3
4 Number of response variables = n-1
5
6 Generate datafile for input data using .spd
7
8 Develop strings for .spk file
9
10     Create axes using a given range for each variable
11
12     Create sample by giving the number of grid points required and axes
13
14     Create surfaces using a given method, kriging or otherwise
15
16     Chose optimization method, global or local
17
18     save information to an .spk file

```

Figure 3.1: Matlab Routine Block Diagram for Generating .spk input files

3.3 Generated Results Comparisons

In order to confirm that the algorithm and coupled Matlab/Fortran methods work a comparison was required. The developers of Surfpak have released a paper that explains the strengths of Surfpak for irregularly spaced sparse matrices [6]. The following sections takes their data and creates a .csv file from it. Then through the algorithm and functions explained in the previous sections a .spd Surfpak data input and .spk Surfpak command input files can be generated. The following sections compares the results from reference [6] with the results from the present methods for one-dimensional, two-dimensional and four-dimensional data sets. After making these comparisons the author can be confident that the functions and usage are correct.

3.3.1 One Dimensional

Given on page 5 and in the appendices of reference [6] are a set of CFD force coefficients for some independent variable. The Matlab code generated the input function, figure 3.2, based on the given parameters from reference [6]. These parameters include the number of predictors, lower and upper bounds, number of points and type of interpolation method, which were 1, -4 to 1.2, 27, and kriging, respectively. From figure 3.3 it can be seen that the results match exactly. This

occurred from using the input file from reference [6]. However, when comparing these results with the results from the paper, figure 3.4, there are some differences in the first and last point. This is likely due to Surfpac’s continual development after 2006, when reference [6] was published. As of 2008 Surfpac was no longer being developed individually and since then, as per a conversation with the developer Keith Dalbey, an improved kriging method was written thus accounting for the slight differences between figures 3.3 and 3.4. The surrogate matches the sampled points exactly since the model uses kriging interpolation rather than regression.

```

1 Load[name = DataFile, file = 'Example_1D_comparison.spd', n_predictors =1, n_responses = 1]
2
3 CreateAxes[name = indep_ax, bounds = '-4 1.2']
4
5 CreateSample[name = sample_indep, axes = indep_ax, grid_points = (27), labels = (Yr)]
6
7 CreateSurface[name = surface_1, data = DataFile, response = y,type = kriging,log_scale = (y),
8 norm_scale = (Yr),response_index =0]
9
10 Evaluate[surface = surface_1, data = sample_indep, label = y1]
11
12 Save[data = sample_indep, file = 'Example_1D_Surfpaccomparison.spd']
13
14 Save[surface = surface_1, file = 'Example_1D_Surfpaccomparison1.sps']
15

```

Figure 3.2: Generated .spk input file for 1-D from Author’s code

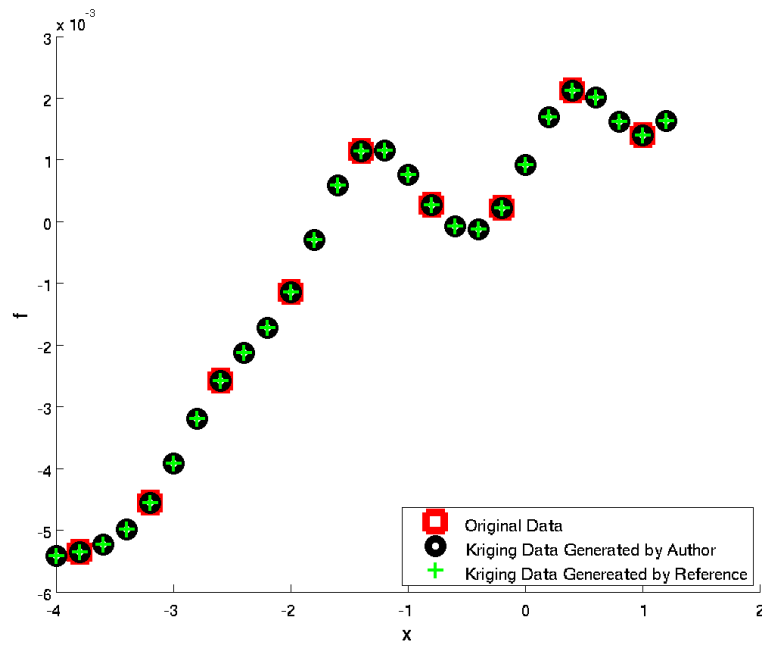


Figure 3.3: Comparative Results between Reference [6] and Author's File Generation

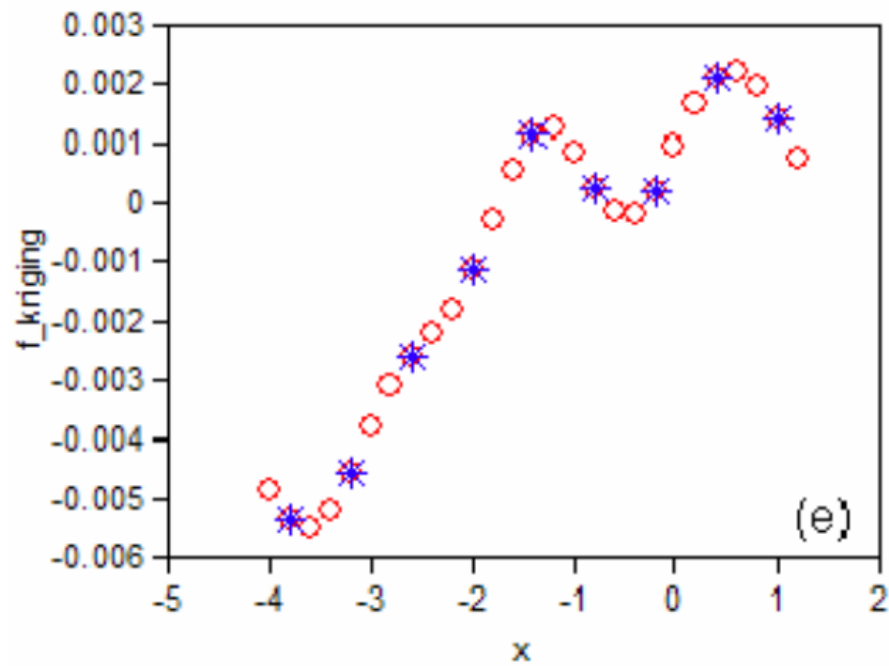


Figure 3.4: Results from Reference [6]

3.3.2 Two Dimensional

Generating the two dimensional case was slightly more difficult but mainly in the plotting sense. The Matlab routine developed by the author was designed to handle any number of input parameters, however there was one modification that was made to the Matlab routine for the multiple dimension cases. By comparing figures 3.2 and 3.5 it can be seen that the local optimization key was added. The default is global optimization but in order to find the multiple maxima and minima local optimization is required. The results for Matlab peaks are given when comparing the author's results with the results from reference [6] as shown in figures 3.7 and 3.8. Here the results are almost exact but based on the results from the one-dimensional solution it seems that the regression surface routines in Surpack have been slightly modified since the publishing on reference [6]. With regards to the actual quality of the comparisons the general patterns are well captured by roughly 17 points. When looking for a global maximum Kriging was able to find where the minimum and maximum points exist in a relatively complicated data set. The actual values for the maxima are off by 15%. The author's local optima matches reference [6], but the slope leading up to that point by up to 20%. However, since the location is the same another run to generate one more point in the maxima will allow the user to quickly and easily find the optimization points.

```

1 Load[name = DataFile, file = 'Example_2D_comparison.spd', n_predictors =2, n_responses = 1]
2
3 CreateAxes[name = indep_ax, bounds = '-3 3 | -3 3']
4
5 CreateSample[name = sample_indep, axes = indep_ax, grid_points = (13,13), labels = (Yr)]
6
7 CreateSurface[name = surface_1, data = DataFile, response = y,type = kriging,log_scale = (y),
8   norm_scale = (Yr),response_index =0, optimization_method = local]
9
10 Evaluate[surface = surface_1, data = sample_indep, label = y1]
11
12 Save[data = sample_indep, file = 'Example_2D_Surpackcomparison.spd']
13
14 Save[surface = surface_1, file = 'Example_2D_Surpackcomparison1.sps']
15

```

Figure 3.5: Generated .spk input file for 2-D from Author's code

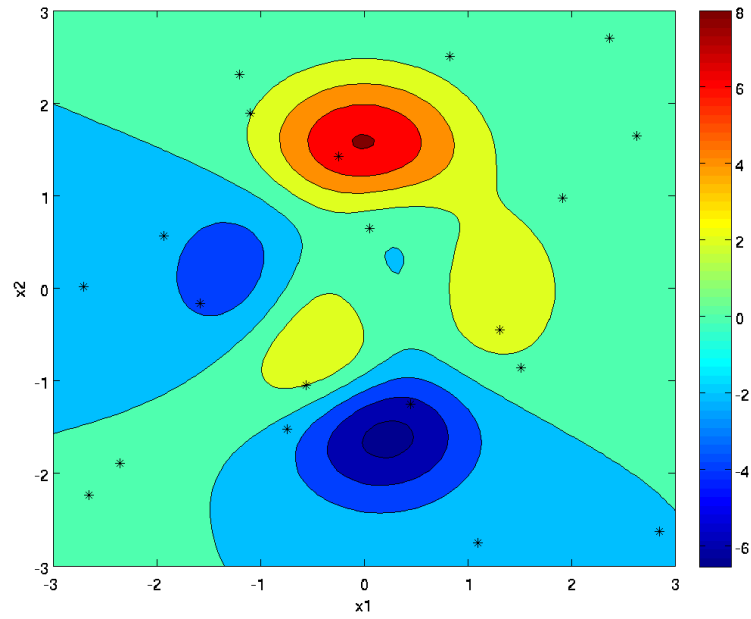


Figure 3.6: Matlab's Results for 'Peaks' Generation

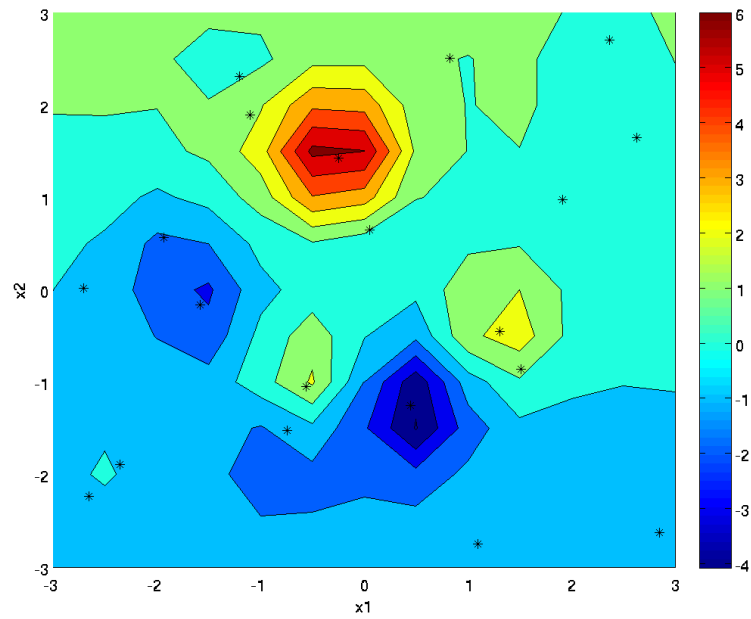


Figure 3.7: Author's Results for Matlab's 'Peaks' Generation

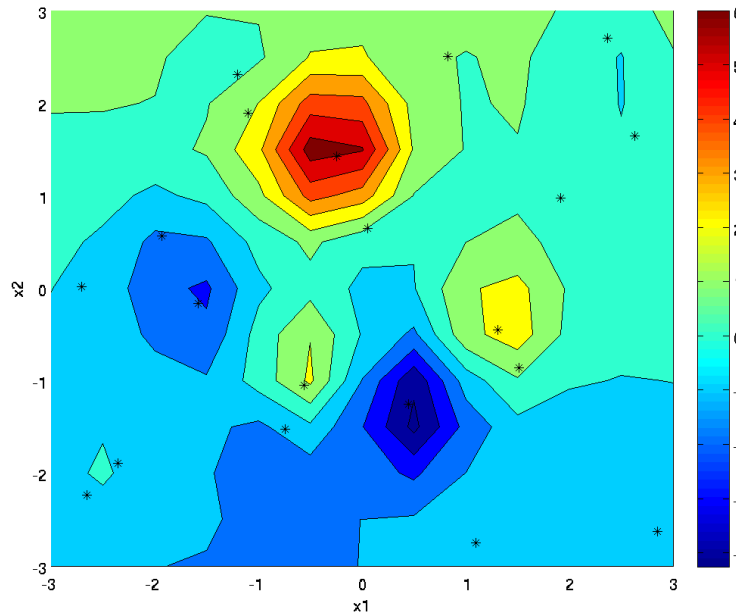


Figure 3.8: Reference [6] Results for 'Peaks' Generation

3.3.3 Four Dimensional

The final referenced dimension from the Sandia Lab's Surfpack group [6] gives a surrogate surface created from real world engineering analysis problem. There were 42 points sampled in four dimensions but for generating the figures 3.3.3 through 3.12 only three dimensions were shown with the 4th being held constant at 0.7. An 11x11x11x1 grid was generated using Surfpack for the input function, figure 3.9, which generated figure 3.11. Figure 3.3.3 shows the 42 points that were used to generate the two surrogate surfaces still using kriging optimization routines. From comparing figures 3.11 and 3.12 it can be seen that the author's Surfpack and input routines match that of the 4D Engineering example given in reference [6].

This chapter was developed in order to confirm that the author's routines and compiled Surfpack executable was working correctly and basic visualization methods could be employed to confirm. By examining the first, second, and fourth dimensional routines it can be determined that the methods employed match that of Sandia Lab's Surfpack team and can be trusted in so much

that any problems with optimization lies more in the choice of points used to generate the model or the kriging optimization routine, rather than in the author's routines and functions used. This is very helpful for analysing results and determining where the error lies.

