

AIRCRAFT MULTIDISCIPLINARY DESIGN  
OPTIMIZATION USING DESIGN OF EXPERIMENTS  
THEORY AND RESPONSE SURFACE MODELING  
METHODS

By  
Anthony A. Giunta

A DISSERTATION SUBMITTED TO THE FACULTY OF  
VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY  
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF  
DOCTOR OF PHILOSOPHY  
IN  
AEROSPACE ENGINEERING

APPROVED:

Bernard Grossman, Chair  
Raphael T. Haftka  
William H. Mason  
Layne T. Watson  
Eugene M. Cliff

May 1997  
Blacksburg, Virginia

**Keywords:** High-Speed Civil Transport, Aerodynamics, Parallel Computing

**AIRCRAFT MULTIDISCIPLINARY DESIGN OPTIMIZATION  
USING DESIGN OF EXPERIMENTS THEORY AND RESPONSE  
SURFACE MODELING METHODS**

by

Anthony A. Giunta

Committee Chairman: Bernard Grossman  
Aerospace and Ocean Engineering

(ABSTRACT)

Design engineers often employ numerical optimization techniques to assist in the evaluation and comparison of new aircraft configurations. While the use of numerical optimization methods is largely successful, the presence of numerical noise in realistic engineering optimization problems often inhibits the use of many gradient-based optimization techniques. Numerical noise causes inaccurate gradient calculations which in turn slows or prevents convergence during optimization. The problems created by numerical noise are particularly acute in aircraft design applications where a single aerodynamic or structural analysis of a realistic aircraft configuration may require tens of CPU hours on a supercomputer. The computational expense of the analyses coupled with the convergence difficulties created by numerical noise are significant obstacles to performing aircraft multidisciplinary design optimization. To address these issues, a procedure has been developed to create two types of noise-free mathematical models for use in aircraft optimization studies. These two methods use elements of statistical analysis and the overall procedure for using the methods is made computationally affordable by the application of parallel computing techniques. The first modeling method, which has been the primary focus of this work, employs classical statistical techniques in response surface modeling and least squares surface fitting to yield polynomial approximation models. The second method, in which only a preliminary investigation has been performed, uses Bayesian statistics and an

adaptation of the Kriging process in Geostatistics to create exponential function-based interpolating models. The particular application of this research involves modeling the subsonic and supersonic aerodynamic performance of high-speed civil transport (HSCT) aircraft configurations. The aerodynamic models created using the two methods outlined above are employed in HSCT optimization studies so that the detrimental effects of numerical noise are reduced or eliminated during optimization. Results from sample HSCT optimization studies involving five and ten variables are presented here to demonstrate the utility of the two modeling methods.

# Acknowledgments

I was fortunate to have a unique educational experience in that I participated in a research project where a group of graduate students and faculty members met weekly to discuss the progress and difficulties of their respective portions of the project. This environment fostered much interaction both among the students, and between each student and the various faculty members. This experience has done much to further my academic and professional growth.

My advisor and committee chairman, Dr. Bernard Grossman, was a constant source of advice and encouragement throughout my graduate studies. He gave me numerous opportunities to author technical papers and to present research results at engineering and scientific conferences. These experiences have done much to enhance my professional growth and I am grateful for all of Dr. Grossman's efforts. Also, I am deeply indebted to Dr. Raphael Haftka who, on occasions too numerous to count, provided helpful insights into an array of difficult research-related problems in engineering and mathematics. For the past two years Dr. Haftka remained active in our weekly group meetings (via teleconferencing, fax machines, and the World Wide Web) while a faculty member at the University of Florida. His extraordinary efforts are greatly appreciated. Dr. William Mason spent much time in his office with me confronting interesting aspects of applied aerodynamics that often appeared in this research project. For all of his assistance and for indelibly marking the phrase "monkey burned, monkey learns" into my memory, I am most grateful. Dr. Layne Watson has played a key role in my academic development through his advice on issues of computer science and mathematics. Through working with him I have become more aware of the tools offered by applied mathematics which may be employed to solve

engineering problems. In addition, Dr. Watson introduced me to the software package *Mathematica*, and its myriad applications. Dr. Eugene Cliff's extensive background in theoretical and applied optimization, as well as his outsider's perspective on this project, have helped me to view this research within the larger scope of numerical optimization. For his assistance I am grateful.

Financial support for this research project was provided through a grant from the High Performance Computing and Communications Program at the NASA Langley Research Center. I am thankful to our contract monitor, Dr. Perry Newman, and the other researchers at Langley who assisted with this project.

It is difficult to express the profound sense of gratitude that I feel towards my family and friends who have been unfailingly supportive throughout my educational career. Foremost, I wish to thank my parents who instilled within me a constant desire for knowledge. Also contributing in no small way to my educational development are the many friends I have made at Virginia Tech over the past several years, especially those past and current residents of the "Sunroom" (AOE Dept. Computer Laboratory) with whom I have shared so many experiences. I also wish to express my deepest thanks and love to my fiancée Rachel Knudsen. Rachel, herself a Ph.D. candidate in Materials Engineering Science at Virginia Tech, has been my friend and confidant as I progressed through graduate school. In the past year she has managed to arrange many of the details of our wedding while diligently pursuing her own research work. She is truly a remarkable woman.

Finally, I wish to particularly acknowledge two people without whom I certainly would not be writing this today. These are my paternal grandfather, Anthony Giunta, for whom I am named, whose original financial planning allowed me to attend Virginia Tech as an undergraduate, and my maternal grandfather, Michael Manno, who first introduced me to the fascinating world of aircraft and flight. It is in their memory that this work is dedicated.

# Contents

<b>List of Tables</b>	<b>xi</b>
<b>List of Figures</b>	<b>xiii</b>
<b>Nomenclature</b>	<b>xvi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Motivation . . . . .	2
1.3 Methodology . . . . .	4
<b>2 HSCT Optimization Problem Formulation</b>	<b>9</b>
2.1 Analysis and Optimization Software . . . . .	10
2.2 HSCT Configuration Definition . . . . .	11
2.3 Constraint Formulation . . . . .	12
2.4 Optimization Problem Formulation . . . . .	13
2.5 Five Variable HSCT Optimization Problem . . . . .	13
2.6 Ten Variable HSCT Optimization Problem . . . . .	15
<b>3 Numerical Noise Issues and the VCRSM Method</b>	<b>28</b>
3.1 Examples of Numerical Noise . . . . .	29
3.2 Description of the VCRSM Method . . . . .	31
3.2.1 VCRSM and the Five Variable Optimization Problem . . . . .	31
3.2.2 VCRSM and the Ten Variable Optimization Problem . . . . .	32

<b>4</b>	<b>Response Surface Modeling Methods</b>	<b>36</b>
4.1	Response Surface Modeling . . . . .	37
4.1.1	RSM and Physical Experiments . . . . .	37
4.1.2	RSM and Computer Simulations . . . . .	38
4.1.3	Polynomial Models for RSM . . . . .	39
4.2	ANOVA and Regression Analysis . . . . .	41
4.3	Design of Experiments Theory . . . . .	42
4.3.1	Full Factorial Experimental Designs . . . . .	42
4.3.2	Central-Composite Experimental Designs . . . . .	43
4.3.3	<i>D</i> -Optimal Experimental Designs . . . . .	43
4.4	Design and Analysis of Computer Experiments . . . . .	45
4.4.1	Mathematics of DACE Modeling . . . . .	46
4.4.2	Estimation of the Correlation Parameter . . . . .	48
4.4.3	Using Multi-Fidelity Data in DACE Modeling . . . . .	49
<b>5</b>	<b>Using the VCRSM Method in HSCT Optimization</b>	<b>54</b>
5.1	Description of Steps in the VCRSM Method . . . . .	54
5.1.1	Initial HSCT Configuration, Step 1 . . . . .	54
5.1.2	Design Space Boundaries, Step 2 . . . . .	55
5.1.3	Full Factorial Experimental Design, Step 3 . . . . .	55
5.1.4	Low Fidelity Analyses, Step 4 . . . . .	56
5.1.5	Reduced Design Space, Step 5 . . . . .	56
5.1.6	<i>D</i> -Optimal Experimental Design, Step 6 . . . . .	56
5.1.7	Medium Fidelity Analyses, Step 7 . . . . .	57
5.1.8	Creating Response Surface Models or DACE Models, Step 8 . . . . .	57
5.1.9	HSCT Optimization, Step 9 . . . . .	57
5.1.10	Analysis of Optimal HSCT Configuration, Step 10a . . . . .	58
5.1.11	Define a New Design Space, Step 10b . . . . .	58
5.2	Variations of the VCRSM Method . . . . .	58
5.2.1	Kaufman's VCRSM Approach . . . . .	59
5.2.2	Crisafulli's VCRSM Approach . . . . .	60

<b>6</b>	<b>Results of HSCT Optimization Trials</b>	<b>63</b>
6.1	Five Variable HSCT Optimization Problem . . . . .	63
6.1.1	Low Fidelity HSCT Analyses . . . . .	63
6.1.2	Medium Fidelity HSCT Analyses . . . . .	63
6.1.3	HSCT Optimization Results . . . . .	65
6.1.4	Computational Expense . . . . .	66
6.2	Ten Variable HSCT Optimization Problem . . . . .	67
6.2.1	Low Fidelity HSCT Analyses . . . . .	67
6.2.2	Medium Fidelity HSCT Analyses . . . . .	68
6.2.3	HSCT Optimization . . . . .	69
6.2.4	Post-Optimality Investigations . . . . .	71
6.2.5	Computational Expense . . . . .	72
6.2.6	VCRSM Versus Traditional Optimization Methods . . . . .	73
<b>7</b>	<b>Test Problems Using DACE Modeling Methods</b>	<b>91</b>
7.1	Creation of Test Problems . . . . .	91
7.1.1	Selection of Functions Employed in the Test Problems . . . . .	91
7.1.2	Evaluation of Modeling Accuracy . . . . .	92
7.2	One Variable Test Problem . . . . .	93
7.2.1	Data Selection . . . . .	93
7.2.2	Results . . . . .	93
7.3	Five Variable Test Problem . . . . .	94
7.3.1	Data Selection . . . . .	94
7.3.2	Model Definition . . . . .	94
7.3.3	Results . . . . .	95
7.4	Ten Variable Test Problem . . . . .	95
7.4.1	Data Selection . . . . .	95
7.4.2	Model Definition . . . . .	95
7.4.3	Results . . . . .	96
7.5	Summary of Test Problem Results . . . . .	96
7.6	Variable-Complexity Modeling Test Problems . . . . .	97



7.6.1	One Variable Test Problem . . . . .	98
7.6.2	Data and Model Selection - Five Variable Test Problem . . . . .	99
7.6.3	Results - Five Variable Test Problem . . . . .	99
<b>8</b>	<b>HSCT Optimization Using DACE Modeling</b>	<b>114</b>
8.1	Five Variable HSCT Optimization Problem . . . . .	114
8.2	Application of DACE Modeling . . . . .	115
8.3	Application of Multi-Fidelity DACE Modeling . . . . .	115
8.4	Discussion of Results . . . . .	117
8.5	Discussion of DACE Model Sensitivity . . . . .	117
8.5.1	Sensitivity of DACE Parameters to Scaling . . . . .	117
8.5.2	Implications for DACE Models Used in this Study . . . . .	119
<b>9</b>	<b>Parallel Computing and the VCRSM Method</b>	<b>130</b>
<b>10</b>	<b>Concluding Remarks</b>	<b>135</b>
10.1	Summary . . . . .	135
10.2	Future Work . . . . .	136
10.2.1	Variable-Complexity Modeling . . . . .	136
10.2.2	Parallel Computing . . . . .	137
10.2.3	Design of Experiments Theory . . . . .	137
10.2.4	Polynomial Response Surface Modeling . . . . .	138
10.2.5	DACE Modeling . . . . .	138
	<b>Bibliography</b>	<b>139</b>
<b>A</b>	<b>User's Guides for HSCT Software</b>	<b>148</b>
A.1	Software Modifications - Five Variable HSCT Optimization Problem . . . . .	148
A.1.1	aero.c . . . . .	149
A.1.2	dataio.c . . . . .	149
A.1.3	ddot1.f - ddot6.f . . . . .	149
A.1.4	dotcnt1.f . . . . .	149
A.1.5	main.c . . . . .	149

A.1.6	<code>functions_05DV.c</code> . . . . .	151
A.1.7	Using the Modified Software in the VCRSM Method - Five Variables . . . . .	151
A.2	Software Modifications - Ten Variable HSCT Optimization Problem .	152
A.2.1	<code>aero.c</code> . . . . .	152
A.2.2	<code>main.c</code> . . . . .	152
A.2.3	<code>functions_10DV.c</code> . . . . .	153
A.2.4	Using the Modified Software in the VCRSM Method - Ten Variables . . . . .	153
A.3	Software Modifications - DACE Modeling . . . . .	153
A.3.1	Using DACE Modeling in the VCRSM Method . . . . .	154
A.3.2	Source Code for <code>dace_eval.f</code> . . . . .	154
<b>B</b>	<b>Error Estimation Methods</b>	<b>164</b>
<b>Vita</b>		<b>166</b>

# List of Tables

2.1	Analysis and optimization tools for HSCT design. . . . .	16
2.2	Twenty-nine HSCT variables and typical values. . . . .	17
2.3	Constraints for the 29 variable HSCT optimization problem. . . . .	18
2.4	Five HSCT wing variables in the example problem. . . . .	19
2.5	Constraints for the five variable HSCT wing optimization problem. . . . .	20
2.6	Initial values for the ten HSCT wing variables and the allowable range on each variable. . . . .	21
2.7	Constraints for the ten variable HSCT optimization problem. . . . .	22
6.1	Accuracy of the response surface models for the five variable HSCT optimization. . . . .	75
6.2	Initial and optimal values in trial 1 of the five variable HSCT optimization problem. . . . .	76
6.3	Important linear and quadratic terms in the response surface models for the ten variable HSCT optimization problem (interaction terms not shown). . . . .	77
6.4	Accuracy of the estimated responses in the ten variable reduced design space based on analyses of the 132 <i>D</i> -optimal HSCT configurations. . . . .	78
6.5	Initial and optimal values in trial 1 of the ten variable HSCT optimization problem. . . . .	79
6.6	Optimization results for the two initial HSCT configurations, both with and with the response surface models. . . . .	80

6.7	Response surface modeling error estimates based on analyses of 150 randomly selected HSCT configurations (10 variable HSCT optimization problem). . . . .	81
7.1	Modeling errors for Cases 1 and 2 for the one variable test problem. .	101
7.2	Modeling errors for Cases 1 and 2 for the five variable test problem. .	102
7.3	Modeling errors for Cases 1 and 2 for the ten variable test problem. .	103
7.4	A comparison of the modeling errors between the $k = 1$ stage DACE model (three samples of the noisy test function) and the $k = 2$ stage DACE model (two samples of the smooth test function) for Cases 1 and 2 of the one variable test problem. . . . .	104
7.5	A comparison of the modeling errors between the $k = 1$ stage DACE model (50 samples of the noisy test function) and the $k = 2$ stage DACE model (three samples of the smooth test function) for Cases 1 and 2 of the five variable test problem. . . . .	105
8.1	Initial and optimal variable values for trials 1-3 of the five variable HSCT optimization problem using the DACE model for $C_{Dwave}$ . . . .	120
8.2	Lower and upper bounds on the HSCT optimization problem using a multi-fidelity DACE model for $C_{Dwave}$ . . . . .	121
8.3	Initial and optimal variable values used in the five variable HSCT optimization problem using the high fidelity DACE model for $C_{Dwave}$ . Both of the optimization cases were started from the trial 1 initial HSCT configuration. . . . .	122
8.4	Sample data to demonstrate the effect of scaling in DACE models. . .	123

# List of Figures

1.1	A typical HSCT configuration (courtesy of Mr. Duane Knill). . . . .	8
2.1	Multidisciplinary optimization method for HSCT design. . . . .	23
2.2	Wing section and planform variables used in the 29 variable HSCT optimization problem. . . . .	24
2.3	Idealized HSCT mission profile. . . . .	25
2.4	Wing section and planform variables used in the five variable HSCT optimization problem. . . . .	26
2.5	Wing section and planform variables used in the ten variable HSCT optimization problem. . . . .	27
3.1	Numerical noise in computed wave drag. . . . .	33
3.2	Numerical noise in computed drag-due-to-lift (two variables). . . . .	34
3.3	Numerical noise in computed drag-due-to-lift (one variable). . . . .	35
4.1	A $3^3$ full factorial experimental design (27 points). . . . .	51
4.2	A three variable central composite experimental design (15 points). . . . .	52
4.3	A three variable $D$ -optimal experimental design (11 points). . . . .	53
5.1	A flowchart showing the stages of the VCRSM method used in HSCT optimization. . . . .	62
6.1	Five variable initial and optimal HSCT configurations obtained using RS models (trial 1). . . . .	82

6.2	Five variable initial and optimal HSCT configurations obtained without using RS models (trial 1). . . . .	83
6.3	Five variable initial and optimal HSCT configurations obtained using RS models (trial 2). . . . .	84
6.4	Five variable initial and optimal designs obtained without using RS models (trial 2). . . . .	85
6.5	Five variable initial and optimal HSCT configurations obtained using RS models (trial 3). . . . .	86
6.6	Five variable initial and optimal HSCT configurations obtained without using RS models (trial 3). . . . .	87
6.7	Ten variable initial and optimal HSCT configurations obtained using RS models (trial 1). . . . .	88
6.8	Ten variable initial and optimal HSCT wings (with RS models) showing airfoil differences (trial 1). . . . .	89
6.9	Ten variable initial and optimal HSCT configurations obtained without using RS models (trial 1). . . . .	90
7.1	A one dimensional view of the Case 1 test function ( $\epsilon = 1.0$ ). . . . .	106
7.2	A two dimensional view of the smooth and noisy variants of the Case 1 test function ( $\epsilon = 1.0$ ). . . . .	107
7.3	A one dimensional view of the Case 2 test function ( $\epsilon = 0.7$ ). . . . .	108
7.4	A two dimensional view of the smooth and noisy variants of the Case 2 test function ( $\epsilon = 0.7$ ). . . . .	109
7.5	The DACE and quadratic polynomial RS models for Case 1 ( $\epsilon = 1.0$ ) of the one variable test function. . . . .	110
7.6	The DACE and quadratic polynomial RS models for Case 2 ( $\epsilon = 0.7$ ) of the one variable test function. . . . .	111
7.7	The $k = 1$ and $k = 2$ stage DACE models for Case 1 ( $\epsilon = 1.0$ ) of the one variable test function. . . . .	112
7.8	The $k = 1$ and $k = 2$ stage DACE models for Case 2 ( $\epsilon = 0.7$ ) of the one variable test function. . . . .	113

8.1	Trial 1 initial and optimal HSCT configurations of the five variable HSCT optimization problem. . . . .	124
8.2	Trial 2 initial and optimal HSCT configurations of the five variable HSCT optimization problem. . . . .	125
8.3	Trial 3 initial and optimal HSCT configurations of the five variable HSCT optimization problem. . . . .	126
8.4	A view of the medium fidelity (stage $k = 1$ ) and high fidelity (stage $k = 2$ ) models for $C_{Dwave}$ used in the five variable HSCT optimization problem. . . . .	127
8.5	Initial and optimal HSCT configurations for the HSCT optimization problem using the high fidelity DACE model. Here the design space was reduced to $\pm 3.0$ percent around the optimal HSCT configurations obtained from trials 1-3. . . . .	128
8.6	Initial and optimal HSCT configurations for the HSCT optimization problem using the high fidelity DACE model. Here the original reduced design space was used. . . . .	129
9.1	Parallel computing speedup for the low fidelity and medium fidelity aerodynamic analyses. . . . .	133
9.2	Parallel computing efficiency for the low fidelity and medium fidelity aerodynamic analyses. . . . .	134

# Nomenclature

ANOVA	analysis of variance
$b_{half}$	wing semi-span
$b_{nacelle}$	spanwise location of the inboard nacelle centerline
$\mathbf{c}$	vector of unknown coefficients in least squares surface fitting
$\hat{\mathbf{c}}$	vector of estimated coefficients in least squares surface fitting
$C_D$	drag coefficient
$C_{Dlift}$	supersonic drag-due-to-lift
$C_{Dwave}$	supersonic volumetric wave drag
$C_l$	wing section lift coefficient
$C_L$	lift coefficient
$C_{L\alpha}$	lift curve slope
$C_{root}$	root chord
$C_{tip}$	tip chord
$C_T$	leading edge thrust coefficient
CFD	computational fluid dynamics
CPU	central processing unit
DACE	design and analysis of computer experiments
DOE	design of experiments
DOF	degrees of freedom
$f(\cdot)$	unknown function
$\hat{f}(\cdot)$	predicted function
$\mathbf{f}$	vector of constants used in DACE modeling
$\mathbf{g}$	vector of the optimization constraint values



GA	genetic algorithm
HSCT	high-speed civil transport
I/O	input/output
$k_t$	attainable leading edge thrust factor
$(L/D)_{max}$	maximum value of the lift to drag ratio
LE	leading edge
M	Mach number
MDO	multidisciplinary design optimization
MSE	mean squared error
$n_c$	number of candidate sample sites
$n_e$	number of sample sites used to calculate modeling error
$n_g$	number of optimization constraints
$n_i$	number of HSCT configurations initially selected for analysis
$n_s$	number of sample sites in design space
$n_{st}$	number of stages used in multi-fidelity DACE modeling
$n_t$	number of terms in a polynomial model
$n_v$	number of design variables
$p$	number of processors used on a parallel computer
PBIB	partially balanced incomplete block experimental design
$\mathbf{r}$	vector of correlation values
$R(\cdot)$	correlation function
$\mathbf{R}$	correlation matrix
$R_{LE}$	leading edge radius parameter
$R^2$	correlation coefficient in least squares surface fitting
$R^2_{adj}$	adjusted correlation coefficient
RMS	root mean square error estimate
$RMS_{ub}$	unbiased root mean square error estimate
RS	response surface
RSM	response surface methodology
SQP	sequential quadratic programming
SSE	error sum of squares in analysis of variance calculations

$SY^2$	total sum of squares in analysis of variance calculations
$t_{max}$	chordwise location of maximum thickness
$t/c$	thickness-to-chord ratio
$T_p$	parallel execution time
$T_s$	serial execution time
TE	trailing edge
TOGW	takeoff gross weight
VCRSM	variable complexity response surface modeling
$W_{bend}$	wing bending material weight
$W_{fuel}$	fuel weight
$W_{non-str}$	non-structural weight
$W_{payload}$	payload weight
$W_{str}$	structural weight
$x$	scalar component of $\mathbf{x}$
$\mathbf{x}$	vector denoting all locations in $n_v$ -dimensional space
$\mathbf{x}_{lower}$	vector of lower bounds on the vector $\mathbf{x}$
$\mathbf{x}_{upper}$	vector of upper bounds on the vector $\mathbf{x}$
$\mathbf{x}^{(p)}$	vector denoting the $p^{th}$ location in $n_v$ -dimensional space
$\bar{\mathbf{x}}^{(p)}$	vector of the polynomial model terms at the $p^{th}$ sample site
$\hat{\mathbf{x}}$	vector of all unscaled locations in $n_v$ -dimensional space
$X$	matrix of sample site locations in least squares surface fitting
$y$	scalar observed response value
$\hat{y}$	scalar predicted response value
$\bar{y}$	mean of observed response values
$\hat{y}(\cdot)$	prediction function
$\mathbf{y}$	vector of observed response values at sample sites
$Z(\cdot)$	Gaussian random function
$\beta$	parameter in DACE modeling
$\hat{\beta}$	estimated parameter in DACE modeling
$\bar{\delta}$	mean modeling error
$\delta_{max}$	maximum modeling error

$\delta_{median}$	median modeling error
$\epsilon$	parameter used in defining the DACE test function
$\Lambda_{LE_I}$	inboard leading edge sweep angle
$\Lambda_{LE_O}$	outboard leading edge sweep angle
$\Lambda_{TE_I}$	inboard trailing edge sweep angle
$\pi_\beta$	prior distribution on $\beta$
$\sigma_\delta$	standard deviation of modeling error
$\sigma^2$	sample variance
$\hat{\sigma}^2$	estimated sample variance
$\theta$	scalar correlation parameter used in DACE modeling

# Chapter 1

## Introduction

### 1.1 Background

“The concept of optimization is basic to much of what we do in our daily lives. The desire to run a faster race, win a debate, or increase corporate profits implies a desire to do or be the best in some sense. In engineering, we wish to produce the ‘best quality of life possible with the resources available.’ Thus in ‘designing’ new products, we must use design tools which provide the desired results in a timely and economical fashion.”

This excerpt from the textbook by engineering optimization author Dr. Garret Vanderplaats [1, page 1] encapsulates the essence of what it means to *optimize*. Traditionally, computational resources limited the scope of the problems to which an engineer could apply numerical optimization. As computing power has grown, so has the complexity of the engineering optimization problems. The progress of aerodynamic optimization provides an interesting illustration of this trend. Numerical optimization methods were initially applied to the design of optimally shaped airfoil sections. As computational capabilities increased, aerodynamicists tackled the design of optimally shaped wings. Recent advances in computer hardware and in parallel computing methods have placed engineers at the threshold of optimizing the shape of an entire aircraft configuration. Furthermore, it is beginning to appear possible to optimize the aircraft not only from an aerodynamic perspective, but also by

incorporating the influences of structural mechanics, dynamics and controls, and propulsion, so that the aircraft is optimized as a *system*. This concept of system engineering and optimization has resulted in the inclusion of a host of engineering and non-engineering disciplines (e.g., manufacturing, accounting) under the mantle of multidisciplinary design optimization (MDO). While the benefits of MDO are apparent, methods for implementing MDO are less well defined and the computational challenges facing MDO researchers are daunting. This research project has resulted in the development of a new methodology for aircraft MDO which attempts to address some of the computational challenges currently at the forefront of MDO research.

## 1.2 Motivation

The aircraft industry has given considerable attention to MDO as manufacturers attempt to reduce the time-to-market of new products. Aircraft MDO practitioners would prefer to use high fidelity analysis methods (e.g., Navier-Stokes fluid flow analyses and detailed finite element structural analyses) as early as possible in the design process where the increased accuracy of the high fidelity methods can most strongly influence aircraft design. However, the existence of finite computational resources and time constraints limit the extent to which the high fidelity analyses may be applied in the early stages of the aircraft design process.

One drawback to using high fidelity analyses is *numerical noise* which occurs as a result of the incomplete convergence of iterative processes, the use of adaptive numerical algorithms, round-off errors, and the discrete representation of continuous physical objects (fluids or solids) [2, 3]. Such numerical noise is typically manifested as a high frequency, low amplitude variation in the results obtained from computer analyses as the design parameters vary. When gradient-based numerical optimization is attempted, this oscillatory behavior creates numerous, artificial local optima and causes slow convergence or even convergence failures [2, 4]. Often the result is that the optimizer identifies many locally optimal designs, each of which is obtained by starting the optimization process from a different initial design. Here, the optimization process is deterministic (i.e., the same optimal design is found for a specific set of initial

variables), but the numerical noise traps the optimizer far from the globally optimal design. The problems associated with numerical noise in optimization are ubiquitous in engineering, with examples appearing in the structural optimization software [5] used in aerodynamic-structural design research related to this study, and in an Euler flow solver used for a nozzle design study [6, 7].

Both gradient-based optimization methods which use finite-difference gradient estimation methods [8, pages 54–56, 339–345] and gradient-free pattern search methods [9] require *hundreds* or *thousands* of function evaluations to converge to an optimal design for complex MDO problems. Thus, when the cost of performing an aerodynamic and structural analysis of an aircraft is computationally expensive, numerous local optima and convergence difficulties are unacceptable in aircraft MDO. While there are several computer programs which perform aircraft MDO using computationally inexpensive methods (e.g., ACSYNT [10], FLOPS [11]), high fidelity aircraft *system* MDO remains computationally intractable.

Clearly, new strategies are needed to alleviate the optimization problems posed by numerical noise, and this has spurred much current research in both the aircraft design and numerical optimization communities [12, 13]. These research efforts may be broadly divided into two categories: (1) novel modeling methods, and (2) novel optimization methods.

Several of the novel modeling methods employ statistical techniques based on *design of experiments theory* and *response surface modeling methodologies* [14, pages 1–67]. Here, the aircraft designer performs a limited number of computational analyses (referred to as numerical experiments) using experimental design theory to prescribe values for the independent variables. With the resulting data, the designer creates mathematical models using some type of function (e.g., polynomial functions, rational functions, interpolating functions, neural networks). The mathematical model is often called a *response surface model*. The designer then uses the response surface (RS) model in subsequent calculations during the optimization process. Response surface models filter out the numerical noise which inhibits gradient-based optimization. Although the computational expense of creating a response surface model may be significant, this cost is incurred prior to the use of the RS model in numerical

optimization. Thus, a RS model may be evaluated hundreds or thousands of times during an optimization process without significant computational expense. There has been considerable interest in the application of RS modeling methods to MDO problems and examples of this research are given in [15, 16, 17, 18, 19, 20, 21, 22]. The category of novel modeling methods also includes the interpolation methods based on Bayesian statistics of Sacks et al. [23].

The novel optimization methods include a variety of both gradient-based and gradient-free techniques. The gradient-based techniques include the collaborative subspace optimization method of Sobieski [24] and the collaborative optimization method developed by Kroo et al. [25]. In both of these methods an MDO problem is decomposed along disciplinary boundaries and a series of discipline-specific sub-optimizations are coordinated by a system level optimizer. Another novel gradient-based method is the implicit filtering technique developed by Gillmore and Kelley [26] which employs a scheme where the step size used in finite-difference gradient estimates is systematically varied until convergence using several step sizes is attained. Also included in the category of novel gradient-based optimization techniques are the trust-region methods of Dennis et al. [27]. The novel gradient-free methods include pattern search schemes such as genetic algorithms [28] and simulated annealing [29, 30], modern simplicial search algorithms [31], and stochastic approximation methods [32].

### 1.3 Methodology

In efforts to develop a methodology that will be successful for aircraft MDO, Giunta et al. [33] proposed a method which is termed *variable complexity response surface modeling* (VCRSM) to overcome the difficulties imposed by numerical noise in aircraft MDO. The multidisciplinary design optimization of the High-Speed Civil Transport (HSCT) aircraft (Figure 1.1) was used as a testbed for the VCRSM method. The design of an HSCT is an ideal MDO test case since the technical and economic feasibility of such an aircraft relies on synergistic interactions between the various aircraft design disciplines.

The *variable complexity* portion of VCRSM denotes the use of low, medium, and high fidelity aircraft analysis methods. Here the low fidelity methods include algebraic expressions for estimating lift and drag for HSCT-type aircraft. These typically require only a few CPU seconds to evaluate on a workstation. The medium fidelity methods include linear theory aerodynamic analyses. These analyses require tens of CPU seconds up to several CPU minutes on a workstation. The high fidelity methods include various subsets of the Navier-Stokes equations including the Reynolds-averaged Navier-Stokes equations, the parabolized Navier-Stokes equations, and the Euler equations. These analyses may require anywhere from tens of CPU minutes on a workstation to tens of CPU hours on a supercomputer. Thus, the computational expense between a single low fidelity analysis and a single high fidelity analysis methods spans several orders of magnitude.

In optimization problems the *design space* refers to the box-like region of interest formed by the lower and upper bounds on each design variable. In VCRSM the low fidelity methods are used to perform aerodynamic analyses for *thousands* or *tens of thousands* of HSCT configurations in the design space. Such a large number of analyses is used to identify a subset of the design space where the HSCT configurations meet minimum values for several performance related criteria. This portion of the design space is termed the *reduced design space*. Note that the reduced design space has the same dimension as the original design space, but has more restrictive bounds. Next, *tens to hundreds* of HSCT configurations are selected from the reduced design space and are evaluated using the medium and/or high fidelity analysis methods. In this manner, computational resources are not squandered in evaluating HSCT configurations outside of the reduced design space which do not meet the minimum performance criteria.

The *response surface modeling* portion of VCRSM denotes the use of statistical methods known as design of experiments theory and response surface modeling [14, pages 16–81, 279–401]. Here, design of experiments (DOE) theory is used to select the HSCT configurations which are analyzed using the low fidelity analysis methods in the original design space and the medium/high fidelity analysis methods in the reduced design space. Once the medium/high fidelity analyses are completed, response surface



models (multidimensional curve fits) are created from the aerodynamic analysis data. The response surface models are then employed as *approximation models* or *surrogate models* which replace the medium/high fidelity analysis methods when performing numerical optimization. The response surface models have two advantages over the medium/high fidelity analysis methods which are (1) the response surface models are computationally inexpensive to evaluate since they are algebraic equations, and (2) the response surface models remove the numerical noise present in the medium and high fidelity analysis results. In addition, response surface models provide a mechanism to incorporate high fidelity analysis data into the optimization process when the traditional approach of linking the high fidelity analysis method directly to the optimizer is not computationally affordable. This research focused on the use of quadratic polynomial response surface models [14, page 55], although interpolating models based on Bayesian statistics [23] were briefly examined as well. The statistical methods employed in response surface modeling are detailed in Chapter 4.

In applying the VCRSM process the size of the design space has been reduced substantially. For example, in a two variable problem with the range of each variable reduced by one half, the reduced design space contains one quarter of the area of the original design space. In a ten variable problem, eliminating half of the range of each variable results in a reduced design space with less than 0.10 percent of the volume of the original design space. The work of Kaufman et al. [5] and Roux et al. [21] showed that the response surface models were considerably more accurate when created for the reduced design space than for the original design space.

Once the reduced design space is identified and the response surface models are constructed, a traditional gradient-based numerical optimization is conducted for the aircraft MDO problem using the response surface models instead of the original computational methods. With the optimizer restricted to the reduced design space and with the noise-free, computationally inexpensive response surface models, aircraft system MDO is possible.

For this study, example HSCT optimization problems involving five and ten variables were created to demonstrate the utility of the VCRSM method. In these optimization problems, the variables controlled the shape of the airfoil sections and

wing planform of the HSCT while the remaining portions of the HSCT (fuselage, tail, engine nacelle location) remained fixed. Polynomial response surface models were used to approximate various aerodynamic quantities as functions of the geometric variables which defined the HSCT wing. These details of these optimization problems are given in Chapter 2 and the optimization results are presented in Chapter 6.

To investigate the use of interpolating models based on Bayesian statistics test problems involving one, five, and ten variable were examined. These results are given in Chapter 7. In addition, a preliminary study of the use of Bayesian interpolating models in the VCRSM method was conducted for the five variable HSCT optimization problem. See Chapter 8 for details on this study.

High fidelity analysis tools are not included in this study due to their computational expense. However, the VCRSM method described here is easily extended to include data from high fidelity analyses. In this study the low fidelity analysis methods are used to identify the reduced design space and medium fidelity analyses are used to construct the response surface models. An Euler/Navier-Stokes solver has been evaluated for use in HSCT design and these efforts are described by Knill, et al. in References [34] and [35].

Parallel computing is crucial in reducing the computational expense of using the VCRSM method for aircraft MDO. An overview of the parallel computing strategies employed in this study is given in Chapter 9. See References [36] and [37] for more details on how parallel computing strategies are employed in the VCRSM method.

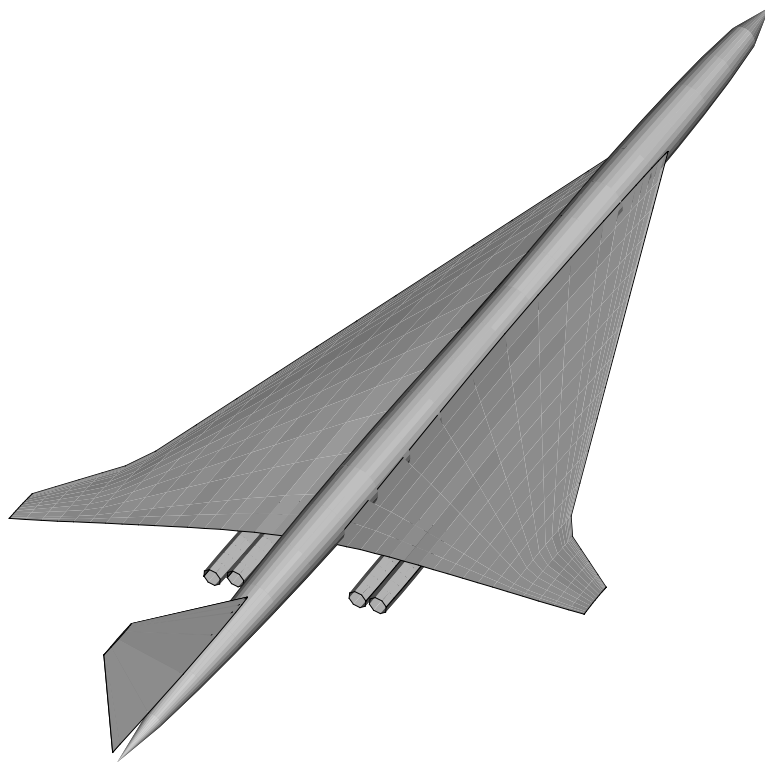


Figure 1.1: A typical HSCT configuration (courtesy of Mr. Duane Knill).

## Chapter 2

# HSCT Optimization Problem Formulation

The design problem is to minimize the takeoff gross weight (TOGW) for a 250 passenger High-Speed Civil Transport with a range of 5500 nautical miles and a cruise speed of Mach 2.4. The idealized mission profile is divided into three segments: takeoff, supersonic cruise/climb at Mach 2.4, and landing. The three mission design variables are fuel weight, starting altitude for the supersonic cruise/climb segment, and rate-of-climb during the supersonic cruise/climb segment. If the HSCT reaches the maximum ceiling of 70000 *ft*, supersonic cruise at Mach 2.4 is maintained at that altitude for the duration of the supersonic mission leg.

