

A Design Space Exploration Process for Large Scale, Multi-objective Computer Simulations

A Thesis
Presented to
The Academic Faculty

by

John Marc Zentner

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Aerospace Engineering
Georgia Institute of Technology
August 2006

A Design Space Exploration Process for Large Scale, Multi-objective Computer Simulations

Approved by:

Prof. Dimitri Mavris
Committee Chair
School of Aerospace Engineering
Georgia Institute of Technology

Prof. Daniel Schrage
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Vitali Volovoi
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Viren Kumar
Principal Engineer - Engineering Product Development
EED Services Engineering
GE Energy Power Generation Technology

Mr. Gene E. Wiggs
Principle Engineer - Probabilistics
Advanced Technology & Preliminary Design Department
General Electric Aviation Engineering Division

Date Approved: 3 July 2006

As this may be the only book that I ever write,

I dedicate this thesis to my mom.

ACKNOWLEDGEMENTS

It's interesting, after writing some 200+ pages for a thesis, I find myself with writers block on the one thing that should be about the easiest part to write, the acknowledgments. It's not that I don't have anyone that I'd like to thank but rather there are so many, I don't even know where to begin!

I guess the first place to start is by thanking the one man who made my whole graduate experience possible, Dr. Dimitri Mavris. Without his willingness to give an average mechanical engineering student from a "third tier" school a chance, I wouldn't be writing this now. Doc, thank you very much for your timely advisement, mentorship and most of all, your patience. The past seven years have truly been the best and most enlightening years of my life.

The second person that deserves more thanks that I can possibly give is Dr. Vitali Volovoi. Dr. Volovoi helped guide me through countless obstacles and road blocks that always arise when doing research, and consequently, things seem to have turned out alright. Thank you, Vitali.

I also need to thank Dr. Viren Kumar at General Electric Power Systems for posing the initial idea to use a partitioned, hierarchical metamodel to analyze the first stage nozzle problem. I'm just happy that I was actually able to bring your idea to complete fruition; thank you, Viren.

I also would like to thank Mr. Gene Wiggs at General Electric Aircraft Engines for sitting on my committee and supplying me with the torsional-vibration system test problem. Gene, your comments were right on the money and made my thesis a more focused document, thank you very.

I also need to thank many of the very good people at the Aerospace Systems Design

Laboratory at Georgia Tech for their help in reading and editing of my thesis and listening to my crack pot ideas and actually thinking they were pretty good! Those excellent people include Drs to-be Frederic Villeneuve, Stephane Dufresne, Simon Brineno, Henry Won, Mandy Goltsch, Brian German, Tommer Ender and Drs Neil Weston and Peter Hollingsworth. Guys, you actually make coming to work everyday enjoyable; thank you for all your help.

I have one last thank you that I need to make, and this thank you is probably the most important. Without Janel Nixon's support and patience over the past six years, I don't think that I would have been able to finish. Having someone to come home to that understands what you are going through when you are trying to finish a Ph.D. dissertation is a benefit that can't be over stated. Janel, thank you very, very much, and I'll be there for you as you finish yours.

Jack Zentner

Atlanta, GA

Contents

DEDICATION	iii
ACKNOWLEDGEMENTS	iv
LIST OF TABLES	x
LIST OF FIGURES	xi
SUMMARY	xv
I INTRODUCTION	1
1.1 The GE H Machine and Enabling Technologies Overview	2
1.1.1 Closed-loop Steam-cooling	4
1.1.2 Single Crystal Materials	6
1.1.3 Dry, Low NOx Combustors	6
1.2 First Stage Nozzle Overview	8
1.3 Operating Conditions and Nozzle Life	10
1.4 Assessing Nozzle Life	12
1.5 A Proposed Lifting Methodology	17
1.6 Initial Research Questions and Hypotheses Review	25
1.7 Outline of the Thesis	27
1.8 Summary	29
II REVIEW OF PROBABILISTIC DESIGN METHODS	32
2.1 Direct Monte Carlo Simulation	33
2.1.1 Traditional Monte Carlo Sampling	33
2.1.2 Stratified Sampling	35
2.1.3 Quasi-Monte Carlo Sampling	36
2.1.4 Adaptive-Importance Sampling	36
2.2 Analytic Reliability Methods	37
2.2.1 First Order Reliability Method	38
2.2.2 Second Order Reliability Method	43

2.3	Indirect Monte Carlo Simulation	44
2.4	Discussion of Probabilistic Design Methods	48
2.5	Summary	55
III	REVIEW OF LARGE SCALE METAMODELING METHODS	57
3.1	Definition of the Large Scale Problem	58
3.2	Non-partitioned Approaches	61
3.2.1	Pre-built Designs	63
3.2.2	Sequential Designs	64
3.3	Partitioned Approaches	65
3.3.1	Non-hierarchical Partitioning	67
3.3.2	Hierarchical Partitioning	71
3.4	Review and Discussion of Current LSM Methods	77
3.5	Formulation of a New LSM Method	81
3.6	Summary	87
IV	REVIEW OF SYSTEM PARTITIONING METHODS	89
4.1	Diakoptics	90
4.2	Overview of Sparse Matrix Methods	92
4.3	Generalized Method of Systems Decomposition	97
4.4	Graph Theoretic Methods for Partitioning	99
4.5	Extending Graph Theory to Metamodeling	104
4.6	Summary	108
V	INTERIM SUMMARY	110
5.1	Review of the Previous Four Chapters	110
5.2	Summary of Research Questions and Hypotheses	116
VI	THE HIERARCHICAL DESIGN SPACE EXPLORATION PROCESS	123
6.1	Development of the Lumped Parameter Model	123

6.2	Development of the Graph-based Partitioning Method	135
6.3	The Hierarchical Design Space Exploration Process	138
6.4	Validation Plan	142
6.4.1	For the Lumped Parameter Metamodeling Process	142
6.4.2	For the Graph-based Partitioning Method	144
6.4.3	The Validation Test Cases	146
6.5	Summary	147
VII FORMULATION OF A PARALLEL, DISTRIBUTED COMPUTING ENVIRONMENT		150
7.1	Need for Integrating Frameworks	150
7.2	History of Integrating Frameworks	153
7.2.1	Hard-Coded Algorithms	153
7.2.2	Commercial Applications	153
7.2.3	Open-Source Applications	155
7.3	Formulation of an Open-Source Alternative	156
7.3.1	Enabling Technologies	157
7.4	Framework Design and Components	160
7.4.1	The Client Side API	162
7.4.2	The Server Side API	166
7.5	Summary	170
VIII VALIDATING THE HIERARCHICAL DESIGN SPACE EXPLORATION PROCESS		172
8.1	Test Case 1	172
8.1.1	The Hollow I-beam Test Model	172
8.1.2	Applying the Hi-DSE Process to the I-beam Problem	173
8.2	Test Case 2	185
8.2.1	The Torsional Vibration System Test Problem	185
8.2.2	Applying the Hi-DSE Process to the TVS Problem	188
8.3	Discussion of the Test Case Results	192

8.3.1	Validating Hypothesis 2.1	192
8.3.2	Validating Hypotheses 3.1 and 3.2	199
IX	CONCLUSIONS AND OBSERVATIONS	203
9.1	Review of Research Questions and Hypotheses	203
9.2	Contributions of this Thesis	210
9.3	Conclusions and Observations	212
9.4	Closing Remarks	215
Appendix A	— I-BEAM TEST CASE RESULTS	219
Appendix B	— TVS TEST CASE RESULTS	272
Appendix C	— THE HOLLOW I-BEAM ANSYS MACRO	277
	REFERENCES	302

List of Tables

Table 1	Ranking Matrix for Probabilistic Design Methods	51
Table 2	Summary of UTE Variables per Partition	84
Table 3	Benchmarked Run Estimates for New Hypothesized Method	84
Table 4	I-beam Variable Definitions and Ranges	174
Table 5	Variable-type and Subsystem Partitions for I-beam Model	181
Table 6	I-beam Model Error Summary	184
Table 7	TVS Model Variable and Range Definitions	190
Table 8	TVS Model Variable-type and Subsystem Partitions	191
Table 9	TVS Model Error Analysis Summary	193
Table 10	Zones 1, 2 and 5 Stress Response Error Summary	228
Table 11	Zones 6, 7 and 8 Stress Response Error Summary	229
Table 12	Zones 1, 2 and 5 Temperature Response Error Summary	230
Table 13	Zones 6, 7 and 8 Temperature Response Error Summary	231

List of Figures

Figure 1	General Electric H Machine [40]	3
Figure 2	Steam-cooling versus Film Cooling [15]	5
Figure 3	First Stage Nozzle CAD Drawing and H Machine Flow Path[21] . .	9
Figure 4	Meshed First Stage Nozzle	15
Figure 5	Nozzle FEA Model Divided into Geographic Zones	19
Figure 6	Projected Runs per Partition Size	24
Figure 7	Taxonomy for Large Scale Metamodeling Techniques	58
Figure 8	Generic Hierarchical Representation	72
Figure 9	Breguet Range Hierarchy	75
Figure 10	Example Sparse Matrix [186]	94
Figure 11	Incidence Matrix of Equation Set 25	95
Figure 12	Reordered Incidence Matrix of Equation Set 25	96
Figure 13	Reordered Incidence Matrix with Feedback	97
Figure 14	Incidence Matrix and Its Graph [138]	100
Figure 15	Graph of Incidence Matrix in Figure 11	101
Figure 16	Graph of Incidence Matrix with Feedback	102
Figure 17	Generic Hierarchical Representation	125
Figure 18	Generic Functional Mapping [138]	126
Figure 19	Generic Composite Function Mapping	127
Figure 20	Hierarchical Mapping in Composite Function Terms	128
Figure 21	Functional Mappings for the Lumped Parameter Model	130
Figure 22	Creation of \tilde{Y} Regression Array	132
Figure 23	Hierarchical Design Space Exploration Process	149
Figure 24	Notional Agent Configuration	160
Figure 25	Client Side UML Diagram	163
Figure 26	Server Side UML Diagram	164
Figure 27	The Hollow I-beam Test Model	174