```

1 Load[name = DataFile, file = 'Example_4D_comparison.spd', n_predictors =4, n_responses = 1]
2
3 CreateAxes[name = indep_ax, bounds = '0.5 1.5 | 1.5 4.5 | 0.7 1.0 | 0.7']
4
5 CreateSample[name = sample_indep, axes = indep_ax, grid_points = (11,11,11,1), labels = (Yr)]
6
7 CreateSurface[name = surface_1, data = DataFile, response = y,type = kriging,
8 log_scale = (y), norm_scale = (Yr),response_index =0, optimization_method = global]
9
10 Evaluate[surface = surface_1, data = sample_indep, label = y1]
11
12 Save[data = sample_indep, file = 'Example_4D_Surfpaccomparison.spd']
13
14 Save[surface = surface_1, file = 'Example_4D_Surfpaccomparison1.sps']
15

```

Figure 3.9: Generated .spk input file for 4-D Engineering Analysis Example from Author's code

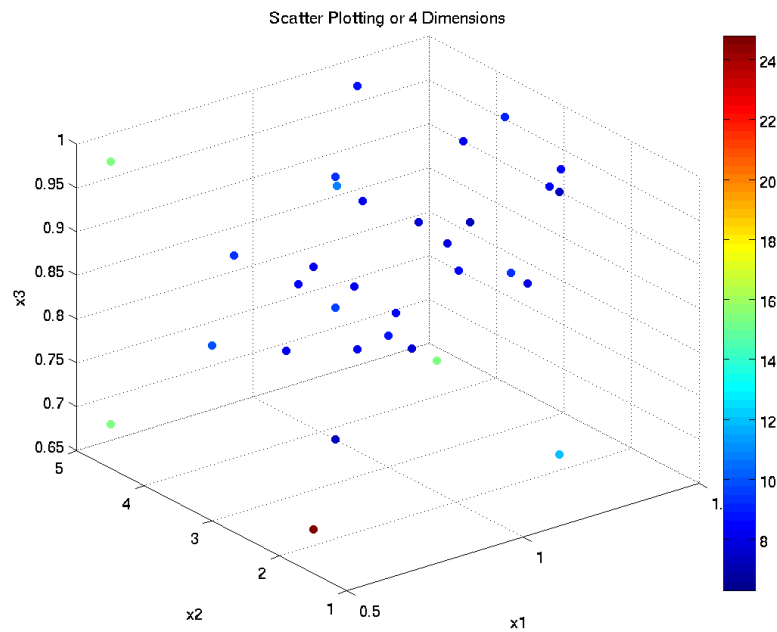


Figure 3.10: Original Points Used to Develop Regression Surfaces

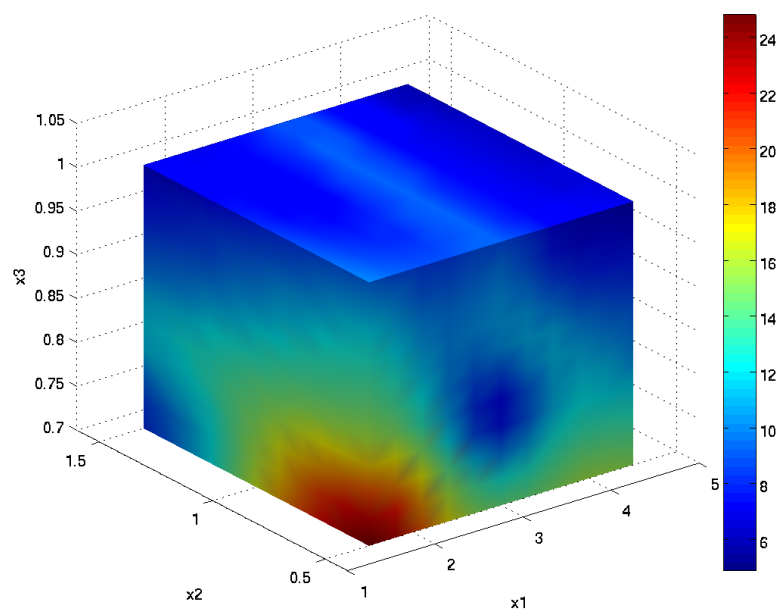


Figure 3.11: Author's Results for the 4-D Engineering Analysis Example

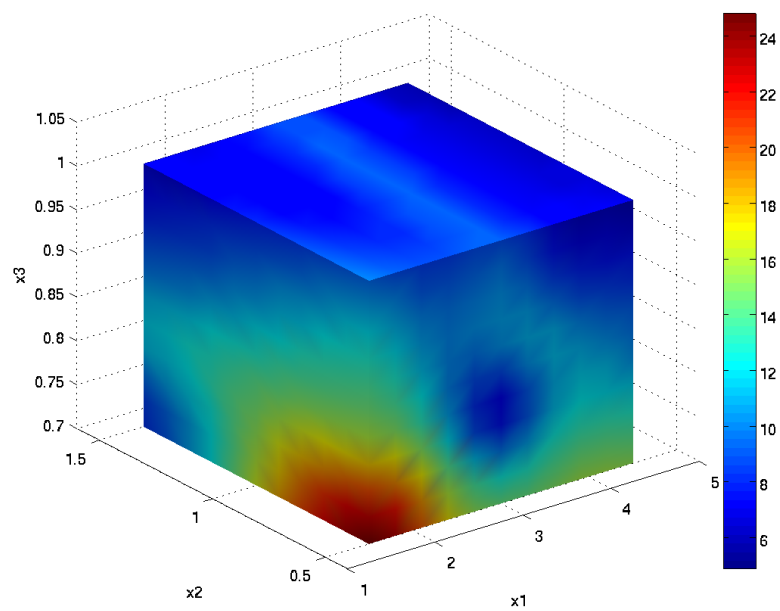


Figure 3.12: Reference [6] Results for the 4-D Engineering Analysis Example

Chapter 4

Sampling Procedures for Optimizing Design

The foremost reason for interest in surrogate modelling is to find the global and local minima and maxima given an existing data set or at the onset of developing a CFD or experimental testing plan. The goal is to find these maxima and minima from the least number of data points thus saving hundreds of hours of computational time. If the data set already exists, surrogate modelling is more focused on giving the designers a better understanding of the interplay of each of the variables and insight into whether or not more data points are required to capture the most efficient, highest thrust, longest range, i.e. maxima or minima, of the system. If no data has been run, the surrogate models can be generated on the fly with each new data point adding to the accuracy of the overall model and the model giving recommendations for which data point should be tried next to increase overall efficiency of each data set. Local and Global optimizers differ with regards to efficiency in capturing specific effects. Local optimizers are usually gradient, conjugate gradient, or quasi-Newton based, and are very effective at capturing smooth data's maxima and minima but get stuck on or in peaks and valleys [3]. Global optimizers are geared towards capturing global maxima and minima by simultaneously examining multiple local maxima and minima while quickly ignoring shallow trenches and short peaks, with common methods as genetic algorithms and simulated annealing [3].

In order to practice the optimization routines, continuing with previous chapters, one, two, and n dimensional algorithms were tested for accuracy of capturing minima and maxima, as well as minimizing the number of required data points in a data set.

4.1 One Dimensional Optimization and Methods

One dimensional methods for optimization are relatively simple. Assuming an original three point initial condition, each additional point can be added by picking the point of no gradient, which corresponds to a maxima or minima. Reference [3] used the function $f(x) = (6x - 2)^2 \sin(12x - 4)$ for their section on exploring and exploiting a surrogate. The function itself is relatively simple with a global minimum and a local maxima and minima in between the boundaries. Using a polynomial approximation the surrogate diverges from the actual function significantly even after 9 iterations, with each iteration adding a new point when the gradient is zero [3]. Using a Gaussian RBF approximation eight points are required and there is still relatively high inaccuracy towards $x = 0.5$ [3]. Figure 4.1 gives the results using kriging surrogate modelling for three to seven points. At six points the general shape of the function is found with only missing the optimum magnitude. Adding one more point captures the original function within a small tolerance. By testing one more point at the minimum would tell the designer that the optimum design point has been found. Thus with only seven points all knowledge of the function is known.

The function $f(x) = (6x - 2)^2 \sin(12x - 4)$ is fairly simple and since it is based on a mathematical function, relatively easy to match using interpolation methods. Instead, consider a more difficult case of an inert 90 mm diameter in-bore flowfield ram accelerator projectile propulsion system [4]. With this data we see significantly more diverse effects which are significantly more difficult to capture. The experimental data was captured from reference [4] by using the program XYextract, which lead to 113 discrete points. When a zero gradient spot was found in the surrogate solution using a gradient based solver took that point to be a possible optimum resulting in evaluating points right next to each other on the flat line. This resulted in a system over populated by data that was outside the region of interest, too many points were placed close to each other leading to an over sampled system causing instabilities in the surrogate model. This effect can be seen in figure 2.11. In order to improve sampling efficiency two methods were used. The first was a basic uniform sampling with 12 and 23 total data points. These results are given in figure 4.2.

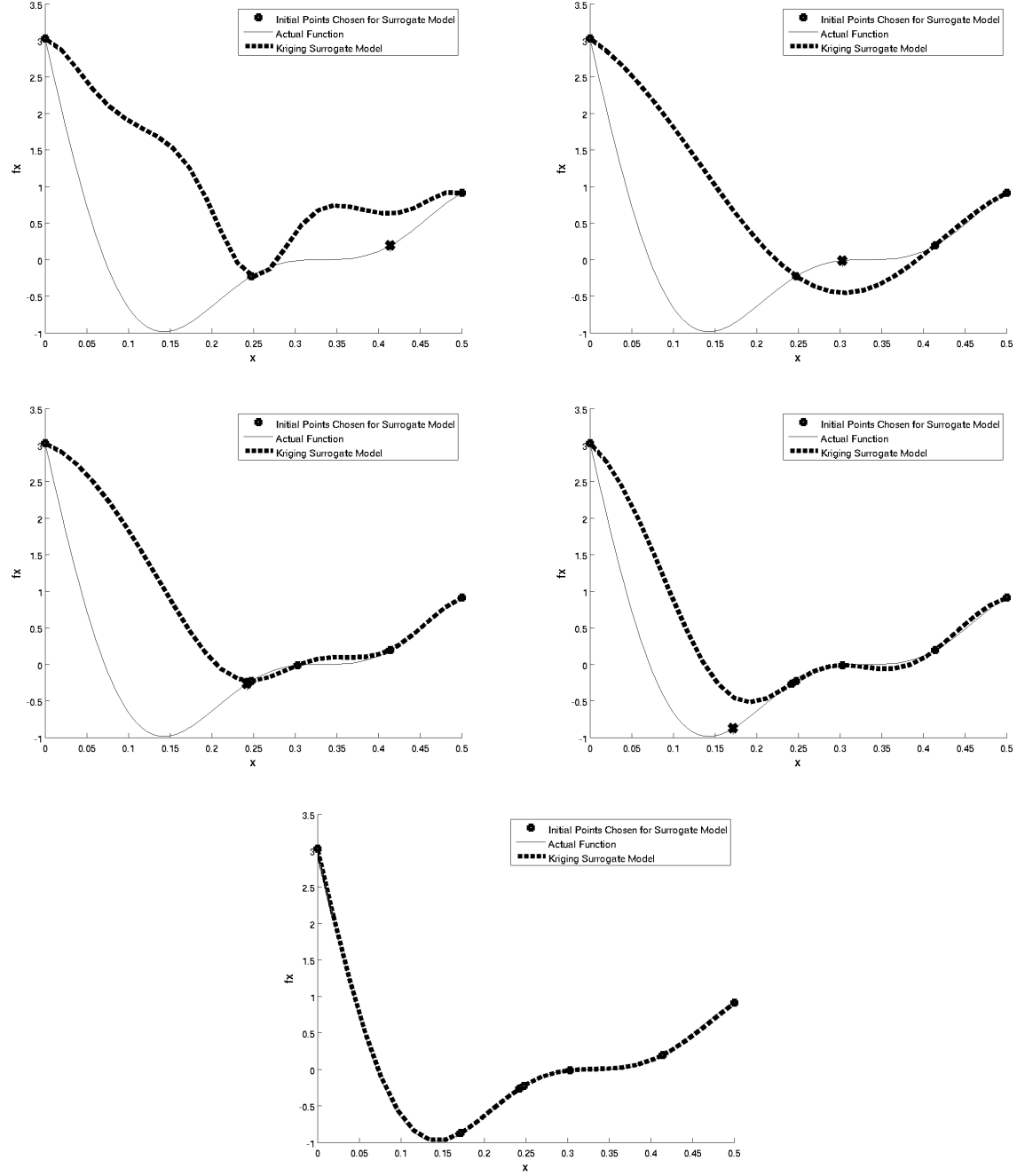


Figure 4.1: Three to Seven Point Kriging Solution for $f(x) = (6x - 2)^2 \sin(12x - 4)$

Using a uniform sampling most of the function is accurately captured, aside from the short constant pressure region at 70 mm, which kriging interpolation forces to be oscillatory. However, the rest of the solution is relatively accurately captured aside from the oscillations in the beginning and end.

Comparing these results with figure 11 from reference [4], figure 4.4, the surrogate model predicts the amplitude and overall effects significantly better than CFD. Now assuming a priori knowledge of the system, either through previous experiments or high fidelity CFD, an improved model can be built using specific sampling which can be seen in figure 4.3. With improved sampling methods using discrete points pulling useful data from less sensors allows for easier experimentation with less adverse effects from inserting sensors into the experimental apparatus.

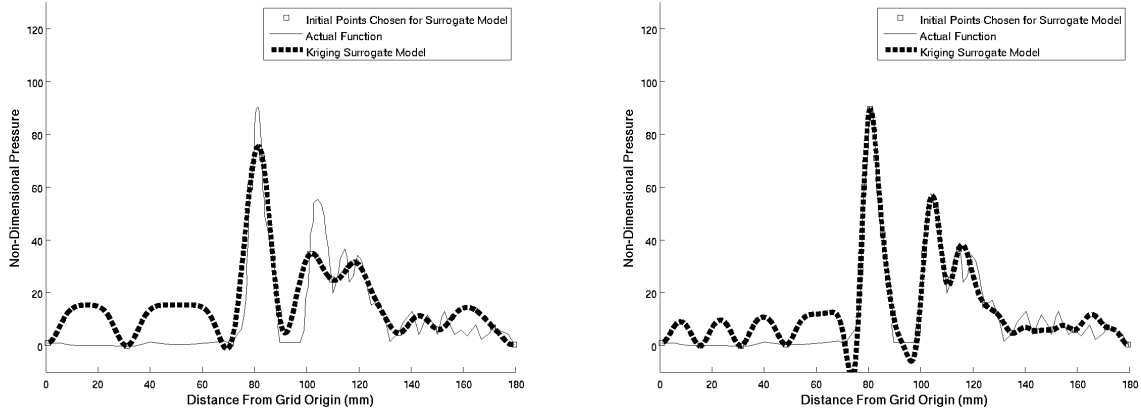


Figure 4.2: Twelve (Left) and Twenty Three (Right) Uniform Sampling Point Kriging Solution for the Experimental Results from reference [4]

Another study was done from the same reference looking at the pressure on a projectile surface for a reacting flowfield. Figure 4.5 shows the results of seven to eleven points using gradient based optimization on the kriging surrogate model. From here seven equispaced initial points does a good job of capturing the relevant shape with added peaks and values. By checking each peak and valley the actual optimums can be identified. With any further points past ten, are just in improving confidence that there are no other optima.

The major conclusions from analysing the one dimensional case are to confirm that the sample set is a minimum before adding gradient based information. To define a minimum there must be obvious peaks and valleys that can be further refined. Without these the iterator will not function correctly. If the data set is over sampled, then the kriging surrogate will lead to sharp gradients and an unreal design space. Thus always start with a minimum to capture some shape of the function

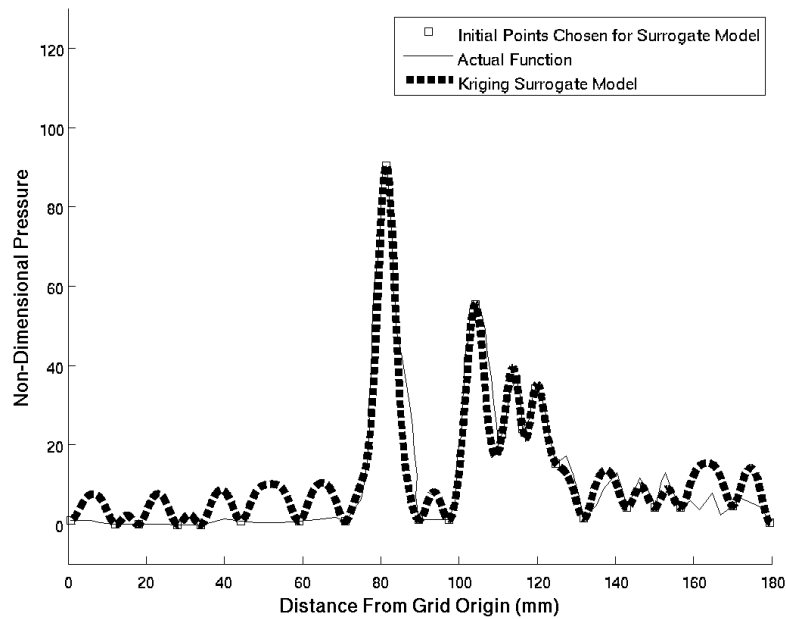


Figure 4.3: Specific Sampling Point Kriging Solution for the Experimental Results from Reference [4]

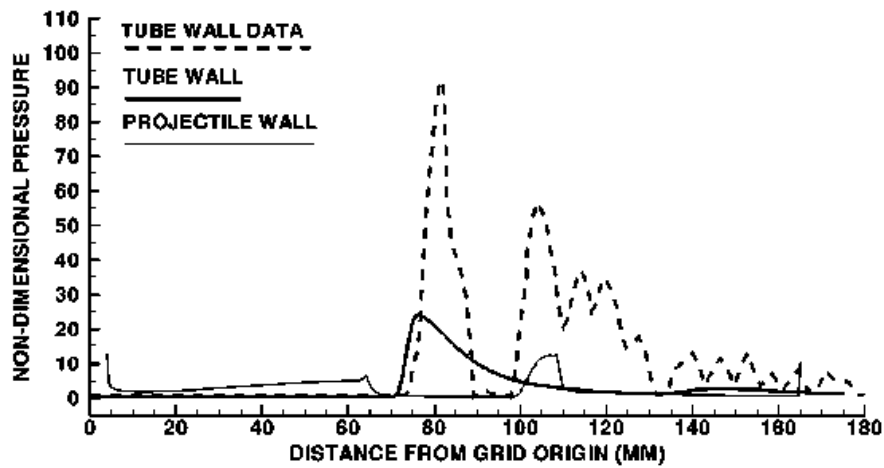


Figure 4.4: Figure 11 from Reference [4], Nondimensional Pressure Distributions Along Tube all for Non Reacting Conditions

and slowly add points to determine if the valleys and peaks are in fact optima or just the kriging surrogate forcing oscillatory behaviour.