TOGW was selected as the objective function for this optimization problem since it represents a composite measure of merit for the aircraft as a *system*. When TOGW is expressed as a sum of the dry weight (i.e., the weight of the aircraft including payload, but without fuel) and the fuel weight, the dry weight can be correlated to the initial acquisition cost of the aircraft and fuel weight represents the yearly recurring costs of aircraft operations. Thus, in some sense a minimum TOGW aircraft roughly corresponds to a minimum cost aircraft, or at least an aircraft for which the costs should be reasonable.

## 2.1 Analysis and Optimization Software

For these research efforts a suite of low fidelity and medium fidelity analysis methods has been developed which includes several software packages obtained from NASA and the USAF, supplemented with some commercial software and in-house software as well [4, 38, 39]. The low fidelity supersonic aerodynamic analysis methods include a modified version of Eminton’s method [40] to calculate supersonic volumetric wave drag, the analytical methods of Cohen and Friedman[41] for estimating supersonic drag-due-to-lift, and the methods of Hopkins and Inouye [42], and Mason [43] for calculating supersonic skin friction drag. The medium fidelity supersonic aerodynamic analysis methods include the Harris program [44] for calculating supersonic volumetric wave drag and Carlson’s supersonic panel program [45] (also known as a “Mach-box” method) for calculating supersonic drag-due-to-lift.

Subsonic aerodynamic performance characteristics, including stability and control derivatives, are estimated using DATCOM-based techniques [46], which are low fidelity analysis methods, and a vortex-lattice program named JKVLM developed by Kay et al. [47], which is a medium fidelity analysis method.

Takeoff gross weight is estimated using historically-based algebraic weight equations obtained from the Flight Optimization System (FLOPS) program [11] developed by McCullers. In FLOPS the takeoff gross weight is defined as

$$TOGW = W_{payload} + W_{fuel} + W_{str} + W_{non-str}, \quad (2.1)$$

where  $W_{payload}$  is the payload weight,  $W_{str}$  is the structural weight, and  $W_{non-str}$  is the non-structural weight.

One component of the structural weight is the wing bending material weight ( $W_{bend}$ ), which is the portion of wing structural weight needed to withstand distributed bending loads. A study by Huang [48] found that the FLOPS estimates of  $W_{bend}$  were inaccurate for HSCT-type planforms for which there is limited historical data. To improve the accuracy of  $W_{bend}$  estimates, the HSCT wing and fuselage structure were modeled using the finite element structural optimization program GENESIS [49] in which the structural members were optimally sized to carry the distributed bending loads created during several loading conditions [50]. Due to the computational

expense of performing an HSCT structural optimization with GENESIS (i.e., where GENESIS performs a sub-optimization within the overall HSCT optimization), various methods for employing the GENESIS estimates for  $W_{bend}$  were investigated. These included a periodic updating procedure that was termed *interlacing* [4], and a response surface modeling approach [51]. While the studies involving GENESIS are related to the HSCT optimization research described here, GENESIS data for  $W_{bend}$  were not included in the optimization cases to be described below.

The numerical optimization software is the commercially available program DOT (Design Optimization Tools) [52]. DOT offers several algorithms for performing constrained optimization and its sequential quadratic programming (SQP) option is employed in this study. See Table 2.1 for a listing of both the analysis and optimization tools used in this HSCT research. Figure 2.1 shows how the analysis and optimization tools are coupled to perform HSCT configuration optimization.

High fidelity analysis tools such as Euler/Navier-Stokes solvers and detailed finite element structural models currently are not used in this HSCT optimization study. However, recent research has been conducted in which the Euler/Navier-Stokes solver GASP [53] has been evaluated for use in HSCT optimization [34, 35]. In addition, work is underway to use more detailed finite element structural models in GENESIS, but results from this work have not yet been published.

## 2.2 HSCT Configuration Definition

The HSCT configuration and mission are defined using 29 variables (Table 2.2). Twenty-six of these variables describe the geometric layout of the HSCT and three variables describe the mission profile. The airfoil and planform variables are shown in Figure 2.2. In this parametrization, eight variables describe the wing planform, eight variables define the area ruled fuselage shape distribution, five variables define the wing leading edge and airfoil section properties, two variables define the engine nacelle locations, two variables define the horizontal and vertical tail areas, and one variable defines engine thrust. For this HSCT design problem the fuselage has a fixed length of 300 *ft* and an internal volume of 23720 *ft*<sup>3</sup>.

The idealized mission profile (Figure 2.3) is divided into three segments: takeoff, supersonic cruise/climb at Mach 2.4, and landing. The three mission design variables are fuel weight, starting altitude for the supersonic cruise/climb segment, and rate-of-climb during the supersonic cruise/climb segment. If the HSCT reaches the maximum ceiling of 70000 *ft*, supersonic cruise at Mach 2.4 is maintained at that altitude for the duration of the supersonic mission leg until 85 percent of the fuel weight is consumed. The remaining 15 percent of the fuel weight is held as reserve.

## 2.3 Constraint Formulation

The HSCT design employs 69 nonlinear inequality constraints which consist of both geometric constraints (e.g., all wing chords  $\geq 7.0$  *ft*), and aerodynamic/performance constraints (e.g.,  $C_L$  at landing  $\leq 1$ , and range  $\geq 5500$  *naut.mi.*). These constraints are listed in Table 2.3.

The geometric constraints are applied to the HSCT configuration to prevent the optimizer from creating grossly unrealistic designs, such as wings with zero thickness or engine nacelles located inside the fuselage. These constraints are computationally inexpensive to evaluate since no aerodynamic or weight calculations are involved.

In contrast, the aerodynamic and performance constraints are considerably more computationally expensive than the geometric constraints. The range constraint is the most expensive of the aerodynamic constraints since the various supersonic aerodynamic analysis methods are used repeatedly in the range calculation. (See Hutchison [54] and MacMillin [39] for more detail on the range calculation methods.) The least expensive of the aerodynamic constraints concern the section  $C_l$  and total  $C_L$  at landing. An elliptical spanwise lift distribution is assumed for the section  $C_l$  constraints, and lift equal to weight is assumed for the total  $C_L$  constraint. For both the section  $C_l$  and  $C_L$  constraints a worst-case emergency landing scenario is simulated with the following conditions

1. a landing speed of 145 *kts*,
2. a landing weight which includes 50 percent of the total fuel weight,

3. an airport elevation of 5000 *ft* above sea level, and
4. high temperatures ( $90^\circ F$ ) at the landing site.

Of moderate computational expense are the remaining performance constraints which encompass landing angle of attack limits, takeoff performance, and stability issues. All of these constraints rely on the calculation of subsonic aerodynamic performance, in particular the subsonic lift curve slope and subsonic stability derivatives. See the work of MacMillin [39] for additional details on the calculation of these constraints.

## 2.4 Optimization Problem Formulation

The HSCT design objective is to minimize TOGW, where TOGW is a nonlinear function of the 29 design variables. In formal optimization terms this problem may be expressed as

$$\begin{aligned} \min_{\mathbf{x} \in R^{29}} TOGW(\mathbf{x}), \quad \text{subject to } \mathbf{g}_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, 69, \\ \mathbf{x}_{lower} \leq \mathbf{x} \leq \mathbf{x}_{upper} \end{aligned} \quad (2.2)$$

where  $\mathbf{x}$  is the vector of  $n_v = 29$  design variables,

$$\mathbf{x} = [x_1, x_2, \dots, x_{n_v}], \quad (2.3)$$

$\mathbf{x}_{lower}$  is the vector of lower bounds on the design variables,  $\mathbf{x}_{upper}$  is the vector of upper bounds on the design variables, and  $\mathbf{g}(\mathbf{x})$  is the vector of  $n_g = 69$  nonlinear inequality constraints,

$$\mathbf{g} = [g_1, g_2, \dots, g_{n_g}]. \quad (2.4)$$

## 2.5 Five Variable HSCT Optimization Problem

A simple five variable HSCT wing design problem is used to develop and validate the variable complexity response surface modeling method. This five variable problem is based on the 29 variable HSCT MDO problem described above. The five variables



are wing root chord ( $C_{root}$ ), wing tip chord ( $C_{tip}$ ), thickness-to-chord ratio ( $t/c$  ratio), inboard leading edge sweep angle ( $\Lambda_{LEI}$ ), and fuel weight ( $W_{fuel}$ ). Their initial values along with the minimum and maximum values are listed in Table 2.4.

The wing planform and airfoil section definitions are shown in Figure 2.4. To uniquely define the wing planform, the length of the leading edge from the location where the wing intersects the fuselage to the leading edge break is held constant at 150.0 *ft*, and the outboard leading edge sweep angle is held constant at 44.2°. Also note that the trailing edge is straight, i.e., there is no trailing edge break in this wing definition. For the airfoil section definitions, the chordwise location of maximum thickness is constant at 40.0 percent and the nondimensional leading edge radius parameter is constant at a value of 2.63. See [54] and [55, pages 113–118] for information on the selection of the leading edge radius parameter. Additional attributes of the HSCT are the vertical tail area which is fixed at 700 *ft*<sup>2</sup>, and the thrust which is constant at 39000 *lb* per engine. The engine nacelles are held at fixed spanwise positions 20.9 *ft* and 26.9 *ft* on each side of the fuselage centerline (four engines total). A horizontal tail is not considered for this aircraft.

The mission profile also is simplified from the 29 variable HSCT design problem to include only a supersonic cruise leg and landing. The altitude for the Mach 2.4 supersonic cruise mission is constant at 65000 *ft*. Landing constraints on both the overall lift coefficient and 18 wing section lift coefficients are examined for emergency landing situations. For the five variable design problem there are 42 constraints (Table 2.5) and the optimization problem may be expressed as

$$\begin{aligned} \min_{\mathbf{x} \in R^5} TOGW(\mathbf{x}), \quad \text{subject to } \mathbf{g}_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, 42, \\ \mathbf{x}_{lower} \leq \mathbf{x} \leq \mathbf{x}_{upper} \end{aligned} \quad (2.5)$$

where  $\mathbf{x}$  is the vector of  $n_v = 5$  design variables, and  $\mathbf{g}(\mathbf{x})$  is the vector of  $n_g = 42$  nonlinear inequality constraints.

The software modifications necessary to convert the original 29 variable HSCT optimization problem to the five variable optimization problem are described in Appendix A.1. In addition, a user’s guide for the HSCT software is also provided in Reference [39].

## 2.6 Ten Variable HSCT Optimization Problem

A ten variable HSCT wing design problem was examined to further develop the VCRSM method. The ten variables for this design problem are wing root chord ( $C_{root}$ ), wing tip chord ( $C_{tip}$ ), wing semi-span ( $b_{half}$ ), inboard leading edge sweep angle ( $\Lambda_{LEI}$ ), outboard leading edge sweep angle ( $\Lambda_{LEO}$ ), chordwise location of maximum thickness ( $t_{max}$ ), leading edge radius parameter ( $R_{LE}$ ), thickness-to-chord ratio ( $t/c$ ), spanwise location of the inboard nacelle ( $b_{nacelle}$ ), and fuel weight ( $W_{fuel}$ ). Their initial values along with the minimum and maximum values are listed in Table 2.6. Figure 2.5 shows the airfoil and wing planform variables for this problem.

As was done in the five variable HSCT optimization problem, the length of the leading edge from the wing/fuselage intersection to the leading edge break is held constant at 150 *ft*, the vertical tail area is fixed at 700 *ft*<sup>2</sup>, the thrust is held constant at 39000 *lb* per engine, the trailing edge of the wing is straight, and no horizontal tail is considered. Additionally, the outboard nacelle is fixed at a spanwise distance of 6 *ft* from the centerline of the inboard nacelle and the Mach 2.4 cruise altitude is constant at 70000 *ft*. The 49 constraints for this optimization problem are listed in Table 2.7. Thus, the ten variable HSCT wing design optimization problem may be expressed as

$$\begin{aligned} \min_{\mathbf{x} \in R^{10}} TOGW(\mathbf{x}), \quad \text{subject to } \mathbf{g}_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, 49, \\ \mathbf{x}_{lower} \leq \mathbf{x} \leq \mathbf{x}_{upper} \end{aligned} \quad (2.6)$$

where  $\mathbf{x}$  is the vector of  $n_v = 10$  design variables, and  $\mathbf{g}(\mathbf{x})$  is the vector of  $n_g = 49$  nonlinear inequality constraints.

The software modifications associated with the ten variable HSCT optimization problem are described in Appendix A.2. Note that many of the software modifications used in the five variable optimization problem also are used in the ten variable optimization problem.

Table 2.1: Analysis and optimization tools for HSCT design.

<b>Method</b>	<b>Description</b>
Subsonic Aerodynamics	In-house Codes
Supersonic Aerodynamics	In-house Codes, WINGDES [45] (NASA), Harris Wave Drag Code [44] (NASA)
Propulsion	In-house Codes
Stability and Control	DATCOM (USAF) [46], JKVLM [47]
Takeoff/Landing Performance	In-house Codes
Mission Performance	In-house Codes
Weights and Structures	FLOPS [11] (NASA), GENESIS [49] (VMA, Inc.)
Optimizer	DOT [52] (VR&D, Inc.)

Table 2.2: Twenty-nine HSCT variables and typical values.

Number	Typical Value	Description
1	181.48	Wing root chord, <i>ft</i>
2	155.9	LE break point, <i>x ft</i>
3	49.2	LE break point, <i>y ft</i>
4	181.6	TE break point, <i>x ft</i>
5	64.2	TE break point, <i>y ft</i>
6	169.5	LE wing tip, <i>x ft</i>
7	7.00	Wing tip chord, <i>ft</i>
8	75.9	Wing semi-span, <i>ft</i>
9	0.40	Chordwise location of maximum thickness
10	3.69	LE radius parameter
11	2.58	Airfoil <i>t/c ratio</i> at root, %
12	2.16	Airfoil <i>t/c ratio</i> at LE break, %
13	1.80	Airfoil <i>t/c ratio</i> at tip, %
14	2.20	Fuselage restraint 1, <i>x ft</i>
15	1.06	Fuselage restraint 1, <i>r ft</i>
16	12.20	Fuselage restraint 2, <i>x ft</i>
17	3.50	Fuselage restraint 2, <i>r ft</i>
18	132.46	Fuselage restraint 3, <i>x ft</i>
19	5.34	Fuselage restraint 3, <i>r ft</i>
20	248.67	Fuselage restraint 4, <i>x ft</i>
21	4.67	Fuselage restraint 4, <i>r ft</i>
22	26.23	Nacelle 1 location, <i>ft</i>
23	32.39	Nacelle 2 location, <i>ft</i>
24	697.9	Vertical tail area, <i>ft</i> <sup>2</sup>
25	713.0	Horizontal tail area, <i>ft</i> <sup>2</sup>
26	39000	Thrust per engine, <i>lb</i>
27	322617	Mission fuel, <i>lb</i>
28	64794	Starting cruise/climb altitude, <i>ft</i>
29	33.90	Supersonic cruise/climb rate, <i>ft/min</i>

Table 2.3: Constraints for the 29 variable HSCT optimization problem.

Number	Geometric Constraints
1	Fuel volume $\leq 50\%$ wing volume
2	Airfoil section spacing at $C_{tip} \geq 3.0ft$
3-20	Wing chord $\geq 7.0ft$
21	LE break $\leq$ semi-span
22	TE break $\leq$ semi-span
23	Root chord $t/c$ ratio $\geq 1.5\%$
24	LE break chord $t/c$ ratio $\geq 1.5\%$
25	Tip chord $t/c$ ratio $\geq 1.5\%$
26-30	Fuselage restraints
31	Nacelle 1 outboard of fuselage
32	Nacelle 1 inboard of nacelle 2
33	Nacelle 2 inboard of semi-span
	<b>Aerodynamic/Performance Constraints</b>
34	Range $\geq 5500$ naut.mi.
35	$C_L$ at landing $\leq 1$
36-53	Section $C_l$ at landing $\leq 2$
54	Landing angle of attack $\leq 12^\circ$
55-58	Engine scrape at landing
59	Wing tip scrape at landing
60	LE break scrape at landing
61	Rudder deflection $\leq 22.5^\circ$
62	Bank angle at landing $\leq 5^\circ$
63	Tail deflection at approach $\leq 22.5^\circ$
64	Takeoff rotation to occur $\leq V_{min}$
65	Engine-out limit with vertical tail
66	Balanced field length $\leq 11000$ ft
67-69	Mission segments: thrust available $\geq$ thrust required

Table 2.4: Five HSCT wing variables in the example problem.

<b>Variable</b>	<b>Initial</b>	<b>Minimum</b>	<b>Maximum</b>
$C_{root}$	185.0 <i>ft</i>	148.0 <i>ft</i>	222.0 <i>ft</i>
$C_{tip}$	10.0 <i>ft</i>	8.0 <i>ft</i>	12.0 <i>ft</i>
$\Lambda_{LEI}$	75.0°	68.3°	81.8°
$t/c$ ratio	2.0%	1.6%	2.4%
$W_{fuel}$	315000 <i>lb</i>	305550 <i>lb</i>	324450 <i>lb</i>

Table 2.5: Constraints for the five variable HSCT wing optimization problem.

Number	Geometric Constraints
1	Fuel volume $\leq$ 50% wing volume
2	Airfoil section spacing at $C_{tip} \geq 3.0ft$
3-20	Wing chord $\geq 7.0ft$
21	LE break $\leq$ semi-span
22	Airfoil $t/c$ ratio $\geq 1.5\%$
	<b>Aerodynamic/Performance Constraints</b>
23	Range $\geq 5500$ naut.mi.
24	$C_L$ at landing $\leq 1.0$
25-42	Section $C_l$ at landing $\leq 2.0$

Table 2.6: Initial values for the ten HSCT wing variables and the allowable range on each variable.

<b>Variable</b>	<b>Initial</b>	<b>Minimum</b>	<b>Maximum</b>
$C_{root}$	174.0 <i>ft</i>	139.2 <i>ft</i>	208.7 <i>ft</i>
$C_{tip}$	8.1 <i>ft</i>	6.5 <i>ft</i>	9.7 <i>ft</i>
$b_{half}$	73.9 <i>ft</i>	66.5 <i>ft</i>	81.3 <i>ft</i>
$\Lambda_{LE_I}$	71.9°	65.4°	78.3°
$\Lambda_{LE_O}$	44.2°	40.2°	48.2°
$t_{max}$	39.6%	31.7%	47.5%
$R_{LE}$	2.6	2.1	3.2
$t/c$	2.3%	1.8%	2.7%
$b_{nacelle}$	20.9 <i>ft</i>	16.7 <i>ft</i>	25.1 <i>ft</i>
$W_{fuel}$	310000 <i>lb</i>	300700 <i>lb</i>	319300 <i>lb</i>



Table 2.7: Constraints for the ten variable HSCT optimization problem.

<b>Number</b>	<b>Geometric Constraints</b>
1	Fuel volume $\leq 50$ wing volume
2	Airfoil section spacing at $C_{tip} \geq 3.0ft$
3-20	Wing chord $\geq 7.0ft$
21	Leading edge break $\leq$ semi-span
22	Airfoil $t/c \geq 1.5$
	<b>Aerodynamic/Performance Constraints</b>
23	Range $\geq 5500$ naut.mi.
24	$C_L$ at landing $\leq 1.0$
25-42	Section $C_l$ at landing $\leq 2.0$
43	Landing angle of attack $\leq 13^\circ$
44-45	Engine clearance $\geq 0.0$ , landing at $0^\circ$ roll
46-47	Engine clearance $\geq 0.0$ , landing at $5^\circ$ roll
48-49	Wing clearance $\geq 0.0$ , landing at $5^\circ$ roll

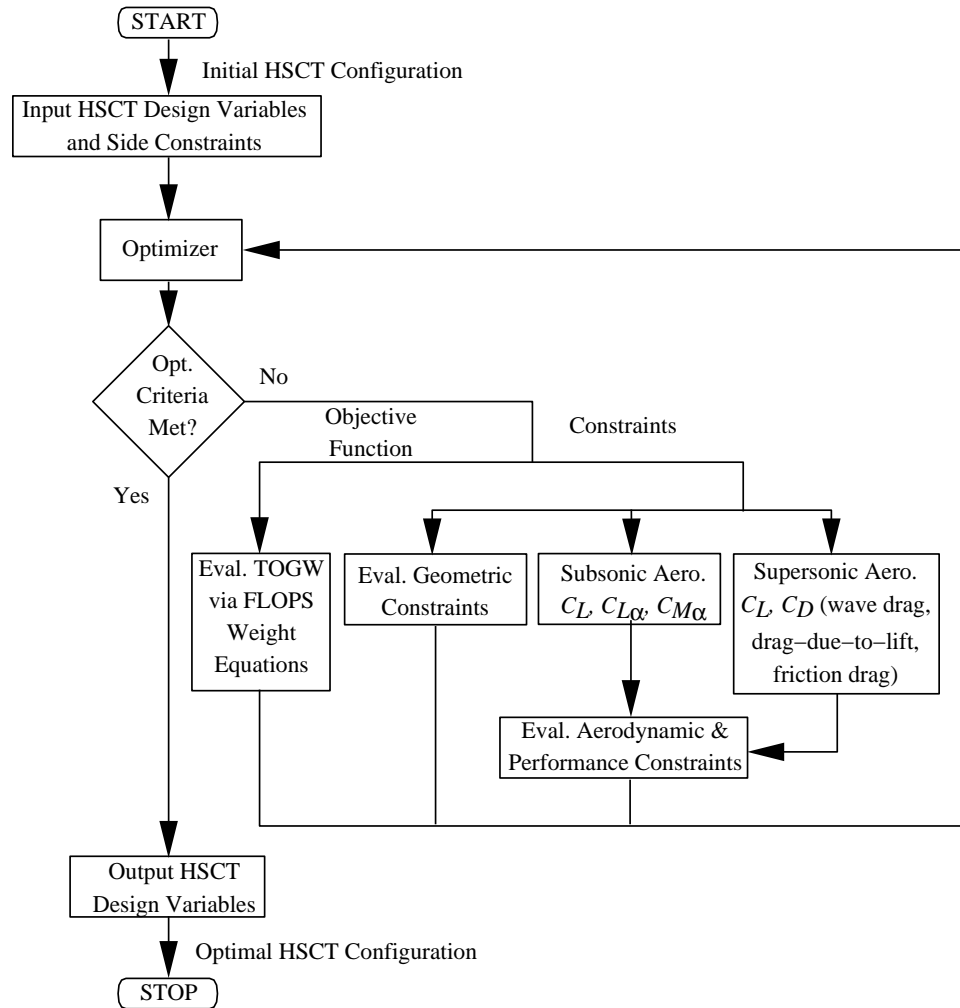


Figure 2.1: Multidisciplinary optimization method for HSCT design.

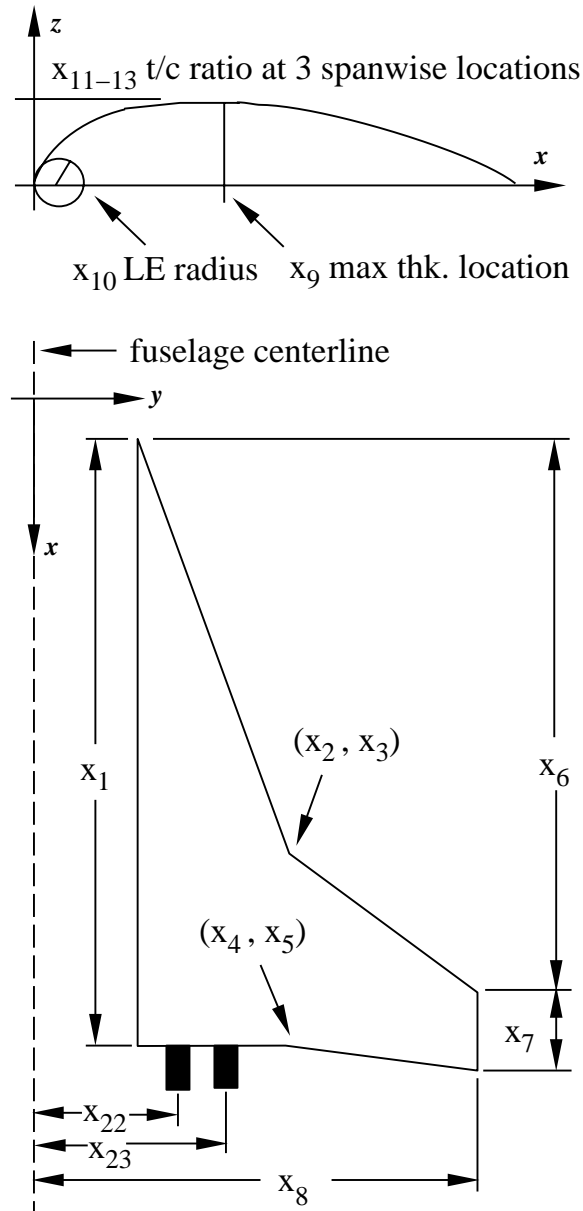


Figure 2.2: Wing section and planform variables used in the 29 variable HSCT optimization problem.

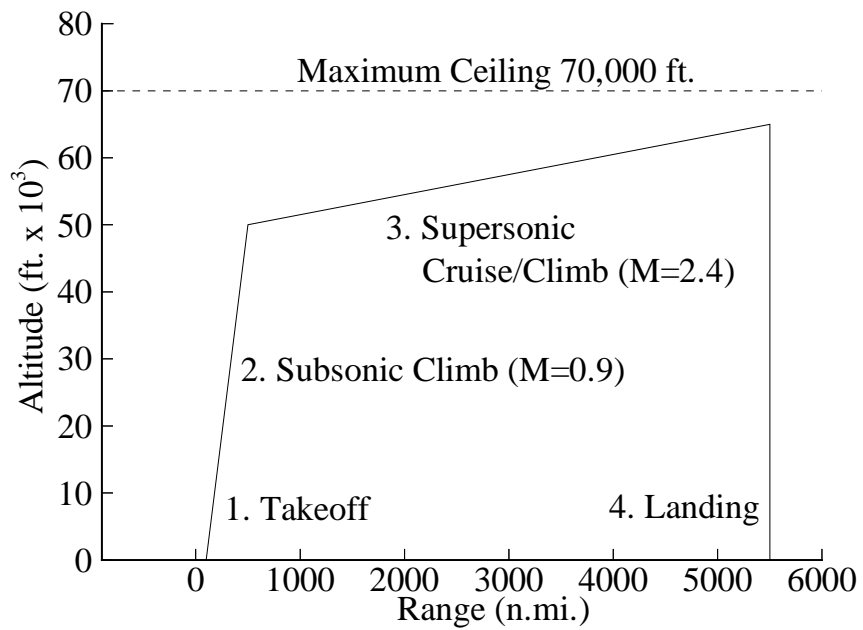


Figure 2.3: Idealized HSCT mission profile.

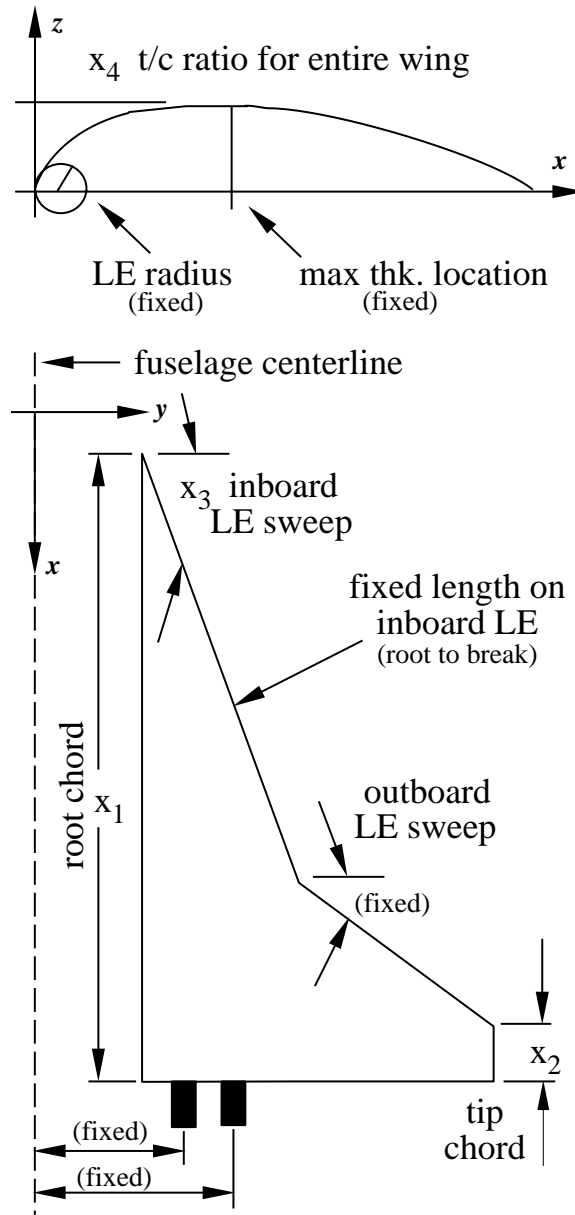


Figure 2.4: Wing section and planform variables used in the five variable HSCT optimization problem.

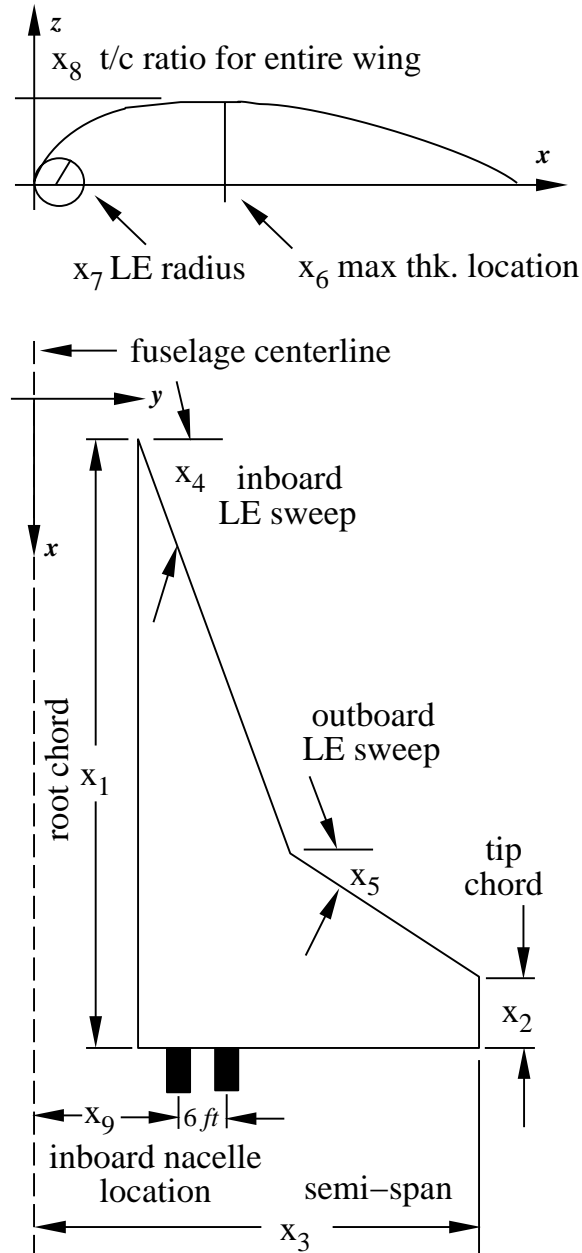


Figure 2.5: Wing section and planform variables used in the ten variable HSCT optimization problem.

## Chapter 3

# Numerical Noise Issues and the VCRSM Method

As noted in Chapter 1, the existence of high frequency low amplitude numerical noise pervades many computational efforts. Numerical noise poses a significant hindrance to gradient-based optimization because it inhibits the use of finite difference methods in gradient estimation. Typical sources of numerical noise include

1. the incomplete convergence of iterative processes,
2. the use of adaptive numerical algorithms,
3. round-off errors, and
4. the discrete representation of continuous physical objects.

When confronted with numerical noise in optimization problems, one may elect to eliminate the noise sources (e.g., problem reformulation or software modification) or one may use optimization methods that are insensitive to numerical noise. While eliminating the noise sources is an attractive option, it is not always possible to reformulate the optimization to accomplish this goal. Eliminating noise sources through software modification is also possible, but this can be an extremely tedious task, particularly so if the user is not the original software developer. Furthermore, in many industrial settings the user does not have access to the source files for

commercially available software. The development of optimization methods which are insensitive to numerical noise is an active area of research in the optimization community. While the efforts of Gilmore and Kelley [26] are promising, their methods are not applicable when the evaluation of the objective function and/or the evaluation of the constraints are computationally expensive.

The VCRSM method is a hybrid of the two approaches used to eliminate numerical noise in optimization problems. See Section 3.2 below and Chapter 5 for more details on the development and use of VCRSM techniques.

### 3.1 Examples of Numerical Noise

An example of item (4) is demonstrated through the calculation of wave drag for a series of similar HSCT configurations. The Harris wave drag code [44] uses a discrete set of about 50 to 60 parallel cutting planes (inclined at the Mach angle in the streamwise direction) at which locations the projected area of the HSCT is calculated. Wave drag is then estimated from the discrete area distribution. Figure 3.1 shows supersonic volumetric wave drag ( $C_{D_{wave}}$ ) calculated for HSCT configurations cruising at Mach 2.4. Here, the fuselage shape of the HSCT is held fixed and only the wing shape is allowed to vary as the wing semi-span ranges from 50 to 100 *ft*. As the semi-span increases, numerical noise is created by a high frequency low amplitude variation in the calculated wave drag values. Physically, wave drag should change smoothly as the wing shape is varied. Note that this numerical noise occurs on an extremely small scale with variations in wave drag on the order of 0.02 to 0.1 drag counts (one drag count corresponds to  $C_D = 10^{-4}$ ). The Harris wave drag program has an accuracy of approximately 0.5 to 1.0 drag counts and was not developed specifically to be used with optimization. Hence, wave drag variations of 0.02 to 0.1 counts were considered inconsequential by the original programmers. Figure 3.1 also shows a quadratic response surface model obtained by performing a least squares curve fit to the noisy data. The quadratic surface smoothes out the various scales of numerical noise present in the data while it captures the global trend of the variation in wave drag. The use of response surface models to smooth out the effects of numerical noise



is addressed below.

Another example of numerical noise is provided by the calculation of supersonic drag-due-to-lift ( $C_{Dlift}$ ). The supersonic panel methods of Carlson et al. [45] utilize a wing discretization scheme that is sensitive to wing planform changes. Thus, slight modifications to the leading and trailing edge sweep angles, along with changes in the location where the Mach angle intersects the leading edge, produce noisy variations in the predicted drag.

Supersonic drag-due-to-lift is expressed by Hutchison [54] as

$$C_{Dlift} = \left( \frac{1}{C_{L\alpha}} - k_t \frac{C_T}{C_L^2} \right) C_L^2, \quad (3.1)$$

where  $C_{L\alpha}$  is the supersonic lift curve slope at  $M = 2.4$ ,  $C_T/C_L^2$  is the leading edge thrust term, and  $k_t$  is the attainable leading edge thrust factor calculated using Carlson's method of "attainable leading edge suction" [56]. Note that Equation 3.1 is dominated by the term  $1/C_{L\alpha}$  which typically has values of approximately 0.6 to 0.7, whereas the term  $k_t(C_T/C_L^2)$  typically ranges from 0.04 to 0.10.

Numerical noise in computed drag-due-to-lift is shown in Figure 3.2 where the inboard leading edge sweep angle ( $\Lambda_{LE_I}$ ) is varied from  $77^\circ$  to  $79^\circ$ , and the inboard trailing edge sweep angle ( $\Lambda_{TE_I}$ ) is varied from  $-55^\circ$  to  $50^\circ$ . As before, the fuselage shape and all other wing shape variables were held constant.

The drag-due-to-lift calculations were performed on a  $21 \times 41$  uniform mesh over the range of the two variables to produce the detailed three dimensional surface map. Figure 3.3 shows a cross sectional cut of this three dimensional surface taken at a trailing-edge sweep angle of approximately three degrees. Note the extremely fine scale of the noise with respect to the variation in leading edge sweep angle. The numerical noise in the drag-due-to-lift values occurs in the estimation of the supersonic lift curve slope and leading edge thrust terms. As was the case for the noise levels in the wave drag analyses, the numerical noise in the drag-due-to-lift calculations is within the intended accuracy of the analysis method.

## 3.2 Description of the VCRSM Method

The VCRSM method removes sources of numerical noise which inhibit numerical optimization. This is accomplished by creating smooth, polynomial response surface models or Bayesian interpolating models which replace the original analysis methods that produce numerical noise. Without numerical noise to inhibit optimization, there is a greater probability that the globally optimal design will be found, regardless of the initial variable values used at the start of the numerical optimization.

One of the most important advantages obtained by using response surface models in optimization is significant reduction in computational expense. This allows the user to perform global optimization and reliability-based optimization, which are otherwise prohibitively computationally expensive. In addition, the use of response surface models allows the design engineer to quickly perform a variety of trade-off studies which provide information on the sensitivity of the optimal aircraft design to changes in performance criteria and to off-design conditions.

The reduction in the computational expense of optimization when using response surface models motivates their use in the modeling of data obtained from expensive analysis methods, even though the expensive analysis methods may not produce numerical noise. One such relatively expensive analysis method is a vortex-lattice method [47] which is used to estimate subsonic aerodynamic performance. In particular, the subsonic lift curve slope  $C_{L\alpha}$  at  $M = 0.2$  is the parameter of interest obtained from the vortex-lattice code, since  $C_{L\alpha}$  at  $M = 0.2$  is needed in the landing angle of attack constraint as well as the runway strike constraints.

### 3.2.1 VCRSM and the Five Variable Optimization Problem

In the five variable HSCT optimization problem the VCRSM method is used to create response surface models for  $C_{Dwave}$ ,  $C_{L\alpha}$  at  $M=2.4$ , and  $C_T/C_L^2$ . Note that only these supersonic aerodynamic performance parameters are needed since there are no constraints on subsonic aerodynamic performance in the five variable problem.