Figure 28	I-beam System Graph and Matrix Plot	176
Figure 29	Interactions Missed and Runs Saved per Partition Count	177
Figure 30	Partitioned Sparse Matrix Plots	179
Figure 31	Torsional Vibration System Test Model	186
Figure 32	TVS Model True Graph and Matrix Plot	189
Figure 33	TVS Model Experimentally Determined Graph and Matrix Plot . .	190
Figure 34	I-beam Model Response Clusters	195
Figure 35	Complete Graph Results	220
Figure 36	Mincut-2 Results	221
Figure 37	Mincut-3 Results	222
Figure 38	Mincut-4 Results	223
Figure 39	Mincut-5 Results	224
Figure 40	Mincut-6 Results	225
Figure 41	Mincut-7 Results	226
Figure 42	All-at-once Min. Run Res. V CCD Error Distributions - Stresses Zone 1, 2, 5	232
Figure 43	All-at-once Min. Run Res. V CCD Error Distributions - Stresses Zone 6, 7, 8	233
Figure 44	All-at-once Min. Run Res. V CCD Error Distributions - Tempera- tures Zone 1, 2, 5	234
Figure 45	All-at-once Min. Run Res. V CCD Error Distributions - Tempera- tures Zone 6, 7, 8	235
Figure 46	All-at-once Optimal Latin Hypercube Error Distributions - Stresses Zone 1, 2, 5	236
Figure 47	All-at-once Optimal Latin Hypercube Error Distributions - Stresses Zone 6, 7, 8	237
Figure 48	All-at-once Optimal Latin Hypercube Error Distributions - Temper- atures Zone 1, 2, 5	238
Figure 49	All-at-once Optimal Latin Hypercube Error Distributions - Temper- atures Zone 6, 7, 8	239
Figure 50	Variable-type Partitioning Error Distributions - Stresses Zone 1, 2, 5	240

Figure 51	Variable-type Partitioning Error Distributions - Stresses Zone 6, 7, 8	241
Figure 52	Variable-type Partitioning Error Distributions - Temperatures Zone 1, 2, 5	242
Figure 53	Variable-type Partitioning Error Distributions- Temperatures Zone 6, 7, 8	243
Figure 54	Subsystem Partitioning Error Distributions - Stresses Zone 1, 2, 5 .	244
Figure 55	Subsystem Partitioning Error Distributions - Stresses Zone 6, 7, 8 .	245
Figure 56	Subsystem Partitioning Error Distributions - Temperatures Zone 1, 2, 5	246
Figure 57	Subsystem Partitioning Error Distributions - Temperatures Zone 1, 2, 5	247
Figure 58	Mincut-2 Partitioning Error Distributions - Stresses Zone 1, 2, 5 . .	248
Figure 59	Mincut-2 Partitioning Error Distributions - Stresses Zone 6, 7, 8 . .	249
Figure 60	Mincut-2 Partitioning Error Distributions - Temperatures Zone 1, 2, 5	250
Figure 61	Mincut-2 Partitioning Error Distributions - Temperatures Zone 6, 7, 8	251
Figure 62	Mincut-3 Partitioning Error Distributions - Stresses Zone 1, 2, 5 . .	252
Figure 63	Mincut-3 Partitioning Error Distributions - Stresses Zone 6, 7, 8 . .	253
Figure 64	Mincut-3 Partitioning Error Distributions - Temperatures Zone 1, 2, 5	254
Figure 65	Mincut-3 Partitioning Error Distributions - Temperatures Zone 6, 7, 8	255
Figure 66	Mincut-4 Partitioning Error Distributions - Stresses Zone 1, 2, 5 . .	256
Figure 67	Mincut-4 Partitioning Error Distributions - Stresses Zone 6, 7, 8 . .	257
Figure 68	Mincut-4 Partitioning Error Distributions - Temperatures Zone 1, 2, 5	258
Figure 69	Mincut-4 Partitioning Error Distributions - Temperatures Zone 6, 7, 8	259
Figure 70	Mincut-5 Partitioning Error Distributions - Stresses Zone 1, 2, 5 . .	260
Figure 71	Mincut-5 Partitioning Error Distributions - Stresses Zone 6, 7, 8 . .	261
Figure 72	Mincut-5 Partitioning Error Distributions - Temperatures Zone 1, 2, 5	262
Figure 73	Mincut-5 Partitioning Error Distributions - Temperatures Zone 6, 7, 8	263
Figure 74	Mincut-6 Partitioning Error Distributions - Stresses Zone 1, 2, 5 . .	264
Figure 75	Mincut-6 Partitioning Error Distributions - Stresses Zone 6, 7, 8 . .	265
Figure 76	Mincut-6 Partitioning Error Distributions - Temperatures Zone 1, 2, 5	266

Figure 77	Mincut-6 Partitioning Error Distributions - Temperatures Zone 6, 7, 8	267
Figure 78	Mincut-7 Partitioning Error Distributions - Stresses Zone 1, 2, 5 . . .	268
Figure 79	Mincut-7 Partitioning Error Distributions - Stresses Zone 6, 7, 8 . . .	269
Figure 80	Mincut-7 Partitioning Error Distributions - Temperatures Zone 1, 2, 5	270
Figure 81	Mincut-7 Partitioning Error Distributions - Temperatures Zone 6, 7, 8	271
Figure 82	All-at-once Min. Run Res. V CCD Error Distributions	273
Figure 83	All-at-once Optimal Latin Hypercube Error Distributions	274
Figure 84	Variable-type Partitioning Error Distributions	275
Figure 85	Subsystem Partitioning Error Distributions	276

SUMMARY

The primary contributions of this thesis are associated with the development of a new method for exploring the relationships between inputs and outputs for large scale computer simulations. Primarily, the proposed design space exploration procedure uses a hierarchical partitioning method to help mitigate the curse of dimensionality often associated with the analysis of large scale systems. Closely coupled with the use of a partitioning approach, is the problem of how to partition the system. This thesis also introduces and discusses a quantitative method developed to aid the user in finding a set of good partitions for creating partitioned metamodels of large scale systems.

The high level motivation for the thesis was tied to the task of finding or developing a method for enabling probabilistic lifing of the first stage nozzle in a gas turbine engine. To accurately assess the life of the nozzle, a complex finite element analysis was created that mimicked the spatial variability of the operating environment. The large scale, complex nature of this model exposed several limitations to current classes of methods typically used to assess the probabilistic nature of the life of gas turbine parts. The proposed lifing methodology was an indirect Monte Carlo simulation technique that combined metamodels of the system with traditional Monte Carlo simulation to assess the probabilistic nature of the nozzle life.

Unfortunately, metamodels based on traditional statistical linear models were not deemed efficient enough to use with the large scale nozzle model. Thus it was hypothesized that the spatially partitioned nature of the nozzle model could be leveraged to create partitioned metamodels of the system.

Partitioned model building allows a model builder to create metamodels of large

scale systems with significantly fewer runs than traditional statistical linear model based metamodeling schemes. The reason being is that the majority of the data required for traditional linear model based metamodels is used to quantify the variability on the response due to interactions between main factors.

A common assumption in statistical model building is that of effect sparsity. Effect sparsity implies that not all potential effects present in a statistical model are actually important to the variability to the response. Thus a partitioned metamodeling scheme realizes the sparsity of the effects by capturing important interactions between variables in a variable subgroup or partition and neglects the potential interactions between variables in disjoint subgroups.

The new hierarchically partitioned metamodeling scheme, the lumped parameter model (LPM), was developed to address two primary limitations to the current partitioning methods for large scale metamodeling. First the LPM was formulated to negate the need to rely on variable redundancies between partitions to account for potentially important interactions. While variable redundancies improve the accuracy of the partitioned metamodel, they do so at the cost of significantly reducing the efficiency of the partitioned metamodeling process. By using a hierarchical structure, the LPM addresses the impact of neglected, direct interactions by indirectly accounting for these interactions via the interactions that occur between the lumped parameters in intermediate to top-level mappings.

Secondly, the LPM was developed to allow for hierarchical modeling of black-box analyses that do not have available intermediaries with which to partition the system around. Current hierarchical metamodeling methods require that the parent analysis have intermediate responses or variables that the user can specify directly as well as use the analysis' lower-level design variables to calculate. For systems that do not have this level of transparency, hierarchical metamodeling was previously not possible. Consequently the LPM enables hierarchical metamodeling of these types of

black-box analyses.

The second contribution of this thesis is a graph-based partitioning method for large scale, black-box systems. When using a partitioned metamodeling method, the very pertinent question of how to partition the system always arises. The current methods used to address this problem rely on either engineering intuition to partition the system or to partition the system along code or process boundaries. When there is initially little known about the system, determining how to partition based on engineering intuition can be unreliable at best. Consequently having a method that is able to quantitatively guide the partitioning process using a small sample of data from the system would be very useful. The second contribution for this thesis does just that.

The graph-based partitioning method combines the graph and sparse matrix decomposition methods used by the electrical engineering community with the results of a screening test to create a quantitative method for partitioning large scale, black-box systems. An ANOVA analysis of the results of a screening test can be used to determine the sparse nature of the large scale system. With this information known, the sparse matrix and graph theoretic partitioning schemes can then be used to create potential sets of partitions to use with the lumped parameter model.