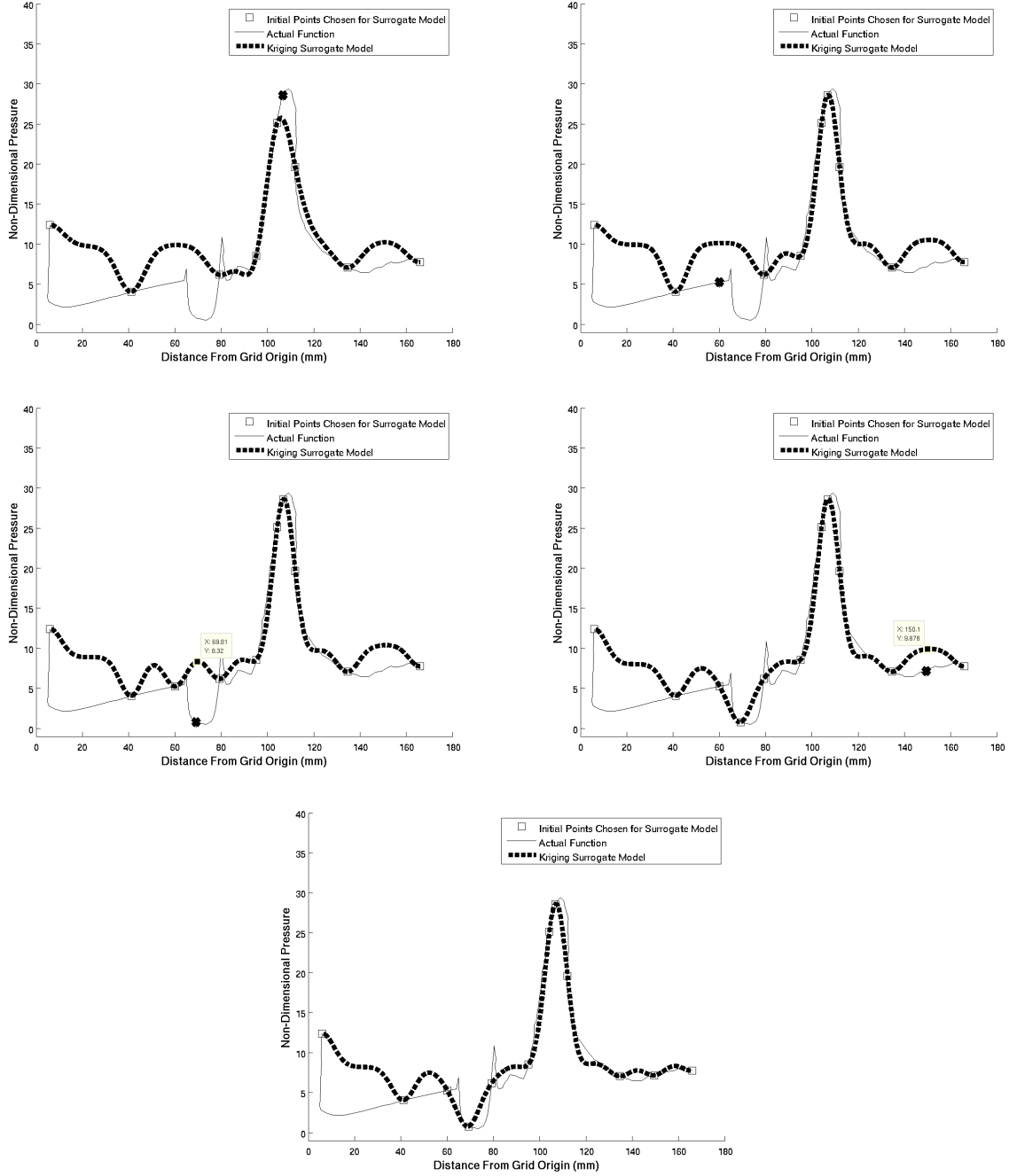


Figure 4.5: Seven to Eleven points sampled using gradient based optimization, Kriging Solution for the Experimental Results from reference [4]

4.2 Two Dimensional Optimization and Methods

In order to develop methods for multi-dimensional problem types, similar to the one-dimensional, a complicated enough function was required. The function was chosen to be $f(x_1, x_2) = 21.5 +$

$x_1 \sin(4\pi x_1) + x_2 \sin(20\pi x_2)$ and came from reference [2]. The plot of the original function can be seen in figure 4.6. As can be seen from the function a number of peaks are developed with varying amplitude in both the x_1 and x_2 directions. Without a priori understanding of the problem accurately capturing the overall function as well as the optima is difficult. But the goal of this section is to develop methods that would allow a user without any understanding of the solution to be able to minimize the number of points and still achieve a good surrogate. Thus as from previous sections, the first step is to pick a few points in a uniform manner.

Figure 4.7 shows the first surrogate model chosen from the minimum required values to create a model, a grid of 5x5 uniformly chosen points. Both global and local optimization methods were used for comparison. Luckily we happened to capture the maxima within a 2.5 percent. However, even though at the outset the maximum value is known, there is no way to be sure this is in fact the maximum without testing many other places, or developing a more confident surrogate model. Also, the x_2 position is off further increasing the need to improve the surrogate model.

Judging from the local and global optimization kriging surrogate models it can be seen that a number of hills and valleys are created. But the fear is that these are not real and were created as a function of the kriging methods similar to the oscillations occurring in figure 4.5. Or, as is the reality, that the function is missing even more peaks and valleys. Thus before applying gradient optimization methods, a good original surrogate must exist. To improve the surrogate another sampling of the optima region is required using a near neighbour method. A single peak and valley was sampled in figure 4.8. The surrogate in figure 4.8 shows that more than one peak exists and thus to better capture the function behaviour a more refined model is required. Refining it one more time gives figure 4.9 and while another sample or two to confirm the frequency would be required for improving confidence in an unknown solution, figure 4.9 is known to be reduced enough from a priori knowledge of the actual solution and further investigation will not be done. Instead, now that the general frequency amplitude and overall patterns is known in the surrogate space and final model will be created combining all previous points, which gives a 12x12 discrete sampling space and is shown in figure 4.10 along with the surrogate model. While the complete

function space is not exact, the general magnitude and frequency is shown, and the optima are found within one percent of the maximum including $x_1, x_2, f(x_1, x_2)$, and within two percent of the minimum independent variable locations x_1, x_2 , and seven percent of the dependent variable, $f(x_1, x_2)$.

The methods developed in the one dimensional section are thus easily applied to a two independent variable space. By first using a sampling space that captures some initial behaviour regions of interest can then be more finely sampled resulting in a final evaluation of the surrogate that accurately captures the optimal points. A priori knowledge of the solution, or the type of the solution, speeds up the process, but without any knowledge the designer can test his model by adding another couple of discrete points, which given a two dimensional complicated space such as that gone over in this section is practically negligible. These results were captured using only uniform sampling plans and thus can be readily reproduced. If random sampling was chosen then more points in the initial surrogate are required to accurately capture the overall shape of the function allowing for the correct choice of the optimum region.

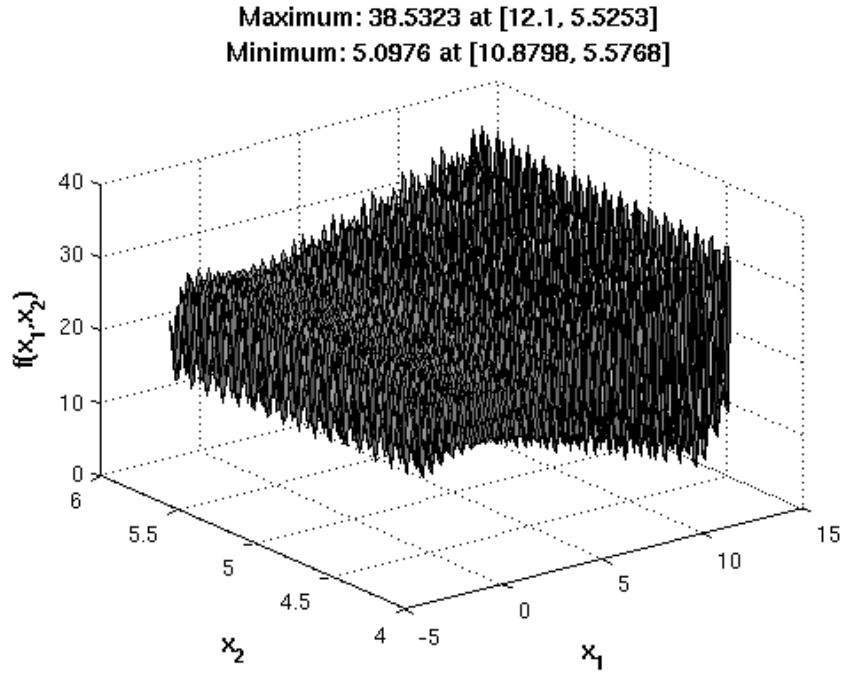


Figure 4.6: Plot of the function $f(x_1, x_2) = 21.5 + x_1 \sin(4\pi x_1) + x_2 \sin(20\pi x_2)$

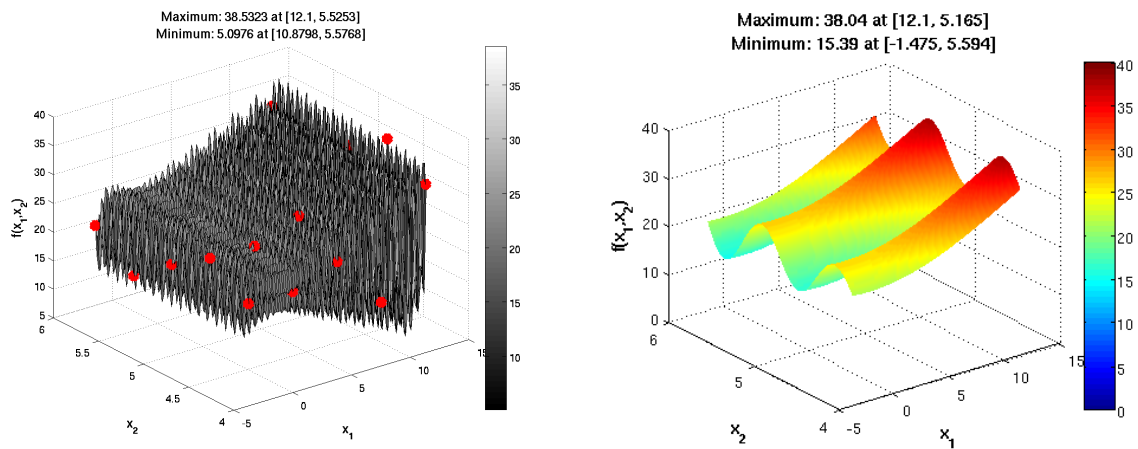


Figure 4.7: Twenty Five Point Optimization of the function $f(x_1, x_2) = 21.5 + x_1 \sin(4\pi x_1) + x_2 \sin(20\pi x_2)$ Point Selection Left, Local Optimization Right

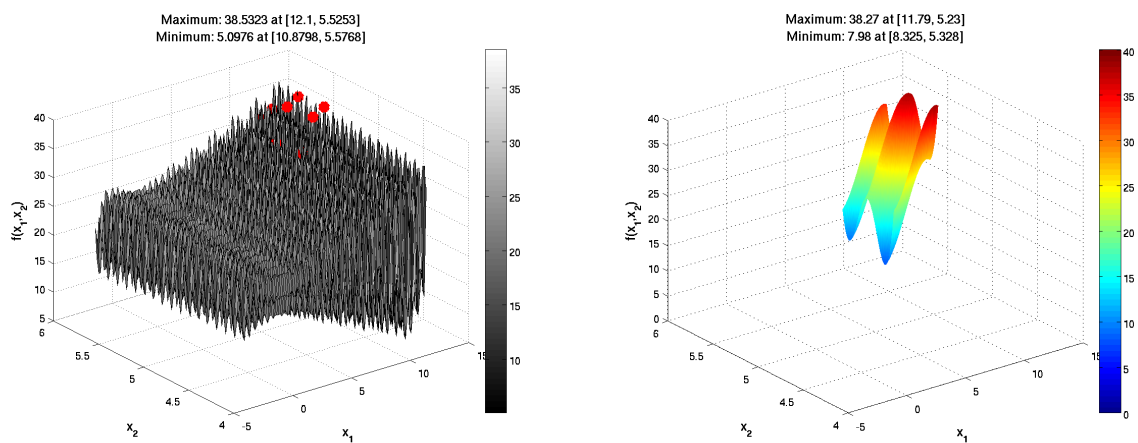


Figure 4.8: Twenty Five Point Optimization of the function $f(x_1, x_2) = 21.5 + x_1 \sin(4\pi x_1) + x_2 \sin(20\pi x_2)$ Point Selection Left, Local Optimization Right

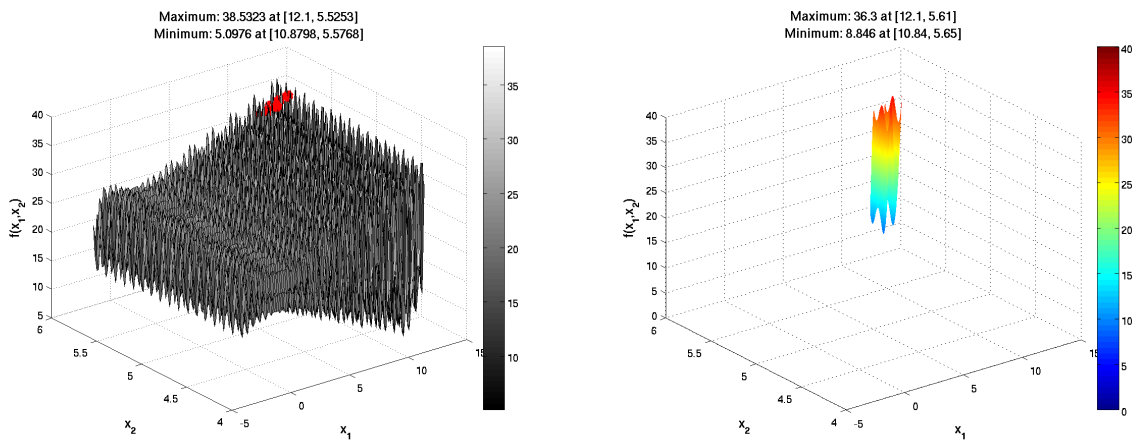


Figure 4.9: Twenty Five Point Optimization of the function $f(x_1, x_2) = 21.5 + x_1 \sin(4\pi x_1) + x_2 \sin(20\pi x_2)$ Point Selection Left, Local Optimization Right

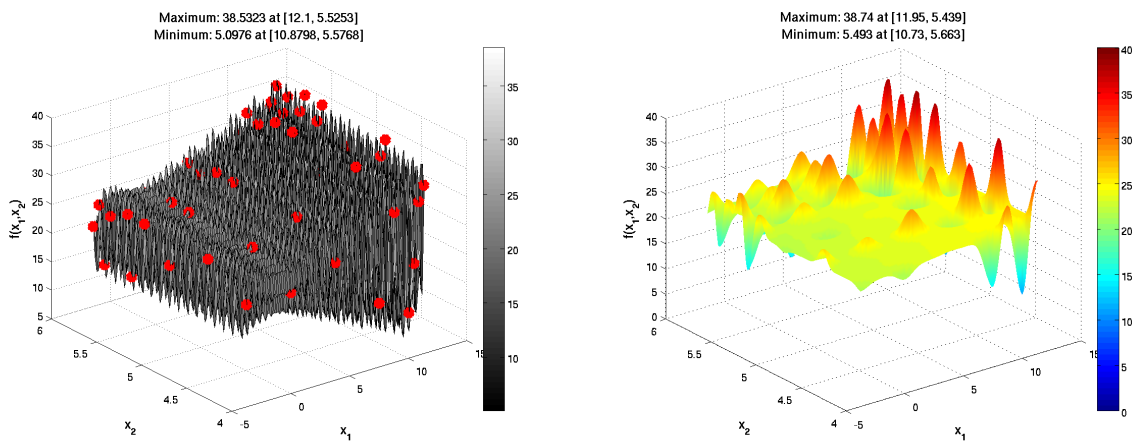


Figure 4.10: 144 Point Optimization (combination of previous three figures) of the function $f(x_1, x_2) = 21.5 + x_1 \sin(4\pi x_1) + x_2 \sin(20\pi x_2)$ Point Selection Left, Local Optimization Right

4.3 N Dimensional Optimization and Methods

Optimizing routines for N dimensions becomes significantly more difficult. For one, sampling schemes are more involved since different independent variables have different effects on the function being optimized. For example, a peak in the dependent variable may be due to any point in the space x_1, x_2, \dots, x_n . Thus choosing the correct variable for analysis becomes very difficult. Furthermore, troubles are increased just in the visualization aspect. As CFD and multi-variable functions become more prevalent in engineering design due to the increase in computational power, visualization methods are improving. This still however, poses difficulties to the design engineer of multi-variable dependent systems.