### 3.2.2 VCRSM and the Ten Variable Optimization Problem

For the ten variable HSCT optimization problem the VCRSM method is used to create response surface models for the same three supersonic parameters as well as for  $C_{L\alpha}$  at  $M=0.2$ . Recall that in this optimization problem subsonic aerodynamic performance is needed to estimate the landing angle-of-attack constraint and the runway scrape constraints for the wing structure and engine pods.

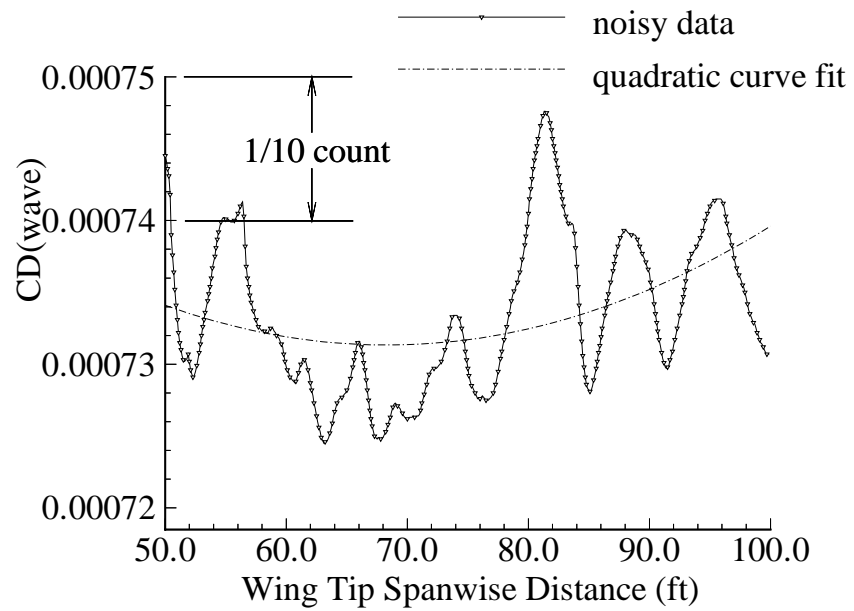


Figure 3.1: Numerical noise in computed wave drag.

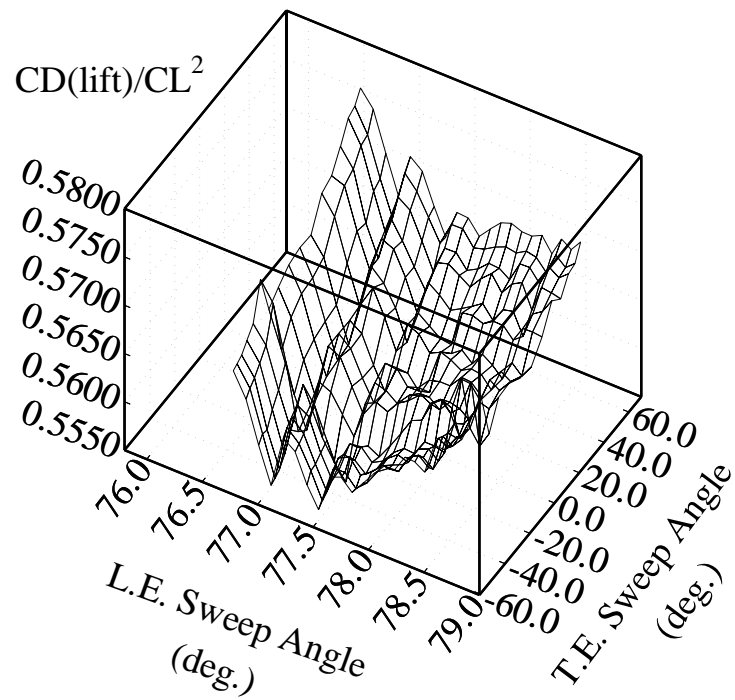


Figure 3.2: Numerical noise in computed drag-due-to-lift (two variables).

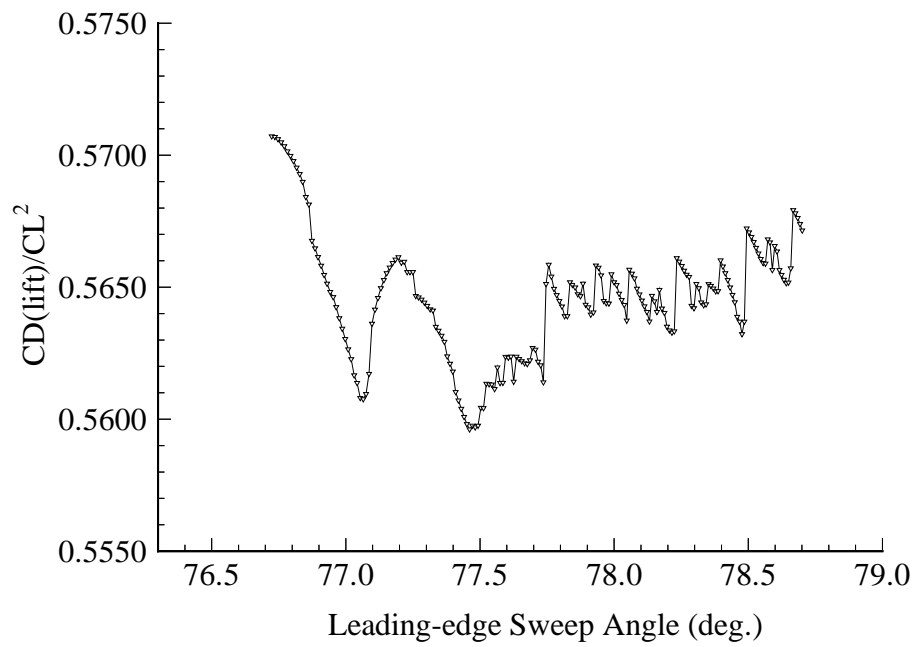


Figure 3.3: Numerical noise in computed drag-due-to-lift (one variable).

# Chapter 4

## Response Surface Modeling Methods

Design of experiment theory (DOE) [14, pages 79–133] is a branch of statistics which provides the researcher with methods for selecting the independent variable values at which a limited number of experiments will be conducted. The various experimental design methods create certain combinations of experiments (analyses) in which the independent variables are prescribed at specific values or *levels*. The results of these planned experiments are used to investigate the sensitivity of some dependent quantity, identified as the *response*, to the independent variables. The statistical tools used to model the sensitivity in the observed data include regression analysis, and analysis of variance (ANOVA) [14, page 28]. The collective use of DOE techniques, regression analysis, and ANOVA is termed a *response surface methodology* (RSM).

The primary objective of this research is to use the statistical methods in RSM to analyze and model computer data which contain numerical noise. The first approach, and the primary focus of this research, was to employ traditional response surface modeling techniques which use polynomial functions computed via the method of least squares to model trends in the observed data. Note that polynomial models have a smoothing effect on the experimental error (i.e., uncertainty or noise) in the observed data. A traditional RSM does not use interpolating functions because

such modeling methods replicate the experimental error in the observed data. The polynomial response surface modeling techniques are described below in Section 4.1.

The secondary objective of this research was to perform a preliminary investigation into the modeling methods employed in a growing field in statistics known as *computer experiments* which Sacks, et al. [23] have termed DACE (design and analysis of computer experiments). DACE modeling methods are based on elements of Bayesian statistics and on the Kriging process often used in the field of Geostatistics. In contrast to the polynomial modeling used in RSM, interpolating models are employed in the DACE modeling methods. While the DACE interpolating techniques will reproduce experimental error in observed data (or numerical noise in computer-generated data), the DACE methods have some attractive modeling properties which are detailed below in Section 4.4.

As an aside, the reader familiar with the work of Taguchi [57] in the field of robust parameter design will recognize many similarities among the design of experiments methods presented below. While Taguchi's methods have been widely adopted and have been quite successful, they lack a rigorous foundation in statistical theory. Myers and Montgomery [14, pages 462–480] discuss the limitations of Taguchi's methods and they suggest an approach whereby Taguchi's methods are rectified with classical design of experiments statistical theory.

## 4.1 Response Surface Modeling

### 4.1.1 RSM and Physical Experiments

Response surface modeling methods originally were developed to analyze experimental data and to create empirical models of the observed response values. The particular forte of RSM is its applicability to investigations where there are few observations because the physical experiment is both very expensive and very time consuming to perform.

In RSM the relationship between observations and independent variables is defined



as

$$y = f(\mathbf{x}), \quad (4.1)$$

where  $y$  is the observed response,  $\mathbf{x}$  is the vector of  $n_v$  independent variables defined as

$$\mathbf{x} = [x_1, x_2, \dots, x_{n_v}], \quad (4.2)$$

and  $f(\mathbf{x})$  is the unknown function. The empirical model of the unknown function found via the application of RSM is defined as

$$\hat{y} = \hat{f}(\mathbf{x}), \quad (4.3)$$

where  $\hat{f}(\mathbf{x})$  typically is a first or second order polynomial in  $\mathbf{x}$ . Note that the random error (uncertainty) present in stochastic experimental data is implicit in both  $f(\mathbf{x})$  and  $\hat{f}(\mathbf{x})$ .

RSM employs the statistical techniques of regression analysis and ANOVA to determine  $\hat{f}(\mathbf{x})$  through a systematic decomposition of the variability in the observed response values. The empirical model is then estimated by assigning portions of the variability to either the effect of an independent variable or to random error.

### 4.1.2 RSM and Computer Simulations

Complex computer simulations are now ubiquitous in science and engineering. In many instances these simulations require enormous amounts of computer time and memory when performed. In addition, a single simulation can necessitate weeks or months of human labor-intensive effort to properly set-up and “debug.” As a consequence, it is attractive to consider applying RSM methods, originally developed for physical experiments, to computer simulations as well.

It should be noted that in this research project RSM methods are applied to the results obtained from *deterministic* computer simulations, where simulations performed with identical starting values and initial conditions produce identical results. In such simulations numerical noise takes the place of random error. The application of RSM methods to *stochastic* computer simulations is beyond the scope of this research project.

### 4.1.3 Polynomial Models for RSM

In many RSM applications, either linear or quadratic polynomials are assumed to accurately model the observed response values. Although this is certainly not true for all cases, RSM becomes prohibitively expensive when cubic and higher-order polynomials are chosen for experiments involving several variables. In addition, cubic and higher-order polynomial models may contain one or more inflection points. In gradient-based numerical optimization schemes the optimizer may converge to an inflection point rather than to a local or global optimum.

If  $n_s$  analyses are conducted and  $p = 1, \dots, n_s$ , then a quadratic response surface (RS) model has the form

$$y^{(p)} = c_o + \sum_{1 \leq j \leq n_v} c_j x_j^{(p)} + \sum_{1 \leq j \leq k \leq n_v} c_{(n_v-1+j+k)} x_j^{(p)} x_k^{(p)}, \quad (4.4)$$

where  $y^{(p)}$  is the response;  $x_j^{(p)}$  and  $x_k^{(p)}$  are the  $n_v$  design variables; and  $c_o$ ,  $c_j$ , and  $c_{(n_v-1+j+k)}$  are the unknown polynomial coefficients. Note that there are  $n_t = (n_v + 1)(n_v + 2)/2$  coefficients (i.e., model terms) in the quadratic polynomial. This polynomial model may be written in matrix notation as

$$y^{(p)} = \mathbf{c}^T \bar{\mathbf{x}}^{(p)}, \quad (4.5)$$

where  $\mathbf{c}$  is the vector of length  $n_t$  of unknown coefficients to be estimated,

$$\mathbf{c} = [c_0, c_1, \dots, c_{n_t-1}], \quad (4.6)$$

and  $\bar{\mathbf{x}}^{(p)}$  is the vector of length  $n_t$  corresponding to the form of the  $x_j^{(p)}$  and  $x_k^{(p)}$  terms in the polynomial model (Equation 4.4). For the  $p^{\text{th}}$  observation this is

$$\bar{\mathbf{x}}^{(p)} = [1, x_1^{(p)}, x_2^{(p)}, \dots, x_1^{(p)} x_2^{(p)}, \dots, (x_{n_v}^{(p)})^2]. \quad (4.7)$$

Note that there is a difference between the  $p^{\text{th}}$  vector of independent variables,  $\mathbf{x}^{(p)}$ , and the  $p^{\text{th}}$  vector of independent variables mapped into the form of the polynomial model,  $\bar{\mathbf{x}}^{(p)}$ .

Estimating the unknown coefficients requires  $n_s$  analyses, where  $n_s \geq n_t$ . Under such conditions, the estimation problem may be formulated in matrix notation as

$$\mathbf{y} \approx \mathbf{X}\mathbf{c}, \quad (4.8)$$

where  $\mathbf{y}$  is the vector of  $n_s$  observed response values,

$$\mathbf{y} = [y^{(1)}, y^{(2)}, \dots, y^{(n_s)}], \quad (4.9)$$

and  $\mathbf{X}$  is the matrix formed by the  $p$  row vectors  $\bar{\mathbf{x}}^{(\mathbf{p})}$  which is assumed to have rank  $n_t$ . Thus,  $\mathbf{X}$  may be expressed as

$$\mathbf{X} = \begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \cdots & (x_{n_v}^{(1)})^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(n_s)} & x_2^{(n_s)} & \cdots & (x_{n_v}^{(n_s)})^2 \end{bmatrix}. \quad (4.10)$$

The unique least squares solution to Equation 4.8 is

$$\hat{\mathbf{c}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}, \quad (4.11)$$

where  $(\mathbf{X}^T \mathbf{X})^{-1}$  exists if the rows of  $\mathbf{X}$  are linearly independent. When  $\hat{\mathbf{c}}$  is substituted for  $\mathbf{c}$  into Equation 4.5, values of the response may be predicted at any location  $\mathbf{x}$  by mapping  $\mathbf{x}$  into  $\bar{\mathbf{x}}^{(\mathbf{p})}$ . In matrix notation this corresponds to

$$\hat{\mathbf{y}} = \hat{\mathbf{c}}^T \bar{\mathbf{x}}^{(\mathbf{p})}. \quad (4.12)$$

Note that if  $n_s > n_t$  the system of equations is overdetermined. Thus, the *predicted* response values (from the polynomial model) at the original sample locations may differ from the *observed* response values at the sampled locations.

In a study conducted early in this research project [2], several different polynomial models were investigated in addition to quadratic polynomials. These other models (expressed as a function of two variables) were the bilinear tensor product

$$y^{(p)} = (c_1 x_1^{(p)} + c_2)(c_3 x_2^{(p)} + c_4), \quad (4.13)$$

which has four terms when expanded, and the biquadratic tensor product,

$$y^{(p)} = (c_1 (x_1^{(p)})^2 + c_2 x_1^{(p)} + c_3) (c_4 (x_2^{(p)})^2 + c_5 x_2^{(p)} + c_6), \quad (4.14)$$

which has nine terms when expanded. In addition, nonlinear regression was used to examine the (1,1) rational function

$$y^{(p)} = \frac{c_1 x_1^{(p)} + c_2 x_2^{(p)} + c_3}{c_4 x_1^{(p)} + c_5 x_2^{(p)} + c_6}, \quad (4.15)$$

where the (1,1) denotes a first order polynomial in both the numerator and the denominator. The results of this investigation indicated that the quadratic polynomial model provided the best compromise between modeling accuracy and computational expense out of all the models studied.

## 4.2 ANOVA and Regression Analysis

In addition to estimating the coefficients in the quadratic polynomial model, ANOVA and regression analysis also yield a measure of the uncertainty in the coefficients. This uncertainty estimation is provided by the  $t$ -statistic which is defined in Myers and Montgomery [14, page 32] as

$$t = \frac{c_{j-1}}{\sqrt{\hat{\sigma}^2(\mathbf{X}^T \mathbf{X})_{jj}^{-1}}}, \quad (4.16)$$

where  $\hat{\sigma}^2$  is the estimate of the variance in the observed response data and  $j = 1, \dots, n_t$ . Note that the reciprocal of the  $t$ -statistic is an estimate of the standard deviation of each coefficient as a fraction of its value. Accordingly, coefficients with “low” values for the  $t$ -statistic are not accurately estimated. It is the prerogative of the user to select the minimum allowable  $t$ -statistic. This choice typically depends on the number of observed response values (also known as *degrees of freedom* (DOF) in the statistical lexicon) used to create the response surface model. For a coefficient estimated with at least of one degree of freedom, a 90 percent confidence interval requires a  $t$ -statistic of greater than 6.31 for that coefficient to be estimated accurately. This value of 6.31 is obtained from tabulated values of the probability density function of the  $t$ -distribution which are found in most text books on statistics (cf., [58, page 449]).

Various procedures exist for removing terms in a polynomial model which are not accurately estimated (i.e., terms for which the coefficients have a low  $t$ -statistic). Allowing these poorly estimated terms to remain in the response surface polynomial model may actually reduce the prediction accuracy of the model. A common statistical measure of the utility of removing terms from a polynomial model is the adjusted  $R^2$

value ( $R^2_{adj}$ ) [14, pages 30–31]. The adjusted  $R^2$  value is calculated as

$$R^2_{adj} = 1 - \frac{SSE/DOF_{SSE}}{SYY/DOF_{SYY}}, \quad (4.17)$$

where  $SSE$  is the error sum of squares,  $SYY$  is the total sum of squares, and the DOF for  $SSE$  and  $SYY$  are obtained from ANOVA calculations [14, pages 28–31]. The maximum value for  $R^2_{adj}$  is 1.0 which occurs when all of the variation in the observed response values is described by the trends of the response surface polynomial model. Typical values for  $R^2_{adj}$  are  $0.9 \leq R^2_{adj} \leq 1.0$  when the observed response values are accurately predicted by the response surface model.

The commercial statistical package JMP [59] was used to perform ANOVA and regression analysis along with other statistical analyses. See the *JMP User's Manual* and the text by Myers and Montgomery [14, pages 28–31] for a description of the statistical methods which underlie ANOVA and regression analysis, including the calculation of  $R^2$  and  $R^2_{adj}$ .

## 4.3 Design of Experiments Theory

### 4.3.1 Full Factorial Experimental Designs

Prior to creating an experimental design, the allowable range of each of the  $n_v$  variables is defined by lower and upper bounds. The allowable range is then discretized at equally-spaced levels. For numerical stability and for ease of notation the range of each variable is scaled to span  $[-1, 1]$  [14, page 22]. The region enclosed by the lower and upper bounds on the variables is termed the design space, the vertices of which determine an  $n_v$ -dimensional cube. If each of the variables is specified at only the lower and upper bounds (two levels), the experimental design is called a  $2^{n_v}$  full factorial. Similarly, a  $3^{n_v}$  full factorial design is created by specifying the lower bound, midpoint, and upper bound (three levels:  $\{-1, 0, 1\}$ ) for each of the  $n_v$  variables. A  $3^3$  full factorial design is shown in Figure 4.1.

The construction of a quadratic response surface model in  $n_v$  variables requires at least  $(n_v + 1)(n_v + 2)/2$  response evaluations. A  $3^{n_v}$  full factorial design provides

ample response evaluations to permit the estimation of the RS model coefficients. For example, fitting a quadratic response surface model in three variables ( $n_v = 3, n_t = 10$ ) requires  $n_s \geq 10$  evaluations, and a  $3^3$  full factorial design provides 27 evaluations. However, as  $n_v$  becomes large ( $n_v > 10$ ) the evaluation of both  $2^{n_v}$  and  $3^{n_v}$  full factorial designs becomes impractical (e.g.,  $2^{29} \approx 5.4 \cdot 10^8$ ). A full factorial design typically is used for ten or fewer variables. For greater than ten variables other experimental designs may be used. These include fractional factorial designs, small composite designs [14, pages 135–141, 351–357], and partially balanced incomplete block (PBIB) designs such as those used by Kaufman [60].

### 4.3.2 Central-Composite Experimental Designs

To reduce the number of needed experiments, another experimental design method known as central composite design (CCD) may be used as is shown in Myers and Montgomery [14, pages 297–305]. In CCD a  $2^{n_v}$  full factorial experimental design is employed along with  $2n_v$  “star” design points and one or more “center” design points. A three variable CCD is shown in Figure 4.2. In this experimental design, the star points lie outside the boundary created by the  $2^{n_v}$  full factorial points. The distance from the star points to the center of the CCD typically varies from 1.0 to  $\sqrt{n_v}$  [14, pages 299]. Using the response data from the  $2^{n_v} + 2n_v + 1$  experiments specified by a CCD, a quadratic response surface model may be constructed. As with the  $2^{n_v}$  and  $3^{n_v}$  full factorial designs, the number of required CCD experiments also becomes impractical as  $n_v$  becomes large.

### 4.3.3 $D$ -Optimal Experimental Designs

Response surface modeling methods often employ a full factorial or a central-composite experimental design. However, full factorial experimental designs are intended for use with rectangular design spaces and not the irregularly shaped (even nonconvex) design spaces that may arise in the HSCT design problems considered here. A previous study by Craig [61] and some investigations performed early in this research study [2, 62] indicated that  $D$ -optimal experimental designs [14, pages 364–366] provided

an attractive method for creating experimental designs inside an irregularly shaped design space. In addition,  $D$ -optimal experimental designs may be constructed which require fewer than the  $2^{n_v} + 2n_v + 1$  response values needed for central-composite experimental designs. A sample  $D$ -optimal design is shown in Figure 4.3.

A  $D$ -optimal experimental design is a collection of sample sites in a design space for which the quantity  $|\mathbf{X}^T \mathbf{X}|$  is maximized over all possible site locations. Box and Draper [63] describe the appealing properties of  $D$ -optimal experimental designs which include

1. the variance (uncertainty) in the estimated coefficients is minimized,
2. the maximum variance of any predicted value,  $\hat{y}(\mathbf{x})$ , is minimized, and
3.  $|\mathbf{X}^T \mathbf{X}|$  is invariant to the scaling of  $\mathbf{x}$ .

Typically, optimization methods are used to create  $D$ -optimal experimental designs. While the optimization problem may be posed in either a continuous or a discrete form, it is attractive to use a discrete formulation when considering design spaces that have irregularly shaped boundaries for which there are no analytical descriptions. In this discrete form, one selects a set of  $n_s$  sample sites (locations) in a design space from a pool of  $n_c$  candidate locations ( $n_c \geq n_s$ ), where the pool of  $n_c$  candidate locations is defined *a priori* by the user. An iterative numerical optimization method is then employed to find the  $n_s$  locations which maximize  $|\mathbf{X}^T \mathbf{X}|$ .

Selecting a set of  $n_s$  locations out of  $n_c$  locations is a combinatorial problem having  $\binom{n_c}{n_s} = n_c! / (n_s!(n_c - n_s)!)$  possible combinations. This combinatorial problem quickly grows as is illustrated in a simple example involving two variables ( $n_v = 2$ ), where the two-variable design space is discretized using an  $11 \times 11$  mesh ( $n_c = 121$ ), and 25 sample locations ( $n_s = 25$ ) are desired. The combinatorial problem is then  $\binom{121}{25}$  for which there are  $5.26 \cdot 10^{25}$  possible combinations, one or more of which are  $D$ -optimal.

Clearly, finding a set of  $D$ -optimal locations requires a numerical optimization method. During these research efforts two methods were used. Earlier research [2, 5, 37, 62] used a genetic algorithm (GA) developed by Dr. Robert Narducci to search through the  $\binom{n_c}{n_s}$  combinations. However, the brute force GA method became too

computationally expensive for more than five variables. Thus, a more computationally inexpensive method employing Mitchell’s “k-exchange” method [64] was developed by Dr. Dan Haim which thus far has been applied to problems of up to 25 variables [65]. Although JMP has the capability to create  $D$ -optimal designs, inherent software limitations and computational expense precluded its use for more than five variables.

## 4.4 Design and Analysis of Computer Experiments

Prior to a description of the mathematical underpinnings of the DACE method, it is useful to compare the philosophy of polynomial modeling methods to that of DACE interpolating methods. To simplify this comparison, the phrase *RS model* will henceforth refer to polynomial models created via least squares surface fitting, and the term *DACE model* will refer to the class of interpolating models based on Bayesian statistics and Kriging. Although both RS models and DACE models are approximations to the true, unknown *response surface* and as such are technically response surface models, the statistical literature tends to reserve the term *response surface model* for polynomial models. The phrase *polynomial RS model* will be used to reinforce this distinction.

Polynomial RS models can be thought of as “global” models in which all of the  $n_s$  observed values of the response are equally weighted in the fitting of the polynomial surface. At an unsampled location in design space,  $\mathbf{x}$ , response observations that are near to  $\mathbf{x}$  (in the sense of Euclidean distance) have an equal influence on the predicted response,  $\hat{f}(\mathbf{x})$ , as do the response observations that are far from  $\mathbf{x}$ . It can be argued that such a global model may not be the best approximator if the true unknown response has many real local optima (as opposed to the artificial local optima created by numerical noise). In such a situation an approximation scheme having “local” modeling properties may be more attractive, i.e., where  $\hat{f}(\mathbf{x})$  is more strongly influenced by nearby measured response values and is less strongly influenced by those further away. Such local modeling behavior is characteristic of interpolation models, of which DACE models are one particular implementation.



### 4.4.1 Mathematics of DACE Modeling

The objective here is to provide an introduction to the statistics and mathematics of DACE modeling. A detailed treatment of the statistical and mathematical methods involved in DACE is both beyond the scope of this dissertation and is unnecessarily confusing to the reader who has a limited knowledge of statistical principles. The interested reader is directed to the work of Sacks et al. [23]; Koehler and Owen [66]; Osio and Amon [67]; and Booker et al. [68] where more detailed discussions of the fundamental statistical and mathematical concepts may be found.

Before addressing the principles underlying DACE modeling methods, it is useful at this point to introduce the term *prior distribution* which is often used in Bayesian statistics. A prior distribution refers to the probability density function which one assigns to a variable of unknown value *before* any experimental data on that variable are collected [69, pages 4,5]. The prior distribution is the mechanism in Bayesian statistics through which one applies past experiences, knowledge, and intuition when performing an experiment. However, one's choice of the prior distribution biases the interpretation of the experimental data. This intentional bias is the source of much controversy in the statistical community. In spite of the differences between classical statistics and Bayesian statistics, Berger [69, pages 109,110] emphasizes that both classical and Bayesian statistics have merit, and he provides examples where the same interpretations of experimental data are obtained using both methods.

In the DACE literature the true, unknown function to be modeled is typically expressed as

$$y(\mathbf{x}) = f(\mathbf{x}) + Z(\mathbf{x}), \quad (4.18)$$

where  $f(\mathbf{x})$  is a known function of  $\mathbf{x}$  and  $Z(\mathbf{x})$  is a Gaussian random function with zero mean and with variance of  $\sigma^2$  (i.e., the behavior of  $Z(\mathbf{x})$  follows a normal or *Gaussian* distribution). The  $f(\mathbf{x})$  term in Equation 4.18 in some sense is a “global” model for the entire design space based on the  $n_s$  response observations, while the  $Z(\mathbf{x})$  term creates a “localized” deviation from the global model.

In much of the current DACE literature the term  $f(\mathbf{x})$  in Equation 4.18 is considered a constant and is indicated using the term  $\beta$  ([23], [66], [68]). Equation 4.18 then

becomes

$$y(\mathbf{x}) = \beta + Z(\mathbf{x}), \quad (4.19)$$

The term  $\beta$  takes on different meanings depending on one's statistical point of view. From the perspective of the Kriging approach  $\beta$  is simply an unknown constant to be estimated based on the  $n_s$  observed response values. From the perspective of Bayesian statistics  $\beta$  is a random variable with a prior distribution denoted as  $\pi_\beta$ . The interpretations of  $\beta$  are identical regardless of the statistical perspective if  $Z(\mathbf{x})$  has a Gaussian distribution and  $\pi_\beta$  has a uniform distribution [23].

The covariance matrix of  $Z(\mathbf{x})$  is expressed as

$$\text{Cov}[Z(\mathbf{x}^{(i)}), Z(\mathbf{x}^{(j)})] = \sigma^2 \mathbf{R}[R(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})], \quad (4.20)$$

where  $\mathbf{R}$  is the *correlation matrix*, and  $R$  is the *correlation function* which is selected by the user. In Equation 4.20  $i = 1, \dots, n_s$  and  $j = 1, \dots, n_s$ . Note that the correlation matrix  $\mathbf{R}$  is symmetric with values of unity along the diagonal.

As noted above, the user may select the form of the correlation function  $R$ . Sacks et al. [23]; Koehler and Owen [66]; and Booker et al. [68] provide a detailed description of various correlation functions that may be used. A choice for  $R$  often found in the statistical literature, and employed in [68], is an exponential correlation function

$$R(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \exp\left[-\sum_{k=1}^{n_v} \theta_k |\mathbf{x}_k^{(i)} - \mathbf{x}_k^{(j)}|^2\right], \quad (4.21)$$

where  $\theta_k$  is the vector of unknown *correlation parameters*. For this research only a single correlation parameter is used instead of a vector of correlation parameters. The scalar correlation parameter is denoted as  $\theta$ . Thus, Equation 4.22 may be rewritten as

$$R(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \exp\left[-\theta \sum_{k=1}^{n_v} |\mathbf{x}_k^{(i)} - \mathbf{x}_k^{(j)}|^2\right], \quad (4.22)$$

The process by which a value for  $\theta$  is estimated is given below.

Another term of interest is the *correlation vector*,  $\mathbf{r}(\mathbf{x})$ , between the response at a location,  $\mathbf{x}$ , and the  $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n_s)}$  response values. The correlation vector is expressed as

$$\mathbf{r}(\mathbf{x}) = R(\mathbf{x}, \mathbf{x}^{(i)}) = [R(\mathbf{x}, \mathbf{x}^{(1)}), R(\mathbf{x}, \mathbf{x}^{(2)}), \dots, R(\mathbf{x}, \mathbf{x}^{(n_s)})]. \quad (4.23)$$

### 4.4.2 Estimation of the Correlation Parameter

While Equation 4.19 represents the true, unknown function to be approximated, the computed (i.e., estimated) DACE model is given the symbol  $\hat{y}(\mathbf{x})$ . In statistical notation, this estimated DACE model is defined as

$$\hat{y}(\mathbf{x}) = E(y(\mathbf{x})|y(\mathbf{x}^{(1)}), \dots, y(\mathbf{x}^{(n_s)})), \quad (4.24)$$

where the expression  $E(\cdot)$  is the statistical symbol for the *expected* value of  $(\cdot)$  and the expression  $E(A|B)$  is interpreted as the expected value of  $A$  given the information  $B$ . The terms  $y(\mathbf{x}^{(1)}), \dots, y(\mathbf{x}^{(n_s)})$  are the  $n_s$  observed values of the response,  $y(\mathbf{x})$  is the true response one is attempting to estimate, and  $\hat{y}(\mathbf{x})$  is the actual estimate of the response (which one hopes is close to  $y(\mathbf{x})$ ). This distinction between  $\hat{y}(\mathbf{x})$  and  $y(\mathbf{x})$  is necessary so that the concept of mean squared error (MSE) may be introduced where

$$MSE = E(\hat{y}(\mathbf{x}) - y(\mathbf{x}))^2. \quad (4.25)$$

This is simply a measure of the amount of error between the DACE model,  $\hat{y}(\mathbf{x})$ , and the true model,  $y(\mathbf{x})$ , at all locations,  $\mathbf{x}$ , in the design space. Since the DACE model performs interpolation there is no error between the DACE model and the true model at the  $n_s$  sites where the values of the response are known.

If MSE is minimized,  $\hat{y}(\mathbf{x})$  becomes

$$\hat{y}(\mathbf{x}) = \hat{\beta} + \mathbf{r}^T(\mathbf{x})\mathbf{R}^{-1}(\mathbf{y} - \hat{\beta}\mathbf{f}), \quad (4.26)$$

where  $\hat{\beta}$  is estimated as

$$\hat{\beta} = (\mathbf{f}^T\mathbf{R}^{-1}\mathbf{f})^{-1}\mathbf{f}^T\mathbf{R}^{-1}\mathbf{y}. \quad (4.27)$$

The vector  $\mathbf{f}$  has length  $n_s$  with all entries equal to unity

$$\mathbf{f} = [1, \dots, 1], \quad (4.28)$$

which is a result of the assumption that all of the variability in  $y(\mathbf{x})$  is accounted for in the  $Z(\mathbf{x})$  term. While the usual notation for a vector with all entries equal to unity is  $\mathbf{e}$ , the vector  $\mathbf{f}$  is retained to maintain similarity with the notation used in Koehler

and Owen [66] and Booker et al. [68]. As a result of minimizing MSE the estimated variance,  $\hat{\sigma}^2$ , becomes

$$\hat{\sigma}^2 = \frac{(\mathbf{y} - \hat{\beta}\mathbf{f})^T R^{-1}(\mathbf{y} - \hat{\beta}\mathbf{f})}{n_s}. \quad (4.29)$$

Note that Equations 4.27 and 4.29 implicitly depend on the correlation parameter  $\theta$ .

As is shown in [68] the choice of  $\theta$  which minimizes MSE also maximizes the expression

$$(-n_s/2) \left[ (\ln \hat{\sigma}^2) + \ln |\mathbf{R}| \right]. \quad (4.30)$$

The problem of finding the value of  $\theta$  which minimizes MSE may be expressed in the form of a simply constrained one-dimensional optimization problem as

$$\max_{\theta \in R^1} (-n_s/2) \left[ (\ln \hat{\sigma}^2) + \ln |\mathbf{R}| \right], \quad \text{subject to } 0 \leq \theta \leq \infty, \quad (4.31)$$

where both  $\hat{\sigma}^2$  and  $\mathbf{R}$  are functions of  $\theta$ . Thus, by solving this one-dimensional optimization problem the DACE approximation model  $\hat{y}(\mathbf{x})$  is completely defined.

### 4.4.3 Using Multi-Fidelity Data in DACE Modeling

An extension of DACE modeling methods is presented by Osio and Amon [67] in which a modeling strategy that emulates the stages of the engineering design process is developed. In this approach the stages correspond to the use of increasingly sophisticated and computationally expensive analysis techniques during the design process; a paradigm which shares much in common with the variable-complexity modeling paradigm described in Chapter 5. The essence of Osio's and Amon's approach is that the design process is divided into  $k = 1, \dots, n_{st}$  stages, where the superscript ( $k$ ) is used to represent the  $k^{th}$  stage. There are  $n_s^{(k)}$  response evaluations at each stage where typically  $n_s^{(1)} \geq \dots n_s^{(k)} \geq \dots \geq n_s^{(n_{st})}$ .

At each of the  $n_{st}$  stages a new DACE model is calculated using the methods described above for the estimation of the parameter  $\theta$ . However, the parameter  $\hat{\beta}$  is treated differently. For the  $k = 1$  stage the term  $\hat{\beta}$  is calculated using Equation 4.27. For stages  $k > 1$  the DACE model uses the estimate of  $\hat{\beta}$  at the  $k - 1$  stage. Thus, the term  $\hat{\beta}$  is used to pass *prior distribution* information to the  $k^{th}$  stage DACE model

from the  $k - 1$  stage. Specifically, the  $k = 1$  stage DACE model is

$$\hat{y}(\mathbf{x})^{(1)} = \hat{\beta}^{(1)} + (\mathbf{r}^T(\mathbf{x})\mathbf{R}^{-1})^{(1)}(\mathbf{y}^{(1)} - \hat{\beta}^{(1)}\mathbf{f}), \quad (4.32)$$

where  $\hat{\beta}$  is calculated using Equation 4.27. The DACE models for the  $k = 2 \dots n_{st}$  stages are

$$\begin{aligned} \hat{y}(\mathbf{x})^{(2)} &= \hat{\beta}^{(1)} + (\mathbf{r}^T(\mathbf{x})\mathbf{R}^{-1})^{(2)}(\mathbf{y}^{(2)} - \hat{\beta}^{(1)}\mathbf{f}), \\ &\vdots \\ \hat{y}(\mathbf{x})^{(k)} &= \hat{\beta}^{(k-1)} + (\mathbf{r}^T(\mathbf{x})\mathbf{R}^{-1})^{(k)}(\mathbf{y}^{(k)} - \hat{\beta}^{(k-1)}\mathbf{f}), \\ &\vdots \\ \hat{y}(\mathbf{x})^{(n_{st})} &= \hat{\beta}^{(n_{st}-1)} + (\mathbf{r}^T(\mathbf{x})\mathbf{R}^{-1})^{(n_{st})}(\mathbf{y}^{(n_{st})} - \hat{\beta}^{(n_{st}-1)}\mathbf{f}). \end{aligned} \quad (4.33)$$

Note that in Equation 4.33 the optimal value of  $\theta^{(k)}$  is found using  $\hat{\beta}^{(k-1)}$ . See Osio and Amon [67] for a more complete description of this application of DACE modeling.