To validate the lumped parameter modeling method and the graph-based partitioning method, two test cases were developed. Unfortunately, the actual nozzle model was not a good model to use with the validation process due to its very long runtime. Thus, the first test case developed for validation was a finite element model of a parametric hollow I-beam. The I-beam model was created to be a medium fidelity stand-in for the motivating nozzle model. It was formulated in as similar a manner as possible to the nozzle model to capture the salient features while at same time having a considerably shorter execution time.

While the first test case was chosen as a stand-in for the nozzle model, the second

test case was chosen to provide a contrast to the black-box nature of models based on finite element analyses. The second test case was a set of equations that described the behavior of a torsional-vibration system. The primary goal of this test case was not necessarily to show how good or bad the lumped parameter metamodels of the system are but rather to demonstrate a system that is not a good candidate for partitioned metamodeling. In addition, since the equations for the system are completely known, the true graph of the system can easily be created and compared to the experimental graph created with the graph-based partitioning method.

The results of the validation process show that for systems which are good candidates for partitioning, such as the I-beam problem, partitioned metamodels built on the lumped parameter model can be quite good. The LPMs created for the I-beam problem were as accurate or more accurate than metamodels created using traditional statistical linear models. In addition LPM based models required approximately half the data to create as compared to traditional statistical linear models.

Chapter I

INTRODUCTION

As this thesis title implies, the work to be introduced and discussed revolves around the development of a method for exploring the relationships between inputs and outputs for large scale computer simulations. Primarily, the proposed design space exploration procedure uses a hierarchical partitioning method to help mitigate the “curse of dimensionality” often associated with the analysis of large scale systems. Closely coupled with the use of a partitioning approach, is the problem of how to partition the system. This thesis will also introduce and discuss a quantitative method developed to aid the user in finding a set of good partitions for a large scale system.

Although the research contribution is intended to stand on its own and be applicable for a wide range of large scale systems, rarely is the research for an engineering thesis performed in a vacuum. This thesis is no different. In actuality all of the research completed can be directly tied to the following motivation: *Find or develop a method that facilitates a means to accurately and efficiently assess the probabilistic life of a first stage nozzle in General Electric’s new H machine gas turbine power plant.*

To see how that motivating statement and the contribution of this thesis are connected, it will be beneficial to start at the beginning and layout the prior development that ultimately lead to the motivation. In fact, dissecting the motivating statement foreshadows the flow of the discussion of the next few sections. If the reader were to essentially read the statement backwards, the order of topics to be covered would be pretty much laid out.

To a first order, the contribution of this thesis is directly tied to the development

of the new H machine by General Electric Energy. Thus to make the connection, this chapter will first start by giving a brief overview of the H machine, why it was developed, what were its primary design goals and what technologies enabled the realization of these design goals.

As we will see, it was the use of the enabling technologies that ultimately spurred the motivating statement that drove the work which forms the core of this thesis. The enabling technologies on the new H machine create a highly variable and destructive operating environment for the first stage nozzle. This operating environment has a direct negative impact on the life of the first stage nozzle. In addition, degradation of nozzle life can be directly linked to the economics of operating a gas turbine engine.

To accurately assess the life of the nozzle, a complex finite element analysis was created that mimicked the spatial variability of the operating environment. The large scale, complex nature of this model exposed numerous limitations to current classes of methods typically used to assess the probabilistic nature of the life of gas turbine parts. From the analysis of these limitations, the actual research motivation of this thesis will be identified which will drive the development of the hierarchical partitioning scheme and the associated method for finding a set of good partitions for large scale systems.

1.1 The GE H Machine and Enabling Technologies Overview

GE's new H series of gas turbine power plants were designed to be the first combined cycle power plants to break the 60% efficiency barrier. The 60% efficiency is considered to be a true milestone for combined cycle power plants [169]. As a comparison, the previous best in class combined cycle power plants, GE's FA machine [40] and Siemens's 6000G [152], topped out at about 57 to 58% efficiency. Although a 2% increase in efficiency does not seem like much, its effect amortized over the life of

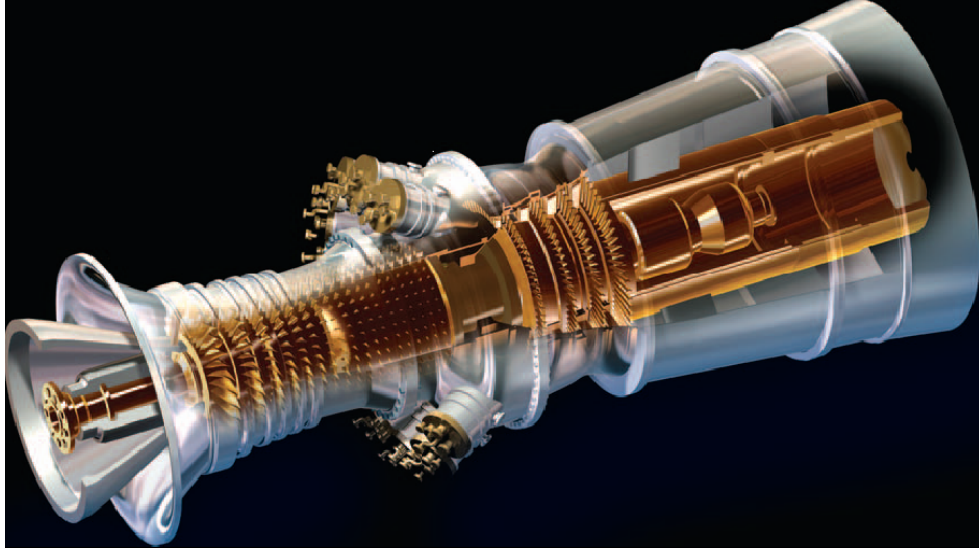


Figure 1: General Electric H Machine [40]

power plant can equate to significant cost saving for the power plant operator. This saving is realized because for gas turbine power plants, fuel usage is single largest direct operating cost [96].

A combined cycle power plant is a system that integrates a gas turbine engine, a steam turbine and a heat recovery steam generator into one package. The combination of these three systems allows the power plant to utilize more of the energy liberated from the fuel. In a typical engine, the majority of the heat generated by burning the fuel is exhausted from the engine as waste heat. However, in a combined cycle power plant the hot exhaust gases exiting the engine are used to create steam in the heat recovery steam generator. This steam is then expanded through a steam turbine to produce additional energy.

The GE engineers are able to reach the remarkable 60% efficiency level by incorporating three primary technologies into the design of the H machine: closed-loop steam-cooling, single crystal materials, and dry low NOx combustors [39]. Figure 1 shows a rendering of the H machine. As it will be discussed, each of these technologies affect the life of the first stage nozzles in different and sometimes detrimental ways.

Of the three technologies that facilitate the reaching of the 60% efficiency, closed-loop steam-cooling is truly the enabling technology. The other two key technologies, single crystal materials and dry, low NOx combustors, can be considered supporting technologies for the closed-loop steam-cooling because of how the use of steam-cooling affects the operating environment of the hot gas path components. Hot gas path in this context refers to all of the engine components that are downstream of the combustor, namely the gas turbine nozzles and buckets.

1.1.1 Closed-loop Steam-cooling

Closed-loop steam-cooling affects the overall efficiency of the engine in two ways. First it allows the engine to operate at a higher effective turbine firing temperatures. Second, it minimizes the chargeable cooling air losses associated with the parasitic cooling air that is extracted from the final stages of the compressor and used to film cool the first few stages of the turbine [96].

From a study of cycle analysis in classical thermodynamics, one knows that there are several ways to increase the thermal efficiency and power output of a gas turbine engine. The two primary ways are to increase the overall pressure ratio of the compressor and to increase the enthalpy or temperature of the combustion gasses flowing through the turbine [155]. For the H machine, GE engineers actually employed both strategies. Comparing the previous FA machine to the new H machine, the pressure ratio was increased from 15.5:1 to 23.0:1, respectively, and the effective turbine firing temperature was increase from approximately 2,420°F to 2,600°F, respectively[21].

In most high performance gas turbine applications, parasitic cooling air is used to film cool the first few stages in the gas turbine. The film cooling process causes an overall temperature drop across the first stage nozzle of roughly 280°F. By utilizing the closed-loop steam-cooling on the first two stages of the turbine, the total amount of parasitic air needed to cool the first stage nozzle is significantly reduced and the

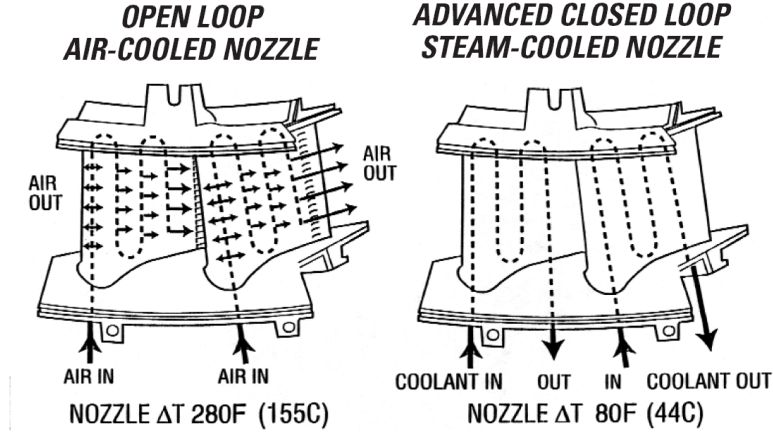


Figure 2: Steam-cooling versus Film Cooling [15]

first stage nozzle temperature drop is reduced to less than 80°F [15]. Reducing the first stage nozzle temperature drop has the effect of increasing the effective firing temperature of the turbine without actually raising the temperature of the combustion gases exiting the combustor. Figure 2 is a schematic depicting the difference between film cooling and steam cooling as applied to a nozzle. In Figure 2, air cooling is synonymous with film cooling.