In order to develop procedures for multi-dimensional optimization equation 4.1 was used. This equation comes from reference [3] and is a conceptual design equation that estimates the weight on an Cessna C172 Skyhawk air plane derived from data fitting schemes adapted into an equation by D.P. Raymer [31]. Table 4.3 gives definitions of each variable as well as its range. A baseline set was used in reference [3] in order to set all other variables constant for plotting contour plots. This visualization method was also employed for ease of comparison. Table 4.3 gives the explanation of each variable in equation 4.1. Although there are ten variables it can be expected that not all of them have a profound effect on the final weight of the vehicle (dependent variable). However, this is only known from a priori understanding of the system and it's functions. It will be assumed that each of these parameters has an unknown effect on the system and in order to develop a understanding of how each effects the overall weight a number of surrogate models were developed.

$$W = 0.036 S_w^{0.758} W_{fw}^{0.0035} \left(\frac{A}{\cos^2 \Lambda} \right)^{0.6} q^{0.006} \lambda^{0.04} \left(\frac{100tc}{\cos^2 \Lambda} \right)^{-0.3} (N_z W_{dg})^{0.49} + S_w W_p \quad (4.1)$$

There are two methods of deriving a surrogate model. One way is to develop a system of two dimensional surrogates to create a figure like figure 4.11 from reference [3]. To develop this type of figure using surrogate models requires keeping all other variables besides two constant and develop-

Symbol	Parameter	Baseline	Minimum Value	Maximum Value
S_w	Wing area (ft ²)	174	150	200
W_{fw}	Weight of fuel in the wing (lb)	252	220	300
A	Aspect ratio	7.52	6	10
Λ	Quarter-chord sweep (deg)	0	-10	10
q	Dynamic pressure at cruise(lb/ft ²)	34	16	45
λ	Taper ratio	0.672	0.5	1
tc	Aerofoil thickness to chord ratio	0.12	0.08	0.18
N_z	Ultimate load factor	3.8	2.5	6
W_{dg}	Flight design gross weight (lb)	2000	1700	2500
W_p	Paint weight (lb/ft ²)	0.064	0.025	0.08

Table 4.1: Equation 4.1 Variable Definition

ing a two dimensional surrogate similar to the previous section. However, to get the required data to perform this operation is expensive. In order to develop an idea of the system it is much easier and computationally cheaper to run a few discrete points and then from there develop a surrogate model from all available resources. The complication with this however is as variable dimension increases, the difficulty of creating a surrogate increases by orders of magnitude. Furthermore as dimensionality increases the need for more discrete points is required further increasing the cost of the surrogate. As mentioned previously, generating figure 4.11 could be easily done using 40 surrogate models of 2D data sets. This would be the cheapest computationally with regards to the surrogate model and would allow the designer to determine which variables create non-linear qualities in the function. This could then allow the designer to reduce the data set to more important independent variables. For example, in figure 4.11, it can be seen that ultimate load factor, aspect ratio, aerofoil thickness to chord ratio, give the greatest change in weight, and quarter-chord sweep is the most non-linear. Using this information the designer can focus on minimizing the overall weight by varying these parameters. This is the limit of the usefulness of these plots, since if these were all the designer had access to, there would still be a lot of CFD/experiment to zero in on the optima. A more useful application of surrogate modelling lies in its cheap computational cost to develop a cross coupled function of multiple independent variables.

Figure 4.12 highlights the surrogates strength's and weaknesses. Fourteen points were ran-

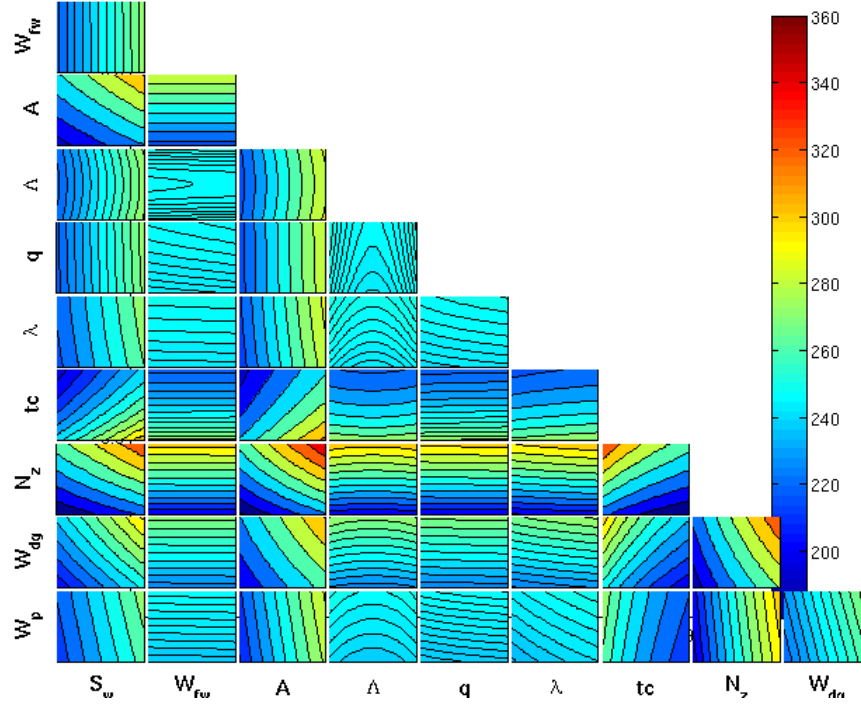


Figure 4.11: Tile plot from reference [3] from equation 4.1. Each tile varies two variables while holding the eight other variables at baseline values as per table 4.3

domly chosen that spanned the function's (equation 4.1) independent variables. Any more points and the cost of generating the surrogate increases, but still far cheaper than CFD results. With fourteen points a surrogate of up to six independent variables can be created. Then as the number of independent variables increase as does the required sampling points. For this simple problem, the first six variables from table 4.3 were used to generate the surrogate model, and resulted in figure 4.12. The minimum of this data set was 180, and occurred at the extrema of each of the six variables (maximum for quarter degree sweep as it appears in the denominator of equation 4.1 and all else are minimized). Sadly, since only the first six variables are used, the weight gain from the final four is not optimized. The actual minimum occurs at 158.28, resulting in a 14% error. This is excellent considering only six of the ten variables were used, and 14 discrete points. This however can not be counted on since this solution was relatively simple. In order to improve the accuracy of the design all independent variables must be accounted for. To do this another surrogate can

be developed using the remaining variables. Such a surrogate's results are shown in figure 4.13. The resulting minimum is 147.2, giving an error of 7 %. Averaging the two together results in an accurate prediction of the final weight, 163.65 or 3 % error, which occurs at the exact place of the actual minimum. One issue to be aware of, if only the variables with the greatest non-linearity are used, then the optima could easily be exaggerated if the end values aren't sampled. Since kriging is basically a basis of sign waves, non-linear effects are exaggerated during extrapolation. The more non-linear a variable acts, the more benefit for optimized design on that variable, but if only those variables are used the surrogate will create a highly non-linear mapping function.

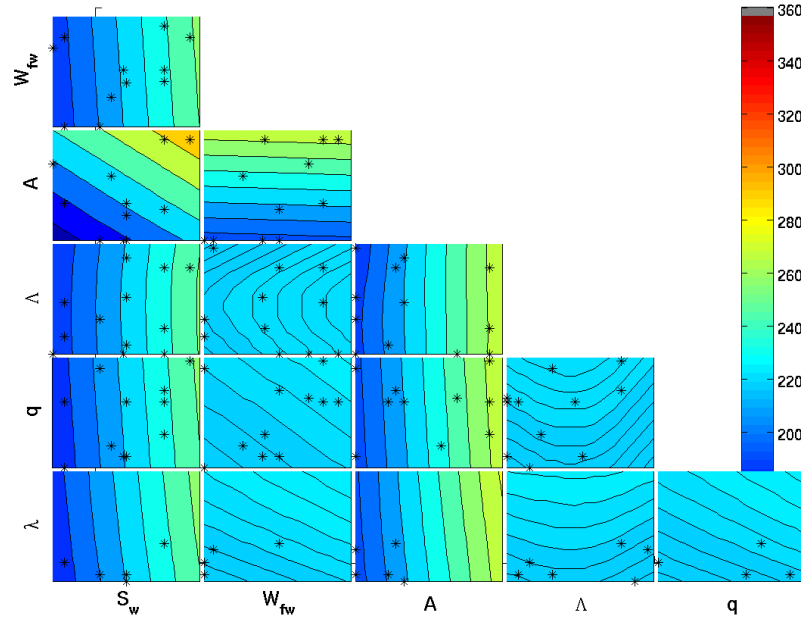


Figure 4.12: Tile plot of surrogate developed from six independent variables using 14 discrete points from equation 4.1. Each tile varies two variables while holding the eight other variables at baseline values as per table 4.3. Also shown are the chosen discrete points.

By combining surrogates in a linear fashion it may be possible to develop an accurate prediction of underlying behaviour of cross coupling of the solution. In future applications when the final result is not known, it is recommended to the reader to look at multiple cross coupling schemes since they are very cheap to build, and can lead to an excellent amount of insight into the system as a whole. Also, as the number of variables continue to increase using cross coupling becomes even

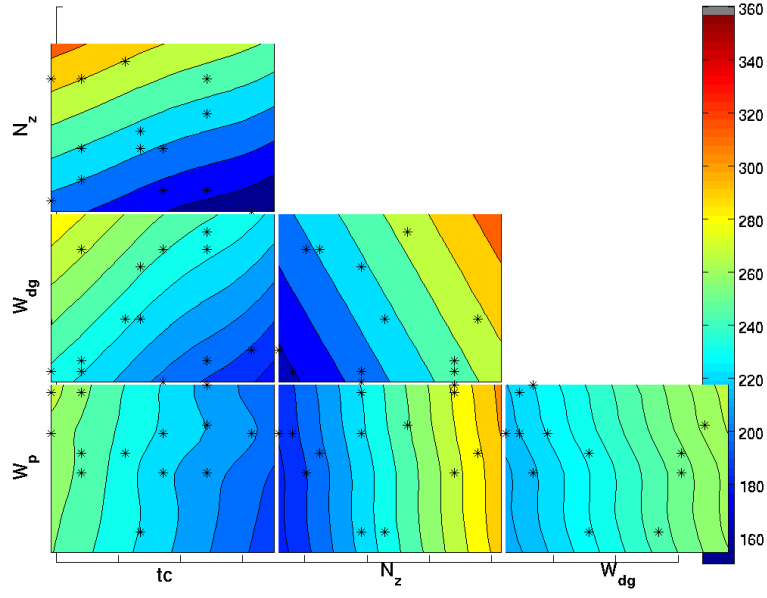


Figure 4.13: Tile plot of surrogate developed from the remaining four independent variables using 14 discrete points from equation 4.1. Each tile varies two variables while holding the eight other variables at baseline values as per table 4.3. Also shown are the chosen discrete points.

more important and requires more knowledge of how the system reacts with each variable.

Chapter 5

Surrogate Reaction Systems for Scramjet Design

5.1 Supersonic Flow Combustion for Various Upstream Conditions

There are a host of different design parameters that can be modified in the design of a supersonic combustion chamber. In order to facilitate the design applications behind surrogate modelling using kriging interpolation only some values were used. There are two different sets of values as well to allow for shorter surrogate computations. Table 5.1 gives the design variables for inlet conditions which depend on inlet geometry and atmospheric flight conditions as well as optimum conditions for robust design. Table 5.2 gives the combustor design conditions that were chosen to modify for this study. The range of each variable was ± 20 %.

Symbol	Description	Constant Value	Optimum Value
M_i	inlet Mach number	2.39	1.91
P	inlet pressure [N/m^2]	202650	202650
T	inlet temperature [K]	1360	1250

Table 5.1: Combustor Inlet Design Variables

As done in chapter 4 only a few parameters were first chosen to develop a solution space. This will be done twice, once for the inlet conditions and the other for the combustor geometry, values from table 5.1 and 5.2 respectively. Another set of surrogates could be made for the injectors however injector design is not the main subject of the design space and is excluded. However, the methods outlined in chapter 4 and again used here can be applied to any design variable. To begin only a few points were run in order to begin developing a surrogate and minimize the number

Symbol	Description	Constant Value
E_L	engine length (isolator/combustor) [m]	1.49
I_L	isolator length [m]	0.75
E_H	height of the engine [m]	0.0169
E_W	width of the engine [m]	0.194
θ_1	first combustor expansion angle [deg]	5.392
θ_2	second combustor expansion angle [deg]	3

Table 5.2: Combustor Design Variables

of points required to characterize the design optimized point as well as a design that is robust when working with off-design parameters. Thus the goal is to optimize a couple of parameters but also try to develop a design where slight deviations from the inlet conditions still results in a robust design. Originally 14 randomly chosen points that were constrained to be cover most of the sample space, are chosen to produce the initial surrogates shown in Figures 5.1 through 5.4 created using surfpack local kriging optimization for exit Mach number, carbon dioxide emissions, CO_2 , total exit temperature, T_O , and the total pressure ratio of exit to inlet, π_b . These parameters were chosen since they have direct effect on final performance. While carbon dioxide may not be a very important parameter for scramjets it is included since it is an important parameter in general aviation emissions and some interest in a scramjets emissions may be of interest, however due to scramjets not yet being common modes of transportation it is probably the least important of the four parameters. Total temperature is important as the total temperature increases less energy can be efficiently added to the system, also due to the material properties at the wall melting may be an issue. The exit Mach number is important because it must remain above 1 otherwise serious transonic losses will occur that are not being taken into account in the scramjet routine used. Also, the freestream Mach and nozzle exit Mach are directly related to the overall efficiency of the engine [32]. Total pressure at the exit divided by total pressure at the inlet is the most telling of all the variables about overall performance. Total pressure losses reduces the efficiency of the engine and thus reduces its overall performance [32].

Since each function is fairly non linear the initial 14 points are unlikely to have accurately captured the function. However, four new points were chosen to test the accuracy. These new points

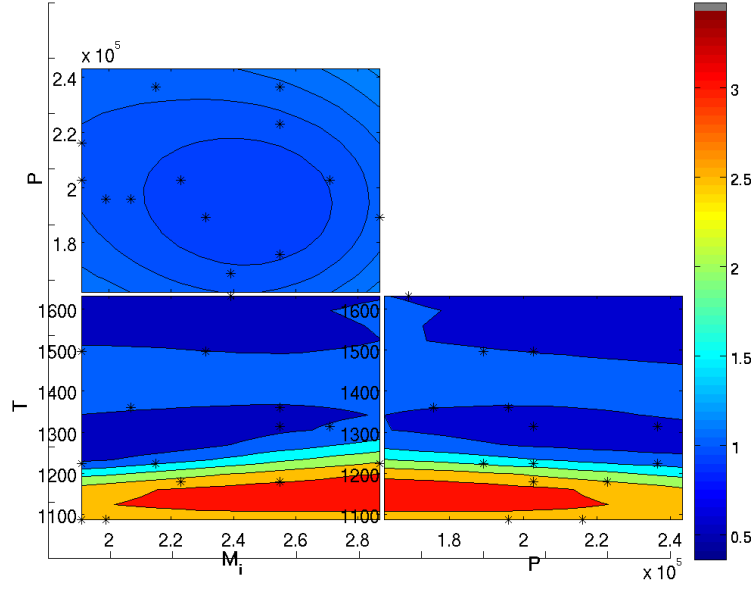


Figure 5.1: Tile plot of exit Mach number surrogate developed from variables in table 5.1 with values from table 5.2 held constant. Also shown are the 14 randomly chosen discrete points.

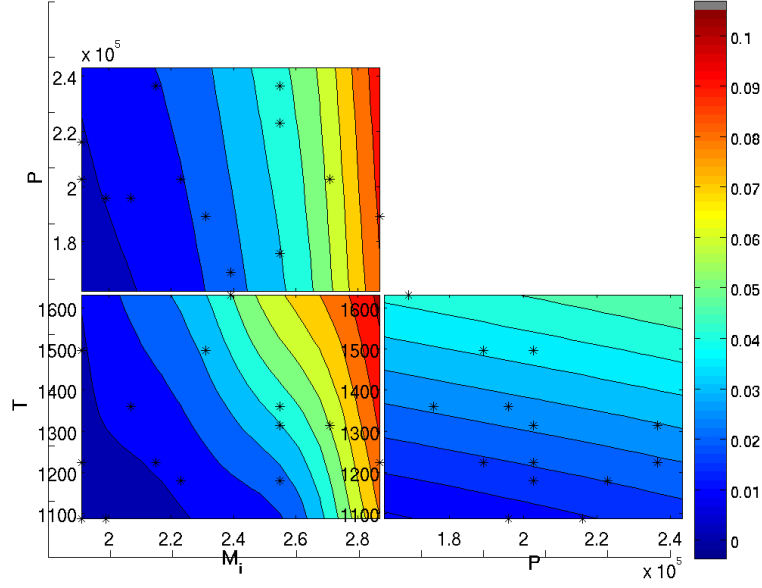


Figure 5.2: Tile plot of exit carbon dioxide emission surrogate developed from variables in table 5.1 with values from table 5.2 held constant. Also shown are the 14 randomly chosen discrete points.