From an engineering design perspective the approach put forth by Osio and Amon is appealing. The modeling function at the  $k^{th}$  level has an explicit dependence on previously collected data, past engineering experience, and design intuition through the term  $\hat{\beta}^{(k-1)}$ . To gain a better understanding of DACE modeling and the methods of Osio and Amon, a series of test problems were examined. These efforts are detailed in Chapters 7 and 8

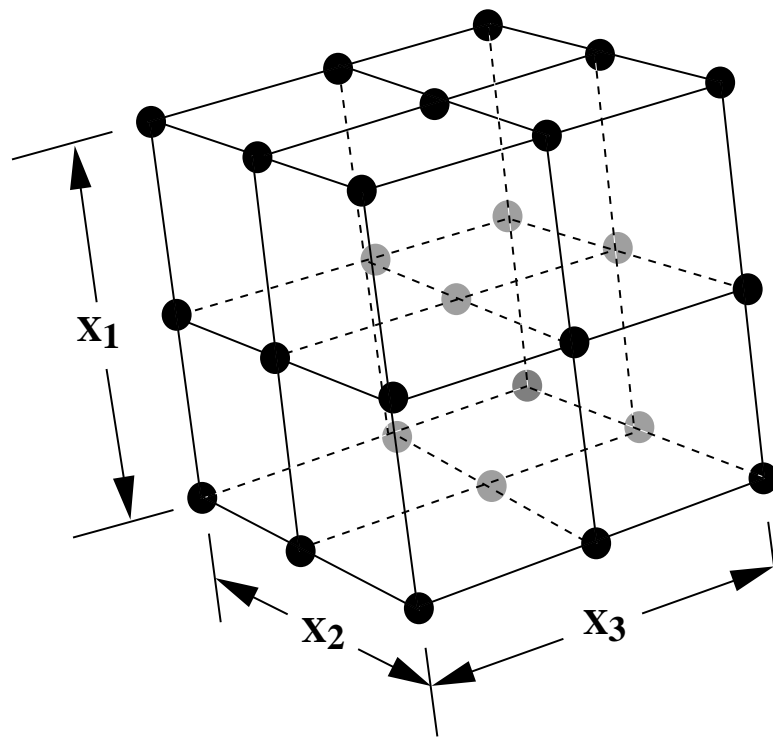


Figure 4.1: A 3<sup>3</sup> full factorial experimental design (27 points).

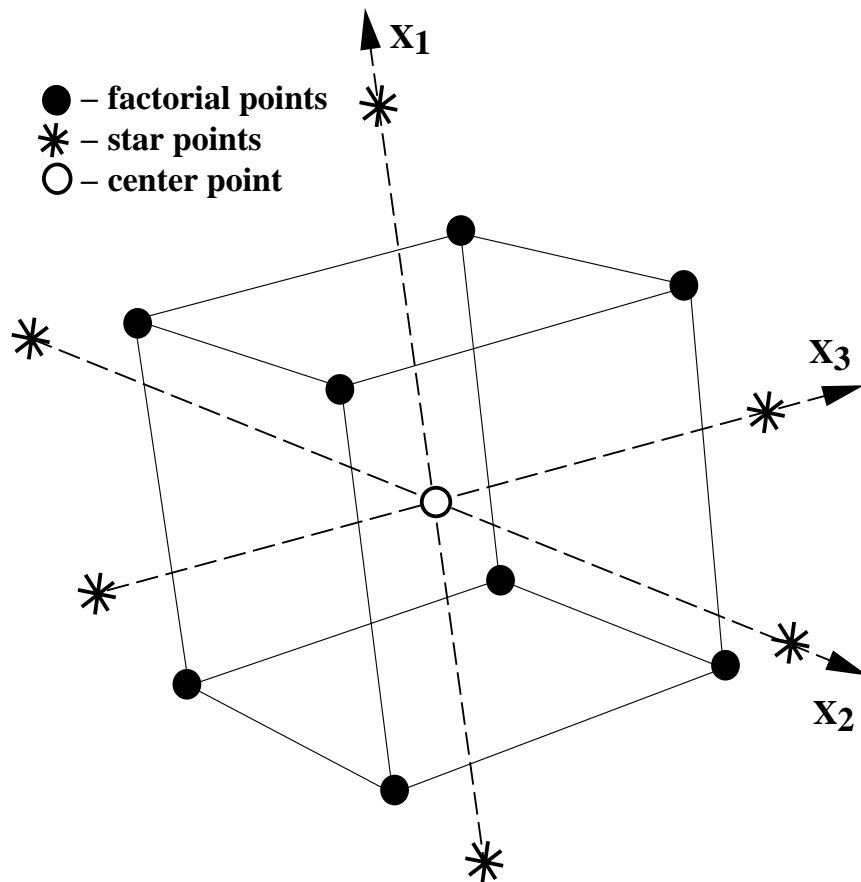


Figure 4.2: A three variable central composite experimental design (15 points).

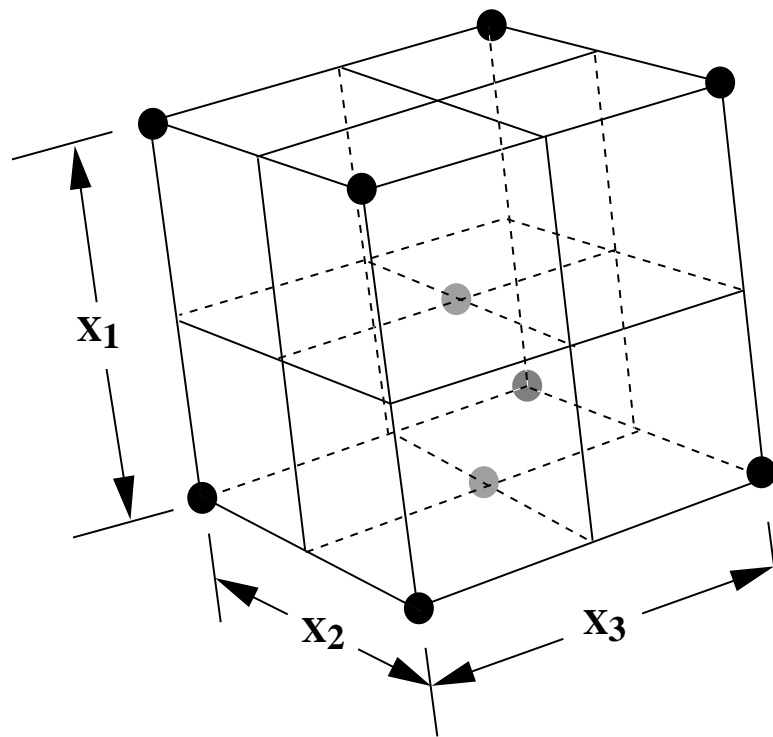


Figure 4.3: A three variable  $D$ -optimal experimental design (11 points).



# Chapter 5

## Using the VCRSM Method in HSCT Optimization

### 5.1 Description of Steps in the VCRSM Method

The construction of response surface models in the VCRSM method is performed in a series of steps which are conducted before the aircraft system optimization is performed. This methodology is illustrated in Figure 5.1 and each step is described below.

#### 5.1.1 Initial HSCT Configuration, Step 1

In this first step of the VCRSM procedure the variables and their initial values (also referred to as nominal values) are selected. Typically these nominal values of the variables are selected based on the user's previous experience or on similar aircraft configurations. In this work, the variables defining the HSCT configuration had already been selected through the work of Hutchison [54]. Nominal values for the variables were selected based on the author's own experience in using the HSCT analysis/optimization code combined with the results obtained by Kaufman [5] and MacMillin [65] in related HSCT optimization work.

### 5.1.2 Design Space Boundaries, Step 2

For each variable a lower and upper bound on its value must be specified prior to optimization. As an example, for the five variable HSCT optimization problem the upper and lower bounds were  $\pm 20$  percent of the nominal value of  $C_{root}$ ,  $C_{tip}$ , and  $t/c$  ratio;  $\pm$  nine percent on  $\Lambda_{LEI}$ ; and  $\pm$  three percent on  $W_{fuel}$ . Note that the user must exercise caution when defining the lower and upper boundaries for each variable since it is possible to create physically unrealistic HSCT configurations that cannot be evaluated using the analysis methods.

Collectively the lower and upper bounds on the  $n_v$  variables define a cube in  $n_v$  dimensions (hypercube) which has  $2^{n_v}$  vertices. Each  $n_v$ -dimensional location in the cube corresponds to a unique HSCT configuration. In optimization problems this hypercube is termed the *design space*. While it is easy to visualize the design space in two dimensions (a square) and in three dimensions (a cube) it is difficult to visualize a design space for  $n_v > 3$ . The visualization of such high dimensional spaces may be accomplished by projecting locations in an  $n_v > 3$  space onto two or three dimensions. However, it is inevitable that some information is lost in this projection process. Also, for an  $n_v$ -dimensional space there are  $\binom{n_v}{n_v-3} = n_v!/((n_v-3)!)3!$  possible three-dimensional projections; a number which quickly becomes large as  $n_v$  grows.

### 5.1.3 Full Factorial Experimental Design, Step 3

In this step the methods from DOE theory are used to discretize the design space into  $n_i$  locations where each location corresponds to a particular HSCT configuration. For the five and ten variable HSCT optimization problems five-level and three-level full factorial experimental designs were used, respectively. Note that for  $n_v > 10$  the use of a three-level full factorial experimental design results in an unacceptably large number of HSCT configurations. See the work of Kaufman [60] for a description of Partially Balanced Incomplete Block experimental designs that may be used in lieu of full factorial experimental designs when  $n_v > 10$ . This topic also is addressed in Section 5.2

### 5.1.4 Low Fidelity Analyses, Step 4

This is the first stage in the variable-complexity modeling portion of VCRSM. Here,  $n_i$  is  $\mathcal{O}(10^3 - 10^4)$  and these HSCT configurations are analyzed using the low fidelity analysis methods which require at most a few CPU seconds per evaluation. From these  $n_i$  analyses the objective function and constraints are obtained, as well as the aerodynamic quantities of interest such as  $C_{Dwave}, C_T/C_L^2, C_{L\alpha}$  at  $M=2.4$ , and  $C_{L\alpha}$  at  $M=0.2$ .

### 5.1.5 Reduced Design Space, Step 5

Using the constraint data collected from the low fidelity analyses, HSCT configurations that violated one or more constraints by an unacceptable amount were eliminated from the  $n_i$  pool. For example, in the ten variable HSCT optimization problem a geometric constraint violation was considered unacceptable if it exceeded the allowable value by more than five percent. The details of this elimination process are provided for the five and ten variable HSCT optimization problems in Chapter 6.

The HSCT configurations remaining after the elimination process comprise what has been termed the *reduced design space*. This is the same as the *approximation domain*, a term which appeared in several past publications relating to this research (cf., [37], [62]). Also, the reduced design space is similar to the term *reasonable design space* used by Kaufman et al. [5]. However, rather than eliminating infeasible HSCT configurations, Kaufman et al. modified the variables of the infeasible HSCT configurations until each HSCT configuration was on the boundary of the feasible design space. Kaufman's procedure is discussed below in Section 5.2.1.

### 5.1.6 D-Optimal Experimental Design, Step 6

The HSCT configurations which comprise the reduced design space become the set of  $n_c$  candidate HSCT configurations from which a subset of size  $n_s$  will be selected for analysis using the medium fidelity methods. These  $n_s$  HSCT configurations are selected to satisfy the  $D$ -optimality criterion (see Chapter 4). For small problems where  $n_v < 10$  and  $n_s$  is  $\mathcal{O}(10^3)$  the statistical analysis program JMP [59] may be

used to select the  $D$ -optimal subset of  $n_s$  HSCT configurations. For larger problems, the memory requirements and computational expense of using JMP on a Macintosh PowerPC were unattractive. Thus for larger problems the GRAMPS software package developed by Dr. Robert Narducci [70] was used initially. The memory limitations and computational expense of GRAMPS forced the migration to the DOPT software package developed by Dr. Dan Haim [71]. See more detailed discussions of JMP, GRAMPS, and DOPT in Chapter 4, Sections 4.2 and 4.3.

### 5.1.7 Medium Fidelity Analyses, Step 7

In this step the  $n_s$  HSCT configurations are evaluated using the medium fidelity analysis methods. As in Step 4, data for the objective function, the constraint values, and the aerodynamic quantities of interest ( $C_{Dwave}$ ,  $C_T/C_L^2$ ,  $C_{L\alpha}$  at  $M=2.4$ , and  $C_{L\alpha}$  at  $M=0.2$ ) are collected. For this stage  $n_s$  is typically  $\mathcal{O}(10^1 - 10^3)$ .

### 5.1.8 Creating Response Surface Models or DACE Models, Step 8

For this step the RS modeling methods or DACE modeling methods of Chapter 4 are applied to the medium fidelity analysis data from Step 7. See Chapter 6 for a detailed discussion of RS modeling for the five and ten variable HSCT optimization problems, and Chapter 8 for a description of DACE modeling used in the five variable HSCT optimization problem.

### 5.1.9 HSCT Optimization, Step 9

At this stage of the VCRSM procedure the RS and/or DACE models are used to replace the original medium fidelity analysis subroutines. Then an HSCT configuration optimization is conducted. For the five and ten variable HSCT optimization problems this subsonic and supersonic aerodynamic analysis methods were replaced with the approximation models and the DOT software package [52] was used to perform the HSCT optimization. The RS and DACE models are substantially less computationally

expensive than the original medium fidelity analysis methods which they replaced. Thus, the CPU time required for an HSCT configuration optimization is significantly reduced compared to an optimization using the original analysis methods. See Chapter 6 for more information on the CPU time required to perform HSCT optimizations using the approximation models.

### 5.1.10 Analysis of Optimal HSCT Configuration, Step 10a

If satisfied with the HSCT configuration optimization results from Step 9 the user may terminate the VCRSM process. Since there usually is some modeling error in the use of RS and DACE models it is prudent to perform a final analysis on the optimal HSCT configuration using the *original* aerodynamic analysis methods. This serves as a “reality check” on the optimization process which can exploit weaknesses (i.e., modeling inaccuracies) in the RS or DACE models. The VCRSM process is concluded if the final analysis results are in good agreement with the results predicted by the optimizer using the RS and/or DACE models, or if all computational resources have been exhausted. It is up to the user to define what is considered “good agreement.”

### 5.1.11 Define a New Design Space, Step 10b

If not satisfied with the optimization results from Step 9, the user can select new boundaries for the design space and then repeat Steps 2-9 of the VCRSM process. Such a scheme has been applied by Toropov [72] for structural optimization problems, by Narducci [6] for an aerodynamic shape optimization problem, and by Crisafulli [73] for an HSCT configuration optimization problem using variable-complexity modeling techniques similar to those described in Chapter 2.

## 5.2 Variations of the VCRSM Method

The VCRSM procedure evolved over a period of four years as experience was gained in the application of DOE theory and RS modeling methods to the HSCT optimization problem. During this time several applications of the evolving VCRSM method were

performed by students in the HSCT research group. These applications are reviewed below to provide the reader with an overall perspective of the various implementations of VCRSM by members of our research group.

### 5.2.1 Kaufman’s VCRSM Approach

Kaufman et al. [5] addressed an HSCT optimization problem involving 25 variables. The goal of this research was to create an RS model for the *wing bending material weight* ( $W_{bend}$ ) portion of the aircraft structural weight using medium fidelity data obtained from GENESIS [49], a finite element analysis/optimization software package. The motivation for this research was that the low fidelity  $W_{bend}$  estimates from FLOPS [11] were inaccurate for HSCT class aircraft and this inaccuracy yielded poor predictions of the aircraft structural weight.

The first major difference between the methods of Kaufman et al. (henceforth denoted as *Kaufman’s VCRSM method*) in comparison to the implementation of VCRSM described above (henceforth denoted as *Giunta’s VCRSM method*) arises in Step 2 of the VCRSM procedure. Kaufman’s method used  $\pm 80$  percent of the nominal variable values to create lower and upper bounds on 24 of the 25 variables, along with bounds of  $\pm 25$  percent on the fuel weight. The intention of this was to create a design space large enough to ensure that all feasible HSCT configurations were contained in its interior. In contrast, Giunta’s VCRSM method employed at most  $\pm 20$  percent lower and upper bounds with the intention of covering most, but not necessarily all, of the feasible design space.

The selection of 25 variables in Kaufman’s VCRSM application created another significant difference from Giunta’s VCRSM implementation in Step 3. Two-level and three-level full factorial experimental design for 25 variables require  $2^{25} \approx 3.36 \cdot 10^7$  and  $3^{25} \approx 8.47 \cdot 10^{11}$  HSCT configurations, respectively. Clearly analyzing such quantities of HSCT configurations is impractical. Thus, in Kaufman’s VCRSM method a variation of a Partially Balanced Incomplete Block experimental design was employed which produced  $n_i = 19651$  HSCT configurations [60]. For comparison, the three-level full factorial experimental designs used in Giunta’s VCRSM method for

the ten variable problem produced  $3^{10} = 59049$  HSCT configurations.

The consequence of choosing  $\pm 80$  percent lower and upper bounds on the variables arose in Step 5 of Kaufman's VCRSM approach where it was found that many of the HSCT configurations were in gross violation of the constraints, some to the point where the HSCT analysis software failed due to floating point exception errors or convergence failures in iterative calculations. Rather than eliminating these flawed HSCT configurations from the set of  $n_i = 19651$ , Kaufman elected to move the offending HSCT configurations in a direction through the design space that would decrease the amount of violation in the geometric constraints (for details of this procedure see [60]). The result of moving the HSCT configurations until they were geometrically feasible was the formation of a type of boundary in 25-dimensional space. This boundary enclosed what Kaufman termed the *reasonable design space*.

Kaufman used a version of Narducci's GRAMPS software [70] to create a  $D$ -optimal experimental design involving 3000 HSCT configurations ( $n_s = 3000$ ) from the set of 19651 HSCT configurations ( $n_c = 19651$ ). Then GENESIS was used to calculate the medium fidelity data for  $W_{bend}$ . Note that parallel computing methods were applied to distribute the computational load of 3000 finite element structural optimization among the processors of an Intel Paragon computer. The details of the use of parallel computing methods for this study may be found in Balabanov et al. [51]. Response surface models were created for  $W_{bend}$  from the GENESIS data, and an HSCT optimization was performed. The VCRSM process was terminated following completion of the HSCT optimization step.

### 5.2.2 Crisafulli's VCRSM Approach

In another application of the VCRSM method Crisafulli et al. [73] performed a nine variable HSCT optimization study which incorporated the modeling of low speed pitching moment characteristics. Crisafulli's VCRSM method used the low fidelity analysis data from Kaufman's set of 19651 HSCT configurations, but Crisafulli reduced this set to 835 HSCT configurations unique to the nine variable HSCT optimization problem. From the  $n_c = 835$  HSCT configurations  $n_s = 138$  were

selected using the  $D$ -optimal experimental design capabilities of JMP. The 138 HSCT configurations were evaluated using a medium fidelity level of analysis which used the pitching moment estimation method of Benoliel and Mason [74]. Six response surface models were constructed from the medium fidelity data and these were incorporated into the HSCT optimization software. After completing an HSCT configuration optimization, Crisafulli elected to follow Step 10b and he repeated the VCRSM process for a design space of reduced size. The boundaries for this *reduced design space* were defined as  $\pm 10$  percent of the variable values for the optimal HSCT configuration found in Step 9.

To examine the reduced design space, Crisafulli created a three-level full factorial experimental design having  $3^9 = 19683$  HSCT configurations. After performing low fidelity analyses on these, he eliminated 12150 HSCT configuration which grossly violated the geometric constraints on the HSCT configuration. Of the remaining  $n_c = 7173$  HSCT configurations,  $n_s = 150$  were selected using the  $D$ -optimal experimental design capabilities of JMP, and these were evaluated using the medium fidelity analysis methods. Response surface models were created from this data and another HSCT optimization was performed. The optimal HSCT configuration obtained in the reduced design space had a TOGW which was 16000 *lb* lower than the optimal HSCT found in the larger design space.



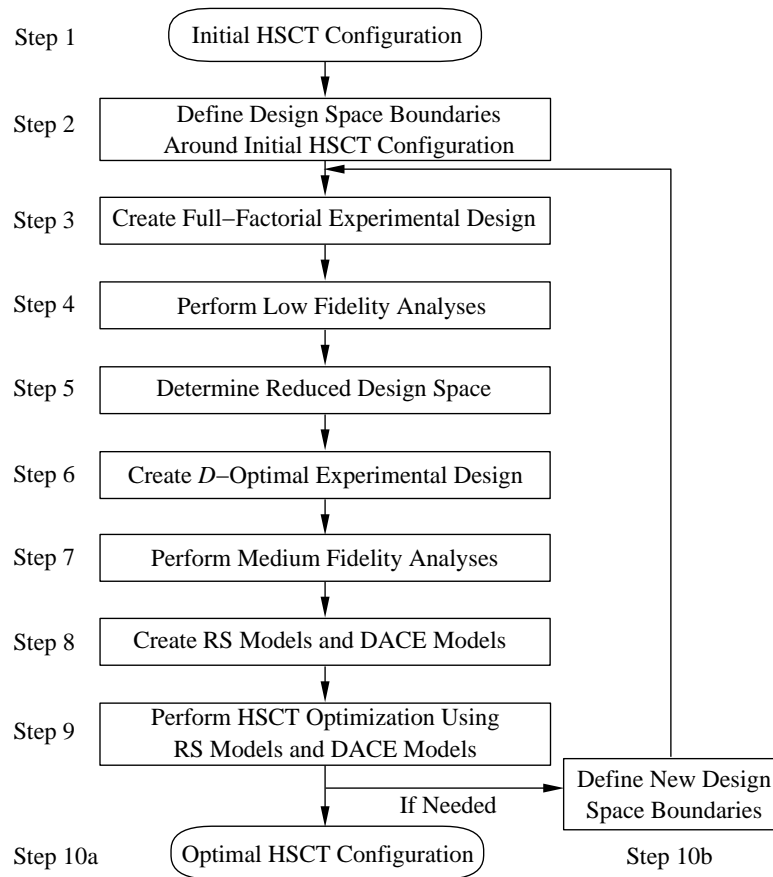


Figure 5.1: A flowchart showing the stages of the VCRSM method used in HSCT optimization.

# Chapter 6

## Results of HSCT Optimization Trials

### 6.1 Five Variable HSCT Optimization Problem

#### 6.1.1 Low Fidelity HSCT Analyses

A  $5^5$  full factorial design (i.e., five levels in each variable) based on DOE methods was constructed around an initial HSCT configuration. The 3125 ( $5^5$ ) full factorial HSCT configurations were analyzed using the inexpensive low fidelity analysis tools. These 3125 HSCT configurations were then screened to eliminate from consideration any grossly infeasible designs, i.e., those which exceeded any of the geometric constraints by more than five percent or any of the aerodynamic constraints by ten percent. After screening, 1860 HSCT configurations remained.

#### 6.1.2 Medium Fidelity HSCT Analyses

In the next step of the VCRSM process, JMP was used to create a  $D$ -optimal experimental design comprised of 50 HSCT configurations out of the 1860 remaining HSCT candidates. Note that at least 21 HSCT configuration analyses are needed to fit a quadratic polynomial in five variables. Choosing 50 HSCT configurations for analysis provides slightly more than double the minimum number of analyses. Note

that past research by Giunta et al. [2] indicated that at least 1.5 to 2.5 times the minimum number of analyses are required. The distribution of the 50  $D$ -optimal HSCT configurations within the design space yields a condition number of 18 for the matrix  $\mathbf{X}$ . This indicates that  $\mathbf{X}$  has full rank which ensures that  $(\mathbf{X}^T\mathbf{X})^{-1}$  exists and that the method of least squares may be performed to create RS models.

Once selected, the 50  $D$ -optimal HSCT configurations were evaluated using the medium fidelity analysis methods and RS models were constructed from the analysis data for  $C_{Dwave}$ ,  $C_{L\alpha}$  (M=2.4), and  $C_T/C_L^2$ . Adjusted  $R^2$  values were calculated as 0.9953, 0.9965, and 0.9908 for  $C_{Dwave}$ ,  $C_{L\alpha}$  (M=2.4), and  $C_T/C_L^2$ , respectively.

Of the 21 coefficients in the  $C_{Dwave}$  RS model, 15 had a  $t$ -statistic above 6.31. All linear terms were significant as were the quadratic terms for  $C_{root}$ ,  $t/c$  ratio, and  $\Lambda_{LEI}$ . The regression analysis procedure in JMP was then used to remove the less significant coefficients from the RS model and  $R^2_{adj}$  was recomputed. However,  $R^2_{adj}$  changed only from 0.9953 to 0.9956. Therefore, the complete (21 coefficient) polynomial RS model for  $C_{Dwave}$  was retained.

Regression analyses also were performed on the  $C_{L\alpha}$  (M=2.4) and  $C_T/C_L^2$  RS models. For the  $C_{L\alpha}$  (M=2.4) model, 11 of the coefficients had  $t$ -statistic values greater than 6.31. The retained coefficients included all linear terms except for  $W_{fuel}$ , and quadratic terms for  $C_{root}$  and  $\Lambda_{LEI}$ . Again, the recomputed  $R^2_{adj}$  value changed only slightly from 0.9965 to 0.9969 so the complete RS model for  $C_{L\alpha}$  (M=2.4) was used. For the  $C_T/C_L^2$  model only seven coefficients were retained which included the linear terms  $C_{root}$ ,  $C_{tip}$ , and  $\Lambda_{LEI}$ , and the quadratic terms for  $C_{root}$  and  $\Lambda_{LEI}$ . The recomputed  $R^2_{adj}$  value improved only slightly from 0.9908 to 0.9913 so the complete RS model also was retained for  $C_T/C_L^2$ .

Another quantitative measure of the quality of the RS model is to evaluate the residual error in the least squares fit which creates the RS model. The residual error is the difference between the calculated response from the numerical experiment data and the predicted response from the RS model. The residual error is a composite measure of error due to numerical noise and error due to bias. Bias error occurs when using a low order polynomial RS model when the data have higher order trends. For the five variable HSCT optimization problem, the average error, root-mean-square

(RMS) error, the unbiased RMS error ( $\text{RMS}_{\text{ub}}$ ), and maximum error are shown in Table 6.1 where the residual error calculations are performed for the data from the 50 HSCT analyses used to create the RS models. See Appendix B for a description of these error estimation terms. In Table 6.1 the errors are expressed as a percentage of the total variation in each RS model. The small magnitude of the errors further indicates that the quadratic polynomials are good models of the data trends.

The quality of the RS models is further shown by comparing the mission range of the initial HSCT computed using both the RS models and the original aerodynamic analysis methods. The range computed using the RS models is 5579.7 *naut.mi.* whereas the range computed using the original methods is 5509.7 *naut.mi.*, a difference of 1.27 percent.

### 6.1.3 HSCT Optimization Results

Figures 6.1 and 6.2 show optimization results for trial 1 of the five variable HSCT optimization problem obtained with and without using the VCRSM method. As shown in Figure 6.1 there were changes in the planform geometry between the initial and optimal HSCT configurations. Specifically, the initial and final variable values are listed in Table 6.2. These design changes result in a savings of approximately 7000 *lb* in the structural weight of the wing and about 10000 *lb* of fuel weight. These weight savings more than offset the decrease in aerodynamic efficiency in the optimal wing which had a maximum lift-to-drag ratio  $(L/D)_{\text{max}}$  of 9.76 compared to  $(L/D)_{\text{max}}$  of 9.85 for the initial wing.

In contrast, Figure 6.2 shows almost no change between the initial and optimal HSCT configurations when the RS models were not used. This occurred because numerical noise in the wave drag and drag-due-to-lift calculations led to numerical noise in the constraint calculations. This numerical noise created local minima in the design space which prevented the optimizer from locating the globally optimal wing design. The consequence of not using the RS models was a 14000 *lb* difference in TOGW between the globally optimal design and the locally optimal design.

For the optimal HSCT configuration, the range computed using the RS models is

5499.2 *naut.mi.* whereas the range computed using the original methods is 5443.2 *naut.mi.*, a difference of 1.03 percent. This discrepancy is a result of modeling error (i.e., inaccuracies) in the RS models. To account for the deficiency in range, a weight penalty is added to the TOGW which represents the additional fuel weight required to meet the 5500 *naut.mi.* constraint. The penalty factor is 90 *lb* of  $W_{fuel}$  per nautical mile of range deficiency. In this case, the range deficiency is approximately 56 *naut.mi.* which adds approximately 5000 *lb* to the TOGW of the HSCT. This is an increase of 0.80 percent in TOGW for the optimal HSCT configuration. Even with the weight penalty, the optimal HSCT configuration found using the RS models has a lower TOGW than the optimal HSCT configuration found using the original aerodynamic analysis methods.

Optimization trials 2 and 3 were performed with each trial starting from a different initial HSCT configuration than was used in trial 1. However, similar results were obtained for these two trials as were obtained in trial 1. With the RS models, the optimizer converged to the same globally optimal wing configuration. Without the RS models, the optimizer became trapped in different, artificial locally optimal wing configurations. The locally optimal HSCT configurations yielded significantly higher values of TOGW than the optimal HSCT configuration and had considerably poorer aerodynamic performance. Figures 6.3 and 6.4 show the optimal HSCT configurations obtained from trial 2, and Figures 6.5 and 6.6 show the optimal HSCT configurations obtained from trial 3.

#### 6.1.4 Computational Expense

The computational expense of performing a complete five variable HSCT optimization using the RS models is approximately 0.75 CPU minutes on a Silicon Graphics Indigo<sup>2</sup> workstation. In comparison, an optimization without using the RS models requires approximately 30 CPU minutes. As described above, the computational expense of the VCRSM method occurs in the creation of the RS models. For the five variable problem the cost of creating the RS models is approximately four CPU hours. Thus, optimization performed using the VCRSM method has a lower computational cost

than traditional optimization for this HSCT optimization problem if more than eight traditional optimizations are performed. In a typical study, the HSCT optimization process is performed many more than eight times. For example, the designer will perform the HSCT optimization starting with several different choices for the initial variable values. In addition, the designer often repeats the optimization to explore different objective criteria (e.g., maximize range, maximize payload). Therefore, the computational expense of creating the RS models is quickly recovered in a typical HSCT optimization study.

The use of a five-level full factorial experimental design in the initial exploration of the design space required  $5^5 = 3125$  HSCT configuration analyses. In comparison, a three-level full factorial experimental design would have required only  $3^5 = 243$  HSCT analyses. While the  $5^5$  experimental design was more computationally expensive to evaluate than a  $3^5$  experimental design, the five-level factorial permitted a more accurate definition of the reduced design space boundary than would have been possible with the three-level factorial. For example, consider a problem having only one variable,  $x$ , which has a design space of  $[0,1]$ . A five-level factorial divides the design space into quarters, whereas a three-level factorial divides the design space into halves. Now consider the inequality constraints  $x \leq 0.8$  and  $x \geq 0.0$  which are applied to create the reduced design space, i.e., the constraints eliminate the  $x = 1$  level. In this example, the reduced design space for the five-level factorial becomes  $[0,0.75]$ , whereas the reduced design space for the three-level factorial becomes  $[0,0.50]$ . Clearly, the reduced design space created from the five-level factorial provides better “coverage” of the feasible design space,  $[0,0.8]$ , than does the reduced design space created from three-level factorial.

## 6.2 Ten Variable HSCT Optimization Problem

### 6.2.1 Low Fidelity HSCT Analyses

A  $3^{10}$  full factorial design (i.e., three levels in each variable) was constructed around an initial HSCT configuration. The 59049 ( $3^{10}$ ) full factorial design points were analyzed

using the inexpensive, low fidelity analysis tools and the HSCT analysis results were screened to eliminate from consideration any grossly infeasible HSCT configurations. An HSCT configuration was eliminated if it violated any of the geometric constraints by more than five percent. After screening, 29163 HSCT configurations remained in the reduced design space; 51 percent of the original HSCT candidate configurations were eliminated.

### 6.2.2 Medium Fidelity HSCT Analyses

The next step in the VCRSM process was to select a set of 132  $D$ -optimal HSCT configurations from the remaining candidate set of 29163. For this problem, an in-house version of Mitchell's k-exchange algorithm was used to create the  $D$ -optimal experimental design. The distribution of the 132  $D$ -optimal HSCT configurations within the design space gives a condition number of 27 for the matrix  $\mathbf{X}$ . This indicates that  $\mathbf{X}$  has full rank and that  $(\mathbf{X}^T\mathbf{X})^{-1}$  exists.

The 132  $D$ -optimal HSCT configurations were evaluated using the medium fidelity analysis methods. Quadratic RS models, each having 65 terms, were created for  $C_{Dwave}$ ,  $C_{L\alpha}$  at  $M = 2.4$ ,  $C_T/C_L^2$ , and  $C_{L\alpha}$  at  $M = 0.2$ . The  $R^2_{adj}$  values for these RS models were 0.9603, 0.9976, 0.9982, and 0.9979, respectively. Regression analyses were performed on these RS models and coefficients that did not have a  $t$ -statistic greater than 6.31 were removed. The recomputed  $R^2_{adj}$  values for the RS models were 0.9667, 0.9978, 0.9986, and 0.9983, respectively. Note that  $C_{tip}^2$  could not be estimated since the reduced design space contained only two levels for that variable. However, the linear term for  $C_{tip}$  and the interaction terms involving  $C_{tip}$  could be estimated. Thus, the quadratic polynomial RS models have 65 terms instead of the 66 terms found in a complete quadratic model for ten variables.

The results of the regression analyses (Table 6.3) indicated that the  $C_{Dwave}$  RS model was most strongly influenced by 17 coefficients. These included all of the linear terms except for  $C_{tip}$ ,  $b_{nacelle}$ , and  $W_{fuel}$ , and included the quadratic terms for  $t/c$  and  $\Lambda_{LEI}$ . The 31 coefficients in the  $C_{L\alpha}$  ( $M = 2.4$ ) RS model included all of the linear terms except for  $t_{max}$ , and included the quadratic terms for  $C_{root}$ ,  $b_{half}$ ,  $R_{LE}$ ,  $t/c$ ,

and  $\Lambda_{LEI}$ . The 17 coefficients in the  $C_T/C_L^2$  RS model contained the linear terms for  $C_{root}$ ,  $C_{tip}$ ,  $b_{half}$ ,  $R_{LE}$ ,  $\Lambda_{LEI}$ , and  $\Lambda_{LEO}$ , along with the quadratic terms for  $C_{root}$ ,  $\Lambda_{LEI}$ , and  $\Lambda_{LEO}$ . The 18 coefficients in the  $C_{L\alpha}$  ( $M = 0.2$ ) RS model included the linear terms for  $C_{root}$ ,  $C_{tip}$ ,  $b_{half}$ ,  $t/c$ ,  $\Lambda_{LEI}$ , and  $\Lambda_{LEO}$ , along with the quadratic terms for  $C_{root}$ ,  $b_{half}$ , and  $\Lambda_{LEI}$ . In summary, all of the linear coefficients are important in at least one of the RS models and six of the quadratic coefficients ( $C_{root}$ ,  $t/c$ ,  $b_{half}$ ,  $R_{LE}$ ,  $\Lambda_{LEI}$ ,  $\Lambda_{LEO}$ ) are important in at least one of the RS models.

The average error, RMS error, unbiased RMS error, and maximum error for the polynomial RS models (65 terms) are shown in Table 6.4 where the residual error calculations are performed for the data from the 132 HSCT analyses used to create the RS models. For  $C_{Dwave}$  the residual error is the result of bias error since the magnitude of the numerical noise in the  $C_{Dwave}$  data is known to be small compared to typical values of  $C_{Dwave}$ . In contrast, for  $C_T/C_L^2$  the residual error is mostly due to numerical noise. It is known that the leading edge thrust estimation methods are extremely sensitive to changes in the HSCT wing planform. This sensitivity produces numerical noise in the  $C_T/C_L^2$  data where the magnitude of the noise may be up to 25 percent of typical  $C_T/C_L^2$  values as was quantified by Hutchison [54]. Although a predicted value for  $C_T/C_L^2$  from the RS model may contain significant error, the supersonic drag-due-to-lift calculation is dominated by the  $C_{L\alpha}$  ( $M = 2.4$ ) term (see Section 3.1). Thus, there is minimal effect on the overall HSCT wing design due to the residual error in the  $C_T/C_L^2$  RS model. Also, note that for the initial HSCT configuration the range computed using the RS models is 5503.7 *naut.mi.* and the range computed using the original methods is 5519.8 *naut.mi.*, a difference of 0.29 percent.

### 6.2.3 HSCT Optimization

Using the RS models, the HSCT configuration was optimized within the allowable limits of the ten variables. To verify that the numerical noise was removed from the optimization problem by using the RS models, two optimization trials were performed where each trial started from a distinctly different initial HSCT configuration. For



the first of these trials, Figure 6.7 shows the wing planform changes from the initial to the optimal HSCT configuration. The initial and optimal variables are listed in Table 6.5. The optimal HSCT has a smaller wing area and a lower aspect ratio than the initial HSCT which results in a wing structural weight savings of 23500 *lb*, a decrease of 18.3 percent. In addition to the structural improvements, the  $(L/D)_{max}$  of the HSCT increased from 9.37 to 9.54, an improvement of 1.8 percent. This contributes to a savings in fuel weight of approximately 18000 *lb*. The wings for the initial and optimal HSCT configurations are also shown in an isometric view in Figure 6.8 where the difference in the chordwise location of maximum thickness is more apparent. Similar weight savings were realized for the optimization trial 2 where the optimizer converged to an HSCT configuration having a wing shape virtually identical to the optimal HSCT configuration found in optimization trial 1. This shows that using the RS models to remove numerical noise allowed the optimizer to find the globally optimal HSCT configuration.

For the optimal HSCT configuration, the range computed using the RS models is 5500.2 *naut.mi.* whereas the range computed using the original methods is 5393.2 *naut.mi.* This difference of 1.98 percent is the result of modeling error in the RS models. Since the true range is below 5500 *naut.mi.*, the HSCT configuration would need to carry additional fuel to meet the range constraint. This would add about 9600 *lb* to the TOGW of the HSCT, an increase of 1.57 percent in TOGW for the optimal HSCT configuration.

To verify that the optimal HSCT configuration was acceptable, the optimization process was repeated with the original analysis methods (i.e., no RS models), but was started from the globally optimal HSCT configuration predicted using the RS models. As expected, the optimizer added about 9600 *lb* of fuel to compensate for the range deficiency, but the wing shape variables remained virtually unchanged. This demonstrated that the globally optimal design predicted using the RS models was reasonable, and that the errors in the RS models had little effect on producing a good HSCT configuration.

When the optimization trials 1 and 2 were attempted without using the RS models, the two HSCT configurations converged to substantially different optimal

configurations. This is consistent with the occurrence of multiple local optima and the convergence difficulties which originally motivated the use of RS modeling methods. One such locally optimal design is shown in Figure 6.9. The locally optimal HSCT design in Figure 6.9 is 17000 *lb* heavier than the globally optimal HSCT configuration. The optimization results, both with and without using the RS models, are shown in Table 6.6 for the two different HSCT optimization trials.