As a point of reference, a 100°F increase in turbine firing temperature can provide a corresponding increase of eight to thirteen percent in power output and a two to four percent increase in simple cycle efficiency [146]. Thus, as fuel usage constitutes the single largest operating expense for a gas turbine operator, even a single point increase in simple cycle efficiency can substantially reduce the direct operating cost over the life of a typical gas fired power plant [96]. Obviously then, any technologies that facilitate an increase in turbine firing temperature are clearly worth pursuing!

Reducing the parasitic cooling air has the additional benefit of increasing the amount of compressor discharge air that is available for heat addition in the combustor. This additional mass flow through the turbine and heat recovery unit results in a two percentage point increase in the overall combined cycle thermal efficiency [21].

1.1.2 Single Crystal Materials

The increase in effective turbine firing temperature drives the need to use single crystal materials technology in the manufacture of the hot gas path components. In single crystal materials, all grain boundaries are eliminated from the material structure which provides the component with improved resistance to damage due to thermal stress, fatigue and creep as compared to previous, directionally solidified components [146]. By eliminating all grain boundaries, single crystal materials exhibit substantially increased melting temperatures which increases the high temperature strength of the material [146]. Single crystal materials also offer increased low cycle fatigue (LCF) life which is a common failure mechanism in gas turbine engines used for peaking cycle operations [115].

In addition to single crystal materials, the hot gas path components are coated with high temperature thermal barrier coatings (TBC). TBCs are required to protect the hot gas path components from corrosion, oxidation and mechanical damage. As the super alloys used in the manufacture of the hot gas path components have become increasingly complex, it has proven to be difficult to obtain both the required levels of corrosion and oxidation resistance as well as the high thermal strength needed to operate in high firing temperature engines [146]. Thus TBCs are a necessary addition to the components in the steam-cooled H machine, especially when the H machine will be used for basing cycle applications where corrosion and oxidation are the primary failure mechanisms [115].

1.1.3 Dry, Low NO_x Combustors

The previous two technologies, closed-loop steam-cooling and single crystal materials were adopted for use on the H machine because of their direct positive impact on the thermal efficiency of the engine. On the other hand, the dry, low NO_x combustor

technology is necessary for the successful adoption of the power plant by power generators. The H machine combustor does not in and of itself have a direct impact on the efficiency of the engine but rather is primarily designed to mitigate the negative environmental effects of operating the engine at very high combustion temperatures. Increasingly stringent environmental regulations are driving engine manufacturers to design combustors with emissions as one of the primary design criteria [26].

It is not particularly difficult to design a combustor that will produce exit temperatures at or above the melting temperature of the turbine components. What is difficult is designing such a combustor to do so while producing low emissions of NOx (nitrous oxides), CO (carbon monoxide), soot, unburned hydrocarbons and at the same time provide an exit temperature profile that maximizes the life of the first stage turbine nozzles and buckets [83]. The primary difficulty arises because of the exponential dependency of the rate of NOx formation with combustion temperature. To minimize the formation of NOx emissions, combustor designers employ a variety of techniques such as staged combustion and dilution of the primary combustion air with uncombusted air [168]. The drive to higher firing temperatures equates to increased engine efficiency which decreases the production of green house gases but unfortunately does so at the cost of higher NOx levels. As such, the implementation of the steam-cooling technology which enables higher firing temperatures requires the use of combustors specially designed to minimize NOx formation.

The previous long diatribe on the adoption and resulting benefits of the three enabling technologies used on the new H machine is not without purpose. In practically every previous paragraph in this section, the connection between increased turbine firing temperature and increased engine cycle efficiency was mentioned, as it turns out the drive to increased efficiency via increased firing temperature has a significant and detrimental impact on the life of the first stage nozzles. In the sections that follow, the function of the nozzles in a gas turbine will be discussed as well as how

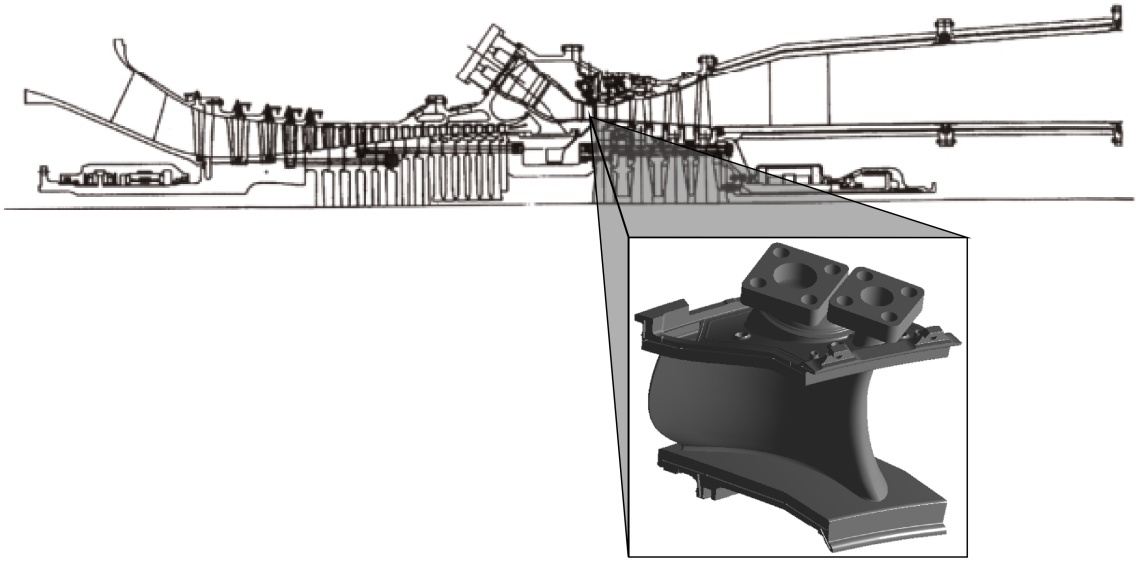
the operating environment imposed on the engine due to the use of steam-cooled technology impacts the life of the first stage nozzles.

1.2 First Stage Nozzle Overview

Up to this point, the term “nozzle” has been thrown around quite a bit but what an actual nozzle is and what it does have not yet been discussed. This section will give an overview of nozzles and their function in a gas turbine engine.

The function of a stationary first stage nozzle is to redirect the axial flow of hot gases from the combustor into the direction of the row of rotating first stage turbine buckets [62]. The resultant vector from the turning action of these hot combustion gases generates considerable forces on the nozzle that must be sustained throughout the life of the part. The combustion gases are normally hotter than the melting temperature of the metal. Thus, to avoid melting or oxidation, the nozzle is internally cooled and gas path surfaces are protected using high temperature thermal barrier coatings (TBC) [146]. There are also internal pressures in the nozzle that result from coolant flow. In addition, the nozzle must also seal the pressurized hot gases from being released into the low-pressure regions of the turbine.

As was eluded to Section 1.1.1, the cooling mechanism developed for H machine first stage nozzle (as well as the first stage buckets and second stage nozzles and buckets) is different than previous nozzle cooling mechanisms used in other GE gas turbine engines. Historically, the initial stages of gas turbine engines have used film cooling to insulate the gas turbine hot gas path parts from the combustor products [13]. The film cooling air is bleed air taken from the last stages of the compressor, routed through the main engine shaft and expelled into the hot gas path through small holes on the airfoil sections of the nozzles and blades. See Figure 2 for a schematic of how film cooling works. The new H machine however uses a revolutionary closed-loop, steam-cooling technique where high pressure steam is routed through the internal



Note - Figure 3 courtesy of the General Electric Company.

Figure 3: First Stage Nozzle CAD Drawing and H Machine Flow Path[21]

cavities of the engine to the first two stages of nozzles and blades to provide cooling. Using steam for cooling provides the additional benefit of super heating the steam for use in the steam turbine. The heat absorbed by the steam is traditionally waste heat but now can be converted into usable energy [96], which further increases the overall efficiency of the system.

The first stage nozzle is the first gas turbine component directly following the combustor. Figure 3 shows a 3-D solid model of a nozzle airfoil section and its location in a cross-sectional view of the H machine flow path. The flow path shown is the cross-sectional view of the H machine shown in Figure 1. The nozzle shown in Figure 3 is only one airfoil section from an array of nozzles that are attached together in a ring around the engine.

1.3 Operating Conditions and Nozzle Life

In Section 1.1, it was stated that the use of closed-loop steam-cooling, single crystal materials and dry, low NOx combustors were the enabling technologies that allowed the H machine to reach the 60% efficiency mark. The down side to using these technologies, however, is that each affects the life of the first stage nozzle in different and often detrimental ways.

Closed-loop steam-cooling increases the effective firing temperature of the gas turbine by decreasing the temperature drop across the nozzle due to film cooling. Increased effective firing temperature through the gas turbine essentially means that the hot gas path components will be operating in increased gas temperatures. Closed-loop steam-cooling eliminates the need to use film cooling to keep the nozzle material's temperature below its melting point. The down side to not using film cooling is that there will be increased thermal gradients through the nozzle walls. Increased thermal gradients equate to increased stresses in the nozzle walls due to thermal loads which negatively impact the nozzle's low cycle fatigue life.

Single crystal materials actually have a positive impact on the life of the nozzle. As the name implies, in single crystal materials all grain boundaries are eliminated. This technology impacts the life of the nozzle by increasing the high temperature strength of the material [146]. Increasing the high temperature strength of the material improves the low cycle fatigue life of the nozzle.