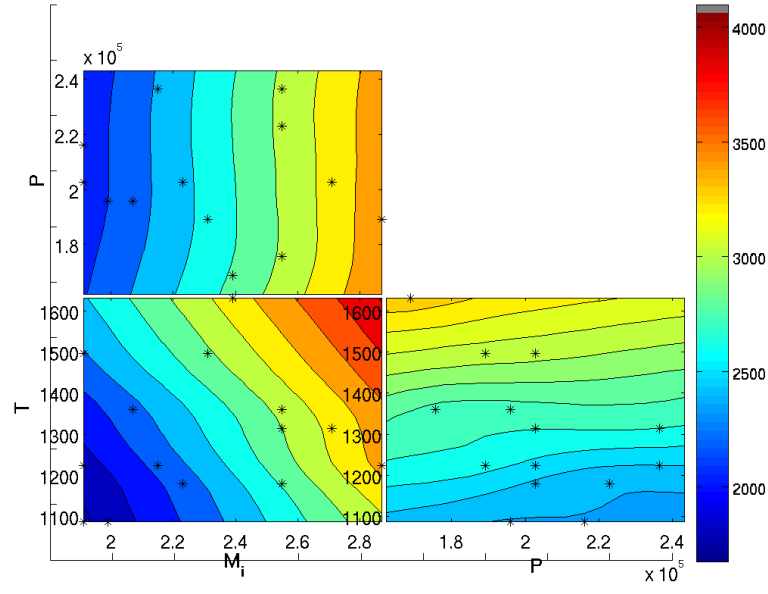


Figure 5.3: Tile plot of exit total temperature surrogate developed from variables in table 5.1 with values from table 5.2 held constant. Also shown are the 14 randomly chosen discrete points.

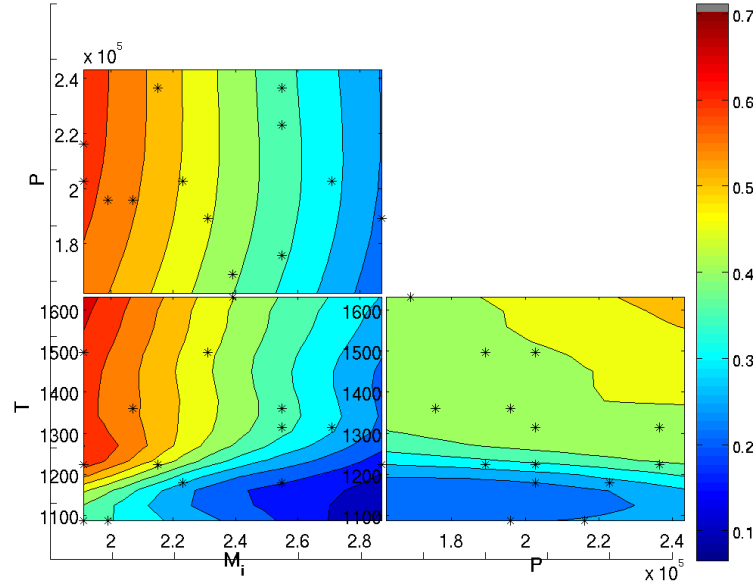


Figure 5.4: Tile plot of exit total pressure inlet to exit ratio surrogate developed from variables in table 5.1 with values from table 5.2 held constant. Also shown are the 14 randomly chosen discrete points.

are shown in figures 5.5 through 5.8, with their accuracy shown in table 5.3. Interestingly enough, not each model had a high level of error from a certain region. Thus as mentioned previously only 14 discrete points is only a good starting point and more points need to be added to develop a successful surrogate.

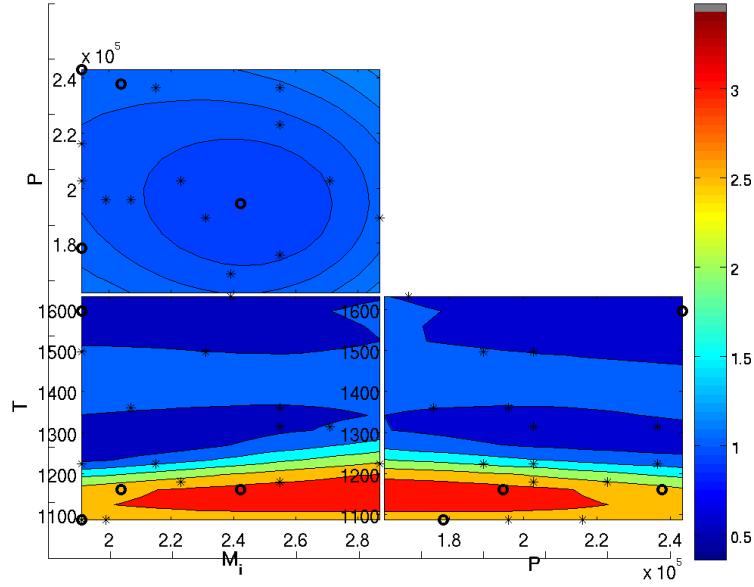


Figure 5.5: Tile plot of exit Mach number surrogate developed from variables in table 5.1 with values from table 5.2 held constant. Also shown are the 14 randomly chosen discrete points (stars) and the added 4 test locations (circles).

M_i	P	T	M_e % error	CO_2 % error	T_O % error	π_b % error
1.912	178320	1088	1	212	3	8
2.0395	237793	1160	9	43	10	19
2.4219	194540	1160	13	44	9	33
1.912	243200	1595	105	26	5	8

Table 5.3: Results for 4 Test Locations from Original 14 Point Altitude Parameter Surrogate

By adding another 4 points in carefully chosen places using near neighbour and sparse regions any periodic values and non-linearities can be captured. These points are shown in table 5.3. These points served the dual purpose of evaluation and due to high error in each of the models at different points these points were added to developing the surrogate. Figures 5.9 through 5.12 give this next

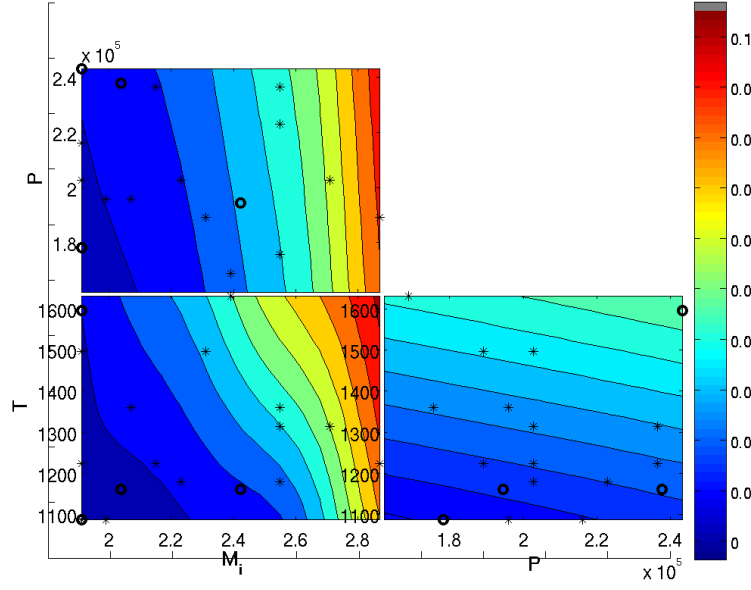


Figure 5.6: Tile plot of exit carbon dioxide emission surrogate developed from variables in table 5.1 with values from table 5.2 held constant. Also shown are the 14 randomly chosen discrete points (stars) and the added 4 test locations (circles).

step of approximation in the surrogate. Table 5.4 show that the average error for exit Mach number, CO_2 , T_O , and π_b dropped from 32% to 9%, 82% to 39%, 7% to 5%, and 17% to 5% respectively. Thus by applying carefully chosen points to a discrete data set the overall error can be significantly decreased. This process could easily be continued, however due to the low errors in the results for all but the CO_2 , which is unsurprising due to the stiff equation set CHEMKIN must solve, emissions the surrogate is accurate enough for characterizing the design space for the inlet condition variables. From here it can be determined that to minimize pressure losses minimizing the entrance Mach number is required with a temperature range of roughly 1200 - 1300 Kelvin. Inlet pressure effects are negligible on the pressure losses. This design choice also limits the total temperature values thus melting walls is less likely. Also while the CO_2 emissions error can be high, the overall trend shows that this design choice minimizes the emissions as well as seen in figure 5.10. However, the final and major caveat of this design choice is that due to the transonic Mach instabilities this design choice is very likely to lead to transonic flow. This region characterized by highly non-linear aerodynamics

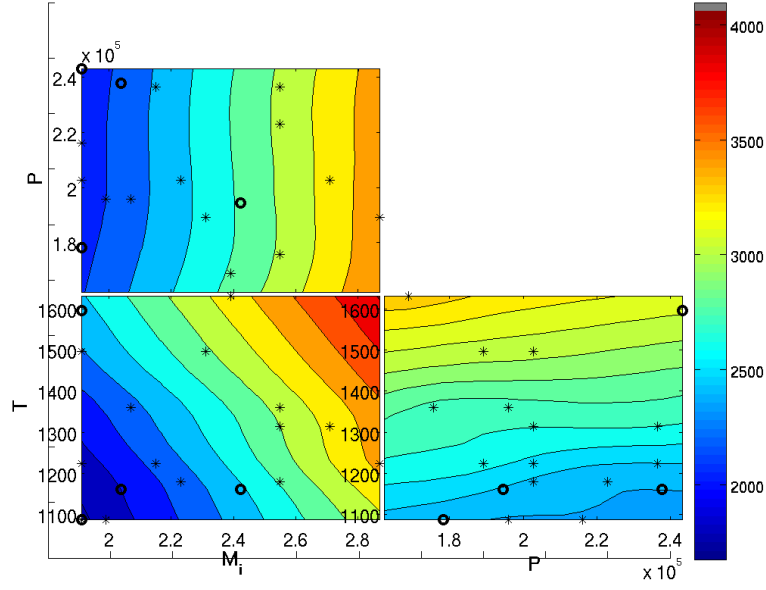


Figure 5.7: Tile plot of exit total temperature surrogate developed from variables in table 5.1 with values from table 5.2 held constant. Also shown are the 14 randomly chosen discrete points (stars) and the added 4 test locations (circles).

and losses [33]. For this reason, which is substantial, the inlet temperature should be restricted to 1250 K, unless higher inlet Mach numbers are necessary. This will prevent the exit Mach number from being reduced to one necessitating more complex analysis. Now that the operational envelope is characterized a specific combustor geometry for the combustion system will be analysed using these constant values as summarized in table 5.1. This process can be summarized by the flow chart 5.21.

M_i	P	T	M_e % error	CO_2 % error	T_O % error	π_b % error
2.39	221573	1523	2	31	3	5
2.678	202650	1487	1	40	3	1
2.7405	202650	1124	22	82	15	13
2	212650	1550	9	4	1	3

Table 5.4: Results for 4 Test Locations from updated 14 Point Altitude Parameter Surrogate with 4 Additional Points

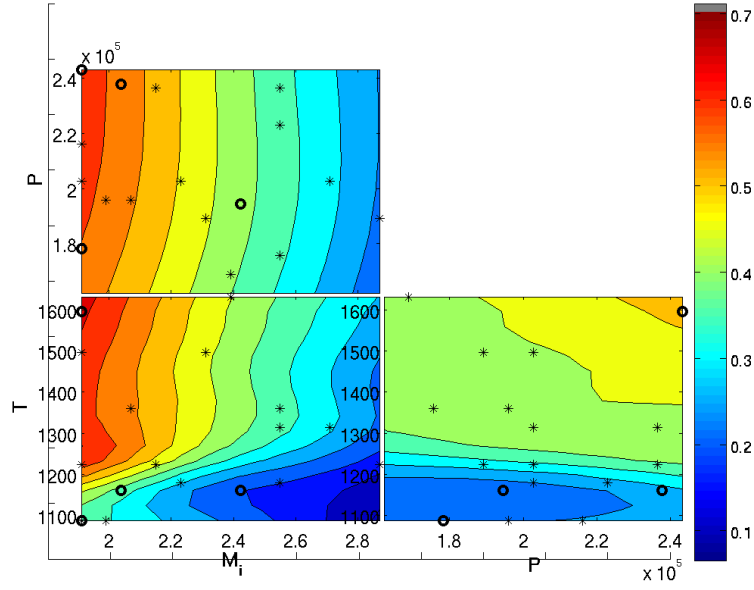


Figure 5.8: Tile plot of exit total pressure inlet to exit ratio surrogate developed from variables in table 5.1 with values from table 5.2 held constant. Also shown are the 14 randomly chosen discrete points (stars) and the added 4 test locations (circles).

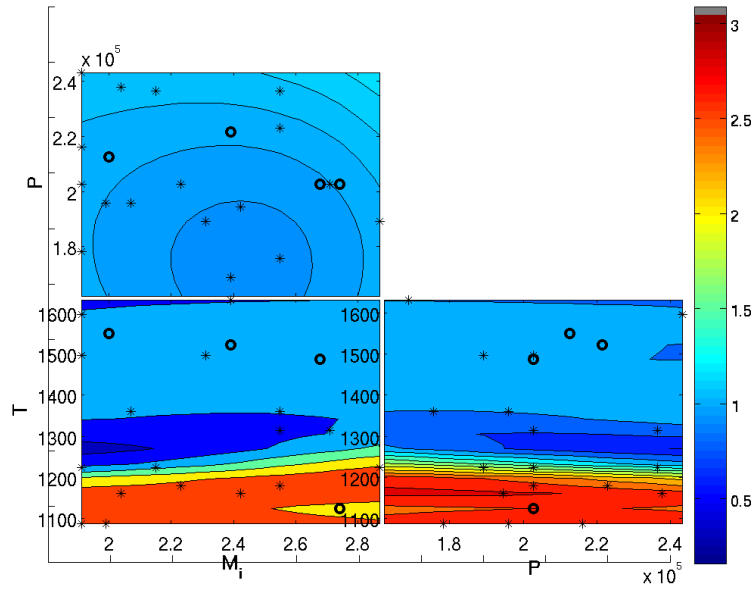


Figure 5.9: Tile plot of exit Mach number surrogate developed from variables in table 5.1 with values from table 5.2 held constant. Also shown are the 14 randomly chosen and 4 evaluated for previous testing discrete points (stars) and the added 4 test locations (circles).

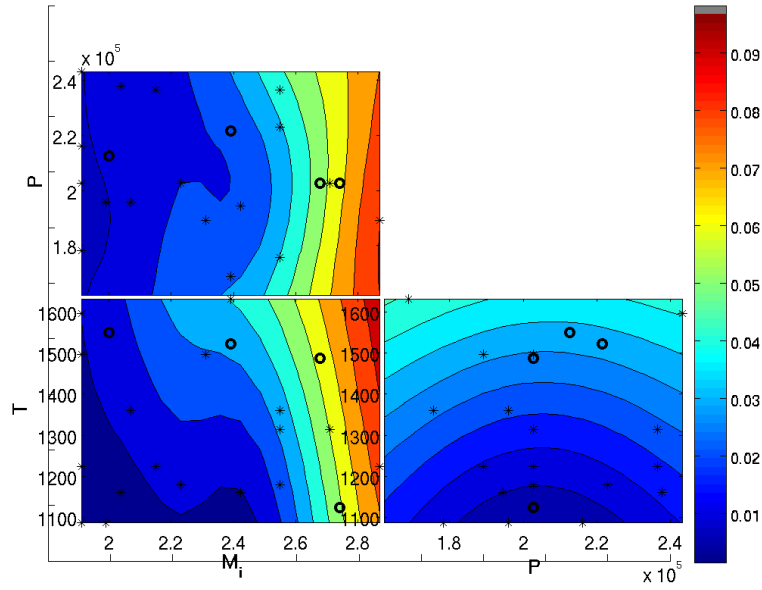


Figure 5.10: Tile plot of exit carbon dioxide emission surrogate developed from variables in table 5.1 with values from table 5.2 held constant. Also shown are the 14 randomly chosen and 4 evaluated for previous testing discrete points (stars) and the added 4 test locations (circles).