### 6.2.4 Post-Optimality Investigations

The optimal HSCT configuration listed in Table 6.5 has four variables ( $t_{max}$ ,  $R_{LE}$ ,  $b_{nacelle}$ , and  $W_{fuel}$ ) for which the side constraints are active. This means that the optimal HSCT configuration lies on the boundary of the reduced design space. While it is tempting to relax the four active side constraints in an effort to examine the sensitivity of the optimal HSCT configuration to the constraints, such a procedure increases the inaccuracy in the RS models which are valid only within the reduced design space. That is, the RS models become *extrapolation* models when used outside the reduced design space, rather than interpolation models (in a least squares sense) when used inside the reduced design space.

As a demonstration of this increased error which occurs when the RS models are used outside the reduced design space, an HSCT optimization was performed with the four side constraints relaxed. The new upper bound on  $t_{max}$  was 50.0 percent, and the new lower bounds on  $R_{LE}$ ,  $b_{nacelle}$ , and  $W_{fuel}$ , were 2.0 (nondimensional), 15.68 *ft*, and 297000 *lb*, respectively. In the resulting “optimal” HSCT configuration, the variable  $R_{LE}$  remained at a value of 2.1 and the variable  $b_{nacelle}$  moved from 16.7 *ft* to 16.1 *ft*. Both of these variables were not active after the re-optimization. However, both  $t_{max}$  and  $W_{fuel}$  were active after the re-optimization (i.e.,  $t_{max} = 50\%$  and  $W_{fuel} = 297000lb$ ). The increase in modeling error in the RS models is demonstrated by comparing the range predicted using the RS models (5500 *naut.mi.*) to the true range (5360 *naut.mi.*). This discrepancy is approximately 30 *naut.mi.* more than the range discrepancy for the optimal HSCT obtained with the original side constraints. Further relaxing the side constraints results in increased RS modeling errors as is

illustrated in an extreme case where the upper bound on  $t_{max}$  and the lower bound on  $W_{fuel}$  were eliminated. In this case the “optimal” HSCT had a value for  $t_{max}$  of 66 percent and a fuel weight of 272000 *lb*, both of which are unrealistic. In addition, the RS modeling error expressed through the range deficiency was 1200 *naut.mi*. These results clearly demonstrate the dangers inherent in allowing the optimizer to search outside the reduced design space.

### 6.2.5 Computational Expense

The computational expense of performing a complete ten variable HSCT optimization using the RS models is approximately 1.5 CPU minutes on a Silicon Graphics Indigo<sup>2</sup> workstation. In comparison, an optimization without using the RS models requires about one CPU hour. Thus, the computational expense of performing an HSCT optimization was significantly reduced. However, this comparison does not include the computational overhead involved in creating the RS models. This issue is discussed below.

The aggregate computational cost of creating the RS models is about 55 CPU hours when the HSCT analyses are performed serially. This includes approximately 50 CPU hours for the  $3^{10}$  low fidelity analyses which require 2-3 CPU seconds per analysis, and five CPU hours for the 132  $D$ -optimal medium fidelity analyses which require 1.5-2.0 CPU minutes per analysis. The majority of the computational expense occurs in the brute force approach of screening the initial design space on the  $3^{10}$  grid. Clearly this approach will be far to computationally expensive in higher dimensional problems. Alternate approaches include the use of partially balanced incomplete block designs as was done by Kaufman et al. [5] or the initial design space may be modeled using cubic polynomials. Either approach requires far fewer than the  $3^{10}$  analyses performed in this study.

Although the overhead cost of the VCRSM method appears substantial, this cost is recovered when one considers that optimizations without the RS models must be repeated numerous times, each with different initial values for the variables, until the general region of the optimal HSCT configuration is located. Further,

in aircraft design engineering, an aircraft configuration is almost never declared “optimal” without performing considerable trade-off studies. If the aircraft design is re-optimized during the trade-off studies, then the computational expense is virtually nil when the RS models are used. However, each re-optimization in the trade-off study requires about one CPU hour if RS models are not used.

### 6.2.6 VCRSM Versus Traditional Optimization Methods

If a traditional DOE and RS modeling approach had been followed in this investigation, RS models would have been calculated for the entire original design space rather than for only the reduced design space. For this application, the reduced design space has about 50 percent of the volume of the original design space. Thus, one would expect the RS approximations for the reduced design space to be more accurate than RS approximations for the entire design space.

To investigate this question of RS model accuracy, 132 HSCT configurations were selected based on the  $D$ -optimal criterion from the set of  $3^{10}$  HSCT configurations defined by the full factorial experimental design in the original design space. These 132 HSCT configurations were evaluated using the medium fidelity analysis methods and response surface models were constructed for the four aerodynamic quantities previously described. These RS models are valid over the entire original design space.

Next, 150 HSCT configurations were randomly selected from the 29163 HSCT configurations in the reduced design space and were evaluated using the medium fidelity analyses. From this *exact* data for the randomly selected HSCT configurations, the average, unbiased RMS, and maximum errors were calculated for each of the four aerodynamic quantities using the RS models obtained for the original design space and for the reduced design space (Table 6.7).

As shown in Table 6.7 the RS models for the reduced design space are more accurate than the RS models for the original design space for  $C_T/C_L^2$ ,  $C_{L\alpha}$  at  $M = 2.4$ , and  $C_{L\alpha}$  at  $M = 0.2$ . Somewhat surprisingly, the error calculations indicate that accuracy of the RS model for  $C_{D_{wave}}$  is slightly lower for the reduced design space as compared to the original design space. Further investigations showed that errors

in the RS model for  $C_{Dwave}$  were particularly sensitive to the location in the design space of the 150 randomly selected points. Thus, the errors calculated for  $C_{Dwave}$  using both sets of RS models are not significantly different. Although the improvements in accuracy shown here for three of the four RS models are not quite as substantial as those shown by Kaufman et al. [5], this investigation demonstrates that a significant improvement in accuracy is obtained by limiting the response surface construction to the reduced design space.

Table 6.1: Accuracy of the response surface models for the five variable HSCT optimization.

<b>Response Surface Model</b>	<b>Average Error</b>	<b>RMS Error</b>	<b>RMS<sub>ub</sub> Error</b>	<b>Maximum Error</b>
$C_{Dwave}$	0.82%	1.06%	1.39%	3.07%
$C_{Dwave}$ (drag counts)	0.10	0.13	0.17	0.35
$C_{L\alpha}$ at M=2.4	0.24%	0.31%	0.41%	0.88%
$C_T/C_L^2$	2.02%	2.46%	3.23%	5.78%

Table 6.2: Initial and optimal values in trial 1 of the five variable HSCT optimization problem.

<b>Variable</b>	<b>Initial</b>	<b>Optimal</b>	<b>Minimum</b>	<b>Maximum</b>
$C_{root}$	185.0 <i>ft</i>	165.2 <i>ft</i>	148.0 <i>ft</i>	222.0 <i>ft</i>
$C_{tip}$	10.0 <i>ft</i>	8.6 <i>ft</i>	8.0 <i>ft</i>	12.0 <i>ft</i>
$\Lambda_{LEI}$	75.0°	72.9°	68.3°	81.8°
<i>t/c ratio</i>	2.0 %	2.2 %	1.6%	2.4%
$W_{fuel}$	315000 <i>lb</i>	305550 <i>lb</i>	305550 <i>lb</i>	324450 <i>lb</i>
<i>TOGW</i>	640724 <i>lb</i>	622804 <i>lb</i>	637448 <i>lb</i>	655700 <i>lb</i>

Table 6.3: Important linear and quadratic terms in the response surface models for the ten variable HSCT optimization problem (interaction terms not shown).

	$C_{Dwave}$	$C_{L\alpha}$ (M=2.4)	$C_T/C_L^2$	$C_{L\alpha}$ (M=2.4)
<b>Linear Terms</b>				
$C_{root}$	X	X	X	X
$C_{tip}$		X	X	X
$b_{half}$	X	X	X	X
$\Lambda_{LEI}$	X	X	X	X
$\Lambda_{LEO}$	X	X	X	X
$t_{max}$	X			
$R_{LE}$	X	X	X	
$t/c$	X	X		X
$b_{nacelle}$		X		
$W_{fuel}$		X		
<b>Quadratic Terms</b>				
$C_{root}^2$		X	X	X
$C_{tip}^2$				
$b_{half}^2$		X		X
$\Lambda_{LEI}^2$	X	X	X	X
$\Lambda_{LEO}^2$			X	
$t_{max}^2$				
$R_{LE}^2$		X		
$(t/c)^2$	X	X		
$b_{nacelle}^2$				
$W_{fuel}^2$				



Table 6.4: Accuracy of the estimated responses in the ten variable reduced design space based on analyses of the 132  $D$ -optimal HSCT configurations.

<b>Response Surface Model</b>	<b>Average Error</b>	<b>RMS Error</b>	<b>RMS<sub>ub</sub> Error</b>	<b>Maximum Error</b>
$C_{Dwave}$	4.36%	5.78%	8.11%	18.31%
$C_{Dwave}$ (drag counts)	0.70	0.92	1.19	3.22
$C_{L\alpha}$ at $M = 2.4$	0.21%	0.27%	0.38%	0.91%
$C_T/C_L^2$	8.39%	13.19%	18.51%	47.45%
$C_{L\alpha}$ at $M = 0.2$	0.24%	0.30%	0.42%	0.90%

Table 6.5: Initial and optimal values in trial 1 of the ten variable HSCT optimization problem.

<b>Variable</b>	<b>Initial</b>	<b>Optimal</b>	<b>Minimum</b>	<b>Maximum</b>
$C_{root}$	150.0 <i>ft</i>	167.7 <i>ft</i>	139.2 <i>ft</i>	208.7 <i>ft</i>
$C_{tip}$	9.0 <i>ft</i>	8.1 <i>ft</i>	6.5 <i>ft</i>	9.7 <i>ft</i>
$b_{half}$	80.0 <i>ft</i>	73.4 <i>ft</i>	66.5 <i>ft</i>	81.3 <i>ft</i>
$\Lambda_{LE_I}$	68.0°	71.1°	65.4°	78.3°
$\Lambda_{LE_O}$	47.0°	40.3°	40.2°	48.2°
$t_{max}$	33.0 %	47.5 %	31.7%	47.5%
$R_{LE}$	2.5	2.1	2.1	3.2
$t/c$	2.1 %	2.3 %	1.8%	2.7%
$b_{nacelle}$	18.0 <i>ft</i>	16.7 <i>ft</i>	16.7 <i>ft</i>	25.1 <i>ft</i>
$W_{fuel}$	319000 <i>lb</i>	300700 <i>lb</i>	300700 <i>lb</i>	319300 <i>lb</i>
$TOGW$	655710 <i>lb</i>	612072 <i>lb</i>	610373 <i>lb</i>	657087 <i>lb</i>

Table 6.6: Optimization results for the two initial HSCT configurations, both with and with the response surface models.

<b>Results</b>	<b>TOGW</b>
Initial HSCT #1 *	655710 <i>lb</i>
Optimal HSCT <sup>a</sup>	612072 <i>lb</i>
Optimal HSCT <sup>b</sup>	629350 <i>lb</i>
Initial HSCT #2 *	626704 <i>lb</i>
Optimal HSCT <sup>a</sup>	612194 <i>lb</i>
Optimal HSCT <sup>b</sup>	626221 <i>lb</i>

<sup>a</sup> Optimization with the response surface models.

<sup>b</sup> Optimization without the response surface models.

\* Violates constraints.

Table 6.7: Response surface modeling error estimates based on analyses of 150 randomly selected HSCT configurations (10 variable HSCT optimization problem).

<b>Response Surface Model</b>	<b>Average Error</b>	<b>RMS<sub>ub</sub> Error</b>	<b>Maximum Error</b>
(a) $C_{Dwave}$	6.89%	8.33%	30.85%
(a) $C_{Dwave}$ (drag counts)	1.20	1.49	4.22
(b) $C_{Dwave}$	7.10%	9.40%	40.91%
(b) $C_{Dwave}$ (drag counts)	1.23	1.66	4.29
(a) $C_{L\alpha}$ at $M = 2.4$	0.66%	0.82%	2.18%
(b) $C_{L\alpha}$ at $M = 2.4$	0.40%	0.58%	1.82%
(a) $C_T/C_L^2$	28.70%	47.28%	157.57%
(b) $C_T/C_L^2$	12.03%	18.93%	73.59%
(a) $C_{L\alpha}$ at $M = 0.2$	1.42%	1.81%	4.59%
(b) $C_{L\alpha}$ at $M = 0.2$	0.45%	0.56%	1.49%

<sup>a</sup> Original design space.

<sup>b</sup> Reduced design space.

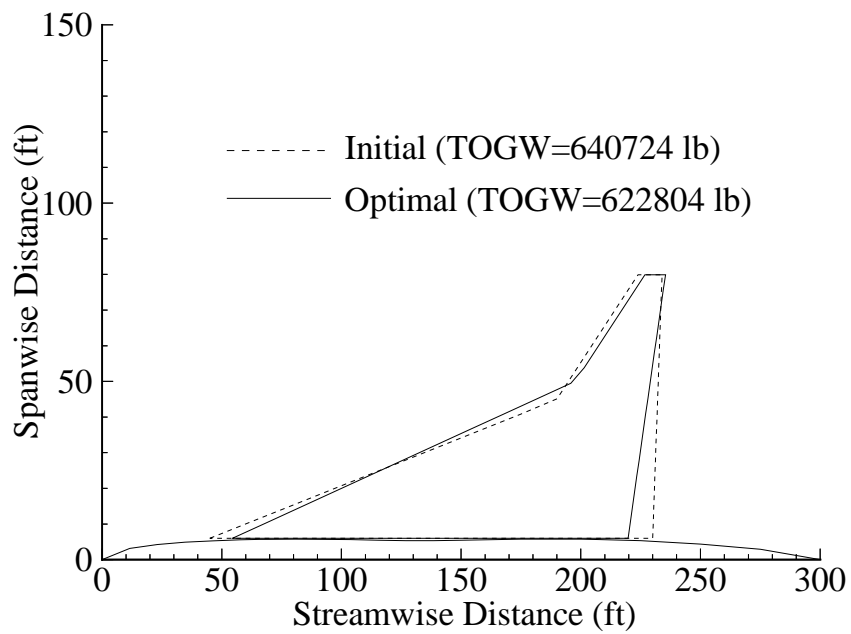


Figure 6.1: Five variable initial and optimal HSCT configurations obtained using RS models (trial 1).

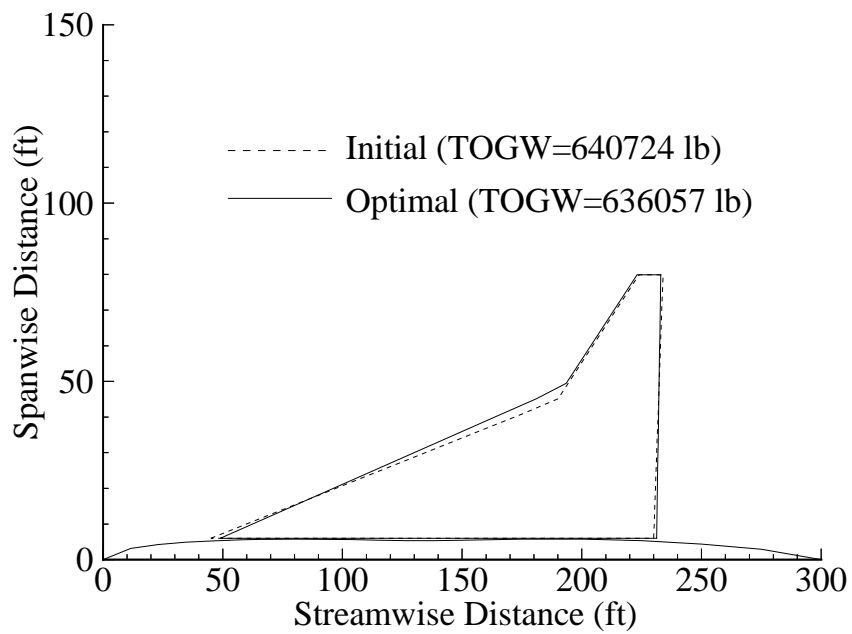


Figure 6.2: Five variable initial and optimal HSCT configurations obtained without using RS models (trial 1).

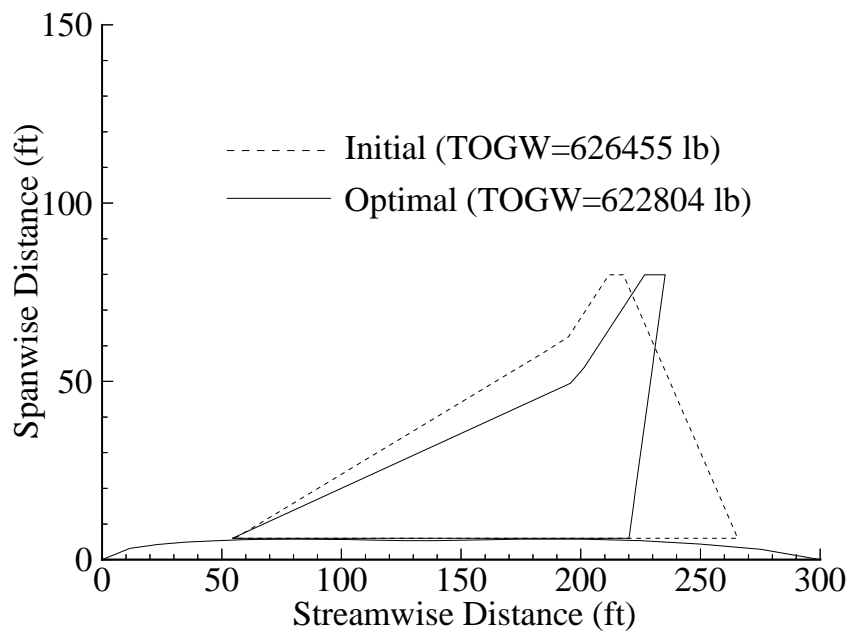


Figure 6.3: Five variable initial and optimal HSCT configurations obtained using RS models (trial 2).

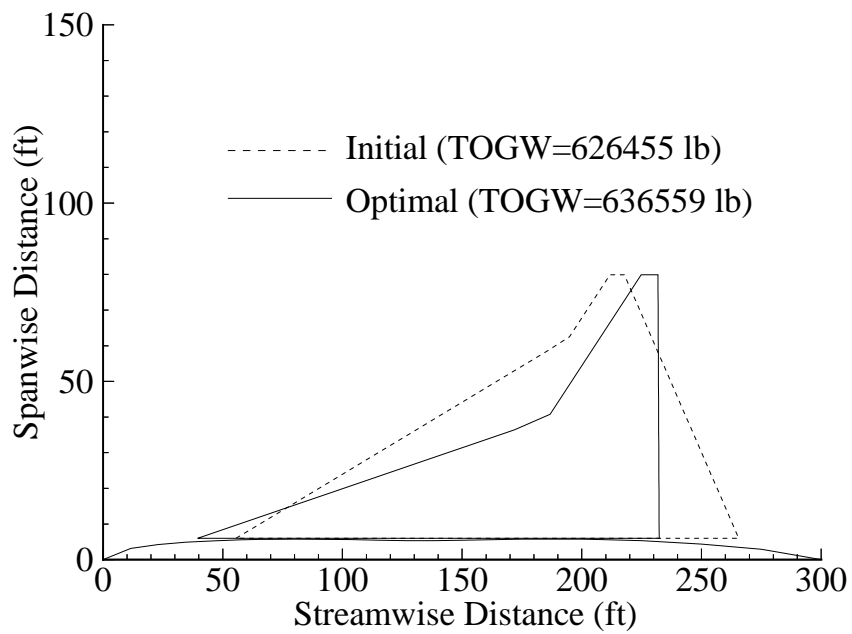


Figure 6.4: Five variable initial and optimal designs obtained without using RS models (trial 2).



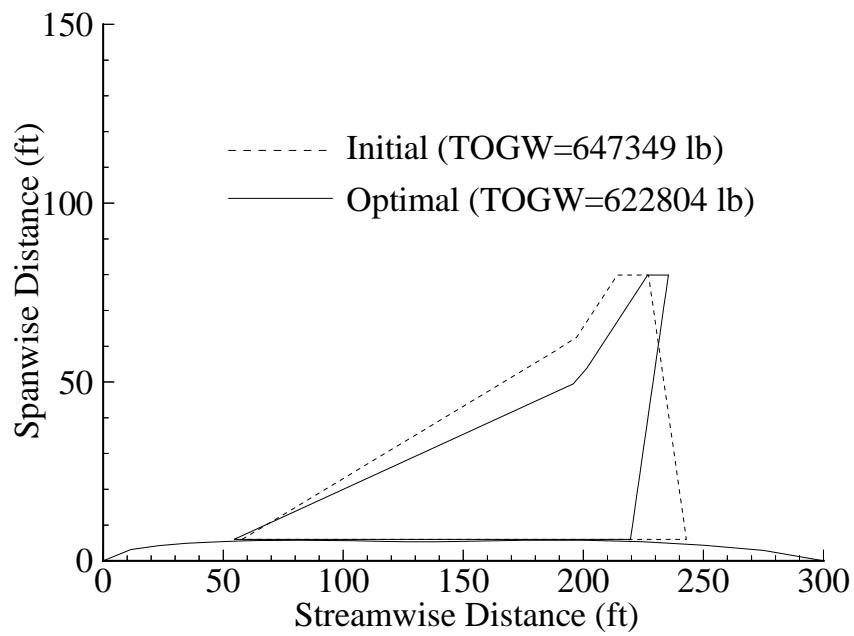


Figure 6.5: Five variable initial and optimal HSCT configurations obtained using RS models (trial 3).

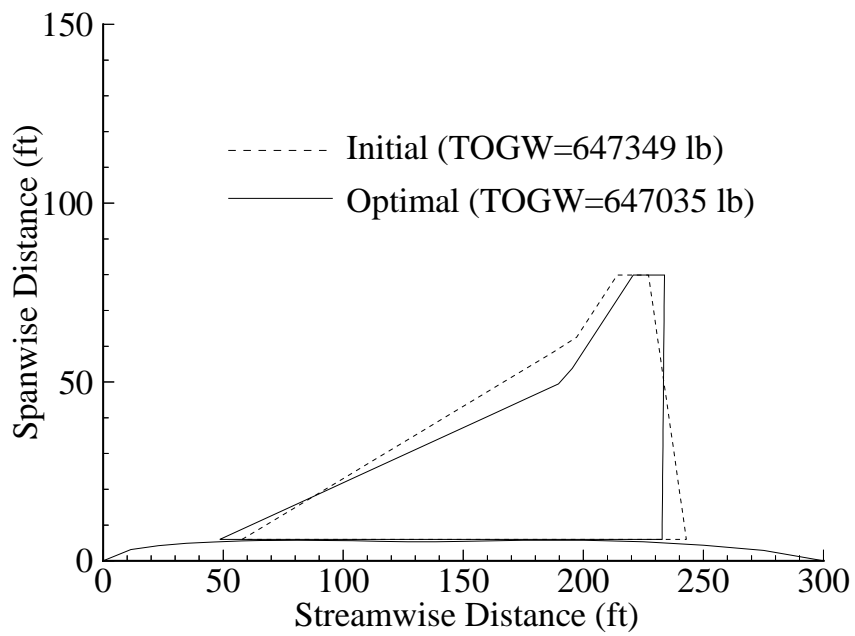


Figure 6.6: Five variable initial and optimal HSCT configurations obtained without using RS models (trial 3).

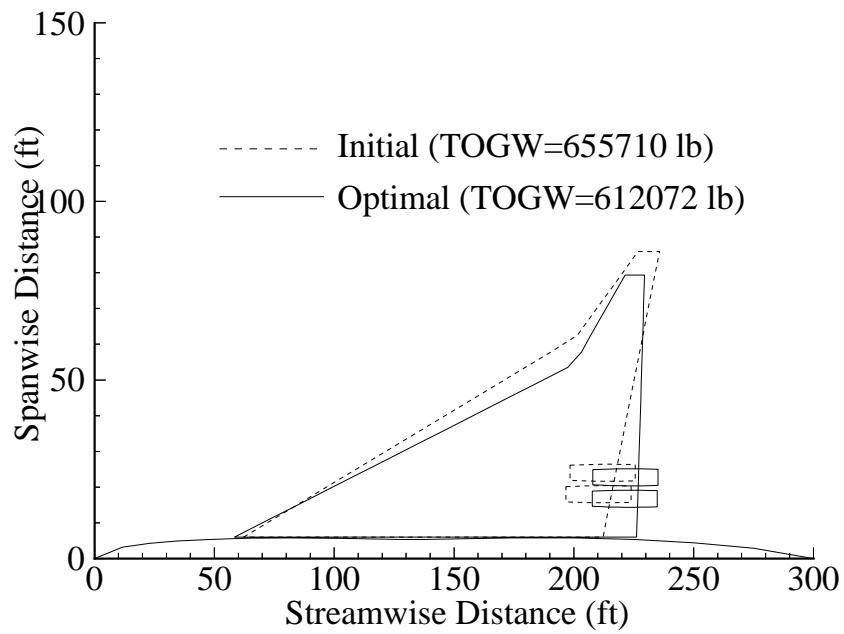


Figure 6.7: Ten variable initial and optimal HSCT configurations obtained using RS models (trial 1).

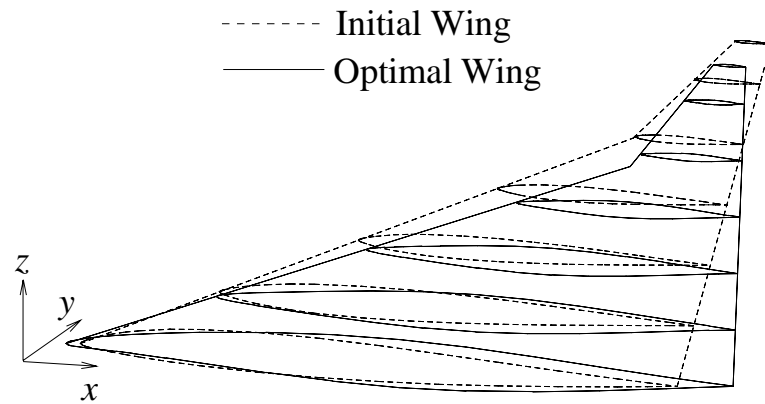


Figure 6.8: Ten variable initial and optimal HSCT wings (with RS models) showing airfoil differences (trial 1).

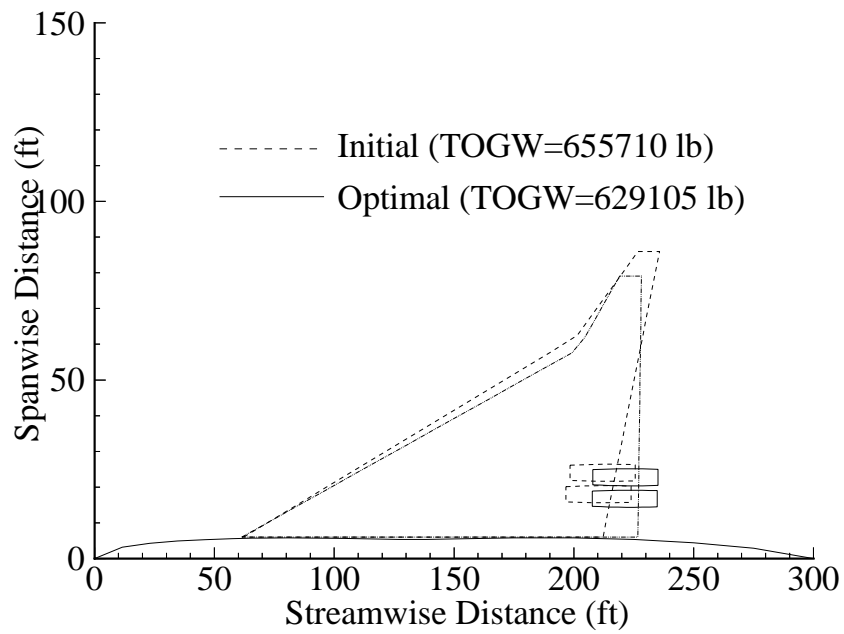


Figure 6.9: Ten variable initial and optimal HSCT configurations obtained without using RS models (trial 1).

# Chapter 7

## Test Problems Using DACE Modeling Methods

The objective of performing the test problems was to gain an understanding of the strengths and weaknesses of DACE modeling as compared to polynomial RS modeling. For these efforts two test problems were formulated where the first test problem was expected to be biased in favor of the DACE modeling method and the second test problem was expected to be biased in favor of the polynomial RS modeling method. A critical element of this comparison is the investigation of how the accuracy of the DACE models and RS models is affected as the number of dimensions,  $n_v$ , increases. To investigate this aspect of modeling accuracy, test problems involving  $n_v = 1$ ,  $n_v = 5$ , and  $n_v = 10$  were examined.

### 7.1 Creation of Test Problems

#### 7.1.1 Selection of Functions Employed in the Test Problems

For this investigation a simple test function was chosen so that it could be exhaustively examined with minimal computational expense. This test function has the form

$$y(\mathbf{x}) = \sum_{i=1}^{n_v} \left[ \frac{3}{10} + \sin\left(\frac{16}{15}x_i - \epsilon\right) + \sin^2\left(\frac{16}{15}x_i - \epsilon\right) \right], \quad (7.1)$$

where the term  $\epsilon$  acts as a shifting mechanism to make the response,  $y(\mathbf{x})$ , appear more or less quadratic on the range  $[-1, 1]^{n_v}$ . The values used for  $\epsilon$  are described below. Since there is no numerical noise inherent in Equation 7.1 it is henceforth referred to as the *smooth test function*.

To simulate the effects of numerical noise, a high frequency, low amplitude sine wave function was added to Equation 7.1. This *noisy test function* has the form

$$y(\mathbf{x}) = \sum_{i=1}^{n_v} \left[ \frac{3}{10} + \sin\left(\frac{16}{15}x_i - \epsilon\right) + \sin^2\left(\frac{16}{15}x_i - \epsilon\right) + \frac{2}{100} \sin\left(40\left(\frac{16}{15}x_i - \epsilon\right)\right) \right], \quad (7.2)$$

where the term on the far right of Equation 7.2 is the high frequency, low amplitude component.

The first test function (Case 1) was created for  $\epsilon = 1.0$  and a plot of the noisy version of this function for  $n_v = 1$  is shown in Figure 7.1. Both the smooth and noisy variants of the Case 1 test functions are shown in Figure 7.2 for  $n_v = 2$ . This function appears quasi-sinusoidal on  $[-1, 1]$ .

The Case 2 test function was created using  $\epsilon = 0.7$  and has a quasi-quadratic trend on  $[-1, 1]$ . The noisy Case 2 test function is shown in Figure 7.3 for  $n_v = 1$  and both the smooth and noisy Case 2 test functions are shown in figure 7.4 for  $n_v = 2$ .

### 7.1.2 Evaluation of Modeling Accuracy

For both Cases 1 and 2, DACE and RS models (denoted as  $\hat{y}(\mathbf{x})$ ) were constructed based on  $n_s$  evaluations (response values) of the noisy test function. These models were then used to estimate the unknown response values of the smooth test function at  $n_e$  locations, where  $n_e \gg n_s$ . These predicted smooth function response values are denoted as  $\hat{y}_{n_e}$ . To evaluate the accuracy of the DACE and RS models, the actual response values of the smooth test function are also calculated for the  $n_e$  locations. These actual smooth function response values are denoted as  $y_{n_e}$ . The discrepancy between  $y_{n_e}$  and  $\hat{y}_{n_e}$  is known as the modeling error. The total modeling error in the DACE and RS models is characterized using five error metrics. These are the mean error,  $\bar{\delta}$ , the median error,  $\delta_{median}$ , the standard deviation,  $\sigma_\delta$ , the maximum error,  $\delta_{max}$ , and the unbiased RMS error  $\text{RMS}_{ub}$ . These statistical quantities are defined

in Appendix B. Values for  $n_s$  and  $n_e$ , as well as the methods used to select the  $n_e$  locations, for the one, five, and ten variable test problems are described below. Note that the definition of the modeling error is different from the *residual error* which is the discrepancy between a polynomial model and the data points in an overdetermined least squares problem. There is no residual error in DACE modeling since the DACE method exactly interpolates the  $n_s$  response values.

## 7.2 One Variable Test Problem

### 7.2.1 Data Selection

In the one variable test problem the test functions were sampled at three locations -0.5, -0.3, and 0.7 on  $[-1, 1]$ . From the response values at these three sites a DACE model and a quadratic polynomial RS model were created. To test the accuracy of the DACE and RS models, the smooth test function was sampled at 201 equally spaced points along  $[-1, 1]$ . Figures 7.5 and 7.6 show the DACE and RS models used in Cases 1 and 2, respectively.

For the Case 1 test problem the DACE model had a correlation parameter of  $\theta = 7.540$  and  $\hat{\beta} = 0.2127$ , while for the Case 2 test problem these values were  $\theta = 28.394$  and  $\hat{\beta} = 0.2777$ . The quadratic response surface polynomial models for both the Case 1 and Case 2 test functions were created using the `Fit[.]` function in *Mathematica* [75, pages 859–861].

In addition to a DACE model and a quadratic polynomial RS model a third model was examined where

$$\hat{y}(\mathbf{x}) = \bar{y}, \quad (7.3)$$

where  $\bar{y}$  is the mean of the  $n_s = 3$  observed response values in  $\mathbf{y}$ . For Case 1,  $\bar{y} = 0.2342$  and for Case 2,  $\bar{y} = 0.2680$ .

### 7.2.2 Results

The modeling errors for DACE, polynomial RS, and mean values models were calculated using the 201 values of the smooth test function. The results for Cases 1 and 2 are



shown in Table 7.1. For Case 1 the DACE model is more accurate than both the polynomial RS model and the mean value model, whereas for Case 2 the polynomial RS model is more accurate than the other two models. This corresponds to what is shown in Figures 7.5 and 7.6.

## 7.3 Five Variable Test Problem

### 7.3.1 Data Selection

For the five variable test problem  $n_s = 50$  and  $n_e = 3125$ . The 50 sample sites were borrowed from the five variable HSCT optimization problem (see Chapter 6) and were located within the five dimensional design space defined by  $[-1, 1]^5$ . The 3125 test sites were created by discretizing the design space into a  $5 \times 5 \times 5 \times 5 \times 5$  mesh where  $5^5 = 3125$ .

### 7.3.2 Model Definition

For the Case 1 test problem the DACE model had a correlation parameter of  $\theta = 0.45$  and  $\hat{\beta} = 1.3516$ , while for the Case 2 test problem these values were  $\theta = 0.08$  and  $\hat{\beta} = 5.9593$ . As above, the quadratic response surface polynomial models for the Case 1 and Case 2 test functions were created using the **Fit**[.] function in *Mathematica*.

In addition to a DACE model and a quadratic polynomial RS model, two other approximation methods were examined. The third model is a combined RS/DACE model of the form

$$\hat{y}(\mathbf{x}) = f(\mathbf{x}) + \beta_{residual} + Z(residual), \quad (7.4)$$

where  $f(\mathbf{x})$  is the quadratic polynomial RS model found using *Mathematica* and  $\beta_{residual} + Z(residual)$  is a DACE model applied to the residual error existing in the least squares surface fit for  $f(\mathbf{x})$ . For the Case 1 RS/DACE model the optimal correlation parameter was  $\theta = 30.0$ , and  $\hat{\beta} = -5.25 \cdot 10^{-7}$ . In the RS/DACE model for Case 2 these parameters were  $\theta = 30.0$ , and  $\hat{\beta} = -5.76 \cdot 10^{-7}$ . The fourth model examined is the mean value model (Equation 7.3) where for Case 1,  $\bar{y} = 1.2512$  and for Case 2,  $\bar{y} = 2.1418$ .

### 7.3.3 Results

The modeling errors for these four approximation models were calculated for Case 1 and Case 2 test functions and are listed in Table 7.2. In the Case 1 results the polynomial RS model and the combined RS/DACE model have nearly identical values for the modeling errors. For the DACE method the modeling error is not as low as for the polynomial-based models, but it is lower than for the constant term model. Similar trends are exhibited in the Case 2 results where the modeling errors for the polynomial RS model and the RS/DACE model are nearly identical, are the modeling errors for the DACE model are only marginally worse. In Case 2 however, the constant term model has considerable higher modeling errors than the other three models.

## 7.4 Ten Variable Test Problem

### 7.4.1 Data Selection

For the ten variable test problem  $n_s = 132$  and  $n_e = 10000$ . The 132 sample sites were obtained from the ten variable HSCT optimization problem (see Chapter 6) and were located within the ten dimensional design space defined by  $[-1, 1]^{10}$ . The 10000 test sites were chosen randomly from within the design space.

### 7.4.2 Model Definition

The four approximation models used in the ten variable test problem were the same as those used in the five variable test problem. For the Case 1 test problem the DACE model had a correlation parameter of  $\theta = 0.50$  and  $\hat{\beta} = 2.6455$ , while for the Case 2 test problem these values were  $\theta = 0.50$  and  $\hat{\beta} = 4.7987$ . As before, the quadratic response surface polynomial models for both the Case 1 and Case 2 test functions were created using the `Fit[]` function in *Mathematica*. For the Case 1 RS/DACE model the optimal correlation parameter was  $\theta = 0.10$ , and  $\hat{\beta} = 4.69 \cdot 10^{-15}$ , and for Case 2, these parameters were  $\theta = 0.10$ , and  $\hat{\beta} = 7.31 \cdot 10^{-15}$ . For the fourth model, Case 1 was  $\bar{y} = 2.6214$  and Case 2 was  $\bar{y} = 4.7190$ .