The dry low NOx combustor technology used on the H machine to reduce emissions does not directly affect the life of the nozzle, what does affect the life of the nozzle is the primary function that a combustor serves in an engine. A combustor's primary function is to add energy to the pressurized gas exiting the compressor before the gas enters the turbine. To add energy to the flow, combustor typically burns natural gas or oil. Burning these fuels in a pressurized oxidizer, air in this case, produces a hot plasma. A plasma is a high temperature, highly chemically reactive gas [184].

To give a reference point, the temperature of the combustion products coming out of the combustor in the H machine is designed to be approximately 2,600°F [169]. Exposure to such extreme operating conditions typically will cause nozzle cracking due to thermal stresses and oxidation due to the highly corrosive combustor effluent [8].

To summarize, excessive thermal stresses and corrosion/oxidation due to combustor effluent are the primary factors that limit the operating life of the first stage nozzles. Fortunately, most gas turbine engines will not be affected by both of these life limiting conditions at the same time. Most gas turbine engines used for power generation are operated in one of two schedules or duties: continuous duty or cyclic duty [8]. Continuous duty schedules are usually associated with power plants used for base load applications and cyclic duty schedules are usually associated with power plants used for peaking load applications.

A power plant operated in a base load cycle is scheduled to run all of the time except during down time due to maintenance or failures [181]. The base load for the energy grid is the amount of power used by the grid that does not fluctuate with time, often times the base load is approximately 30 to 40% of the maximum energy load consumed by the grid. When used in base load operations, the primary failure mechanism for first stage nozzles is failure due to corrosion and oxidation [115].

A power plant operated in a peaking cycle is scheduled to run only when there is an energy demand exceeding the base load [180]. A peaking load operating schedule is cyclic with the peaking plant usually operating in the afternoons in the United States. Power plants used for peaking applications must be able to be brought online and operating at maximum power output in a relatively short time. When used in peaking load operations, the primary failure mechanism for the first stage nozzles is low cycle fatigue failure due to thermal stresses [115]. As it turns out, base load applications are usually assigned to nuclear and coal fired power plants, while gas

turbine engines are used primarily in peaking cycle applications. Thus the design of the first stage nozzle in a gas turbine power plant is often driven by the induced thermal stresses and low cycle fatigue life.

Earlier it was stated that over the operating life of a gas turbine power plant, fuel is the single largest direct operating cost incurred by the power plant operator [96]. However an even more significant impact on the bottom line of power plant operator is any time the power plant is not available when needed. Obviously any time the machine is under going repairs, it can not generate electricity which means the power generator is not generating income, consequently down time has a significant, negative impact to the bottom line of the power generating facility. The time between major overhauls in industrial gas turbines is often driven by the thermal fatigue life of the first stage nozzle [34], especially when the power plant is used in peaking cycle application. Thus, as the first stage nozzles and blades are often times the life limiting parts in a gas turbine power plant, there is a definite need to identify or create new methods that allow for better estimation of the life of these hot gas path parts.

1.4 Assessing Nozzle Life

Up to this point in the motivation for the work of this thesis, we have discussed a very top-level driver for the work. Namely, how the operating conditions of a gas turbine engine affect the life of the nozzle, and how nozzle life is directly tied to the economics of engine operation. As nozzles can often be the life limiting part in an engine, they will have a tendency to drive the schedule for down times due to maintenance and failures [34]. Both of these types of down times have a direct negative impact on the bottom line of power plant operators.

We have also explored how one of the primary design goals of the new H machine also impacts the economics for the power plant operator. Designing the H machine to produce the highest combined cycle power plant efficiency on the market equates to

lower direct operating cost for the power plant operator. The reason being is because over the operating life of a gas turbine engine, fuel is the single largest direct operating cost incurred by a power plant operator [96]. The development of the H machine for increased efficiency requires the use of several key enabling technologies: closed-loop steam cooling, single crystal materials and dry, low NOx combustors. We have examined how these enabling technologies, like the operating schedule of the engine, can also have a negative impact on the nozzle life.

Given these two completely uncoupled sources of effects that adversely affect nozzle life (new technologies and operating schedule), it is obvious that having the ability to accurately predict the life of the nozzle and to design nozzles for prolonged life becomes very attractive [91]. Thus to address this high level motivation, the work for this thesis primarily revolves around developing a means to enable fast and accurate assessments of the probabilistic life of the first stage nozzle in the new H machine.

The design of a first stage nozzle for successful long life in adverse operating conditions depends on a number of interacting factors. For instance, material properties are dramatically affected by the metal temperature. The metal temperature depends on not only the gas path wall temperature but the interior wall temperature, wall thickness and TBC thickness. Wall temperatures depend on gas and coolant temperatures and as well as the heat transfer coefficients. Heat transfer coefficients depend on flow rates, which depend on gas and coolant pressures. Stresses depend on material properties, temperatures, wall thicknesses and mechanical loads. Fatigue life depends on temperature, stress and operating cycle [173]. In addition, most of these factors can and do vary at different locations on the nozzle and the conditions at each location can themselves affect the environment at other locations. Consequently we are faced with a very complex, spatially varying system of interactions that is difficult to analyze.

Simulation of the physics involved in such complex interactions is quite a difficult challenge. Such simulations require the use of complicated finite element models (FEM) that require considerable resources to characterize even a single loading condition for the model [175]. The finite element analysis (FEA) method is a numerical analysis technique that allows an engineer to obtain approximate solutions to the complex partial differential equations (PDEs) that describe the response of a system to external influences [18].

The finite element method works by first discretizing the solution domain of the system into a set of smaller sub-domains called elements. For each element, the PDEs that describe the response of the system are approximated by algebraic, finite difference equations. Numerical analysis techniques are then used to solve the algebraic equations for each element in the solution domain [128].

As an example of a discretized solution domain, Figure 4 is an example of a 3-D solid model with an applied finite element mesh. The model is of one airfoil section from the nozzle array in the first stage of the H machine’s gas turbine section. Using this mesh, an FEA program solves the governing equations to predict the stress, strain and temperature for each element. The solution obtained is a function of the specified set of external loads or boundary conditions.

The estimation of the life of the nozzle is determined first by calculating the temperature and stress on the nozzle using the finite element model, then using these estimates, calculations of the life are obtained via proprietary lifing models. The life of the nozzle can be predicted at many “interesting” locations on the nozzle. Typical locations represent historically low-life regions observed on actual gas turbine first stage nozzles operating in the real world.

Accurately assessing the life of a complex part such as a nozzle operating in the highly variable environment of a gas turbine power plant poses substantial problems for traditional deterministic design methods [92]. This is primarily due to the fact that

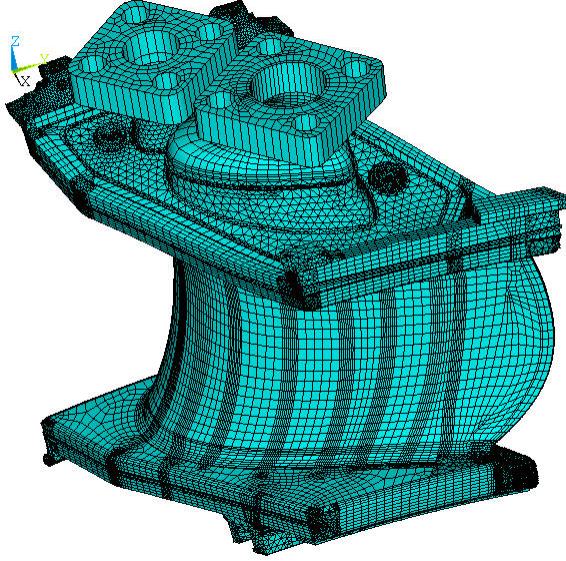


Figure 4 courtesy of the General Electric Company.

Figure 4: Meshed First Stage Nozzle

deterministic design methods are not able to adequately account for this variability. To address this limitation, traditional methods rely on the application of conservative factors of safety (i.e. worst-case scenarios, worst-case material properties, worst-case tolerances, etc.) when estimating the life of the part [122]. Research has shown that the factor of safety approach often leads to overly conservative life estimates which translate into unnecessarily high maintenance and replacement costs for the engine components [93]. Therefore, more recent design methods have begun to incorporate probabilistics into the design of the engine components [174].

Probabilistic design methods allow an engine designer to estimate the non-deterministic behaviour of the life of the nozzle. A non-deterministic estimate of life is essentially a distribution of the life. The distribution of a response captures all of the variability of the response due to the variability in the inputs. To give an example of the informative difference between deterministic and non-deterministic estimates, consider the comparison between the information provided to an engine designer by a factor of safety estimated life value and the information provided to an economist trying

to assess the income equality of the United States with only a quote of the average income in the U.S. [86]. In both cases, the single numbers give some measure of the response in question but neither are all that informative.

If the quoted values are average values (average life, average income), what type of average is implied? Since an average is a measure of the central tendency of the estimate, and there are several measures of central tendency, a quoted average is an ambiguous value. In such a situation, one should quickly ask, does the average value equal the arithmetic mean, the geometric mean, the median, the mode [66]? Does the quoted value represent the lowest expected value, if so is there also a corresponding upper bound on the value? What is the expected range of the value? Does the value vary over a large range, or small range? All of these questions are answered when one has the distribution of values as compared to a single value estimate. Probabilistic design methods provided the designer with the response's distribution. A response's distribution, as discussed in any introductory probability and statistics text [59], provides a complete description of its probabilistic behavior. The probability distribution of the response allows the designer to know the probability of a response exceeding a certain value.

The methods used to quantify the probabilistic nature of a response can be roughly grouped into one of three categories, direct Monte Carlo Methods [140], indirect Monte Carlo methods [11], and analytic reliability methods [198]. Although all three classes of methods create assessments of the probabilistic nature of a response in different ways, each of the classes fall under the common heading of design space exploration methods. A design space exploration method is any method that selects samples from the input space and tracks the resultant sample from the output or response space. For a given analysis, a sample of the input space means one unique setting of the input values to the analysis. A sample of the response space means the response value that resulted from transforming a sample from the input space

through the analysis. The analysis in this case could be a mathematical function, it could be an engineering design code or it could be a series of engineering design codes linked together.