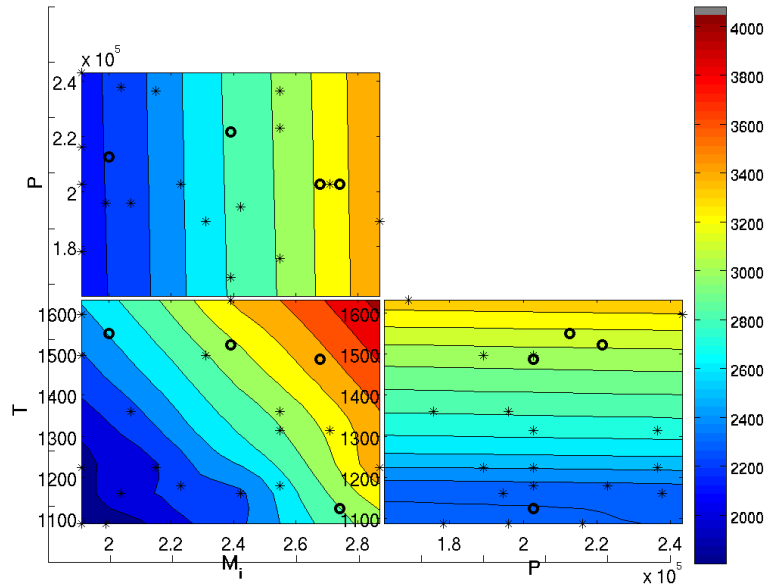


Figure 5.11: Tile plot of exit total temperature surrogate developed from variables in table 5.1 with values from table 5.2 held constant. Also shown are the 14 randomly chosen and 4 evaluated for previous testing discrete points (stars) and the added 4 test locations (circles).

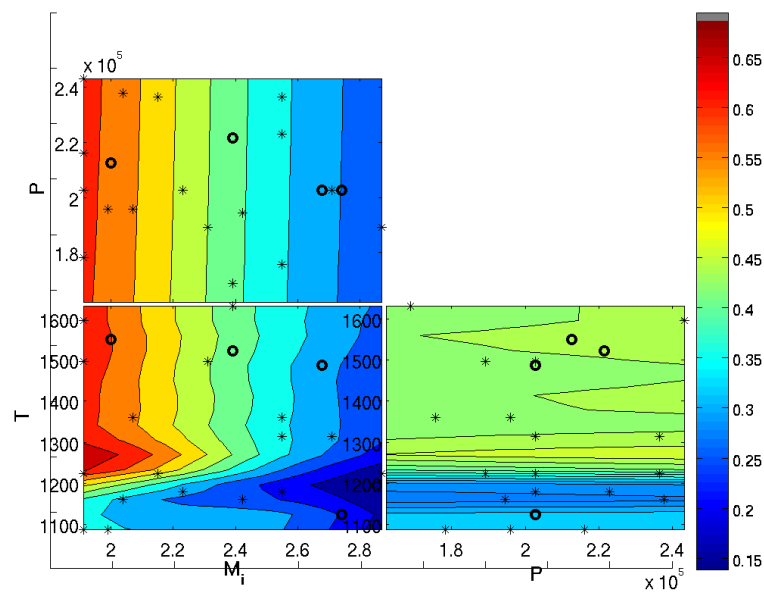


Figure 5.12: Tile plot of exit total pressure inlet to exit ratio surrogate developed from variables in table 5.1 with values from table 5.2 held constant. Also shown are the 14 randomly chosen and 4 evaluated for previous testing discrete points (stars) and the added 4 test locations (circles).

5.2 Supersonic Combustor Geometry Design

The second set of variables were limited to six in table 5.2 in order to reduce computational time in the surrogate. While running the program to get the discrete points is quick, the purpose of surrogate modelling is for use in computationally expensive models that take orders of hours or greater to compute. Using six variables on a personal laptop required roughly twenty minutes to generate the surrogates appearing in figures 5.13 to 5.16. Using 14 points an excellent understanding of the model can immediately become apparent. Based on these models the isolator gives the most dramatic changes in the response variables. Since there is little activity in most of the other variables, both near neighbour methods and far away points will be checked to insure accuracy and confirm there is not periodicity in the results. Multiple points were placed where the isolator was less than 0.7 since that is a region of large gradients in the model. The results in table 5.5 show that the overall error for each point is relatively high. This high error is unsurprising due to only 14 points were used with 6 independent variables. As done with the inlet conditions model these added four points will become the basis for a 18 point surrogate.

Figures 5.17 through 5.20 give the updated surrogate. The shape of the function has changed with these added points with regards to the sharp gradients due to the isolator. Based on the results from table 5.6, the surrogate is likely to accurately capture the overall function. The overall low errors could easily be due to chance, but since the points were randomly chosen and then examined to confirm they are different from other points the overall error in the system is expected to be averaged around 10%. To confirm this table 5.7 was generated which gives an average error of 13 %, 9 %, 1%, and 6% for exit Mach number, carbon dioxide, total exit temperature, and the

E_L	I_L	E_H	E_W	θ_1	θ_2	M_e % error	CO_2 % error	T_O % error	π_b % error
1.788	0.9	0.0135	0.1552	6.47	3.6	91	6	2	10
1.192	0.6	0.0203	0.2328	4.889	2.88	13	38	4	24
1.7694	0.8965	0.02101	0.18557	5.1415	3.1914	45	1	1	6
1.4374	0.74088	0.01413	0.16992	5.4417	2.4639	12	3	0	2

Table 5.5: Results for 4 Test Locations from Original 14 Point Engine Parameters Surrogate

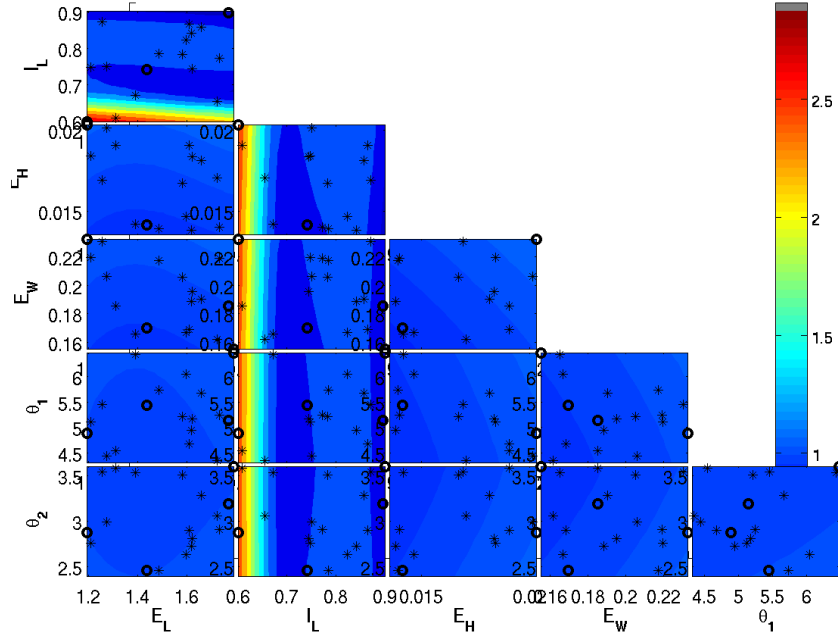


Figure 5.13: Tile plot of exit Mach number surrogate developed from variables in table 5.2 with values from table 5.1 held constant. Also shown are the 14 randomly chosen points (stars) and evaluation points (circles) from table 5.5.

exit to inlet pressure ratios respectively shown in table 5.8. Overall the error is low, however some regions of the are as much as 50 to 100 % off. The trust of the surrogate is relatively high due to the 4 points tested, but in reality some regions are far off from what the surrogate says. Since these regions are significantly less frequent resulting in a low average error the use of a surrogate to determine a solution space for 6 variables using only 18 points is exceptional. Normally this exhaustive of a confirmation would not be applicable due to the high cost of points generated from CFD or experiment. Instead when the error decreases to an acceptable margin depending on the circumstances the desired design point would be run to confirm its properties. In the case of robust designs for off design performance, multiple points would need to be run on and off design specifications. The design process thus can be significantly eased by using a sparse population set with a surrogate. Without the surrogate the number of computational points would be further apart, or take more hours, as the whole design space would need to be characterized with enough fidelity to allow the designer a optimum choice.

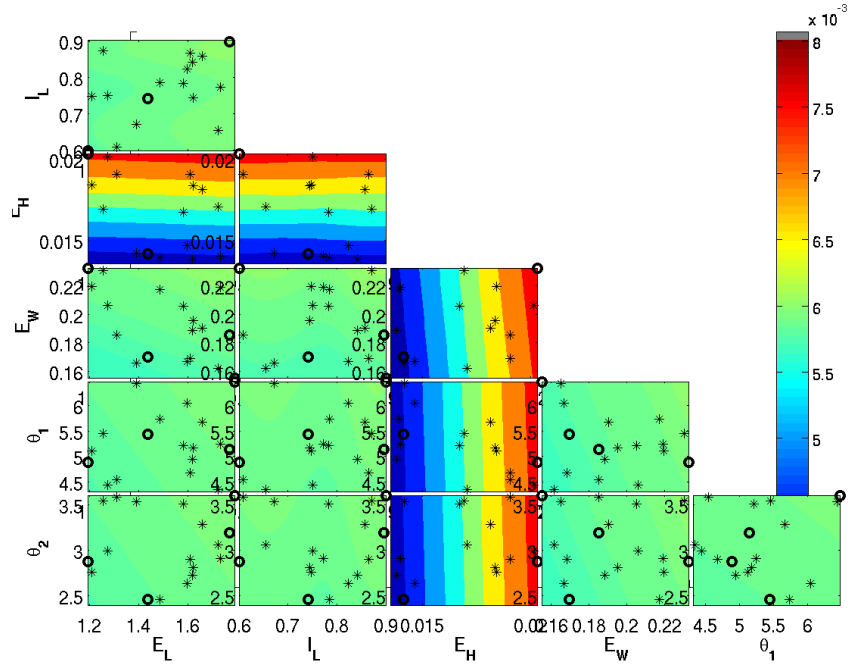


Figure 5.14: Tile plot of exit carbon dioxide emission surrogate developed from variables in table 5.2 with values from table 5.1 held constant. Also shown are the 14 randomly chosen points (stars) and evaluation points (circles) from table 5.5.

E_L	I_L	E_H	E_W	θ_1	θ_2	M_e % error	CO_2 % error	T_O % error	π_b % error
1.3508	0.85332	0.014552	0.22855	4.8847	3.3162	0.3	1.1	0.1	0.1
1.2836	0.70334	0.015826	0.17141	4.9996	3.1535	14.3	0.9	1.2	4.5
1.7137	0.71603	0.019722	0.19013	5.6925	2.5666	9.1	3.2	0.1	2.4
1.5062	0.60201	0.018512	0.16447	5.7059	3.5673	7.2	1.4	0.9	9.5

Table 5.6: Results for 4 Test Locations from 14 Point Engine Parameters Surrogate with 4 Additional Points

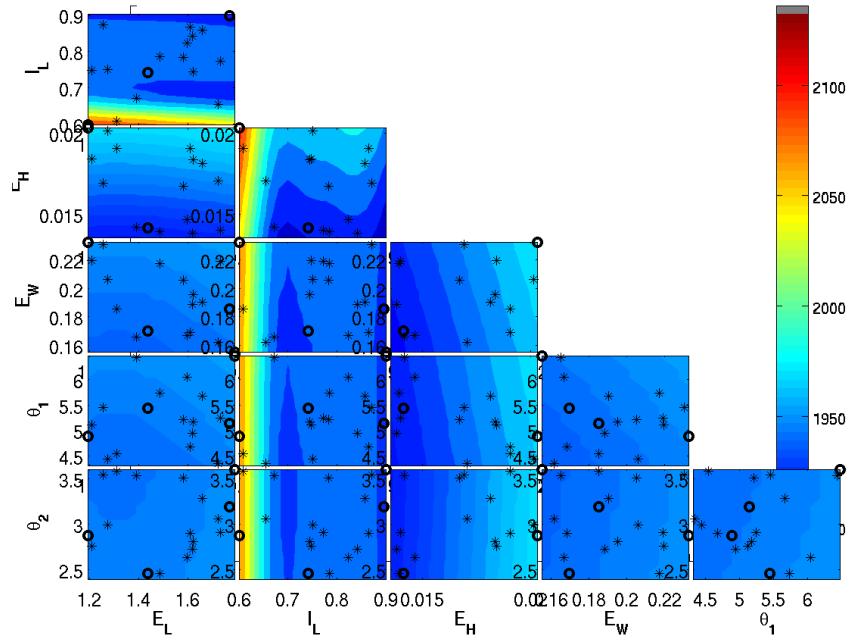


Figure 5.15: Tile plot of exit total temperature surrogate developed from variables in table 5.2 with values from table 5.1 held constant. Also shown are the 14 randomly chosen points (stars) and evaluation points (circles) from table 5.5.

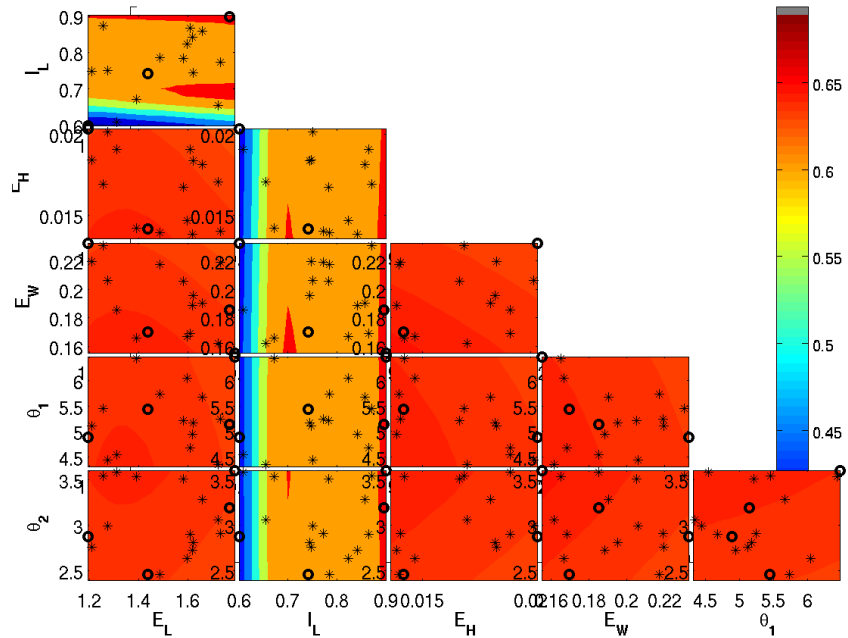


Figure 5.16: Tile plot of exit total pressure inlet to exit ratio surrogate developed from variables in table 5.2 with values from table 5.1 held constant. Also shown are the 14 randomly chosen points (stars) and evaluation points (circles) from table 5.5.

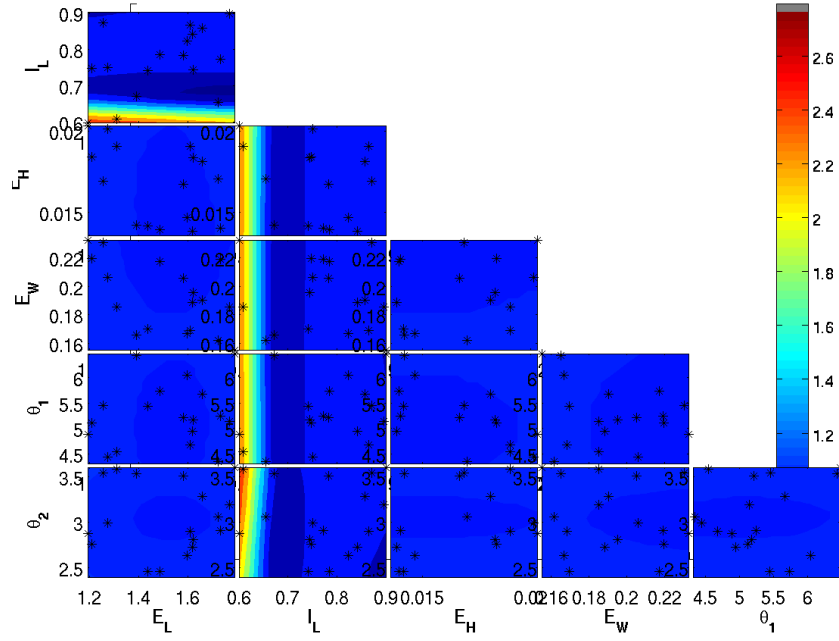


Figure 5.17: Tile plot of exit Mach number surrogate developed from variables in table 5.2 with values from table 5.1 held constant. Also shown are the 14 randomly chosen points and evaluation points (stars) from table 5.6.