### 7.4.3 Results

The results for the Case 1 and Case 2 test problems are listed in Table 7.3. In Case 1, the polynomial RS model and the RS/DACE model exhibit nearly identical modeling errors and provide the best approximations to the test function. The modeling error for the DACE model is somewhat worse than for the polynomial-based models, and the modeling error for the constant term model is the largest.

While the results for the Case 1 test problem are similar for the five and ten variable versions of the test problem, this is not true for the Case 2 test problem. Here, the polynomial RS model is a far more accurate modeling technique than the other three models and in particular has a lower mean modeling error by a factor of four when compared to the accuracy of the DACE model. Note that once again the DACE model is only slightly more accurate than the constant term model.

## 7.5 Summary of Test Problem Results

Note that some caution must be exercised in interpreting these results as the modeling accuracy data and observations are applicable only to the Case 1 and Case 2 test functions considered here. As may be expected, if different test functions had been investigated, the results may have been different. In fact it is quite easy to create a test function for which the constant term model is the most accurate modeling method, as the author discovered in some initial DACE modeling work.

The results from the one variable test problem showed the expected trends, i.e., where the DACE model was more accurate for the Case 1 test function and the polynomial RS model was more accurate for the Case 2 test function. However, the five and ten variable versions of the Case 1 test problem did not yield the expected results. For the quasi-sinusoidal test function in Case 1 the polynomial RS model was slightly more accurate than the DACE model, although both the polynomial RS model and the DACE model were only marginally more accurate than the constant term model. Thus, the sinusoidal features of the test problem posed difficulties for both the polynomial RS and DACE models.

For the one, five, and ten variable versions of the Case 2 test problem, it is clear that the polynomial RS model provides the highest modeling accuracy of the approximation methods considered in this study. These results were expected since the test function is quasi-quadratic. However, the most startling results are shown in the modeling error data for the DACE model as compared to the constant term model for the ten variable test problem. Here, the DACE model is only slightly more accurate than the constant term model.

From the results obtained for the five and ten variable test problems it appears that the modeling accuracy provided by DACE methods is not quite as high as that provided by polynomial RS modeling. However, the use of DACE modeling becomes attractive in applications where the number of  $n_s$  response evaluations needed to construct a quadratic polynomial in  $n_v$  variables is computationally expensive. Such an application arises when employing the variable-complexity modeling paradigm for HSCT optimization. The one and five variable test problems were modified to employ such a variable-complexity modeling paradigm. The results obtained from this investigation are described below.

## 7.6 Variable-Complexity Modeling Test Problems

To simulate the variable-complexity modeling process the methods of Osio and Amon [67] were applied to the one and five variable test problems. Here, two stages of modeling were employed where the first stage ( $k = 1$ ) denotes the use of  $n_s^{(1)}$  samples of the noisy test function and the second stage ( $k = 2$ ) denotes the use of  $n_s^{(2)}$  samples of the smooth test function. The  $n_s^{(1)}$  samples of the noisy test function represent data obtained from a computer simulation having low computational expense and low accuracy, whereas the  $n_s^{(2)}$  samples of the smooth test function represent data from a computer simulation involving high computational expense and high accuracy.

### 7.6.1 One Variable Test Problem

For the one variable test problem the  $k = 1$  stage uses  $n_s^{(1)} = 3$  samples of the noisy test function at -0.50, -0.30, and 0.70 on  $[-1, 1]$ . These locations were chosen arbitrarily and were the same for both Cases 1 and 2. The sample locations for the  $k = 2$  stage were different for Case 1 and Case 2. In Case 1,  $n_s^{(2)} = 2$  samples sites of the smooth test function were at 0.69, and 0.0, where 0.69 was the location of the minimum of the DACE model on  $[-1, 1]$ , and 0.0 was chosen arbitrarily. In Case 2, the two samples sites of the smooth test function were -0.29, and 0.0 where -0.29 was the location of the minimum of the DACE model on  $[-1, 1]$ , and 0.0 was chosen arbitrarily.

The  $k = 1$  stage DACE models for Cases 1 and 2 have the same model parameters as those given above in Section 7.2. For Case 1 the  $k = 2$  DACE model employed  $\hat{\beta}^{(1)} = 0.2127$  (i.e., the  $k = 1$  stage  $\hat{\beta}$  value) and the correlation parameter is  $\theta^{(2)} = 0.1331$ . The Case 2 DACE model for  $k = 2$  uses  $\hat{\beta}^{(1)} = 0.2777$  and the correlation parameter is  $\theta^{(2)} = 1.9574$ .

The  $k = 1$  and  $k = 2$  stage DACE models for the one variable test problem are shown in Figures 7.7 and 7.8 for Cases 1 and 2, respectively. The modeling errors calculated at 201 equally spaced intervals on  $[-1, 1]$ , are listed in Table 7.4. Surprisingly, for Case 1 the  $k = 2$  stage DACE model is less accurate than the  $k = 1$  stage DACE model. As shown in Figure 7.7, the  $k = 2$  stage DACE model is shifted away from the two of the response values from the  $k = 1$  stage at -0.5, and -0.3. Because of this shift the  $k = 2$  stage DACE model is less accurate than the  $k = 1$  stage model. However, for Case 2 the  $k = 2$  stage DACE model is more accurate than the  $k = 1$  stage model as illustrated in Figure 7.8. While the  $k = 2$  stage DACE model becomes more accurate than the  $k = 1$  stage DACE model on  $[-1, 0.5]$ , the  $k = 2$  model becomes less accurate on  $[0.5, 1]$ .

The results of applying multi-fidelity modeling techniques to the one variable test problem demonstrate the sensitivity of the DACE models to the location of the sample sites. Also, it appears that higher stage DACE models (i.e.,  $k = 2$ ) are not uniformly more accurate than the lower stage DACE models (i.e.,  $k = 1$ ). Such trends are encountered in the five variable multi-fidelity DACE modeling example as well.

### 7.6.2 Data and Model Selection - Five Variable Test Problem

In the five variable test problem the  $k = 1$  stage uses of  $n_s^{(1)} = 50$  samples of the noisy test function and the  $k = 2$  stage employs  $n_s^{(2)} = 3$  samples of the smooth test function. At the  $k = 1$  stage, the 50 sample sites are the same as those described above in Section 7.3. For the  $k = 2$  stage the three sample sites were selected in the region of the optimum of the Case 1 and Case 2 test functions.

The DACE model for the  $k = 1$  stage is the same for the DACE model described above in Section 7.3, i.e., for Case 1,  $\theta = 0.45$  and  $\hat{\beta} = 1.3516$ , and for Case 2,  $\theta = 0.08$  and  $\hat{\beta} = 5.9593$ . The DACE model for the three samples in the  $k = 2$  stage has Case 1 values of  $\theta = 0.046$  and  $\hat{\beta} = 1.3516$ , and Case 2 values of  $\theta = 0.030$  and  $\hat{\beta} = 5.9593$ . Note that the  $k = 2$  stage DACE models exactly interpolate the three response values, but do not necessarily exactly interpolate the 50 response values used to create the  $k = 1$  stage DACE model.

### 7.6.3 Results - Five Variable Test Problem

The accuracy of the  $k = 1$  stage models for Cases 1 and 2 are the same as those in the five variable test problem. For ease of comparison, Case 1 and Case 2 modeling errors have been included in Table 7.5. The modeling errors for the  $k = 2$  stage models were calculated at the same  $n_e = 3125$  test sites, and the errors are also shown in Table 7.5.

For Case 1 the  $k = 2$  stage DACE model is slightly less accurate than the  $k = 1$  stage model over the 3125 test sites in  $[-1, 1]^5$ . This is expected since the  $k = 1$  stage DACE model is based on response values at sample sites that cover the design space while the  $k = 2$  stage DACE model is based on response values sampled at three closely-spaced sites in the design space. However, in the vicinity of the three sample locations, the  $k = 2$  stage DACE model is more accurate than the  $k = 1$  stage DACE model. Similar results were obtained for the Case 2 test function. However, the difference in accuracy between the  $k = 1$  stage and  $k = 2$  stage DACE models is more pronounced than in the Case 1 results.

From the data presented in Table 7.5 it appears that the use of three samples of

the smooth function in the  $k = 2$  stage actually decreases the accuracy of the DACE models over the entire design space. This occurs even though the  $k = 2$  stage DACE models are locally more accurate than the  $k = 1$  stage models in the vicinity of the three sample sites. Thus, if the  $k = 2$  stage models are to be used for predictive purposes, the size of the design space over which they are applicable must be restricted. The size of this region of influence will most likely be problem dependent.

Table 7.1: Modeling errors for Cases 1 and 2 for the one variable test problem.

Model	Mean Error	Median Error	Std. Dev.	RMS <sub>ub</sub> Error	Max. Error
<b>Case 1</b>					
$\hat{y}(\mathbf{x}) = \beta + Z(\mathbf{x})$	0.051	0.027	0.045	0.068	0.203
$\hat{y}(\mathbf{x}) = \hat{\mathbf{c}}^T \bar{\mathbf{x}}^{(\mathbf{p})}$	0.076	0.057	0.076	0.107	0.328
$\hat{y}(\mathbf{x}) = \bar{y}$	0.081	0.063	0.058	0.100	0.184
<b>Case 2</b>					
$\hat{y}(\mathbf{x}) = \beta + Z(\mathbf{x})$	0.115	0.093	0.112	0.160	0.502
$\hat{y}(\mathbf{x}) = \hat{\mathbf{c}}^T \bar{\mathbf{x}}^{(\mathbf{p})}$	0.080	0.065	0.081	0.114	0.393
$\hat{y}(\mathbf{x}) = \bar{y}$	0.130	0.117	0.110	0.170	0.519



Table 7.2: Modeling errors for Cases 1 and 2 for the five variable test problem.

<b>Model</b>	<b>Mean Error</b>	<b>Median Error</b>	<b>Std. Dev.</b>	<b>RMS<sub>ub</sub> Error</b>	<b>Max. Error</b>
<b>Case 1</b>					
$\hat{y}(\mathbf{x}) = \beta + Z(\mathbf{x})$	0.211	0.180	0.159	0.264	0.992
$\hat{y}(\mathbf{x}) = \hat{\mathbf{c}}^T \bar{\mathbf{x}}^{(\mathbf{p})}$	0.202	0.171	0.154	0.254	0.963
$\hat{y}(\mathbf{x}) = f(\mathbf{x}) + \beta_{res.} + Z(res.)$	0.203	0.181	0.153	0.254	0.963
$\hat{y}(\mathbf{x}) = \bar{y}$	0.241	0.210	0.175	0.298	0.989
<b>Case 2</b>					
$\hat{y}(\mathbf{x}) = \beta + Z(\mathbf{x})$	0.225	0.190	0.171	0.282	0.945
$\hat{y}(\mathbf{x}) = \hat{\mathbf{c}}^T \bar{\mathbf{x}}^{(\mathbf{p})}$	0.210	0.178	0.158	0.263	0.944
$\hat{y}(\mathbf{x}) = f(\mathbf{x}) + \beta_{res.} + Z(res.)$	0.211	0.179	0.158	0.264	0.944
$\hat{y}(\mathbf{x}) = \bar{y}$	0.696	0.651	0.425	0.815	1.793

Table 7.3: Modeling errors for Cases 1 and 2 for the ten variable test problem.

<b>Model</b>	<b>Mean Error</b>	<b>Median Error</b>	<b>Std. Dev.</b>	<b>RMS<sub>ub</sub> Error</b>	<b>Max. Error</b>
<b>Case 1</b>					
$\hat{y}(\mathbf{x}) = \beta + Z(\mathbf{x})$	0.651	0.636	0.362	0.745	2.010
$\hat{y}(\mathbf{x}) = \hat{\mathbf{c}}^T \bar{\mathbf{x}}^{(\mathbf{p})}$	0.524	0.477	0.355	0.633	1.964
$\hat{y}(\mathbf{x}) = f(\mathbf{x}) + \beta_{res.} + Z(res.)$	0.524	0.479	0.348	0.629	1.823
$\hat{y}(\mathbf{x}) = \bar{y}$	0.698	0.696	0.283	0.753	1.801
<b>Case 2</b>					
$\hat{y}(\mathbf{x}) = \beta + Z(\mathbf{x})$	2.090	1.920	0.531	2.157	4.071
$\hat{y}(\mathbf{x}) = \hat{\mathbf{c}}^T \bar{\mathbf{x}}^{(\mathbf{p})}$	0.380	0.326	0.218	0.473	1.646
$\hat{y}(\mathbf{x}) = f(\mathbf{x}) + \beta_{res.} + Z(res.)$	0.544	0.475	0.416	0.693	2.566
$\hat{y}(\mathbf{x}) = \bar{y}$	2.344	2.385	0.528	2.403	3.914

Table 7.4: A comparison of the modeling errors between the  $k = 1$  stage DACE model (three samples of the noisy test function) and the  $k = 2$  stage DACE model (two samples of the smooth test function) for Cases 1 and 2 of the one variable test problem.

<b>Model</b>	<b>Mean Error</b>	<b>Median Error</b>	<b>Std. Dev.</b>	<b>RMS<sub>ub</sub> Error</b>	<b>Max. Error</b>
<b>Case 1</b>					
<b>Stage k=1</b>					
$\hat{y}(\mathbf{x}) = \beta + Z(\mathbf{x})$	0.051	0.027	0.045	0.068	0.203
<b>Stage k=2</b>					
$\hat{y}(\mathbf{x}) = \beta + Z(\mathbf{x})$	0.076	0.075	0.046	0.089	0.253
<b>Case 2</b>					
<b>Stage k=1</b>					
$\hat{y}(\mathbf{x}) = \beta + Z(\mathbf{x})$	0.115	0.093	0.112	0.160	0.502
<b>Stage k=2</b>					
$\hat{y}(\mathbf{x}) = \beta + Z(\mathbf{x})$	0.076	0.010	0.133	0.153	0.558

Table 7.5: A comparison of the modeling errors between the  $k = 1$  stage DACE model (50 samples of the noisy test function) and the  $k = 2$  stage DACE model (three samples of the smooth test function) for Cases 1 and 2 of the five variable test problem.

<b>Model</b>	<b>Mean Error</b>	<b>Median Error</b>	<b>Std. Dev.</b>	<b>RMS<sub>ub</sub> Error</b>	<b>Max. Error</b>
<b>Case 1</b>					
<b>Stage k=1</b>					
$\hat{y}(\mathbf{x}) = \beta + Z(\mathbf{x})$	0.211	0.180	0.159	0.264	0.992
<b>Stage k=2</b>					
$\hat{y}(\mathbf{x}) = \beta + Z(\mathbf{x})$	0.231	0.201	0.171	0.288	0.925
<b>Case 2</b>					
<b>Stage k=1</b>					
$\hat{y}(\mathbf{x}) = \beta + Z(\mathbf{x})$	0.225	0.190	0.171	0.282	0.945
<b>Stage k=2</b>					
$\hat{y}(\mathbf{x}) = \beta + Z(\mathbf{x})$	0.997	0.885	0.731	1.236	4.127

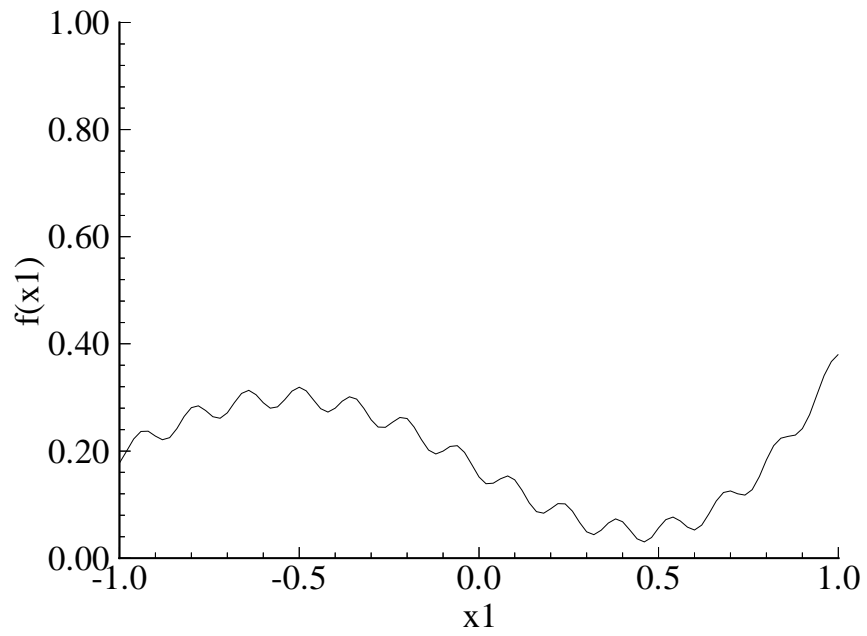


Figure 7.1: A one dimensional view of the Case 1 test function ( $\epsilon = 1.0$ ).

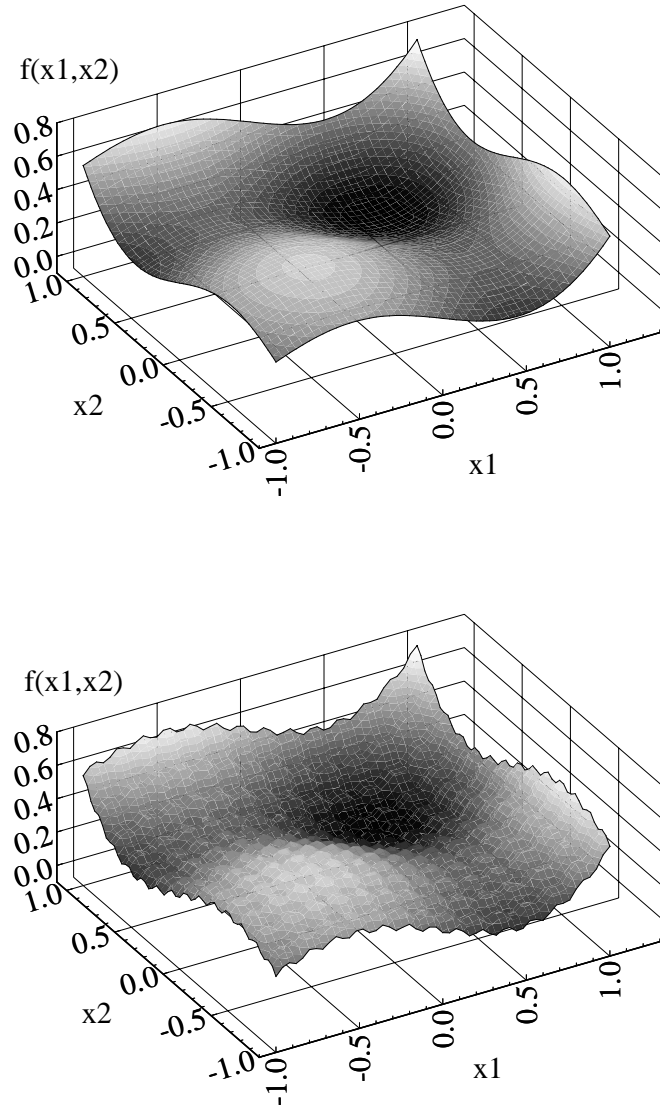


Figure 7.2: A two dimensional view of the smooth and noisy variants of the Case 1 test function ( $\epsilon = 1.0$ ).

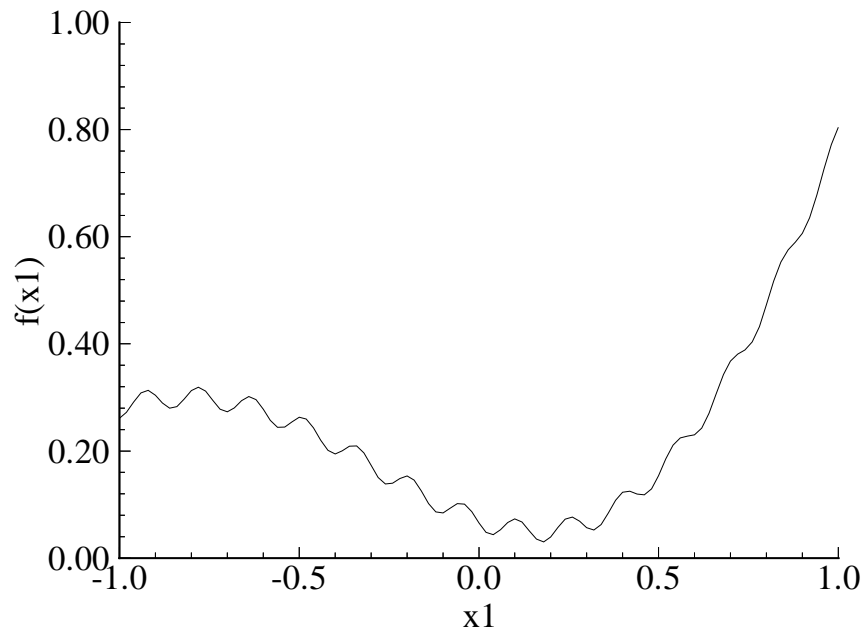


Figure 7.3: A one dimensional view of the Case 2 test function ( $\epsilon = 0.7$ ).

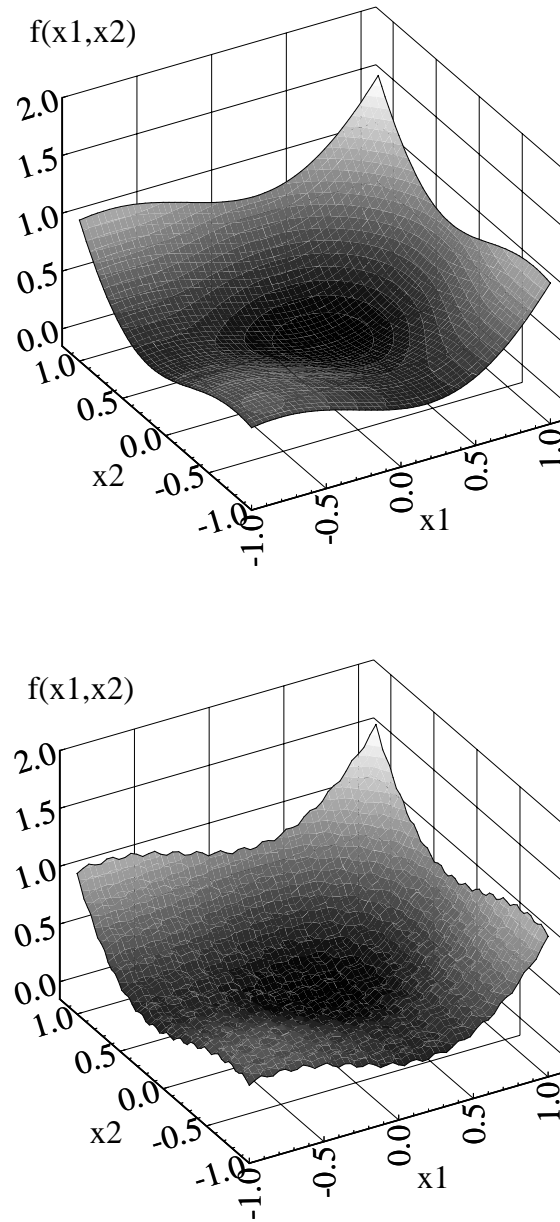


Figure 7.4: A two dimensional view of the smooth and noisy variants of the Case 2 test function ( $\epsilon = 0.7$ ).



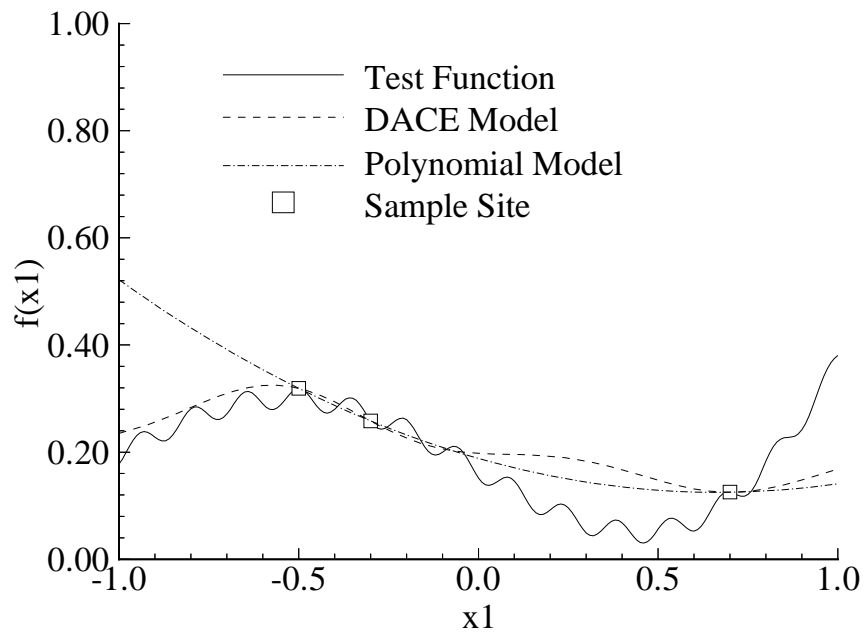


Figure 7.5: The DACE and quadratic polynomial RS models for Case 1 ( $\epsilon = 1.0$ ) of the one variable test function.

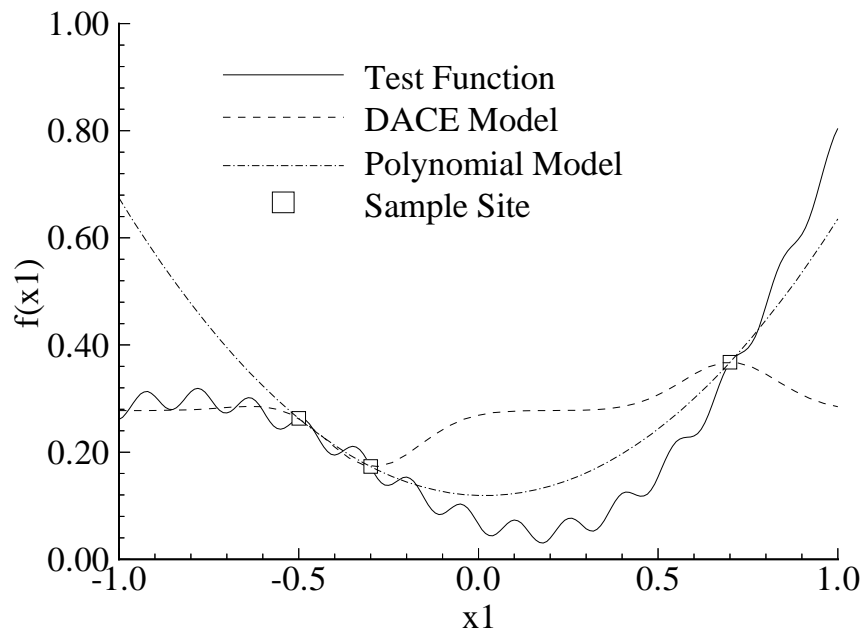


Figure 7.6: The DACE and quadratic polynomial RS models for Case 2 ( $\epsilon = 0.7$ ) of the one variable test function.

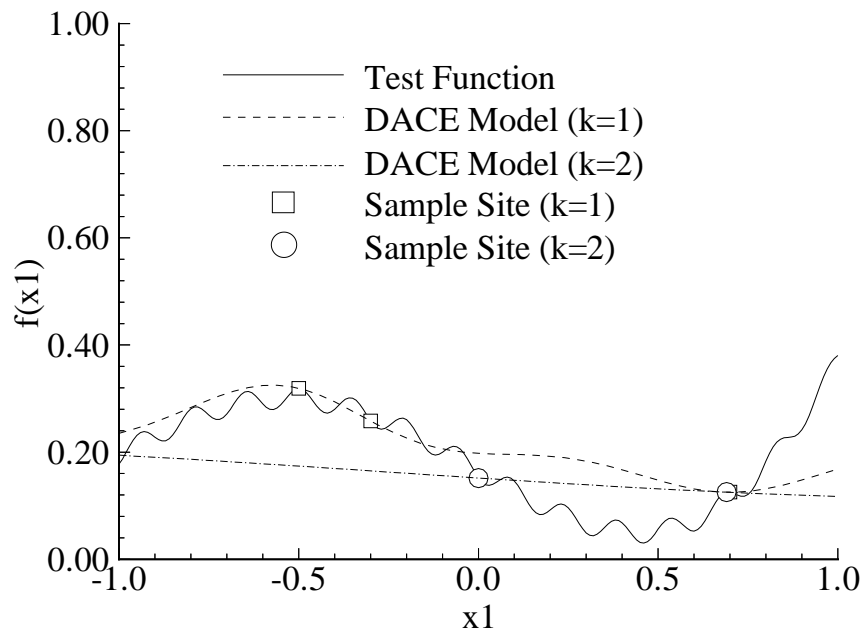


Figure 7.7: The  $k = 1$  and  $k = 2$  stage DACE models for Case 1 ( $\epsilon = 1.0$ ) of the one variable test function.

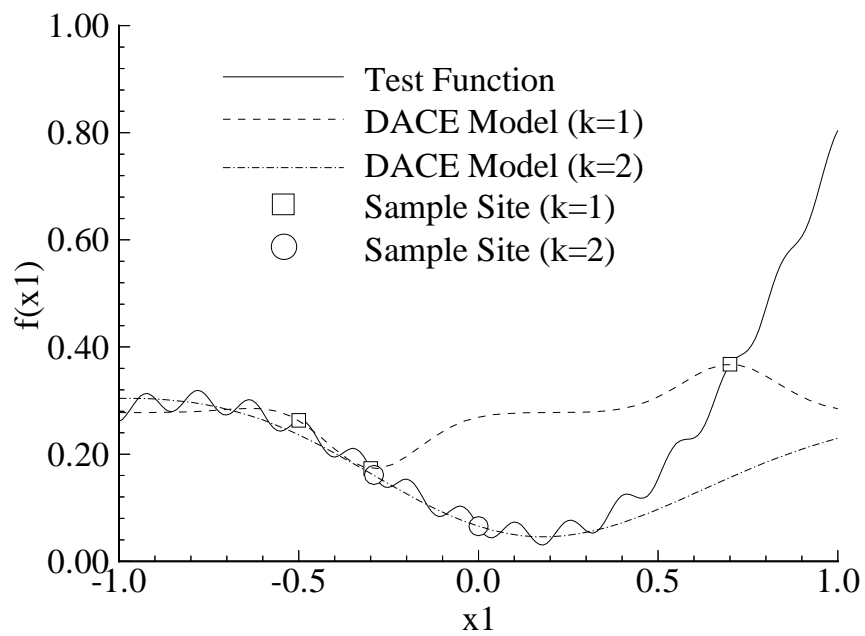


Figure 7.8: The  $k = 1$  and  $k = 2$  stage DACE models for Case 2 ( $\epsilon = 0.7$ ) of the one variable test function.

# Chapter 8

## HSCT Optimization Using DACE Modeling

### 8.1 Five Variable HSCT Optimization Problem

As noted in Chapter 7, DACE modeling methods are well-suited to applications in which computational expense prohibits the evaluation of  $n_s$  response values needed to construct quadratic polynomial models. The variable-complexity modeling paradigm for HSCT optimization is an application in which DACE modeling may be of use. Here, the high fidelity aerodynamic analysis methods typically employ Euler or Navier-Stokes flow solvers which are orders of magnitude more computationally expensive than low fidelity (algebraic) and medium fidelity (linear theory) analysis methods. Because of the computational expense of the high fidelity analyses, it is impractical to perform more than a few,  $\mathcal{O}(10^1 - 10^2)$ , Euler/Navier-Stokes analyses in the HSCT design process. The  $k$ -stage DACE modeling paradigm as employed by Osio and Amon [67] provides a framework through which such a low number of computationally expensive analyses may be used in the HSCT design process. As an initial exploration of the possibilities of using DACE modeling methods in HSCT optimization, the methods of Osio and Amon have been applied to the five variable HSCT optimization problem.

## 8.2 Application of DACE Modeling

In this version of the five variable HSCT optimization problem Steps 1-7 of the VCRSM process were identical to those described in Chapters 5 and 6. In Step 8 however, DACE methods were used to model the supersonic volumetric wave drag,  $C_{Dwave}$ , instead of a quadratic polynomial model. The DACE model for  $C_{Dwave}$  was constructed from the wave drag values computed at 50  $D$ -optimal sample sites in the design space. For the DACE model the correlation parameter was  $\theta = 56.0$  and  $\hat{\beta} = 0.00152$ . Note that the Harris wave drag code [44] was used to calculate these 50 medium fidelity analyses used in the DACE model. Also note that quadratic RS polynomial models were retained for  $C_{L\alpha}$  (M=2.4) and  $C_T/C_L^2$  in this version of the five variable HSCT optimization problem.

Using the DACE model for  $C_{Dwave}$  and the polynomial RS models for  $C_{L\alpha}$  (M=2.4) and  $C_T/C_L^2$ , HSCT optimizations were performed using the same three initial HSCT configurations used for trials 1-3 in Chapter 6. Also, the same lower and upper bounds were placed on the optimized variables as in Table 2.4. The results of these three optimization trials are listed in Table 8.1 and the planforms for the initial and optimal HSCT configurations are shown in Figures 8.1-8.3. Note that a DACE model may have several local minima in contrast to a polynomial RS model which has a unique minimum. For this reason, trials 1-3 converge to slightly different HSCT configurations, each of which is locally optimal. However, all of the optima produced in trials 1-3 are similar. Further, these local optima are also similar to the optimal HSCT configuration obtained when only polynomial RS models were used in HSCT optimization (Chapter 6).

## 8.3 Application of Multi-Fidelity DACE Modeling

Following the approach of Osio and Amon [67] and the methods explored in Chapter 7, the DACE model constructed from the 50  $C_{Dwave}$  values was designated the  $k = 1$  stage DACE model. This  $k = 1$  stage model is also referred to as the *medium fidelity DACE model*. Next, the three optimal HSCT configurations found in trials 1-3 were

selected as the sample sites for the  $k = 2$  stage. To simulate high fidelity analysis data, two counts of drag were added to the wave drag predicted using the Harris wave drag code at the three sample sites. This difference of two drag counts is consistent with the discrepancy between the Harris wave drag code and the Euler/Navier-Stokes solver GASP [53] as observed in an investigation by Knill et al. [34]. Using this simulated high fidelity data for  $C_{Dwave}$ , the  $k = 2$  stage DACE model was constructed where the correlation parameter was calculated as  $\theta^{(1)} = 671.76$  and a value of  $\hat{\beta}^{(0)} = 0.00152$  was used from the  $k = 1$  stage. This  $k = 2$  stage model is referred to as the *high fidelity DACE model*.

Based on experience gained in using multi-fidelity DACE modeling for the five variable test problem (Chapter 7), the allowable design space was restricted to  $\pm 3$  percent around the point in design space defined by  $C_{root} = 166.1ft$ ,  $C_{tip} = 8.6ft$ ,  $\Lambda_{LEI} = 73.0^\circ$ ,  $t/c \text{ ratio} = 2.3\%$ , and  $W_{fuel} = 306588lb$ . This point corresponds to the mean of each of the five variables found in optimization trials 1-3 using the medium fidelity DACE model. The lower and upper bounds defined by the range of  $\pm 3$  percent are listed in Table 8.2.

Optimization using the high fidelity DACE model for  $C_{Dwave}$  was performed starting from the initial HSCT configurations used in trials 1-3. In all cases the optimizer was not able to find a feasible HSCT configuration within the bounds specified by the  $\pm 3$  percent limits. In particular, the range constraint (range  $\geq 5500 \text{ naut.mi.}$ ) was not satisfied in all three optimization trials.

The source of the convergence difficulties was traced to the sensitivity of the mission performance of the HSCT to slight increases in drag. More specifically, the addition of two counts of drag to create the simulated high fidelity analysis resulted in a  $k = 2$  stage DACE model for  $C_{Dwave}$  that behaved completely differently than did the  $k = 1$  stage DACE model. This behavior is shown in Figure 8.4 where  $C_{Dwave}$  was calculated as  $\Lambda_{LEI}$  varies while  $C_{root}$ ,  $C_{tip}$ ,  $t/c \text{ ratio}$ , and  $W_{fuel}$  were held constant.

To reduce the apparent performance penalty created by the high fidelity wave drag values in the  $k = 2$  stage model, the optimizer moved away from the location of the  $k = 1$  stage optima and toward HSCT wing configurations having low structural weight characteristics (i.e., high  $t/c$  ratios and low  $\Lambda_{LEI}$  angles). While the structural

weight of the HSCT was decreased, the fuel weight was increased as much as allowable to overcome the range deficiency. These trends are illustrated in Table 8.3 which lists the optimization results for trial 1, both for  $\pm 3$  percent bounds and the original  $k = 1$  stage bounds. The corresponding optimal (although infeasible) HSCT configurations are shown in Figures 8.5 and 8.6, respectively.

## 8.4 Discussion of Results

While this particular use of DACE modeling was hampered by optimization difficulties, these problems are partially attributable to the nature of the HSCT optimization problem and its sensitivity to increases in drag. For example, if the simulated high fidelity values for  $C_{Dwave}$  were created by subtracting two counts of drag instead of adding two counts of drag, it is suspected that the results of the  $k = 2$  stage optimization would have been quite different. The use of DACE methods, for both single level of fidelity modeling and for multi-fidelity modeling, remains attractive in situations where computational expense precludes the use of quadratic polynomial modeling in optimization studies.

## 8.5 Discussion of DACE Model Sensitivity

### 8.5.1 Sensitivity of DACE Parameters to Scaling

During this investigation it was noticed that the DACE model predictions for the optimal correlation parameter,  $\theta$ , and the constant term in the DACE model,  $\hat{\beta}$ , were sensitive to the scaling of the independent variables. For example, in the DACE model for  $C_{Dwave}$  described above the variables were scaled to be  $\mathcal{O}(0.1)$  to coincide with the existing variable scaling used in the HSCT software. For this DACE model the values for  $\theta$  and  $\hat{\beta}$  were 56.0 and 0.00152, respectively. However, if the variables were scaled to span  $[-1, 1]^5$ , the DACE model values for  $\theta$  and  $\hat{\beta}$  became 0.066 and 0.00159, respectively.