We have now identified the two enablers that will allow an engine designer to estimate the probabilistic life of a nozzle. One is the use of a multi-physics finite element analysis that maps a set of operating conditions to an estimation of the stresses, temperatures and life of the nozzle. The other is the application of a design space exploration procedure which will systematically produce samples mapping the input to the output space through the finite element analysis to characterize the probabilistic nature of the nozzle life. The next section will go into some detail discussing the development of an appropriate finite element model created to capture the influences of the highly variable environment that the H machine imposes on the nozzle. The characteristics of this model will be discussed, focusing how they drive the selection of the design space exploration procedure used to estimate the life of the nozzle.

1.5 A Proposed Lifting Methodology

In Section 1.4, it was stated that nozzles operate in a highly variable environment and in Section 1.2, it was stated that the operating environment of a nozzle has a direct impact on the life of the nozzle. Consequently if we want to accurately estimate nozzle life we need to develop a model that is able to describe or capture as much of that variability as possible.

In Section 1.4 it was also stated that to adequately represent the nozzle life we need a probabilistic description of the life, that is, we need its distribution. To estimate the distribution of the life, we will need to use some sort of probabilistic design method.

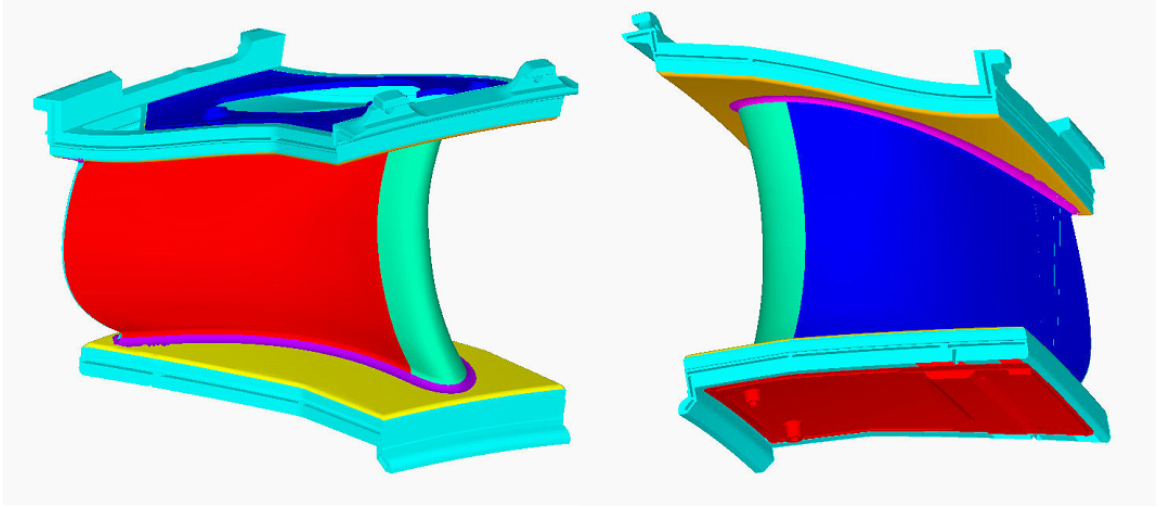
Now just because we have a model and a probabilistic design method, does not

mean that we can simply apply the method to the model and realize the probabilistic nature of the system. Rather, the model must be formulated in such a manner as to enable the use of a design space exploration (DSE) procedure in conjunction with the model. To use a DSE procedure with a given model, the model must allow for parametric design. Parametric design implies that attributes of the model are under control of the designer [29]. A model that enables parametric design is a model that exposes a set of parameters or design variables to the DSE method. The DSE method systematically varies these parameters and tracks the response(s) of the system.

In addition to exposing a set of design variables to the DSE method, the parametrized model must also be formulated such that it mimics the physics of the system that it is representing. In our case we are trying to capture how the complex operating environment imposed on a nozzle affects the nozzle life. This operating environment consists of spatially varying thermal and mechanical loads. This translates into a finite element model that should allow the DSE procedure to parametrically vary the temperature and heat transfer boundary conditions, as well as the wall thicknesses from one location on the nozzle model to another.

The model formulated to capture the spatially varying thermal boundary conditions and wall thickness is a zonally subdivided version of the finite element model shown in Figure 4. Each zone represents a different geographic location on the model. Figure 5 shows the subdivided nozzle finite element model with each zone color coded. The zones of the nozzle include areas such as the suction and pressure sides of the nozzle airfoil, the inner and outer fillet regions between the nozzle airfoil and the inner and outer bands, etc.

The model is parametrized by allowing the internal and external temperature and heat transfer coefficients; wall and TBC thickness to be independently varied across each of the spatial zones. “External” in this context refers to the parts of the nozzle exposed to the hot gas path, and “internal” refers to the parts of the nozzle located



Note - Figure 5 courtesy of the General Electric Company.

Figure 5: Nozzle FEA Model Divided into Geographic Zones

in the cooling cavity. In all, the zonally subdivided nozzle model provided the design space exploration procedure with 52 independent variables.

In typical finite element analyses for lifing prediction, the life response is usually estimated as the overall life or global life of the part [91], however for this formulation, a zonal approach is also used for the responses. As was stated in Section 1.4, the life of the nozzle can be estimated at historically low-life regions on the nozzle. In this formulation, each of the low-life regions corresponded to a cluster of nodes rather than one single node. Within the cluster, responses for the life, temperature and stress will be recorded at the node with the lowest life. In all, there are 85 total clusters on the nozzle model where the three response types are evaluated at, resulting in a total of 255 responses.

In addition to having 52 design variables and 255 responses, the spatially subdivided finite element model is characterized by having a relatively long run time. The total wall clock time for one run is approximately 3.5 to 4 hours. This run time is primarily due to the large number of elements that make up the finite element model.

The model shown in Figure 4 has approximately 750,000 elements. In general, this model can be considered a large scale model.¹

Section 1.4 stated that there are three general classes of probabilistic design methods suitable for realizing the probabilistic nature of the life of the nozzle. Those three classes are direct Monte Carlo simulation, indirect Monte Carlo simulation and analytic reliability methods. With these three methods in mind one of the most important initial research questions can be asked. Namely, which of these three primary probabilistic design methods is the best to use with the nozzle model? The initial hypothesis is, given the nature of the problem, an indirect Monte Carlo simulation technique is most likely the best probabilistic design method to use with the model. The primary reason is that given the combinations of model run time and the large numbers of inputs variables and responses; an indirect Monte Carlo simulation approach is deemed the most computationally efficient method.

Assertion: A spatially partitioned finite element model was created to address the complex operating environment that a nozzle is exposed to. This model has 52 variables, 255 responses and takes 3.5 hours to run.

Assertion: To capture the probabilistic nature of the life of nozzle, a probabilistic design method is needed. The primary methods are direct Monte Carlo, indirect Monte Carlo and analytic reliability methods.

Research Question 1: Which of the three primary probabilistic design methods is most applicable for the nozzle problem?

Research Hypothesis 1: Given the computational nature of the nozzle model, it is hypothesized that an indirect Monte Carlo simulation technique is most likely the best probabilistic design method to use.

¹See Section 3.1 for the definition used in this thesis to characterize a large scale model.

An indirect Monte Carlo simulation method is a probabilistic design method that enables the determination of a response's probabilistic nature by first creating a metamodel or surrogate model of the response the system. Then the metamodel is used with Monte Carlo simulation to transform random samples from the input space into probability distributions of the output space. A metamodel in this case is a simplified representation of the actual model that captures the salient predictive capability of the full model but executes in a fraction of a second. Chapter 2 goes into much more detail on Monte Carlo simulation techniques and other probabilistic design methods. Chapter 2 also provides a qualitative down select of the probabilistic design method most applicable for the problem at hand.

However, even though the Monte Carlo/metamodeling combination is believed to be the most efficient probabilistic design method to apply to the problem, in its traditional form this combination also is deemed to be not computationally efficient enough to apply to the problem. For instance, based on the run time of the model and using a run estimate based on the minimum runs required to create a full quadratic response surface of the system [109], over 5,000 hours would be required to create the set of metamodels. The majority of runs required for this estimate are runs used to quantify the variability of the response due to interactions between the design variables. An interaction is an effect whose magnitude is a function of the current setting of two or more design variables. Consequently, the large scale nature of the model is driving the need to employ the use of a large scale metamodeling technique to create the metamodels. But now the question becomes, which large scale metamodeling method should be used?

To alleviate as much of the computational burden as possible, it is hypothesized that perhaps there is a way to use the spatially partitioned nature of the finite element model to our advantage. Perhaps a partitioned metamodeling approach could be used to possibly reduce the total number of runs required. The idea being that if you

could partition the set of design variables into several smaller sets you could reduce the total number of runs required to make the metamodels. The run reduction comes from the notion that interactions between design variables within a set are captured but interactions between design variables in disjoint sets are not captured.

Assertion: Given the constraints imposed by the nozzle model’s complexity, traditional statistical linear metamodeling methods are not efficient enough to be used with the model. Consequently a large scale metamodeling method is necessary to reduce the total computational expense as much as possible.

Research Question 2: Which large scale metamodeling procedure is best to address the characteristics of the nozzle model?

Hypothesis 2: Given the spatially partitioned nature of the finite element model, it is hypothesized that a partitioned metamodeling scheme seems like a logical choice for reducing to the total number of runs required to create an accurate metamodel.