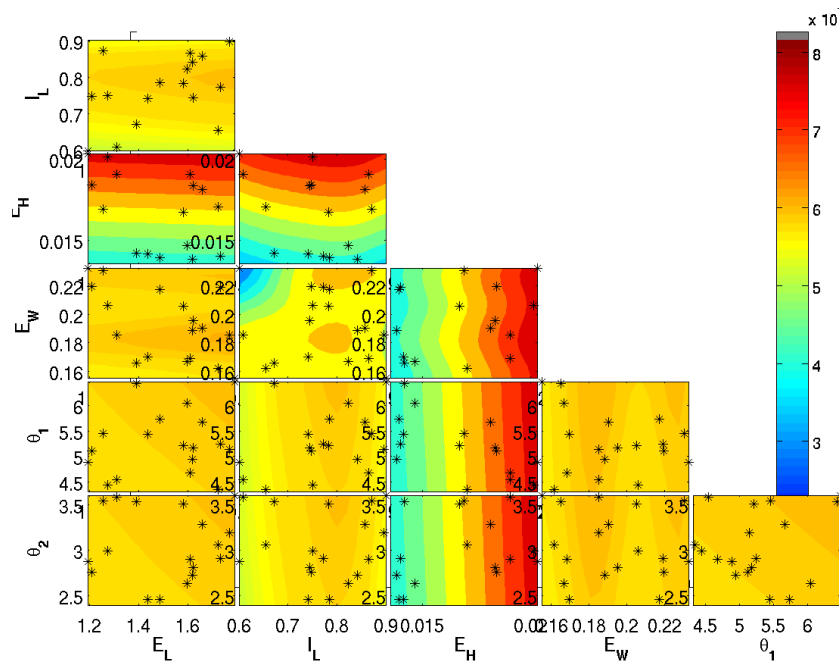


Figure 5.18: Tile plot of exit carbon dioxide emission surrogate developed from variables in table 5.2 with values from table 5.1 held constant. Also shown are the 14 randomly chosen points and evaluation points (stars) from table 5.6.

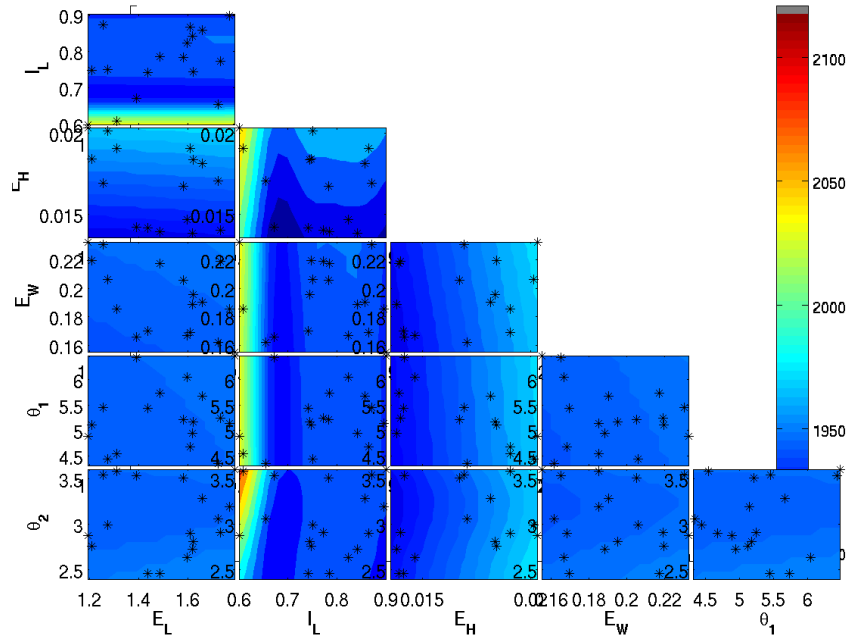


Figure 5.19: Tile plot of exit total temperature surrogate developed from variables in table 5.2 with values from table 5.1 held constant. Also shown are the 14 randomly chosen points and evaluation points (stars) from table 5.6.

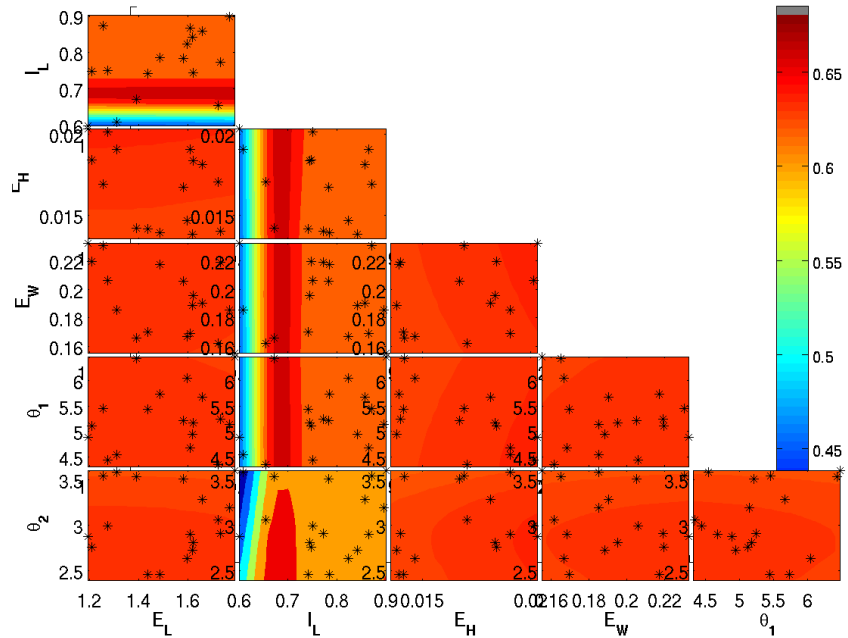


Figure 5.20: Tile plot of exit total pressure inlet to exit ratio surrogate developed from variables in table 5.2 with values from table 5.1 held constant. Also shown are the 14 randomly chosen points and evaluation points (stars) from table 5.6.

E_L	I_L	E_H	E_W	θ_1	θ_2	M_e % error	CO_2 % error	T_O % error	π_b % error
1.3482	0.78441	0.013883	0.20203	4.542	2.6733	1.1	2.3	0.1	0
1.2185	0.70872	0.016506	0.23004	5.7836	3.3653	8.7	10.8	0.9	3
1.6419	0.61486	0.01361	0.18876	5.3128	3.5833	50.4	2	5.3	37.8
1.3367	0.74687	0.019585	0.20911	4.7712	2.436	0.4	3.2	0.1	0.2
1.4557	0.65775	0.014849	0.21403	4.5261	3.0428	2	27.2	0.4	1.9
1.6019	0.63693	0.014151	0.18877	6.0899	2.5045	18	3.2	0.2	1.4
1.4061	0.66165	0.015598	0.20607	4.6911	3.3625	13.4	17.2	0.4	1.7
1.6309	0.64395	0.016603	0.16372	4.6664	3.587	33.9	3.2	3.3	18.1
1.4272	0.65672	0.014207	0.22766	5.75	2.4803	3.7	49.3	0.9	5.3
1.5993	0.6128	0.020249	0.16975	6.2426	3.5273	35.2	30.9	1.6	23.3
1.6116	0.79056	0.015765	0.17586	5.4277	2.4218	0.1	2.5	0.1	0
1.4556	0.68456	0.01553	0.21711	5.8292	3.2206	20.8	21.7	1.1	4.4
1.2037	0.76158	0.013939	0.19304	4.6449	3.3405	2.9	0	0.2	0.3
1.3892	0.80855	0.015536	0.21487	6.37	3.041	0.7	1.1	0.2	0.4
1.4449	0.74973	0.013833	0.18593	5.4802	3.4624	1.3	5.6	0.2	0.5
1.3531	0.76074	0.016937	0.17638	5.7797	3.4788	2.4	4.1	0	0.6
1.3094	0.73355	0.018667	0.15809	4.3925	3.1511	0.9	0.4	0.3	0.1
1.6817	0.63718	0.017786	0.20745	6.0589	2.5654	17	20.8	0.1	2.3
1.4482	0.74711	0.014128	0.18853	5.9282	2.6614	1.4	0.6	0.1	0.1
1.7211	0.8559	0.014067	0.19025	4.5728	2.6186	1.2	6.1	0.2	0.4
1.4251	0.86218	0.018774	0.20252	5.446	2.4502	4	0	0.1	0.8
1.6504	0.68109	0.019639	0.15981	5.0164	2.5283	35.6	3.9	1	7.3
1.4285	0.66254	0.017128	0.17971	5.4922	3.1397	5.7	1.1	0.3	1
1.6739	0.76949	0.014258	0.21516	5.1739	3.5276	0.6	3.4	0.1	0.7
1.642	0.79209	0.019102	0.20924	5.2089	2.8253	0	1.8	0	0.2
1.4169	0.72511	0.015806	0.16493	4.7034	2.8928	8.8	3.6	0.7	2.5
1.3207	0.66179	0.015507	0.1653	4.8644	3.5812	20	1.1	1.1	5.3
1.6631	0.88438	0.018565	0.16237	4.3579	3.5347	4.3	2.1	0	0.7
1.7578	0.62462	0.01359	0.15581	6.3058	3.212	44.2	1.7	3.7	26.9
1.3872	0.63171	0.013848	0.18803	5.7235	3.586	39.4	1.8	3.2	18.2
1.5921	0.64261	0.018035	0.20607	6.3251	3.3202	88	99.6	9.4	38.3
1.4534	0.64994	0.017599	0.2113	4.6663	2.804	27.5	26.2	0.6	5.2
1.6888	0.78629	0.017076	0.19642	6.3002	3.1949	1.6	4.2	0	0
1.6502	0.77211	0.018453	0.16364	6.0275	2.693	0	3.7	0.2	0.3
1.2917	0.61562	0.018301	0.20423	5.5589	2.7546	31.5	28.3	5.7	23.6
1.7057	0.87936	0.018802	0.16502	5.2627	3.2162	2.1	0.6	0	0.1
1.782	0.8186	0.015467	0.16562	4.8692	3.0334	1	0	0.1	0.1
1.4986	0.82135	0.018202	0.16285	5.9354	2.8939	0.9	1.4	0.1	0.1
1.719	0.61902	0.017283	0.16622	4.8068	3.1232	45.6	1.1	3.4	25.8
1.5425	0.85813	0.0162	0.16826	4.452	3.3006	0.6	0.3	0	0.3
1.2842	0.88032	0.013936	0.17043	5.9686	3.1002	2	6.1	0.3	0.6
1.3111	0.89532	0.018794	0.17984	5.7612	3.0622	4	3.1	0.4	0.8
1.4345	0.85768	0.015802	0.17975	5.8562	3.1003	1.6	0.5	0.1	0.1
1.6382	0.83567	0.017629	0.17208	5.6984	3.0142	1.3	2.1	0	0.1
1.684	0.75401	0.018531	0.17468	5.2174	2.4991	2	3.6	0.2	0.6

Table 5.7: Results for 45 Additional Test Locations from Original 14 Point Engine Parameters Surrogate with 4 Additional Points

	M_e % error	CO_2 % error	T_O % error	π_b % error
Minimum Error	0	0	0	0
Maximum Error	88	99.6	9.4	38.3
Average Error	13	9	1	6

Table 5.8: Average, Maximum, and Minimum, Results for 45 Additional Test Locations from Original 14 Point Engine Parameters Surrogate with 4 Additional Points

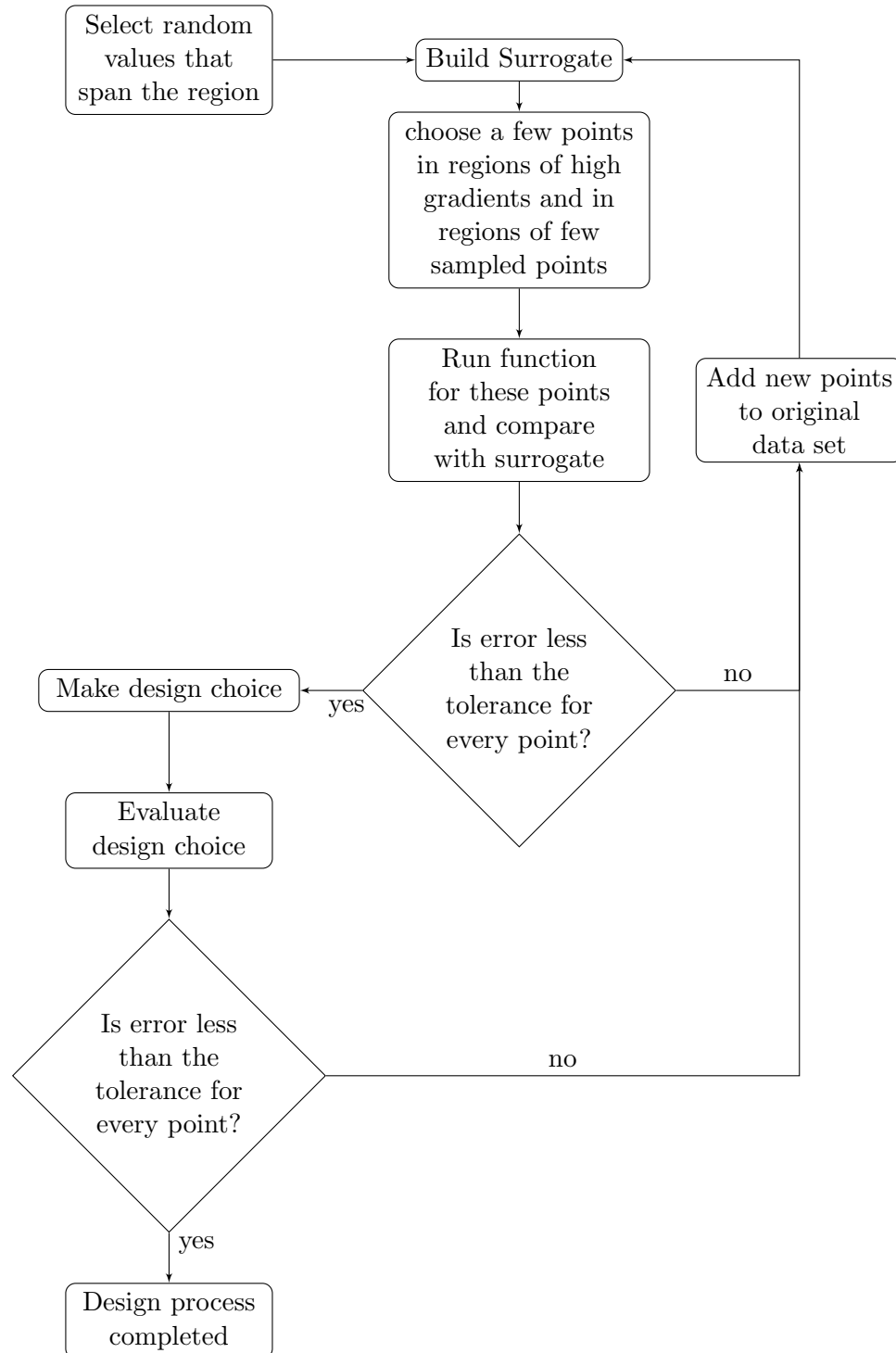


Figure 5.21: Logic Flow Diagram for Genetic Algorithm

5.3 Uncertainty Quantification

The concepts behind generating a design space can be directly applied to uncertainty quantification. Often the results themselves from CFD or experimental sources can be questionable due to uncertainty in the input parameters. Understanding uncertainty and choosing a design point that is unlikely to be affected by the inherent uncertainty in the design variables is necessary in order to design a robust scramjet. Sometimes a small deviation from a known parameter can have non linear results. In order to determine whether or not the Arrhenius reaction coefficients display highly non linear behaviour 14 chemical reaction input files were sent into CHEMKIN with the coefficients modified by up to plus or minus 5% to exhibit uncertainty. Then the scramjet code was run using unique points similar to the design space for the altitude operational envelope in section 5.1. From these results surrogates were generated and shown in figures 5.22 through 5.25. The samples forming these surrogates came from a uniform sparse data set and tested in areas of likely high error. Table 5.9 shows that the carbon dioxide is the only design parameter with low confidence. As seen in previous surrogates this variable is the most difficult to capture due to the chemistry involved and small magnitudes. Otherwise, the error captured by the tested points conclude that the surrogate is of acceptable accuracy to begin evaluations. Based on these figures it can be determined that a 5% plus or minus uncertainty in the Arrhenius reaction rates does not cause a major effect as the design parameters only vary few percent for the uncertain variables.