Initially this caused some concern since it was thought that  $\hat{\beta}$  should have been



independent of the scaling method. This would be true if all the variables were scaled using the same constant factor (e.g., if  $100 \leq x_1 \leq 200$  and  $50 \leq x_2 \leq 60$  are scaled using a factor of 200). However, the method used to scale the variables to  $[-1, 1]^{n_v}$  as given by Myers and Montgomery [14, page 22] scales each variable by a different amount. This scaling function is

$$x_i^{(p)} = \frac{\hat{x}_i^{(p)} - [\max(\hat{\mathbf{x}}_i) + \min(\hat{\mathbf{x}}_i)]/2}{[\max(\hat{\mathbf{x}}_i) - \min(\hat{\mathbf{x}}_i)]/2}, \quad (8.1)$$

where  $\hat{x}_i^{(p)}$  is the  $i^{\text{th}}$  component of the *unscaled* variable  $\hat{\mathbf{x}}^{(p)}$ ;  $\max(\hat{\mathbf{x}}_i)$  and  $\min(\hat{\mathbf{x}}_i)$  represent the maximum and minimum in each of the  $i = 1, \dots, n_v$  unscaled variables; and  $p = 1, \dots, n_s$ .

The result of using Equation 8.1 is that some variables are “stretched” or “shrunk” more than others as all of the variables are scaled to span  $[-1, 1]$ . Because of this, Equation 8.1 does not preserve the *distances* between the  $\hat{\mathbf{x}}^{(\mathbf{p})}$  locations in the original (unscaled) design space.

The effect of scaling in DACE models with a single correlation parameter  $\theta$  can be significant. Consider the sample data listed in Table 8.4 where the unscaled variables  $x_1$  and  $x_2$  have ranges of  $[1.0, 2.0]$  and  $[0.5, 1.0]$ , respectively. If a DACE model with a single value of theta is created using the unscaled variables, the resulting value for  $\hat{\beta}$  is 0.5246. However, if the DACE model is constructed using the scaled variables  $\hat{\beta}$  becomes 0.5467.

However, this scaling problem may be overcome if a vector of correlation parameters is used (see Equation 4.21), where the correlation parameter  $\theta_1$  is applied to the variable  $x_1$  and the correlation parameter  $\theta_2$  is applied to the variable  $x_2$ . Using  $\theta_1$  and  $\theta_2$  in the construction of the correlation matrix  $\mathbf{R}$  conserves the distances between the  $\hat{\mathbf{x}}^{(\mathbf{p})}$  locations as they are scaled to  $[-1, 1]$ .

When the two correlation parameters are used when creating DACE models of the sample data given in Table 8.4, the value obtained for  $\hat{\beta}$  is 0.5248. This value of  $\hat{\beta}$  is the same for the DACE models constructed using both the unscaled and the scaled variables.

### 8.5.2 Implications for DACE Models Used in this Study

Recall that the DACE model for  $C_{Dwave}$  used in the five variable HSCT optimization study was created using unscaled variables supplied by the HSCT software. Since the variables were not scaled to  $[-1, 1]$ , there was no need to use the vector of correlation parameters as was done with the two variable sample data described above.

As a final comment on the issue of scaling, note that the least squares surface fitting method used to create the polynomial response surface models *does not* depend on the distances between the  $n_s$  sample sites. Thus, the use of Equation 8.1 to scale the independent variables to  $[-1, 1]^{n_v}$  is perfectly valid for the polynomial RS modeling performed in this research.

Table 8.1: Initial and optimal variable values for trials 1-3 of the five variable HSCT optimization problem using the DACE model for  $C_{Dwave}$ .

<b>Variable</b>	<b>Initial</b>	<b>Optimal</b>
<b>Trial 1</b>		
$C_{root}$	185.0 <i>ft</i>	164.7 <i>ft</i>
$C_{tip}$	10.0 <i>ft</i>	8.7 <i>ft</i>
$\Lambda_{LEI}$	75.0°	73.0°
<i>t/c ratio</i>	2.0 %	2.2 %
$W_{fuel}$	315000 <i>lb</i>	306279 <i>lb</i>
<i>TOGW</i>	640724 <i>lb</i>	622338 <i>lb</i>
<b>Trial 2</b>		
$C_{root}$	210.0 <i>ft</i>	167.4 <i>ft</i>
$C_{tip}$	6.0 <i>ft</i>	8.3 <i>ft</i>
$\Lambda_{LEI}$	68.0°	72.9°
<i>t/c ratio</i>	2.5 %	2.2 %
$W_{fuel}$	295000 <i>lb</i>	305550 <i>lb</i>
<i>TOGW</i>	626455 <i>lb</i>	622166 <i>lb</i>
<b>Trial 3</b>		
$C_{root}$	185.0 <i>ft</i>	163.5 <i>ft</i>
$C_{tip}$	13.0 <i>ft</i>	8.9 <i>ft</i>
$\Lambda_{LEI}$	68.0°	73.1°
<i>t/c ratio</i>	2.0 %	2.4 %
$W_{fuel}$	315000 <i>lb</i>	307934 <i>lb</i>
<i>TOGW</i>	647349 <i>lb</i>	622557 <i>lb</i>

Table 8.2: Lower and upper bounds on the HSCT optimization problem using a multi-fidelity DACE model for  $C_{Dwave}$ .

<b>Variable</b>	<b>Minimum</b>	<b>Maximum</b>
$C_{root}$	161.1 <i>ft</i>	171.1 <i>ft</i>
$C_{tip}$	8.4 <i>ft</i>	8.9 <i>ft</i>
$\Lambda_{LEI}$	70.8°	75.2°
<i>t/c ratio</i>	2.2%	2.3%
$W_{fuel}$	305550 <i>lb</i>	312720 <i>lb</i>

Table 8.3: Initial and optimal variable values used in the five variable HSCT optimization problem using the high fidelity DACE model for  $C_{Dwave}$ . Both of the optimization cases were started from the trial 1 initial HSCT configuration.

Variable	Initial	Optimal
<b><math>\pm 3\%</math> Design Space</b>		
$C_{root}$	185.0 <i>ft</i>	161.1 <i>ft</i>
$C_{tip}$	10.0 <i>ft</i>	8.4 <i>ft</i>
$\Lambda_{LEI}$	75.0°	71.4°
<i>t/c ratio</i>	2.0 %	2.2 %
$W_{fuel}$	3155000 <i>lb</i>	312720 <i>lb</i>
<i>TOGW</i>	640724 <i>lb</i>	629879 <i>lb</i>
<b>Reduced Design Space</b>		
$C_{root}$	185.0 <i>ft</i>	149.1 <i>ft</i>
$C_{tip}$	10.0 <i>ft</i>	9.6 <i>ft</i>
$\Lambda_{LEI}$	75.0°	69.5°
<i>t/c ratio</i>	2.0 %	2.4 %
$W_{fuel}$	315000 <i>lb</i>	312895 <i>lb</i>
<i>TOGW</i>	640724 <i>lb</i>	628175 <i>lb</i>

Table 8.4: Sample data to demonstrate the effect of scaling in DACE models.

$\mathbf{x}_1$	$\mathbf{x}_2$	$y(\mathbf{x}_1, \mathbf{x}_2)$
<b>Unscaled Variables</b>		
1.000	0.500	0.10
1.000	1.000	0.60
2.000	0.500	0.70
1.750	0.875	0.90
2.000	1.000	0.80
1.500	0.500	0.50
1.250	0.750	0.40
1.500	1.000	0.50
<b>Variables Scaled to <math>[-1, 1]</math></b>		
-1.0	-1.0	0.10
-1.0	1.0	0.60
1.0	-1.0	0.70
0.5	0.5	0.90
1.0	1.0	0.80
0.0	-1.0	0.50
-0.5	0.5	0.40
0.0	1.0	0.50

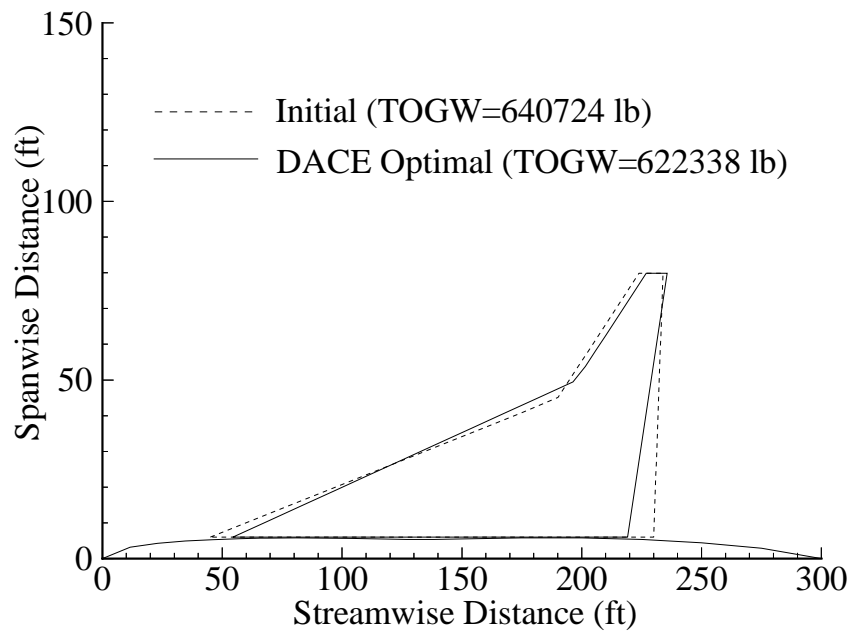


Figure 8.1: Trial 1 initial and optimal HSCT configurations of the five variable HSCT optimization problem.

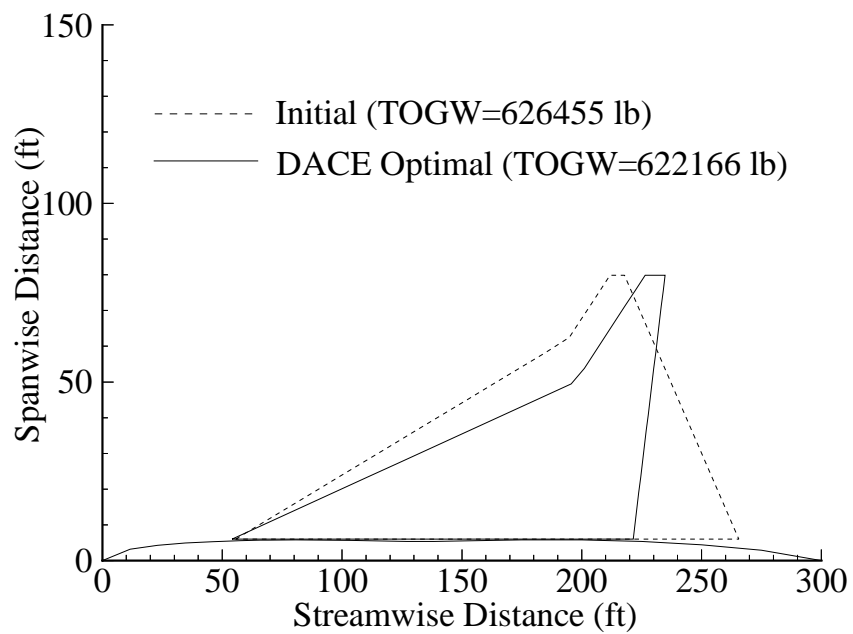


Figure 8.2: Trial 2 initial and optimal HSCT configurations of the five variable HSCT optimization problem.



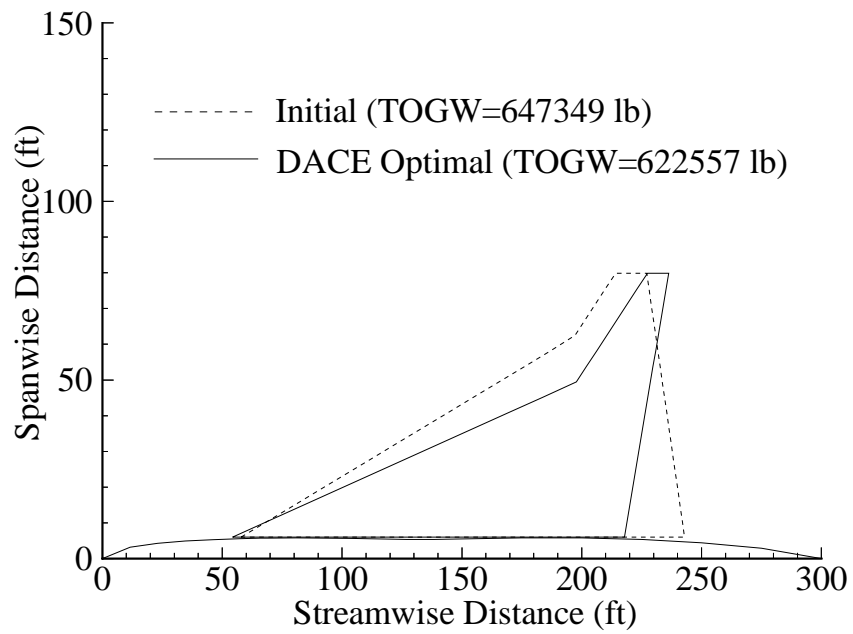


Figure 8.3: Trial 3 initial and optimal HSCT configurations of the five variable HSCT optimization problem.

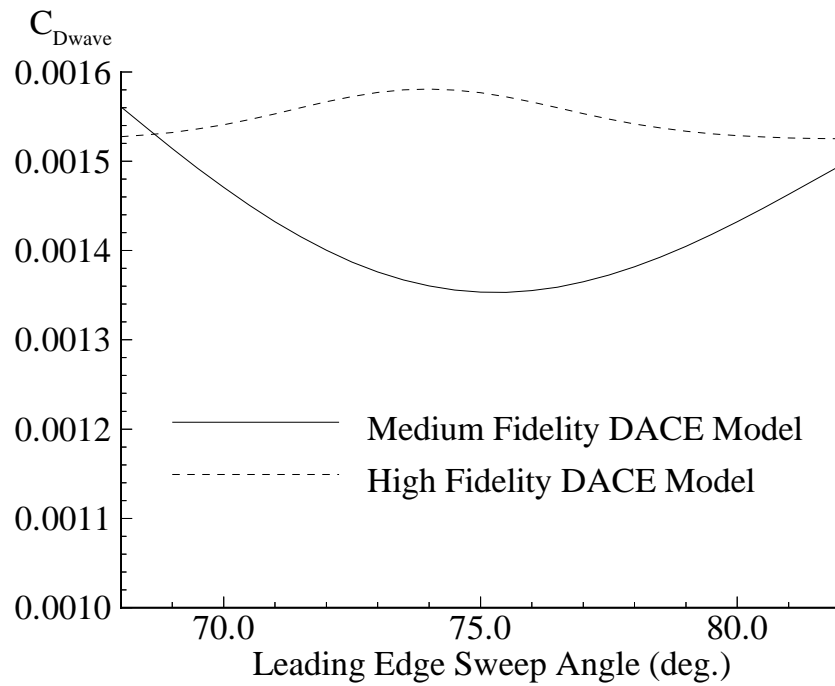


Figure 8.4: A view of the medium fidelity (stage  $k = 1$ ) and high fidelity (stage  $k = 2$ ) models for  $C_{D_{wave}}$  used in the five variable HSCT optimization problem.

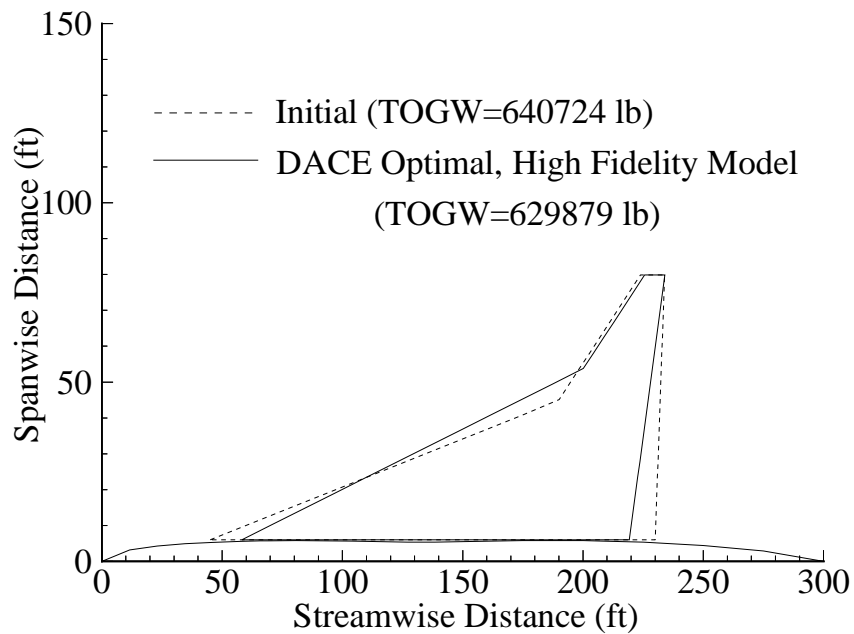


Figure 8.5: Initial and optimal HSCT configurations for the HSCT optimization problem using the high fidelity DACE model. Here the design space was reduced to  $\pm 3.0$  percent around the optimal HSCT configurations obtained from trials 1-3.

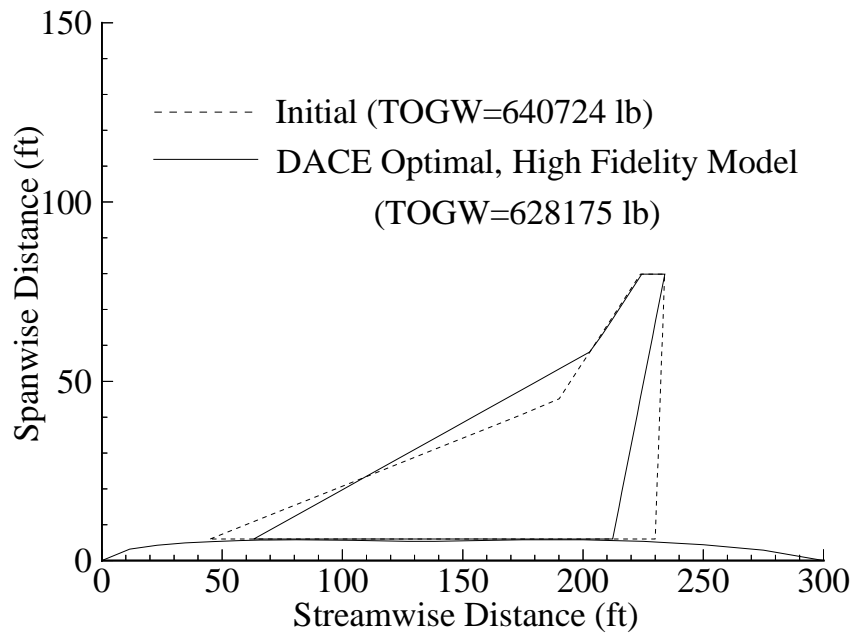


Figure 8.6: Initial and optimal HSCT configurations for the HSCT optimization problem using the high fidelity DACE model. Here the original reduced design space was used.

## Chapter 9

# Parallel Computing and the VCRSM Method

The parallel computing work described in this section was primarily performed by Susan Burgee with the assistance of the author. Thus only a cursory overview of the parallel computing efforts for this research project will be presented here. The reader is directed to References [36] and [37] for a more detailed explanation of these efforts. Our use of parallel computing originally involved a 28-node (i.e., processor) Intel Paragon at Virginia Tech and subsequently a 96-node Intel Paragon at the NASA Langley Research Center (which now resides at Virginia Tech). The Intel Paragon is a distributed memory parallel computer that uses message passing to transfer data among the various processors. See Reference [76] for a detailed description of this computer and its capabilities.

The VCRSM methodology is particularly well suited to a strategy of coarse grained parallel computing as  $n_s$  HSCT configurations specified by DOE theory must be evaluated prior to the construction of polynomial or DACE approximation models. In coarse grained parallel computing, each processor performs numerous unique HSCT configuration evaluations. This is in contrast to a fine grained application of parallel computing where each processor would perform a unique portion of a single HSCT analysis.

To implement the coarse grained parallelization strategy, a master-slave paradigm

was applied on the Intel Paragon computers. In this arrangement, there is one *master* processor that controls both the data transfer and file input/output (I/O) of the other *slave* processors. This method of parallel computing is particularly useful for computers such as the Intel Paragon which have only a few channels, or only a single channel, through which file I/O may occur.

To start the parallel evaluation of the  $n_s$  HSCT configurations the user supplies a file containing a list of the  $n_v$  design variables which uniquely define the  $n_s$  aircraft. The master processor reads in this file and distributes an approximately equal number of HSCT configurations to each slave processor. For example, if  $n_s = 99$  and there are five processors including the master, then four processors would receive input data for 20 HSCT configurations and the fifth processor would receive input data for 19 HSCT configurations. Both the master and slave processors then analyze their respective subsets of the  $n_s$  HSCT configurations while storing the analysis results in an array local to each processor. When each processor has finished its portion of the HSCT analyses, it sends the array of output data back to the master processor. The master node then combines the various arrays of output data into a single array and returns the combined output data file to the user.

Susan Burgee's work focused on a four variable HSCT optimization problem that is similar to the five variable HSCT optimization problem described in Chapter 2. In the four variable problem there were 1296 HSCT configurations to be evaluated using the low fidelity analysis methods (i.e., algebraic models), and 157 HSCT configurations to be evaluated using the medium fidelity analysis methods (linear theory aerodynamics). Once more, the reader is referred to [36] and [37] which covers specific information particular to the four variable HSCT optimization problem.

The results of the parallel computing efforts for the four variable HSCT optimization problem are shown in Figures 9.1 and 9.2 where the speedup and efficiency of the parallelized HSCT analyses are plotted versus the number of processors used on an Intel Paragon. *Speedup* is defined as

$$\frac{T_s}{T_p}, \quad (9.1)$$

where  $T_s$  is the serial execution time (i.e., the time required to perform all  $n_s$  HSCT

evaluations on a single processor) and  $T_p$  is the parallel execution time (i.e., the time required when splitting up the  $n_s$  HSCT evaluations among the processors). Parallel *efficiency* is defined as

$$\frac{T_s}{(p T_p)}, \quad (9.2)$$

where  $p$  is the number of processors used to perform the  $n_s$  HSCT evaluations. Typically  $T_s$  and  $T_p$  are measured in units of time (e.g., seconds, minutes, or hours).

In Figures 9.1 and 9.2 the actual speedup and efficiency are compared to ideal speedup and ideal efficiency, respectively. This ideal value is the upper bound on parallel computing performance since it represents the perfect division of labor among the  $p$  processors. These figures also provide some measure of *scalability*, i.e., the ability of a parallel computer program to perform efficiently as  $p$  increases. Thus, by comparing the actual results to the ideal linear limit, one can quantify how effectively the parallelization of work has been accomplished.

The actual results deviate from the ideal due to unavoidable file I/O demands during the HSCT evaluation process which must be executed serially, and due to unavoidable communication overhead between the master and slave processors. The results shown here, where efficiency values are in the range of 80 percent, are considered to be quite good. However, there is some question regarding the scalability of this implementation of coarse grained parallel computing. Note that as  $p$  increases both the speedup and efficiency deviate further from the ideal level. This occurs because the unavoidable serial execution and inter-processor communication time begin to dominate the total CPU time as  $p$  increases. Thus, while this particular implementation of parallel computing will suffice for tens of processors, it appears that it will not be acceptable for implementation on massively parallel computer having hundreds or thousands of processors.

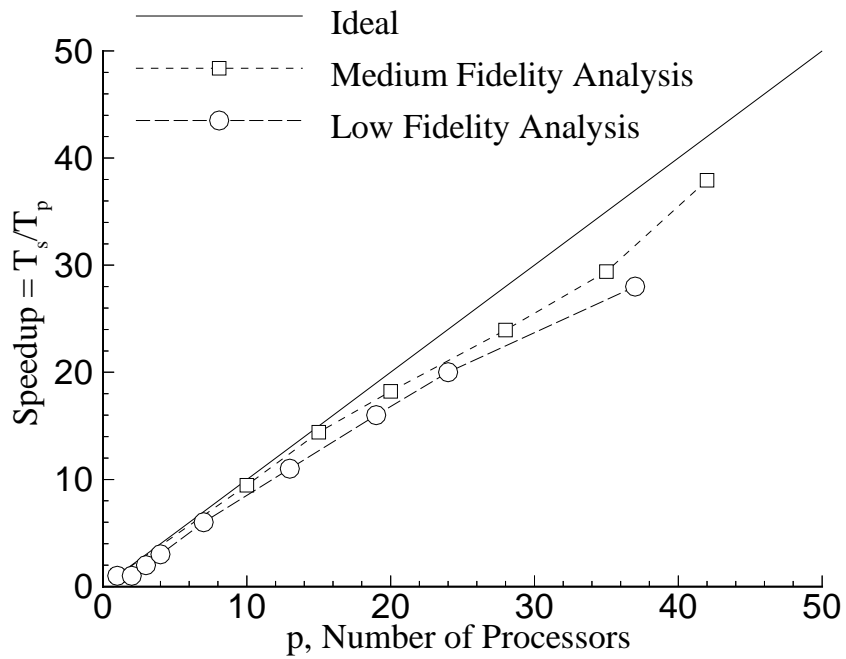


Figure 9.1: Parallel computing speedup for the low fidelity and medium fidelity aerodynamic analyses.



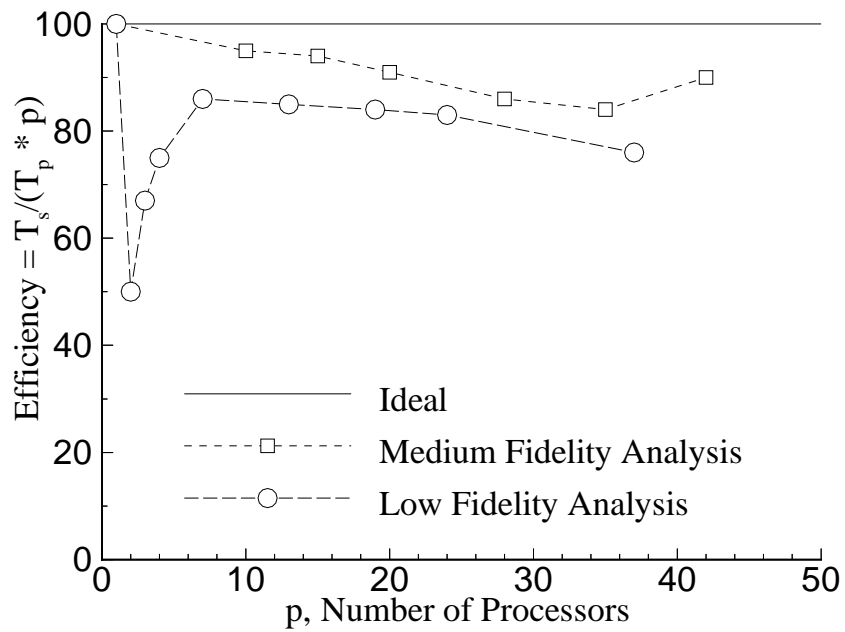


Figure 9.2: Parallel computing efficiency for the low fidelity and medium fidelity aerodynamic analyses.

# Chapter 10

## Concluding Remarks

### 10.1 Summary

The primary objective of this research was to develop a methodology which would enable aircraft multidisciplinary design optimization using analysis methods of varying computational expense, in a manner which leverages the power of parallel computing. Clearly the variable-complexity response surface modeling method meets these goals.

As detailed in Chapter 5, the variable-complexity portion of the VCRSM method incorporates increasingly sophisticated computational models in successive stages of the design process. Thus, the VCRSM method functions in a manner similar to the aircraft design methodology used in the aerospace industry where the design process is comprised of stages in which more accurate, and more computationally expensive, analysis techniques are employed as an aircraft design is refined.

The response surface modeling portion of VCRSM provides a mechanism through which parallel computing may be applied to MDO problems. As described in Chapter 9, coarse grained parallel computing techniques make response surface modeling computationally practical for problems involving numerous variables. The RSM portion of VCRSM provides the benefit of removing the effects of numerical noise which often inhibit the use of gradient-based optimization in MDO problems. Another advantage of the VCRSM approach is that the response surface models facilitate the numerous trade-off studies performed in the aircraft design process since the RS

models can be used repeatedly for a negligible computational cost.

The initial research into the development of the VCRSM method focused on understanding the role of numerical noise in the HSCT optimization problem and on applying statistical modeling methods from design of experiments theory and regression analysis to remove the effects of numerical noise. The basic format of the VCRSM method was then refined through a series of HSCT optimization problems of increasingly complexity. These five and ten variable MDO problems were simple enough to allow for detailed examination, but retained sufficient complexity to preserve the multidisciplinary aspects of aircraft design.

The preliminary investigation into the use of DACE interpolating methods provided a useful comparison to polynomial response surface modeling, particularly with regard to the modeling accuracy of both techniques. From the data garnered from the two test cases it appears that quadratic polynomial response surface methods were more accurate than the DACE methods. This was true even for the test case which was highly non-quadratic. However, this cursory investigation is not intended to serve as an exhaustive study of DACE modeling methods, and the results obtained here have identified aspects of DACE modeling which merit additional study.

The development of the VCRSM method has been one of virtually continuous improvement. Certainly this is still true, and areas for future research are identified below.

## 10.2 Future Work

### 10.2.1 Variable-Complexity Modeling

As of this writing there is already some progress in the use of an Euler/Navier-Stokes solver to provide high fidelity aerodynamic data for use in HSCT optimization [34, 35]. This is in addition to the work of Kaufman [60] and Balabanov et al. [51] which has focused on the use of high fidelity structural analysis data in HSCT optimization. This work should be coupled with future research issues in parallel computing and DACE modeling which are discussed below.

### 10.2.2 Parallel Computing

The use of parallel computing techniques to provide high fidelity aerodynamic data remains challenging due to the per-node memory limitations on many current parallel computers. Until the availability of vendor-supplied CFD software that employs fine grained parallel computing strategies, one is faced with the difficult choice of parallelizing pre-existing software, or developing a new CFD solver in-house.

Other challenging aspects of parallel computing arise when considering the use of the VCRSM method on massively parallel computers where the number of processors is  $\mathcal{O}(10^2 - 10^3)$ . The first issue concerns the scalability of the VCRSM method. Burgee et al. [37] indicated that the parallelization strategies developed for VCRSM may not be efficient for massively parallel computers. A second issue concerns data management for the VCRSM method which currently relies on substantial user interaction between steps. While this is acceptable, and even desirable, for serial computing or modest parallel computing, handling data files of gigabyte size or larger is impractical for massively parallel computing from both a file input/output perspective and a data storage perspective. Thus, most steps in the VCRSM process would require automation.

### 10.2.3 Design of Experiments Theory

Design of experiments theory offers many alternatives to the full factorial and  $D$ -optimal experimental designs used in this research. Some of the DOE methods not investigated here, such as minimum bias experimental designs and space filling designs, may become attractive as the creation of  $D$ -optimal experimental designs becomes prohibitively expensive. Also, as the number of variables in a problem grows much above ten, even two-level full factorial experimental designs cannot be used to provide an initial sampling of the design space. The use of fractional factorial and similar experimental designs should be investigated for use in such situations.

### 10.2.4 Polynomial Response Surface Modeling

The procedures for estimating the modeling error in polynomial response surface models is another area of interesting future work. In particular, the PRESS method described in Myers and Montgomery [14, pages 45–48] appears to be a promising method with respect to both accuracy and computational efficiency.

### 10.2.5 DACE Modeling

The preliminary exploration of DACE modeling methods in Chapters 7 and 8 raises numerous questions. Clearly, additional research should be focused in this area to better understand both the benefits and drawbacks of DACE modeling. Particular emphasis should be placed on the use of multi-fidelity DACE modeling methods. In many MDO problems polynomial modeling using computationally expensive data will not be possible. DACE modeling offers what appears to be an attractive nonlinear modeling capability which is well-suited to problems where polynomial modeling is not an option.

# Bibliography

- [1] Vanderplaats, G. N. *Numerical Optimization Techniques for Engineering Design: With Applications*, p. 1, McGraw Hill, New York, NY (1984).
- [2] Giunta, A. A., Dudley, J. M., Narducci, R., Grossman, B., Haftka, R. T., Mason, W. H., and Watson, L. T. “Noisy Aerodynamic Response and Smooth Approximations in HSCT Design,” in *Proceedings of the 5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, pp. 1117–1128, Panama City Beach, FL, AIAA Paper 94–4376 (Sept. 1994).
- [3] Venter, G., Haftka, R. T., and Starnes, J. H. “Construction of Response Surfaces for Design Optimization Applications,” in *Proceedings of the 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, pp. 548–564, Bellevue, WA, AIAA Paper 96–4040 (Sept. 1996).
- [4] Dudley, J., Huang, X., MacMillin, P. E., Grossman, B., Haftka, R. T., and Mason, W. H. “Multidisciplinary Optimization of the High-Speed Civil Transport,” AIAA Paper 95–0124 (1995).
- [5] Kaufman, M., Balabanov, V., Burgee, S., Giunta, A. A., Grossman, B., Mason, W. H., Watson, L. T., and Haftka, R. T. “Variable-Complexity Response Surface Approximations for Wing Structural Weight in HSCT Design,” AIAA Paper 96–0089 (1996).
- [6] Narducci, R., Grossman, B., Valorani, M., Dadone, A., and Haftka, R. T. “Optimization Methods for Non-smooth or Noisy Objective Functions in Fluid

- Design Problems,” in *Proceedings of the AIAA 12th Computational Fluid Dynamics Conference*, pp. 21–32, San Diego, CA, AIAA Paper 95–1648 (June 1995).
- [7] Wu, X., Cliff, E. M., and Gunzburger, M. D. “An Optimal Design Problem for a Two-Dimensional Flow in a Duct,” *Optimal Control Applications & Methods*, 17, 329–339 (1996).
- [8] Gill, P. E., Murray, W., and Wright, M. H. *Practical Optimization*, pp. 54–56, 339–345, Academic Press, Inc., San Diego, CA (1981).
- [9] Nelder, J. A. and Mead, R. “A Simplex Method for Function Minimization,” *Comput. J.*, 7, 308–313 (1965).
- [10] Jayaram, S., Myklebust, A., and Gelhausen, P. “ACSYNT – A Standards-Based System for Parametric Computer Aided Conceptual Design of Aircraft,” AIAA Paper 92–1268 (1992).
- [11] McCullers, L. A. “Aircraft Configuration Optimization Including Optimized Flight Profiles,” in Sobieski, J., ed., *Proceedings of the Symposium on Recent Experiences in Multidisciplinary Analysis and Optimization*, pp. 396–412, Hampton, VA, NASA CP 2327 (1984).
- [12] Frank, P. D., Booker, A. J., Caudell, T. P., and Healy, M. J. “A Comparison of Optimization and Search Methods for Multidisciplinary Design,” in *Proceedings of the 4th AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization*, pp. 1040–1057, Cleveland, OH, AIAA Paper 92–4827 (Sept. 1992).
- [13] Sobieszczanski–Sobieski, J. and Haftka, R. T. “Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments,” AIAA Paper No. 96–0711 (1996).
- [14] Myers, R. H. and Montgomery, D. C. *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, pp. 1–141, 279–401, 462–480, John Wiley & Sons, New York, NY (1995).

- [15] Healy, M. J., Kowalik, J. S., and Ramsay, J. W. “Airplane Engine Selection by Optimization on Surface Fit Approximations,” *J. Aircraft*, 12(7), 593–599 (1975).
- [16] Englund, W. C., Stanley, D. O., Lepsch, R. A., McMillin, M. M., and Unal, R. “Aerodynamic Configuration Design Using Response Surface Methodology Analysis,” AIAA Paper 93–3967 (1993).
- [17] Tai, J. C., Mavris, D. N., and Schrage, D. P. “Application of a Response Surface Method to the Design of Tipjet Driven Stopped Rotor/Wing Concepts,” AIAA Paper 95-3965 (1995).
- [18] Chen, W., Allen, J. K., Schrage, D. P., and Mistree, F. “Statistical Experimentation Methods for Achieving Affordable Concurrent Systems Design,” *AIAA J.*, 35(5), 893–900 (1997).
- [19] Yesilyurt, S., Ghaddar, C. K., Cruz, M. E., and Patera, A. T. “Bayesian-Validated Surrogates for Noisy Computer Simulations; Application to Random Media,” *SIAM J. Sci. Comp.*, 17(4), 973–992 (1996).
- [20] Sellar, R. S., Stelmack, M. A., Batill, S. M., and Renaud, J. E. “Response Surface Approximations for Discipline Coordination in Multidisciplinary Design Optimization,” AIAA Paper 96–1383 (1996).
- [21] Roux, W. J., Stander, N., and Haftka, R. T. “Response Surface Approximations for Structural Optimization,” in *Proceedings of the 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, pp. 565–578, Bellevue, WA, AIAA Paper 96–4042 (Sept. 1996).
- [22] Toropov, V. V. “Simulation Approach to Structural Optimization,” *Structural Optimization*, 1, 37–46 (1989).
- [23] Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. “Design and Analysis of Computer Experiments,” *Statistical Science*, 4(4), 409–435 (1989).