This idea of leveraging the spatially partitioned nature of the system to create metamodels is a direct realization of the concept of “Motivated Metamodels” by Davis and Biglow (2002, 2003)[27, 28]. Davis and Bigelow’s motivated metamodeling is essentially the notion of using one’s phenomenological knowledge about the system to create better metamodels. “Better” in this case is not only more accurate but also implies that the metamodels “tell the story” of the system. That is, they not only capture the behavior of the system but they also capture the structure of the system. In our case, partitioned metamodels would be capturing the partitioned nature of our system in a way that traditional statistical linear metamodeling schemes can not and at the same time allowing us to create metamodels more efficiently than these traditional approaches.

To give an idea as to the potential run savings associated with using a partitioned metamodeling approach, Figure 6 shows estimates of the total number of runs required for several different sizes of problems. The y-axis is the number of partitions that the set of factors is divided into. In Figure 6, each partition is assumed to be of equal size. The x-axis shows the number of runs required for a given partition size and total factor count. The estimated number of runs, R , for a problem of size, n , partitioned into p subsets is based on the following equation:

$$R = p \frac{(n/p + 2)(n/p + 1)}{2} \quad (1)$$

where,

R = estimated number of runs required

p = number of partitions

n = total factor count

Equation 1 is a modified version of the equation which estimates the minimum number of runs required to create a full quadratic model based on a saturated design space exploration [110]. Saturated in this context means that there is one run per term in the model. In general one always collects more data than the minimum required for a given model structure, so Equation 1 is merely used as a baseline for comparison.

Using Figure 6 as a guide, the estimated number of runs required for a 50 parameter system partitioned into two sets is 702 runs. The estimated number of runs for the complete 50 parameter system is 1,326 runs. Thus there could potentially be a run savings of approximately 47% by partitioning a 50 parameter system into two disjoint sets. If we partition the system into three sets, the run savings jumps to 62%. Granted these run savings projections are merely estimates, most likely the run savings will not be as great as this, however there does seem to be some impetus to

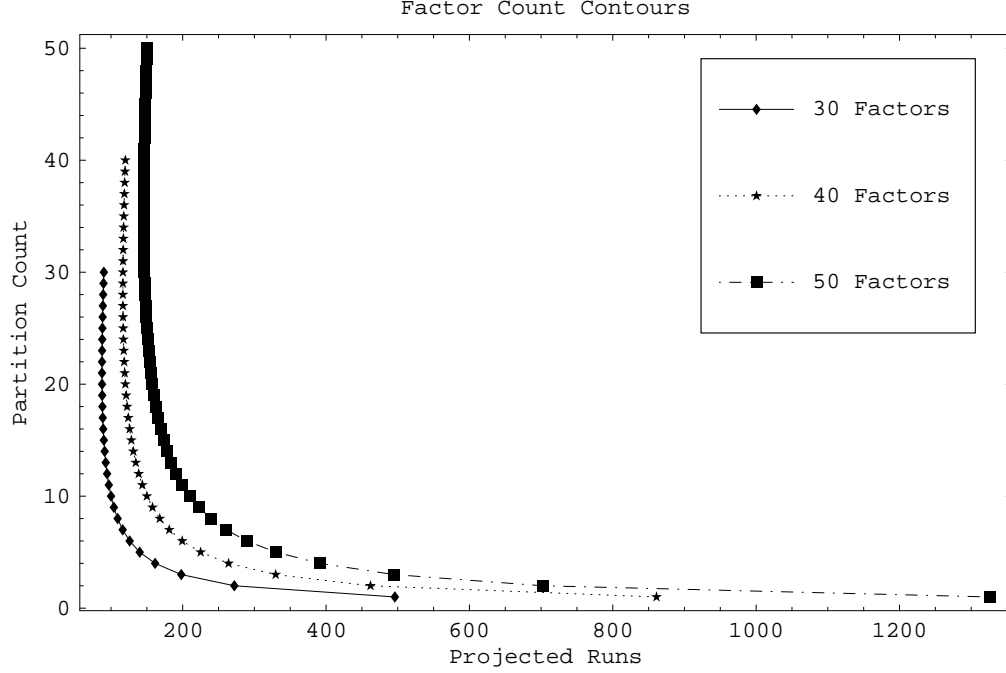


Figure 6: Projected Runs per Partition Size

explore a partitioned approach for creating metamodels of a large scale system.

Now that we have hypothesized that a partitioned metamodeling approach is a potential candidate for reducing the total number of run required, we are now faced with the problem of how should the design variables be partitioned? This is actually a very important question because one could certainly envision how a poorly chosen set of partitions for a large scale system could significantly impact the accuracy of the partitioned metamodels. Consequently, choosing an good set of partitions is very important.

Unfortunately as we will see in Chapter 3, there does not seem to be any currently available means to quantitatively determine an optimal set of partitions for metamodeling. The currently demonstrated methods are either based on engineering intuition or partitioning along process or code boundaries. However, the electrical engineering community has been using partitioning approaches for many years to simply the

analysis of VLSI (very large-scale integrated) circuits. Perhaps one of their methods could be adopted to determine an optimal set of partitions for use with metamodeling.

Assertion: A partitioned metamodeling approach is proposed to reduce to total number of runs required to create metamodels of the nozzle model.

Assertion: The electrical engineering community has been using partitioning techniques for many years to analyze VLSI circuits.

Research Question 3: Is there a way to quantitatively find an good set of partitions to use with the partitioned metamodeling process?

Research Hypothesis 3: It is hypothesized that a method from the electrical engineering community for partitioning VLSI circuits could potentially be adopted for finding an optimal set of partitions for use with partitioned metamodeling.

To summarize, the method proposed to estimate the life of the first stage nozzle is based on the spatially partitioned finite element model shown in Figures 4. It is hypothesized that creating a partitioned metamodel of the system that leverages the spatially partition nature of the model will significantly reduce the computation expense to create an accurate metamodel. This partitioned metamodel can then be used with a traditional Monte Carlo simulation to estimate the life of the nozzle. To find the set of partitions, a technique from the electrical engineering community can be used.

1.6 Initial Research Questions and Hypotheses Review

The proposed method for predicting the life of the first stage nozzle presented in Section 1.5 immediately raises three very important research questions and hypotheses that need to be investigated. In fact the process of supporting or refuting these

hypotheses will form the bulk of the next three chapters. Even more importantly, the results of these inquiries will drive the majority of the work that this thesis will contribute to the engineering community.

Section 1.5 hypothesized that given the nature of the finite element model created to predict the life of the nozzle, an indirect Monte Carlo technique is the most appropriate method to realize the probabilistic nature of the nozzle life. In actuality however, this hypothesis is merely based on the author's understanding of the system and experience with other large scale systems. There may be other probabilistic design methods available that are perhaps more appropriate for the system in question. These could include new advances in quasi-Monte Carlo and adaptive, stratified sampling techniques or the accurate and efficient analytic reliability methods. This question definitely needs to be resolved as selection of the wrong method could preclude any ability to accurately and efficiently assess the probabilistic life of the nozzle. Chapter 2 will review each of the three primary classes of probabilistic design methods and using the characteristics of the system, show that indeed an indirect Monte Carlo approach is a logical choice for analyzing the nozzle system.

The next burning question that needs resolving is, assuming that the indirect Monte Carlo method is appropriate for the system, does a partitioned metamodeling approach makes sense? Alternatively, is there another large scale metamodeling method that already exists that can be readily applied to the system? Section 1.5 hypothesized that the spatially partitioned nature of the nozzle model could be used with a partitioned metamodeling method to create accurate metamodels of the system. To support or refute this statement, Chapter 3 will review the currently available large scale metamodeling techniques and show that a partitioned approach does in fact make sense. However, we will also see that there is not a currently available large scale metamodeling technique that meets all the requirements for the system. This lack of an appropriate method will drive one of the primary contributions of this

thesis, namely, the development of a partitioned, hierarchical metamodeling scheme for large scale, multi-objective systems.

Closely tied to the idea of partitioned metamodels for large scale systems, is the very practical question of how to partition the system. One could certainly envision how a poorly chosen set of partitions for a large scale system could significantly impact the accuracy of the partitioned metamodels. As we will see in Chapter 3, the only currently available partitioning methods for use with metamodeling are based on engineering intuition and partitioning along code or process boundaries. This lack of a quantitative means to partition a large scale systems for use with metamodeling is viewed by this author as a key missing element in the process.

Fortunately, the quantitative partitioning of large scale systems has largely been solved by the electrical engineering community. It is hypothesized that perhaps one of their methods could be used to find an optimal set of partitions for our problem. However, as we will see in Chapter 4, their methods are not 100% applicable to the problem at hand. Chapter 4 will follow the development of partitioning techniques for large scale circuit analysis and identify the missing elements necessary to solve the partitioning problem for large scale metamodeling. The adaptation and modification of the VLSI circuit partitioning methods for quantitatively partitioning large scale metamodeling problems will comprise another of the primary contributions for this thesis.

1.7 Outline of the Thesis

Chapter 2: Chapter 2 is written to address research question 1 and hypothesis 1 introduced in Section 1.5. Chapter 2 will review each of the primary probabilistic design methods and map their characteristics against the characteristics of the nozzle problem. This mapping will then be used to qualitatively determine which probabilistic design method is most appropriate to use with the nozzle

problem.

Chapter 3: Chapter 3 is written to address research question 2 and hypothesis 2 introduced in Section 1.5. Chapter 3 will review each of the large scale meta-modeling techniques currently stated in the literature and use their strengths and weaknesses to formulate a new large scale metamodeling method.

Chapter 4: Chapter 4 is written to address research question 3 and hypothesis 3 introduced in Section 1.5. Chapter 4 will review common systems decomposition methods used by the electrical engineering community analyze VLSI circuits. Based on the limitations of using these methods for metamodeling, a new method will be formulated that allows one to determine the sparse nature of a black-box system.