M_i	P	T	E_a	A_i	β_i	M_e % error	CO_2 % error	T_O % error	π_b % error
2.4856	201573	1270	0.1097	1.05	1.003	1.003	1.1819	2.6087	1.2829
2.35	199946	1160	0.3301	0.96	1.01	1.01	114.6808	1.5408	1.3173
2.74	221573	1523.2	0.0425	0.98	0.9998	1.03	8.9771	4.0355	1.4669
2.4856	201573	1270	0.0139	1.0004	0.997	1.03	271.319	0.668	0.0075

Table 5.9: Results for 4 Test Locations from 14 Point Uncertainty Parameters Surrogate. Variables E_a , A_i , β_i give the Uncertain Change from Initial CHEMKIN Input Files

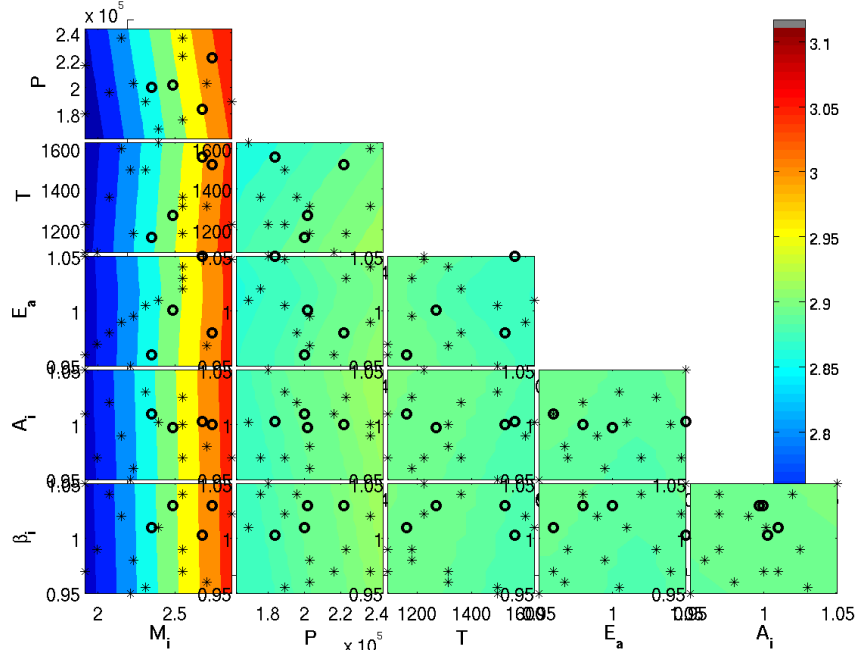


Figure 5.22: Tile plot of exit Mach number surrogate developed from variables in table 5.1 with values from table 5.2 held constant along with the uncertainty in the reaction parameters. Also shown are the 14 chosen points(stars) and evaluation points (circles) from table 5.9.

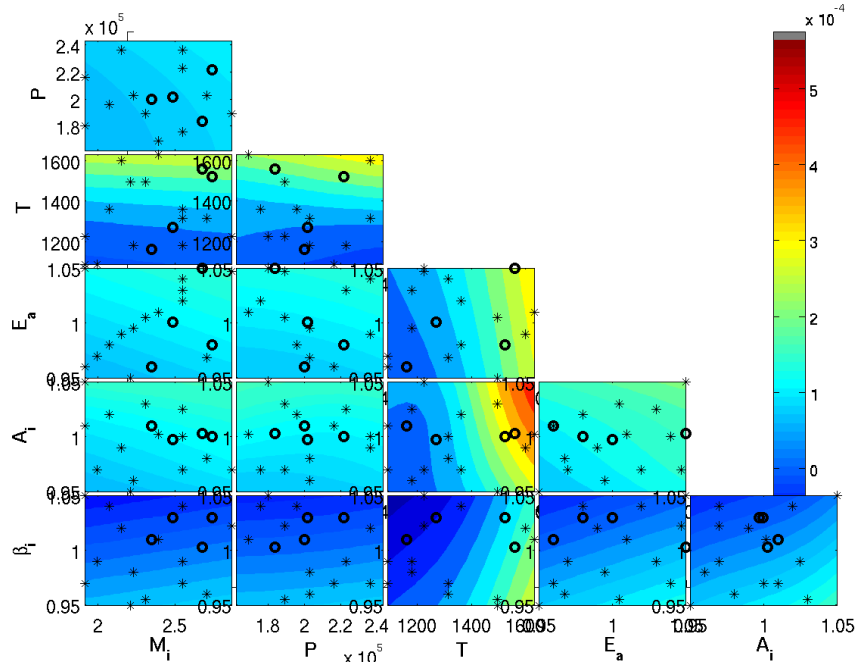


Figure 5.23: Tile plot of exit carbon dioxide emission surrogate developed from variables in table 5.1 with values from table 5.2 held constant along with the uncertainty in the reaction parameters. Also shown are the 14 chosen points(stars) and evaluation points (circles) from table 5.9.

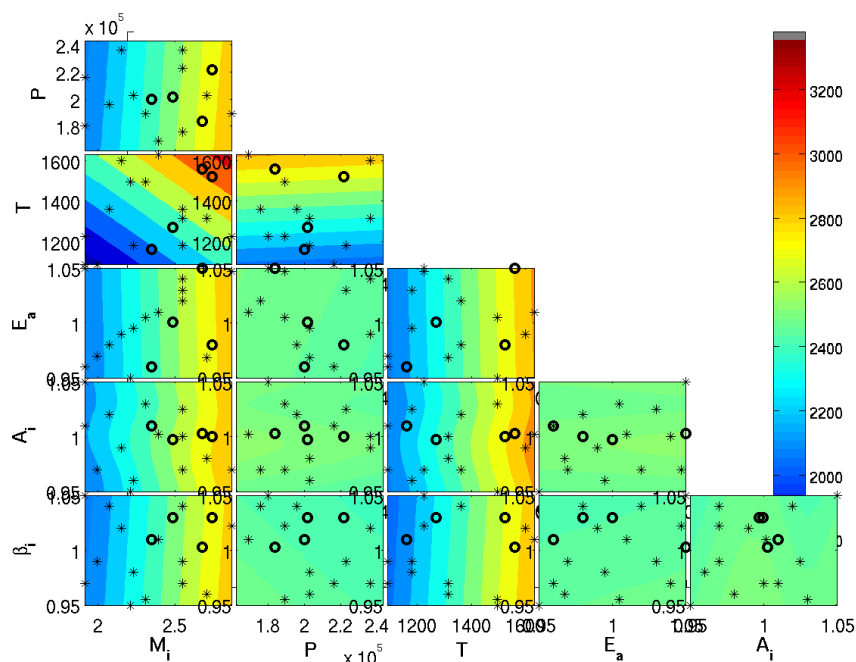


Figure 5.24: Tile plot of exit total temperature surrogate developed from variables in table 5.1 with values from table 5.2 held constant along with the uncertainty in the reaction parameters. Also shown are the 14 chosen points(stars) and evaluation points (circles) from table 5.9.

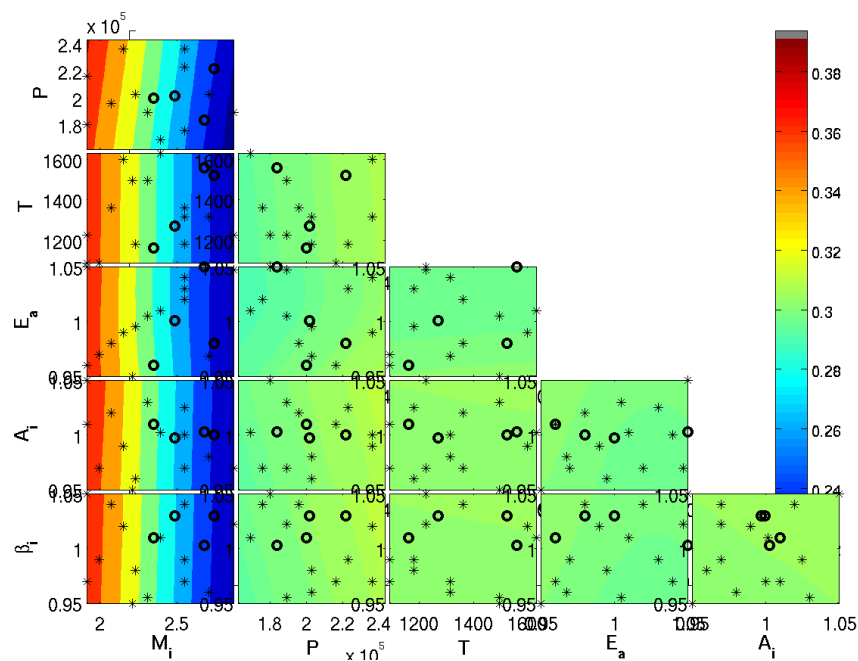


Figure 5.25: Tile plot of exit total pressure inlet to exit ratio surrogate developed from variables in table 5.1 with values from table 5.2 held constant along with the uncertainty in the reaction parameters. Also shown are the 14 chosen points(stars) and evaluation points (circles) from table 5.9.

Chapter 6

Conclusions

This thesis has provided the methods of developing surrogate models of a variety of different function types for any number of design variables. Including characterizing uncertainty in the known parameters. Using these methods a designer can take expensive data sets and fill in a design space to determine where an optimum design choice is and how it will behave outside its design envelope. Combining the surrogate model with good initial sampling with specific choices in evaluation points in regions of sparse data or high gradients will allow a minimum set of expensive data points for the development of a surrogate function.

In general sparsely populated data sets work best for kriging interpolation surrogate models, since too many points in a region lead to non linearities as the basis functions try to interpolate every point using a mathematical model. Thus noisy data sets with closely populated regions would be best modelled using regression schemes instead. Variables that have the most amount of non linearity are the most important for accurate characterization, however if only those variables are modelled the surrogate itself will be very non linear and extrapolated values can not be trusted. Capturing periodic behaviour is difficult for kriging surrogates if the frequency of the data set is not captured. Therefore in regions of apparent lack of gradients, near-neighbour sampling should be used. Overall, the more independent variables used in the surrogate require at least twice as many sparse data points for kriging, 6 variables need a minimum of 12 points. But two surrogates of three variables require two six variable surrogates but the surrogate convergence time is much shorter. The largest surrogate should be used in order to maximize cross coupling effects but too many and

the surrogate may take too much computational time or fail to converge. Surrogate modelling is an effective and accurate way to decrease design time and cost of any engineering project.

The next step in this work would be to continue to develop and understand the strengths and weaknesses of surrogate modelling. By developing surrogates of chemistry mechanisms using many reactions or empirical data would allow their use in computational chemistry to speed up computational fluid dynamics. Also, multi-fidelity models would allow the development of cheap computational routines, such as Reynolds Averaged Navier-Stokes, and combining it with a few expensive Large Eddies Simulations, effectively further decreasing the cost of each data set. Finally combining analytical models of combustion systems with discrete experimental inputs could allow full characterization of all 3 dimensions inside a test chamber.

Bibliography

- [1] Bishop, C. M., "Pattern Recognition and Feed-forward Networks," The MIT Encyclopedia of the Cognitive Sciences MIT Press, 629-632, 1999.
- [2] Michalewicz, Z., Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, Berlin Heidelberg New York, 1996.
- [3] Forrester, A. I., Sobester, A., and Keane, A. J., Engineering Design via Surrogate Modelling A Practical Guide, American Institute of of Aeronautics and Astronautics Inc and The Atrium,, Va, 2008.
- [4] Nusca, M. J., "Numerical Simulation of the Ram Accelerator Using a New Chemical Kinetics Mechanism," Journal of Propulsion and Power Vol3 18, No. 1 January-February, 2002.
- [5] Kundo, K., Penko, P., and Yang, S., "Reduced Reaction Mechanisms for Numerical Calculations in Combustion of Hydrocarbon Fuels," AIAA Paper No, 98-0803, 1997.
- [6] Giunta, A. A., Swiler, L. P., Brown, S. L., and Eldred, M. S., "The Surpack Software Library for Surrogate Modeling of Sparse Irregularly Spaced Multidimensional Data," AIAA Paper No, 2006-7049, 2006.
- [7] Zang, T. A., Hemsch, M. J., Hilburger, M. W., Kenny, S. P., Luckring, J. M., Maghami, P., Padula, S. L., and Stroud, W. J., "Needs and Opportunities for Uncertainty Based Multidisciplinary Design Methods for Aerospace Vehicles," NASA/TM-2002-211462, 2002.
- [8] Leslie, J. D. and Marren, D. E., "Hypersonic Test Capabilities Overview," U.S. Air Force T and E Days AIAA 2009-1702, 2009.
- [9] Warwick, G., "X-51A Team Eyes Results Of Scramjet Flight," Aviation Week June 1, 2010.
- [10] Covault, C., "X-51 scramjet flights poised to bridge air and space propulsion for space launch, Prompt Global Strike," Spaceflight Now, May 16, 2010.
- [11] Andrew R. Crowell, J. J. M. and Miller, B. A., "Surrogate Based Reduced-Order Aerothermodynamic Modeling for Structural Response Prediction at High Mach Numbers," 52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference AIAA 2011-2014, 2011.
- [12] Kim, H., Mullins, J., Ragon, S., Soremekun, G., and Sobieszczanski-Sobieski, J., "A Parallel Trade Study Architecture for Design Optimization of Complex Systems," 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference AIAA 2005-2201, 2005.

- [13] Waltrup, P. J., White, M. E., Zarlingo, F., and Gravlin, E. S., "History of U.S. Navy ramjet, scramjet, and mixed-cycle propulsion development," American Institute of Aeronautics and Astronautics, Inc. AIAA Paper 96-3152, 1996.
- [14] Tenne, Y. and Goh, C.-K., Adaptation, Learning, and Optimization, Volume 2 Computational Intelligence in Expensive Optimization Problems, 2010 Springer-Verlag Berlin Heidelberg, Singapore, 2010.
- [15] Zumberge, J., Doty, J. H., Wu, T., and Bash, M., "Development of Surrogate Models for an Aircraft Synchronous Generator," 50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition AIAA 2012-0492, 2012.
- [16] Viana, F. A. C., Picheny, V., and Haftka, R. T., "Using Cross Validation to Design Conservative Surrogates," AIAA JOURNAL Vol. 48, No. 10, October, 2010.
- [17] Aha, D. W., Kibler, D., and Albert, M. K., "Instance-Based Learning Algorithms," Machine Learning, 6, 37-66, 1991.
- [18] Farlow, J., Hall, J. E., McDill, J. M., and West, B. H., Differential Equations and Linear Algebra, Prentice Hall Second Edition, Upper Saddle River, 2007.
- [19] Selby, R. W., "Machine Learning Models for Predicting Attributes of Large-Scale Systems," 46th AIAA Aerospace Sciences Meeting and Exhibit AIAA 2008-886, 2008.
- [20] Valsa, B., Nair, R. V., and Kaimal, M. R., "A Neural Network Model for Assessing Software Quality for Mission Critical Applications," 54th International Astronautical Congress of the International Astronautical Federation, the International Academy of Astronautics, and the International Institute of Space Law, Bremen, Germany IAC-03-U.2.a.05, 2003.
- [21] Mendenhall, W. and Sincich, T., Statistics for Engineering and the Sciences, 3rd Edition, Dellen Publishing Company, New York, New York, 1992.
- [22] Taylor, J. R., An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurement, 2d Edition, University Science Books, Sausalito, CA, 1997.
- [23] Martin, J. D., "Robust Kriging Models," 51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference AIAA 2010-2854, 2010.
- [24] Giunta, A. A., Richards, M. D., Swiler, L. P., Brown, S. L., and Eldred, M. S., "Surfpack Version 1.0 User Manual," Sandia National Laboratories, 2011.
- [25] Lewis, P. and Stevens, J., "Nonlinear Modeling of Time Series using Multivariate Adaptive Regression Splines (MARS)," Naval Postgraduate School Monterey, California, 1990.
- [26] Kuo, K. K., Principles of Combustion, John Wiley and Sons, Inc second edition, New Jersey, 2005.
- [27] Toro, E. F., Riemann SOLvers and Numerical Methods for Fluid Dynamics, Springer third edition, New York, 2009.
- [28] Warner, D. D., "The Numerical Solution of the Equations of Chemical Kinetics," The Journal of Physical Chemistry, Vol. 81, No. 25, 1977.

- [29] Kee, R. J., Rupley, F. M., and Miller, J. A., “CHEMKIN-II: A Fortran Chemical Kinetics Package for the Analysis of Gas-phase Chemical Kinetics,” UC-405 Sandia National Laboratories SAND89-8009 Unlimited Release, 1989.
- [30] Lee, C.-M., Kundu, K., and Ghorashi, B., “Simplified Jet Fuel Reaction Mechanism for Lean Burn Combustion Application,” NASA Technical Memorandum 105940 AIAA-93-0021, 1993.
- [31] Raymer, D., Aircraft Design: A Conceptual Approach, Educational Series, 4th edition, American Institute of Aeronautics and Astronautics, Washington, DC, 2006.
- [32] Heiser, W. H. and Pratt, D. T., Hypersonic Airbreathing Propulsion, AIAA Education Series American Institute of Aeronautics and Astronautics, Inc., Washington, DC, 1994.
- [33] Anderson, J. D., Modern Compressible Flow, McGraw Hill Third Edition., New York, NY, 2004.