- [24] Sobieszczanski–Sobieski, J. “Optimization by Decomposition: A Step from Hierarchic to Non-Hierarchic Systems,” NASA Conference Publication 3031, Part 1, Second NASA/Air Force Symposium on Recent Advances in Multidisciplinary Analysis and Optimization, Hampton, VA (1988).
- [25] Kroo, I., Altus, S., Braun, R., Gage, P., and Sobieski, I. “Multidisciplinary Optimization Methods for Aircraft Preliminary Design,” in *Proceedings of the 5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, pp. 697–707, Panama City Beach, FL, AIAA Paper 94–4325 (Sept. 1994).
- [26] Gilmore, P. and Kelley, C. T. “An Implicit Filtering Algorithm for Optimization of Functions with Many Local Minima,” *SIAM J. Opt.*, 5(2), 269–285 (1995).
- [27] Dennis, J., El-Alem, M., and Maciel, M. C. “A Global Convergence Theory for General Trust-Region-Based Algorithms for Equality Constrained Optimization,” *SIAM J. Opt.*, 7(1), 177–207 (1997).
- [28] Gage, P. and Kroo, I. “A Role for Genetic Algorithms in a Preliminary Design Environment,” AIAA Paper 93–3933 (1993).
- [29] Aly, S., Marconi, F., Ogot, M., Pelz, R., and Siclari, M. “Stochastic Optimization Applied to CFD Shape Design,” in *Proceedings of the 12th AIAA Computational Fluid Dynamics Conference*, pp. 11–20, San Diego, CA, AIAA Paper 95–1647 (June 1995).
- [30] Siclari, M. J., Van Nostrand, W., and Austin, F. “The Design of Transonic Airfoil Sections for an Adaptive Wing Concept Using a Stochastic Optimization Method,” AIAA Paper 96–0329 (1996).
- [31] Torczon, V. “On the Convergence of Pattern Search Algorithms,” *SIAM J. Opt.*, 7(1), 1–25 (1997).
- [32] Spall, J. C. “Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation,” *IEEE Transactions on Automatic Control*, 37(3), 332–341 (1992).

- [33] Giunta, A. A., Balabanov, V., Haim, D., Grossman, B., Mason, W. H., Watson, L. T., and Haftka, R. T. “Wing Design for a High-Speed Civil Transport Using a Design of Experiments Methodology,” in *Proceedings of the 6th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, pp. 168–183, Bellevue, WA, AIAA Paper 96–4001 (Sept. 1996).
- [34] Knill, D. L., Balabanov, V., Grossman, B., Mason, W. H., and Haftka, R. T. “Certification of a CFD Code for High-Speed Civil Transport Design Optimization,” AIAA Paper 96–0330 (1996).
- [35] Knill, D. L., Balabanov, V., Golovidov, O., Mason, W. H., Grossman, B., Haftka, R. T., and Watson, L. T. “Accuracy of Aerodynamic Predictions and Its Effects on Supersonic Transport Design,” *MAD Center Report 96-12-01*, Multidisciplinary Analysis and Design Center for Advanced Vehicles, Virginia Tech, Blacksburg, VA (1996).
- [36] Burgee, S. *A Coarse Grained Variable-Complexity MDO Paradigm for HSCT Design*, Master’s thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA (1995).
- [37] Burgee, S., Giunta, A. A., Balabanov, V., Grossman, B., Mason, W. H., Narducci, R., Haftka, R. T., and Watson, L. T. “A Coarse Grained Parallel Variable-Complexity Multidisciplinary Optimization Paradigm,” *Intl. J. Supercomputing Applications and High Performance Computing*, 10(4), 269–299 (1996).
- [38] Hutchison, M. G., Unger, E. R., Mason, W. H., and Grossman, B. “Variable-Complexity Aerodynamic Optimization of an HSCT Wing Using Structural Wing-Weight Equations,” *J. Aircraft*, 31(1), 110–116 (1994).
- [39] MacMillin, P. E. *Trim, Control, and Performance Effects in Variable-Complexity High-Speed Civil Transport Design*, Master’s thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA (1996).

- [40] Eminton, E. “On the Numerical Evaluation of the Drag Integral,” ARC Reports and Memoranda No. 3341 (1961).
- [41] Cohen, D. and Friedman, M. D. “Theoretical Investigations of the Supersonic Lift and Drag of Thin, Sweptback Wings with Increased Sweep Near the Root,” NACA TN 2959 (1953).
- [42] Hopkins, E. J. and Inouye, M. “An Evaluation of Theories for Predicting Turbulent Skin Friction and Heat Transfer on Flat Plates at Supersonic and Hypersonic Mach Numbers,” *AIAA J.*, 9(6), 993–1003 (1971).
- [43] Mason, W. H. *Aerodynamic Calculation Methods for Programmable Calculators and Personal Computers*, Pak #4 – Boundary Layer Analysis Methods, Aerocal, Huntington, NY (1981).
- [44] Harris Jr, R. V. “An Analysis and Correlation of Aircraft Wave Drag,” NASA TM X-947 (1964).
- [45] Carlson, H. W. and Walkley, K. B. “Numerical Methods and a Computer Program for Subsonic and Supersonic Design and Analysis of Wings with Attainable Thrust Considerations,” NASA CP 3808 (1984).
- [46] Hoak, D. E. et al. “USAF Stability and Control DATCOM,” Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio, Revised (1978).
- [47] Kay, J., Mason, W. H., Durham, W., and Lutze, F. “Control Authority Assessment in Aircraft Conceptual Design,” AIAA Paper 93–3968 (1993).
- [48] Huang, X., Haftka, R. T., Grossman, B., and Mason, W. H. “Comparison of Statistical-based Weight Equations with Structural Optimization for Supersonic Transport Wings,” AIAA Paper 94–4379 (1994).
- [49] Vanderplaats, Miura and Associates, Inc., Goleta, CA. *GENESIS Users Manual, Version 1.3* (1993).

- [50] Kaufman, M., Balabanov, V., Burgee, S. L., Giunta, A. A., Grossman, B., T., H. R., H., M. W., and Watson, L. T. “Variable-Complexity Response Surface Approximations for Wing Structural Weight in HSCT Design,” *J. Computational Mechanics*, 18(2), 112–126 (1996).
- [51] Balabanov, V., Kaufman, M., Knill, D. L., Haim, D., Golovidov, O., Giunta, A. A., Haftka, R. T., Grossman, B., Mason, W. H., and Watson, L. T. “Dependence of Optimal Structural Weight on Aerodynamic Shape for a High Speed Civil Transport,” in *Proceedings of the 6th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, pp. 599–612, Bellevue, WA, AIAA Paper 96–4046 (Sept. 1996).
- [52] Vanderplaats Research & Development, Inc., Colorado Springs, CO. *DOT Users Manual, Version 4.20* (1995).
- [53] AeroSoft, Inc., Blacksburg, VA. *GASP Users Manual, Version 2.2* (1993).
- [54] Hutchison, M. G. *Multidisciplinary Optimization of High-Speed Civil Transport Configurations Using Variable-Complexity Modeling*, Ph.D. thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA (1993).
- [55] Abbott, I. H. and von Doenhoff, A. E. *Theory of Wing Sections – Including a Summary of Airfoil Data*, pp. 113–118, Dover, New York, NY (1959).
- [56] Carlson, H. W., Mack, R. J., and Barger, R. L. “Estimation of Attainable Leading-Edge Thrust for Wings at Subsonic and Supersonic Speeds,” NASA TP 1500 (1979).
- [57] Taguchi, G. *System of Experimental Design, vol. I,II*, Quality Resources, White Plains, NY, English edition translated by D. Clausing (1991).
- [58] Hogg, R. V. and Ledolter, J. *Applied Statistics for Engineers and Physical Scientists*, p. 449, Macmillin, New York, NY, 2nd edition (1992).
- [59] SAS Institute, Inc., Cary, NC. *JMP Users Guide, Version 3.1* (1995).

- [60] Kaufman, M. *Variable-Complexity Response Surface Approximations for Wing Structural Weight in HSCT Design*, Master's thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA (1996).
- [61] Craig Jr, J. A. *D-Optimal Design Method: Final Report and User's Manual*, General Dynamics Report FZM-6777 for Air Force Contract F33615-78-C-3011 (1978).
- [62] Giunta, A. A., Narducci, R., Burgee, S., Grossman, B., Mason, W. H., Watson, L. T., and Haftka, R. T. "Variable-Complexity Response Surface Aerodynamic Design of an HSCT Wing," in *Proceedings of the 13th AIAA Applied Aerodynamics Conference*, pp. 994–1002, San Diego, CA, AIAA Paper 95–1886 (June 1995).
- [63] Box, M. J. and Draper, N. R. "Factorial Designs, the  $|\mathbf{X}^T\mathbf{X}|$  Criterion, and Some Related Matters," *Technometrics*, 13(4), 731–742 (1971).
- [64] Mitchell, T. J. "An Algorithm for the Construction of *D*-Optimal Experimental Designs," *Technometrics*, 16(2), 203–210 (1974).
- [65] MacMillin, P. E., Golovidov, O., Mason, W. H., Grossman, B., and Haftka, R. T. "An MDO Investigation of the Impact of Practical Constraints on an HSCT Configuration," AIAA Paper 97–0098 (1997).
- [66] Koehler, J. R. and Owen, A. B. *Computer Experiments*, volume 13 of *Handbook of Statistics*, pp. 261–308, Elsevier Science, eds. S. Ghosh and C. R. Rao (1996).
- [67] Osio, I. G. and Amon, C. H. "An Engineering Design Methodology with Multistage Bayesian Surrogates and Optimal Sampling," *Research in Engineering Design*, 8, 189–206 (1996).
- [68] Booker, A. J., Conn, A. R., Dennis, J. E., Frank, P. D., Trosset, M., and Torczon, V. "Global Modeling for Optimization: Boeing/IBM/Rice Collaborative Project," Text provided by Paul D. Frank of Boeing (1995).

- [69] Berger, J. O. *Statistical Decision Theory and Bayesian Analysis*, pp. 4,5,109,110, Springer-Verlag, New York, NY (1985).
- [70] Narducci, R. *Selected Optimization Procedures for CFD-Based Shape Design Involving Shock Waves or Computational Noise*, Ph.D. thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA (1995).
- [71] Haim, D., *Personal Communications*. (1996).
- [72] Toropov, V. V., Filatov, A. A., and Polynkin, A. A. "Multiparameter Structural Optimization Using FEM and Multipoint Explicit Approximations," *Structural Optimization*, 6, 7–14 (1993).
- [73] Crisafulli, P. J., Kaufman, M., Giunta, A. A., Mason, W. H., Grossman, B., Watson, L. T., and Haftka, R. T. "Response Surface Approximations for Pitching Moment, Including Pitch-Up, in the MDO Design of an HSCT," in *Proceedings of the 6th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, pp. 1308–1322, Bellevue, WA, AIAA Paper 96–4136 (Sept. 1996).
- [74] Benoliel, A. M. and Mason, W. H. "Pitch-Up Characteristics for HSCT Class Planforms: Survey and Estimation," AIAA Paper 94–1819 (1994).
- [75] Wolfram, S. *The Mathematica Book*, pp. 859–861, Wolfram Media and Cambridge University Press, Champaign, IL, 3rd edition (1996).
- [76] Intel Corporation, Beaverton, OR. *Paragon System User's Guide* (1996).
- [77] Carlson, C. B. "Description of a Digital Computer Program for Airplane Configuration Plots," NASA TM X-2074 (1970).
- [78] Grandhi, R. V., Thareja, R., and Haftka, R. T. "NEWSUMT-A: A General Purpose Program for Constrained Optimization Using Constraint Approximation," *ASME J. Mechanisms, Transmissions and Automation in Design*, 107, 94–99 (1985).

# Appendix A

## User's Guides for HSCT Software

Reference [39] contains a detailed User's Guide for the software employed in the HSCT analysis and optimization studies conducted by members of our research group. The information provided here details the modifications made to the HSCT analysis and optimization software to facilitate the process of creating and employing response surface models. Note that this software makes use of files written in ANSI standard C, and FORTRAN77 languages. The C language files are denoted with the `.c` filename extension and the FORTRAN files are denoted with the `.f` filename extension. Both C functions and FORTRAN subroutines are identified as `subroutine_name()`, where the symbol `()` denotes function or subroutine arguments.

### A.1 Software Modifications - Five Variable HSCT Optimization Problem

The following files were modified from the baseline HSCT analysis/optimization software, (1) `aero.c`, (2) `dataio.c`, and (3) `main.c`. The following files were added to perform specific functions associated with the creation and use of RS models, (1) `dotcnt1.f`, (2) `ddot1.f` - `ddot6.f`, and (3) `functions_05DV.c`.

### A.1.1 `aero.c`

The only modifications to this file were made in the C function `dragpolar()` where the leading edge suction term was fixed at a constant value of 0.77 rather than using the value calculated in the supersonic drag-due-to-lift analysis.

### A.1.2 `dataio.c`

Modifications were made to the C function `writeDesHistory()` to calculate the mission range using both RS models for the supersonic drag and the original supersonic analysis methods. This information is printed in the output file `design.hist` when optimization is completed.

### A.1.3 `ddot1.f - ddot6.f`

These six files contain the double precision version of the DOT optimization software [52]. No modifications were needed to these files.

### A.1.4 `dotcntl.f`

This file provides an interface between the HSCT analysis software and the DOT optimization software through the FORTRAN subroutine `dotcntl()`. The format for this file was obtained from a sample test case file provided with the DOT software.

### A.1.5 `main.c`

This file contains the most extensive modifications out of all the files used in HSCT analysis and optimization. Three C functions are contained in the file `main.c`. These are (1) `main()` which contains all of the C case choices which perform the various analysis and optimization operations, (2) `analys_()` [sic] which directs the evaluation of the objective function and constraints as needed by the optimizer, and (3) `constraints()` which contains the list of constraints affecting the HSCT geometry and performance. The modifications made to each of these C functions are detailed below.



**main()**

**USER5** - This C case choice was created to calculate the TOGW and constraint values for numerous HSCT configurations. Input data are obtained from the file **hsct\_flags**, in which the level of fidelity of the aerodynamic analyses is specified, and the file **new\_points.dat**, which contains the geometric data for the HSCT configurations to be analyzed. Note that these two input files are used in addition to the files **design.variables**, **hsct.command**, and **wavin**. When employing the case **USER5** the file **design.variables** represents the HSCT configuration at the “center” of the design space. As described in [39], the file **wavin** contains a geometric representation of the HSCT configuration in the Craidon format [77]. Output results are written to the file **cd\_data**.

**EVAL5DV** - This C case choice is similar to the **TEST** choice in the baseline HSCT software. However, **EVAL5DV** calculates both the TOGW and the constraint values for the HSCT configuration given in **design.variables**, with output written to the file **cd\_data**.

**OPT5DV** - This C case choice is an extensively modified version of the **OPTIM** choice. In **OPT5DV** the DOT optimization software is employed rather than the original NEWSUMT-A optimization software [78]. Primarily, the modifications in **OPT5DV** provide a conversion between the 26 variables used in the definition of the HSCT configuration and the five variables used in the optimization problem.

**analys\_()**

This C function was modified to convert the five variables employed by the optimizer to an HSCT configuration defined by 26 variables. Essentially, this is the reverse of the conversion process performed in **OPT5DV**.

**constraints()**

This C function was modified to calculate the 42 constraints used in the five variable HSCT optimization problem. Constraints not needed in this optimization (e.g., constraints relating to subsonic aerodynamic performance) were “turned off” through

the use of flag variables defined at the top of the `constraints()` function. Note that the flag variable `RS_eval_flag` is used to control the type of supersonic aerodynamic analysis methods used in the calculation of the range constraint. If `RS_eval_flag = 1` the RS models for supersonic drag are used, and if `RS_eval_flag = 0` the original supersonic drag evaluation methods are used.

### A.1.6 `functions_05DV.c`

This file contains the C functions (1) `doAnalysis_05DV()` which is used by `USER5` and `EVAL5DV` to evaluate TOGW and the constraints, (2) `Range_3detail05DV()` which calculates the mission range using the RS models for supersonic drag, and (3) `Read_Command_File_05DV()` which reads in the input file `hsct.command`.

### A.1.7 Using the Modified Software in the VCRSM Method - Five Variables

The modifications to the HSCT software are used in several stages of the VCRSM method which is described in Chapter 5. In particular, Step 4 - Low Fidelity Analyses, employs the case `USER5` to evaluate the  $\mathcal{O}(10^3 - 10^4)$  HSCT configurations described by a full factorial experimental design (listed in the input file `new_points.dat`). For the HSCT analyses conducted in Step 4, the values in the input file `hsct_flags` should be set to `SCALED`, and the variable flag `RS_eval_flag` in the function `constraints()` (in `main.c`) should be set to a value of zero.

Step 7, Medium Fidelity Analyses, also employs the case `USER5`. However, in this step, the flags in the input file `hsct_flags` are set to `EXACT`.

In Step 9, HSCT Optimization, the case `OPT5DV` is used. Here, the RS models for the supersonic drag components would be used in the routine `Range_3detail05DV()` (in `functions_05DV.c`) and the flag `RS_eval_flag` in `constraints()` would be set to a value of unity.

## A.2 Software Modifications - Ten Variable HSCT Optimization Problem

Generally, HSCT software modifications needed to perform the ten variable HSCT optimization problem are similar to the modifications made for the five variable HSCT optimization problem. However, there is one significant difference which concerns the implementation and use of the RS models. For the ten variable HSCT design problem the RS models are implemented in distinct C language functions. These subroutines are included in the file `functions_10DV.c` and are discussed below. This change in approach streamlines the interface between the original (unmodified) HSCT software and the RS models.

Software modifications differing from those implemented for the five variable HSCT optimization problem are contained in the files `aero.c` and `main.c`, along with the introduction of a new file `functions_10DV.c` which replaces `functions_05DV.c`. Modifications to these files are described below.

### A.2.1 `aero.c`

In addition to the modification to the C function `dragpolar()` as described in Section A.1.1, modifications were made to the functions `Lift_Drag()`, `Wave_Drag()`, and `lowSpeedCLa()`. In all three of these functions the case `RSMODEL` was created to supplement the existing aerodynamic analysis case types of `EXACT`, `SCALED`, `GLA`, and `LINEAR`. The use of `RSMODEL` is best described in a simple example involving the calculation of the supersonic volumetric wave drag. If the `EXACT` case is specified, the Harris wave drag code is employed, whereas if the `RSMODEL` case is specified, the RS model for wave drag is used. Thus, the name of the case acts as a switch in the `Wave_Drag()` function.

### A.2.2 `main.c`

In the function `main()` the C case choices `EVAL10DV` and `OPT10DV` replace `EVAL5DV` and `OPT5DV` created for the five variable HSCT optimization problem. The function

`analys()` is modified to convert between the ten variables of the optimization problem and the 26 variables of the HSCT geometric definition. Also modified was the function `constraints()` in which seven constraints concerning subsonic aerodynamic performance were activated. This yields a total of 49 constraints for the ten variable HSCT optimization problem.

### A.2.3 `functions_10DV.c`

In this file the function `doAnalysis_10DV()` replaces `doAnalysis_05DV()` and `Read_Command_File_10DV()` replaces `Read_Command_File_05DV()`. In place of the function `Range_3detail05DV`, there is a separate C function for each of the four RS models used in this optimization problem. These functions are `cdwave_10dv()`, `cla_supersonic_10dv()`, `ctcl2_10dv()`, and `cla_subsonic_10dv()`.

### A.2.4 Using the Modified Software in the VCRSM Method - Ten Variables

Using the modified software in the VCRSM method is essentially the same as described above in Section A.1.7. Only step 9 changes where the case `OPT10DV` is employed rather than `OPT5DV`. If the flag variable `RS_eval_flag` in `constraints()` is set to a value of unity, the response surface models in the file `functions_10DV.c` are employed during the optimization process.

## A.3 Software Modifications - DACE Modeling

The DACE modeling methods of [68], [66] and [67] are implemented in the file `dace_eval.f`. This file contains three subroutines `daceinter()`, `correlate()`, and `dace_eval()`. The subroutine `daceinter()` provides an interface between the HSCT analysis software function `Range_3detail05DV()` (in `functions_05DV.c`) and the DACE modeling software. The subroutine `correlate()` calculates all of the terms needed to construct a DACE interpolating model based on the user-supplied data file.

The subroutine `dace_eval()` performs the DACE model evaluation at a user-supplied sample site in the design space. The file `dace_eval.f` is listed below in Section A.3.2.

In addition to `dace_eval.f`, the DACE software employs the BLAS subroutine DGEMM for matrix multiplication along with the LAPACK subroutines `dgefa` and `dgedi`, which are used to calculate the determinant and inverse of the correlation matrix, respectively.

### A.3.1 Using DACE Modeling in the VCRSM Method

The HSCT software which implements the DACE model for  $C_{Dwave}$  in the five variable optimization problem is virtually identical to the software modifications described in Section A.1. The only change is the inclusion of the flag variable `DACEFLAG` in `Range_3detail05DV` which causes the software to use switch between the DACE model (`DACEFLAG=1`) and the polynomial RS model (`DACEFLAG=0`) for  $C_{Dwave}$ .

To use the DACE modeling option in the five variable HSCT optimization problem, the user must supply a data file containing the sample locations and the corresponding response values. Also, the user must supply the terms listed in the `parameter statement` in the FORTRAN subroutine `daceinter()` (in `dace_eval.f`). See the source code listing for `dace_eval.f` in Section A.3.2 for more detail on the required parameters.

### A.3.2 Source Code for `dace_eval.f`

```
c234567
*
*       subroutine daceinter(xnew,ynew)
*
* This is the interface subroutine which connects the HSCT
* analysis software with the DACE modeling software. This
* routine (daceinter) calls the correlation subroutine
* (correlate) to evaluate the correlation matrix and DACE
* model parameters.
*
* Tony Giunta, 12 May 1997
*
*****
*
* Input Variables:
* -----
```

```

*   xnew      = vector of length 'numdv' where the DACE model is
*               to be evaluated
*
* Output Variables:
* -----
*   ynew      = vector of length 'numnew' (usually length = 1) of
*               the DACE model response value(s)
*
* Parameter Variables (to be set by user at run time):
* -----
*   numdv     = number of variables
*   numsamp   = number of data samples from which the correlation
*               matrix and the DACE parameters are calculated
*   numnew    = number of sample site where the unknown response
*               value will be calculated
*   ibeta     = flag variable:
*               = 0 --> calculate 'beta' using from the response
*                   values in the supplied data file
*               = 1 --> use the supplied value of 'beta' specified
*                   in the parameter statement in the
*                   variable 'oldbeta'
*   theta     = user-supplied value of the correlation
*               parameter 'theta'
*   oldbeta   = user-supplied value of 'beta' for use in
*               multi-fidelity (i.e., multi-stage) DACE modeling
*
* Local Variables:
* -----
*   DOUBLE PRECISION
* -----
*   xmat      = numdv x numsamp of sample site locations
*   cormat    = correlation matrix (numsamp x numsamp)
*   invmat    = inverse of the correlation matrix
*               (numsamp x numsamp)
*   Fvect     = matrix (1 x numsamp) of constant terms
*               (all = 1 in 'correlate')
*   FRinv     = matrix product of 'Fvect' and 'invmat'
*   yvect     = matrix (1 x numsamp) of response values
*   yfb_vect  = matrix (1 x numsamp) resulting from
*               ('yvect' - 'Fvect'*'betahat')
*   yfbRinv   = matrix (1 x numsamp) resulting from
*               ('yvect' - 'Fvect'*'betahat')*'invmat'
*   RHStems   = matrix product of 'invmat' and 'yfb_vect'
*   r_xhat    = matrix (1 x numsamp) created by using the
*               vector 'xnew' in the correlation function
*   betahat   = estimate of the constant term in the DACE
*               model (beta)
*   sigmahat  = estimate of the variance (sigma) term in the data
*   MLE       = scalar valued function determined by choice
*               of 'theta' used for comparing to maximum MLE
*               value found from prior analysis of the
*               response data
*   work      = vector of length 'numsamp' used as temporary storage

```

```

*           by the LAPACK subroutine DGEDI
*
*   INTEGER
*   -----
*   iflag    = flag variable:
*             = 1 --> calculate DACE modeling terms and
*                   correlation matrix
*             = 2 --> calculate predicted DACE model response
*                   using previously calculated DACE model
*                   parameters found when iflag=1
*   ipvt     = vector of length 'numsamp' of pivot locations used
*             in LAPACK subroutines DGEDI and DGEFA
*
* Notes:
* 1. User must supply variable values in parameter statement
*    prior to executing this code.
* 2. User must supply the data from which the correlation
*    matrix is calculated in the matrix form
*    (numdv + 1 columns by numsamp rows).
*
*   x(1,1)      ... x(1,numdv)      response_value(1)
*   |           ... |               |
*   |           ... |               |
* x(numsamp,1) ... x(numsamp,numdv) response_value(numsamp)
*
* *****
*
*   parameter( numdv=5, numsamp=3, numnew=1, ibeta=1 )
*   parameter( theta = 671.756d0, oldbeta = 0.00152469d0 )
*
*   double precision xmat(numsamp,numdv), cormat(numsamp,numsamp),
* &   invmat(numsamp,numsamp), Fvect(1,numsamp), FRinv(1,numsamp),
* &   yvect(1,numsamp), yfb_vect(1,numsamp), yfbRinv(1,numsamp),
* &   RHSTerms(1,numsamp), r_xhat(1,numsamp), betahat, sigmahat,
* &   MLE, work(numsamp), xnew(numnew,numdv), ynew(numnew)
*
*   integer i,j,k,ipvt(numsamp),iflag
*
* open data file
*
*   open(21,file='dopt_data_6col_cdwave.dat')
*
* initialize the DACE modeling parameters
*
*   betahat = 0.0d0
*   MLE     = 0.0d0
*   sigmahat = 0.0d0
*
* read in matrix of data points locations in (numsamp by numdv
* matrix) and the corresponding response values

```

```

*
  do 100 i=1,numsamp
    read (21,*) (xmat(i,j),j= 1, numdv), yvect(1,i)
100 continue
  close(21)

*
* initialize y_hat(x) and select x_new, i.e., an unsampled location
* where Y(x) is to be predicted
*
  do 130 i = 1,numnew
    ynew(i) = 0.0d0
130 continue

*
* initialize all other vectors and matrices
*
  do 200 i=1,numsamp
    do 210 j = 1,numsamp
      cormat(i,j) = 0.0d0
      invmat(i,j) = 0.0d0
210   continue
200 continue

  do 220 i=1,numsamp
    Fvect(1,i) = 0.0d0
    FRinv(1,i) = 0.0d0
    yfb_vect(1,i) = 0.0d0
    yfbRinv(1,i) = 0.0d0
    r_xhat(1,i) = 0.0d0
    work(i) = 0.0d0
    RHStems(1,i) = 0.0d0
    ipvt(i) = 0
220 continue

*
* call subroutine to calculate the inverse of the correlation
* matrix and the correlation parameters
*
* iflag = 1, for the given 'theta' find the inverse correlation
* matrix
* iflag = 2, evaluate Y(x), given x, theta, inverse correlation
* matrix, etc.
*
*
* ibeta = 1, use user supplied value of 'betahat' in DACE model
* ibeta = 0, use the calculated value of 'betahat' from data
*
  iflag = 1
  if( ibeta .eq. 1 ) then
    call correlate (iflag,ibeta,xmat,cormat,
&      invmat,Fvect,FRinv,
&      yvect,yfb_vect,yfbRinv,
&      RHStems,r_xhat,

```



```

&      numsamp, numdv, numnew,
&      oldbeta, sigmahat, MLE,
&      work, theta, ipvt, xnew, ynew )
  else
    call correlate (iflag, ibeta, xmat, cormat,
&      invmat, Fvect, FRinv,
&      yvect, yfb_vect, yfbRinv,
&      RHStems, r_xhat,
&      numsamp, numdv, numnew,
&      betahat, sigmahat, MLE,
&      work, theta, ipvt, xnew, ynew )
  endif

  iflag = 2
  if( ibeta .eq. 1 ) then
    call correlate (iflag, ibeta, xmat, cormat,
&      invmat, Fvect, FRinv,
&      yvect, yfb_vect, yfbRinv,
&      RHStems, r_xhat,
&      numsamp, numdv, numnew,
&      oldbeta, sigmahat, MLE,
&      work, theta, ipvt, xnew, ynew )
  else
    call correlate (iflag, ibeta, xmat, cormat,
&      invmat, Fvect, FRinv,
&      yvect, yfb_vect, yfbRinv,
&      RHStems, r_xhat,
&      numsamp, numdv, numnew,
&      betahat, sigmahat, MLE,
&      work, theta, ipvt, xnew, ynew )
  endif

  return
end

*
*****
*
  subroutine correlate (iflag, ibeta, xmat, cormat,
&      invmat, Fvect, FRinv, yvect, yfb_vect, yfbRinv,
&      RHStems, r_xhat, numsamp, numdv, numnew,
&      betahat, sigmahat, MLE, work, theta, ipvt, xnew, ynew )
*
*
* This subroutine calculates the DACE correlation matrix
* using an exponential correlation function (see Boeing/IBM/Rice
* Report) along with a user-supplied value of the correlation
* parameter 'theta.'
*
* Tony Giunta, 12 May 1997
*
*****
*
* Needed External Files:
* -----

```

```

*   LAPACK subroutines DGEFA and DGEDI (FORTRAN77)
*   BLAS subroutine DGEMM (FORTRAN77)
*
* Inputs: (variables defined in subroutine 'daceinter()')
* -----
*   iflag
*   ibeta
*   xmat
*   cormat
*   Fvect
*   FRinv
*   yvect
*   yfb_vect
*   yfbRinv
*   RHStems
*   r_xhat
*   numsamp
*   numdv
*   numnew
*   sigmahat
*   MLE
*   work
*   ipvt
*   xnew
*   ynew
*
* Outputs: (variables defined in subroutine 'daceinter()')
* -----
*   invmat
*   betahat
*   theta
*
*****
*
*   integer numdv,numsamp,numnew,i,j,k,ipvt(numsamp),
*   &   iflag,ibeta
*
*   double precision xmat(numsamp,numdv),
*   &   cormat(numsamp,numsamp),
*   &   invmat(numsamp,numsamp),Fvect(1,numsamp),FRinv(1,numsamp),
*   &   yvect(1,numsamp),beta_num,beta_den,betahat,sigmahat,MLE,
*   &   yfb_vect(1,numsamp),yfbRinv(1,numsamp),
*   &   work(numsamp),RHStems(1,numsamp),
*   &   xnew(numnew,numdv), r_xhat(1,numsamp), theta,
*   &   det(2),detR,sum,delta,ynew(numnew)
*
* if IFLAG=1 find correlation matrix, its inverse, and
* correlation parameters
*
*   if( iflag .eq. 1 ) then

```

```

*
* calculate terms in the correlation matrix
*
      do 200 i = 1,numsamp
        do 210 j = i,numsamp
          if( i .eq. j ) then
            cormat(i,j) = 1.0d0
            invmat(i,j) = 1.0d0
          else
            sum = 0.0d0
            do 220 k = 1,numdv
              sum = sum +
&                (xmat(i,k)-xmat(j,k))*(xmat(i,k)-xmat(j,k))
220            continue
            cormat(i,j) = exp( -1.0d0*theta*sum )
            cormat(j,i) = cormat(i,j)
            invmat(i,j) = cormat(i,j)
            invmat(j,i) = cormat(i,j)
          endif
210        continue
200      continue
*
* calculate the determinant of the correlation matrix
*
      call dgefa( invmat, numsamp, numsamp, ipvt, info)
      if( info .ne. 0 ) then
        write(*,*)"Error in DGEFA, info = ",info
        stop
      endif
*
* Note: in DGEDI the last flag is: 1 (inverse only),
* 10 (Det only), 11 (both)
*
      iflag = 11
      call dgedi( invmat, numsamp, numsamp, ipvt, det,
&              work, iflag)
      detR = det(1) * 10.0d0**det(2)
*
* calculate the terms needed to determine the MLE value
*
      do 300 i = 1,numsamp
        Fvect(1,i) = 1.0d0
300      continue
*
      call DGEMM('n','n',1,numsamp,numsamp,1.0d0,Fvect,1,
&              invmat,numsamp,0.0d0,FRinv,1)
      call DGEMM('n','n',1,1,numsamp,1.0d0,FRinv,1,Fvect,
&              numsamp,0.0d0,beta_den,1)
      call DGEMM('n','n',1,1,numsamp,1.0d0,FRinv,1,yvect,

```

```

        &      numsamp,0.0d0,beta_num,1)
*
* calculate beta if flag 'ibeta' is set to zero, otherwise
* use the supplied value of beta
*
      if( ibeta .eq. 0 ) betahat = beta_num / beta_den
      do 310 i = 1,numsamp
        yfb_vect(1,i) = yvect(1,i) - betahat*Fvect(1,i)
310    continue

      call DGEMM('n','n',1,numsamp,numsamp,1.0d0,yfb_vect,1,
&      invmat, numsamp,0.0d0,yfbRinv,1)
      call DGEMM('n','n',1,1,numsamp,1.0d0,yfbRinv,1,yfb_vect,
&      numsamp,0.0d0,sigmahat,1)
      sigmahat = sigmahat / numsamp

      MLE = -0.5d0*(numsamp*log(sigmahat) + log(detR))
*
      endif
*
* if IFLAG=2 estimate Y(x) at an unsampled location 'x'
*
      if( iflag .eq. 2 ) then
        call DGEMM('n','n',numsamp,1,numsamp,1.0d0,invmat,
&      numsamp,yfb_vect,numsamp,0.0d0,RHStems,numsamp)
        call dace_eval(xnew,xmat,r_xhat,betahat,RHStems,
&      numsamp,numdv,numnew,theta,ynew)
      endif

      return
      end

*****
*
      subroutine dace_eval(xnew,xmat,r_xhat,betahat,RHStems,
&      numsamp,numdv,numnew,theta,ynew)
*
*
* Use DACE interpolating model to predict response values at
* unsampled locations.
*
* Tony Giunta, 12 May 1997
*
*****

```

```

*
* Inputs: (variables defined in subroutine 'daceinter()')
* -----
*   xnew
*   xmat
*   r_xhat
*   betahat
*   RHStems
*   numsamp
*   numdv
*   theta
*
* Outputs: (variables defined in subroutine 'daceinter()')
* -----
*   ynew
*
* Local Variables:
* -----
*   sum   = temporary variable used for calculating the
*           terms in the vector 'r_xhat'
*   yeval = scalar value resulting from matrix multiplication
*           of 'r_xhat' * 'RHStems'
*
*****
*
*   double precision xnew(numnew,numdv),r_xhat(1,numsamp),
*   &   xmat(numsamp,numdv),RHStems(1,numsamp),betahat,
*   &   theta,sum,ynew(numnew),yeval
*
*   integer i,j,k,numdv,numsamp,numnew
*
* calculate the vector r(x)
*
*   do 200 i = 1,numnew
*
*       do 110 j = 1,numsamp
*           sum = 0.0d0
*           do 120 k = 1,numdv
*               sum = sum +
*   &               (xnew(i,k)-xmat(j,k))*
*   &               (xnew(i,k)-xmat(j,k))
*       120   continue
*           r_xhat(1,j) = exp( -1.0d0*theta*sum )
*       110   continue
*
* calculate the estimate of Y, i.e., Y_hat(x)
*
*       yeval = 0.0d0
*       call DGEMM('n','n',1,1,numsamp,1.0d0,r_xhat,1,
*   &           RHStems,numsamp,0.0d0,yeval,1)
*
*       ynew(i) = yeval + betahat
*
*   200 continue

```

```
return  
end
```

# Appendix B

## Error Estimation Methods

In Chapter 7 the notation  $\hat{y}_{n_e}$  was introduced to define the predicted response values from a polynomial RS model or a DACE model at  $n_e$  locations where the true response,  $y_{n_e}$ , is known. The discrepancy between  $y_{n_e}$  and  $\hat{y}_{n_e}$  is known as the modeling error and is defined as

$$\delta_i = |\hat{y}_i - y_i|, \quad (\text{B.1})$$

for  $i = 1, \dots, n_e$ . Various statistical metrics are available for quantifying these modeling errors. These include the mean modeling error

$$\bar{\delta} = \frac{1}{n_e} \sum_{i=1}^{n_e} \delta_i, \quad (\text{B.2})$$

the standard deviation of the modeling error

$$\sigma_\delta = \sqrt{\frac{\sum_{i=1}^{n_e} (\delta_i - \bar{\delta})^2}{n_e - 1}}, \quad (\text{B.3})$$

and the median modeling error,  $\delta_{median}$ , which is defined as the midpoint value of the series  $\delta_k$  in which the values for  $\delta_i$  are ordered such that  $\delta_1 \leq \delta_2 \leq \dots \leq \delta_k \leq \dots \leq \delta_{n_e}$ . Also of use is the maximum modeling error which is defined as

$$\delta_{max} = \max(\delta_i). \quad (\text{B.4})$$

In addition to these metrics, the root mean squared modeling error is

$$\text{RMS} = \sqrt{\frac{\sum_{i=1}^{n_e} \delta_i^2}{n_e}}. \quad (\text{B.5})$$

If the  $n_e$  locations are not the same as the sample sites,  $n_s$ , used to construct the polynomial response surface model, then Equation B.5 is an *unbiased* estimator of the RMS modeling error and is identified as  $\text{RMS}_{\text{ub}}$ . However, if  $n_e$  and  $n_s$  are the same, then Equation B.5 is *biased* and it underestimates the true error. When  $n_e$  and  $n_s$  are the same the unbiased RMS error must be calculated using

$$\text{RMS}_{\text{ub}} = \sqrt{\frac{\sum_{i=1}^{n_e} \delta_i^2}{n_e - n_t}}, \quad (\text{B.6})$$

where  $n_t$  is the number of terms in the polynomial model. See Myers and Montgomery [14, page 26] for more information on unbiased estimators.



# Vita

Anthony A. Giunta was born on April 9, 1970, in Salisbury, Maryland. In 1993 he completed the B.S. degree, *magna cum laude*, in aerospace engineering with a minor in mathematics from Virginia Polytechnic Institute and State University in Blacksburg, Virginia. While an undergraduate, he worked as a co-operative education student at NASA Langley Research Center in Hampton, Virginia. Continuing his education he earned the M.S. degree in aerospace engineering in 1995 and the Ph.D. degree in aerospace engineering in 1997, both from Virginia Polytechnic Institute and State University. Currently he holds a National Research Council postdoctoral fellowship at NASA Langley Research Center.