Chapter 5: Chapter 5 is an interim summary of the first four chapters. Chapter 5 will finish with a complete summary of all of the research questions and hypotheses mapped to the section that they were posed in as well as the section that they are addressed in.

Chapter 6: Chapter 6 presents the formal development of the primary contributions of this thesis. In addition, Chapter 6 will also formulate the validation plan that will be used to assess the quality of the developed methods.

Chapter 7: Chapter 7 outlines the computational framework that was developed to implement and validate the methods developed in Chapter 6.

Chapter 8: Chapter 8 summarizes the results of applying the methods developed in Chapter 6 to the two test cases discussed in the validation plan. Chapter 8 will conclude by addressing the remaining unanswered research questions and hypotheses.

Chapter 9: Chapter 9 will start off with a review of all the research questions and hypotheses posed throughout the thesis and summarize the validity of each hypothesis. In addition, Chapter 9 will formally state the contributions of this thesis.

1.8 Summary

This chapter started off with a direct motivating statement that drives the primary contribution of this thesis to the engineering community. The contribution of this thesis revolves around the development of a method for exploring the relationships between input and outputs for large scale computer simulations. The proposed design space exploration procedure uses a hierarchical partitioning scheme to help mitigate the “curse of dimensionality” often associated with the analysis of large scale systems. Closely coupled with the use of a partitioning approach, is the problem of how to partition the system. Thus, the other main contribution of this thesis is the development of a quantitative means to identify a good set of partitions for a large scale system.

The motivation for the development of these new methods can be directly tied to the development of General Electric’s new H machine. In the design of the new H machine, several key enabling technologies were required to facilitate reaching the H machine’s primary design goal: creation of a combined cycle gas turbine power plant capable of reaching the 60% efficiency barrier. This chapter discussed how the use of closed-loop steam-cooling, single crystal materials and dry, low NO_x combustors have an adverse effect on the life of the first stage nozzle. Degradation of the life of the first stage nozzle was then linked to the economics of operating the gas turbine engine.

Given that the nozzle life is adversely affected by the operating conditions imposed by the H machine’s enabling technologies and poor life has a direct negative impact

on the bottom line of gas turbine operators, there becomes a definite need to design a nozzle that is robust to these operating conditions. To do so, the nozzle life is typically analyzed using a complex finite element analysis in conjunction with a probabilistic design method. The finite element model created for this system was a spatially partitioned model comprised of 52 independent design variables and 255 independent responses. The model took approximately 3.5 hours to compute one set of loading and boundary conditions.

Based on the nature of this analysis, the proposed probabilistic design method used to estimate the probabilistic nature of the life is an indirect Monte Carlo simulation. Although indirect Monte Carlo methods are typically considered efficient probabilistic analysis methods, the large scale partitioned nature of the system identified the need for a new metamodeling method. Thus, it was hypothesized that a new metamodeling procedure could be created that was able to exploit the partitioned nature of model to create metamodels even more efficiently but just as accurate as the currently available metamodeling methods.

Stemming from this hypothesis, were three research questions that needed to be answered before it could even be justified to spend the resources to develop the new hypothesized partitioned metamodel. First, even though an indirect Monte Carlo was chosen as the appropriate probabilistic design method, is this probabilistic analysis method actually the most appropriate for the system? Secondly, assuming that an indirect Monte Carlo method is the appropriate probabilistic design method, does the proposed use of a partitioned metamodeling scheme make sense? Alternatively, is there another large scale metamodeling scheme that is more appropriate for the system. Finally, assuming that a partitioned metamodeling approach is the best approach for the system, how does one actually determine how to partition the system?

The investigation of the necessary background information to support or refute the hypotheses and to answer the research questions will comprise the bulk of the

next three chapters.

Chapter II

REVIEW OF PROBABILISTIC DESIGN METHODS

The goal of this chapter will be support or refute the first research hypothesis stated in Section 1.6; that hypothesis was: Given the computational nature of the nozzle model, it is hypothesized that an indirect Monte Carlo simulation technique is most likely the best probabilistic design method to use. This hypothesis was made based on the characteristics of the nozzle model as well as the experience of the author with applying probabilistic design methods to other similar problems. The pertinent nozzle model characteristics that drove this hypothesis were the large numbers of input variables (52), the large number of responses (255) and the long run time of the model (3.5 hours).

To support or refute this hypothesis, the three primary classes of probabilistic design methods will be reviewed. The review will be comprised of several direct Monte Carlo simulation methods, the two primary analytic reliability methods and several indirect Monte Carlo simulation methods. The review of each of the methods will follow a roughly chronological order, starting with a basic history of the methods (i.e. what field did they originate in and what problems were they initially intended to solve), and finishing with an overview of the fundamental equations for each method.

After reviewing each of these probabilistic design methods, the characteristics of these methods will then be compared to a suite of requirements. The selected requirements are based on the needs of the overall analysis, as well as the constraints that the nozzle model imposes on the analysis. Based on the results of this comparison,

the validity of the previous research hypothesis will then be assessed.

2.1 Direct Monte Carlo Simulation

2.1.1 Traditional Monte Carlo Sampling

Direct Monte Carlo Simulation is the original and simplest of all the probabilistic design methods. The technique was originally developed during the early 1940s by Stan Ulam and John von Neumann to solve neutron shielding and other problems at Los Alamos [151]. The first documented use of these techniques appeared in 1949 in a paper by Metropolis and Ulam [103]. Unlike many other probabilistic design methods, implementation of a Monte Carlo Simulation requires only a limited understanding of probability and statistics.

In the most basic formulation of direct Monte Carlo Simulation, each input variable to a deterministic analysis is assigned a distribution. Random samples from each of the distributions are selected and evaluated in the analysis, each individual analysis is considered one simulation. Many thousands of simulations are performed to determine or realize the resultant probabilistic characteristics of the outputs from the deterministic analysis [47].

In addition to being simple to implement, Monte Carlo simulation is also the most accurate of the probabilistic methods. It is considered the “Gold Standard” by which all other probabilistic methods are judged [130]. This is because as the number of simulation cycles tends to infinity, the true distributions of the outputs as a function of the inputs will be realized. This result follows from a direct frequentist definition of Monte Carlo simulation [172], namely that Monte Carlo Simulation is simply a means of collecting random samples from the underlying population of all possible outcomes of the analysis. Given an infinite number of samples, the sample space becomes equal to the population of outcomes.

While Monte Carlo Simulation is simple to implement and accurate, it is not a

particularly efficient method for determining the probabilistic nature of a system. The inefficient nature of a Monte Carlo Simulation stems from how the technique calculates the probability of a certain event occurring in the underlying population. The following equation is the fundamental equation for calculating the probability of an event, P_{event} , based on the total number of samples, N , taken from the population:

$$P_{event} = \frac{1}{N} \sum_{i=1}^N I_{event}(\mathbf{x}_i) = \frac{n_{event}}{N} \quad (2)$$

where, $I_{event}(\mathbf{x}_i)$ is a simple indicator function defined as:

$$I_{event}(\mathbf{x}_i) = \begin{cases} 1, & \text{if event occurs} \\ 0, & \text{if event does not occur} \end{cases} \quad (3)$$

Given Equation 2, to accurately estimate the probability of an event there is some question as to how many samples, N , to take from the population. Various authors give differing methods and rules of thumb for calculating the required number of samples [48, 132, 140]. Empirical observations of applying Monte Carlo simulation to large scale systems indicate that to get good resolution on the probability of an event occurring, one needs roughly ten times the inverse of the event probability times the number of random variables. As an example, in the design of a system for long life, a typical desired probability of failure is often on the order of 1 failure in 10,000 or 100,000 simulations. As such if the system has ten random input variables, a conservative estimation of the number of simulations required is one to ten million samples. Unless one's simulation code runs extremely fast, this is often an unreasonable number of samples to collect.¹

To address the excessive number of samples required to perform a Monte Carlo

¹However, if one is not necessarily interested in estimating probability of failure but rather in estimating the shape of a response's probability distribution, then many fewer samples are needed. Five to ten thousand samples is often a good starting point when using a Monte Carlo simulation to realize an adequate histogram of probability distribution.

simulation, there have been several other direct simulation techniques developed over the years. In general they can be categorized under the general heading of “Variance Reduction Techniques” and divided into three basic classes: stratified sampling [100], quasi-Monte Carlo [53, 75] and importance or adaptive sampling [102]. In general these techniques can often reduce the number of simulations required by several orders of magnitude as compared to traditional Monte Carlo methods. However they do so at expense of losing one of the most attractive features of classic Monte Carlo - its simplicity.

2.1.2 Stratified Sampling

Stratified sampling is very similar to classic Monte Carlo with one variation. With traditional Monte Carlo, points from the input domain are selected at random with the possibility of missing some regions of the input space. Stratified sampling provides that all areas of the design space are guaranteed to be represented. The sample space is divided into m mutually exclusive regions, R_m , and random samples from each region are taken based on each input variable’s assigned distribution. The probability of a specific event occurring can thus be calculated via Equation 4.

$$P_{event} = \sum_{j=1}^m \left[P(R_j) \frac{1}{N_j} \sum_{i=1}^{N_j} I_{event}(\mathbf{x}_i) \right] \quad (4)$$

where,

$P(R_j)$ = the probability of region, R_i

N_j = the number of samples takes from region, R_j

$I_{event}(\mathbf{x}_i)$ = the indicator function as defined in Equation 3

Of the stratified sampling techniques, latin hypercube sampling is arguably the most popular version. See Chapter six in “*Sensitivity Analysis*” [143] for an excellent discussion of latin hypercube sampling. Also, the papers by McKay (1979)[100],

Iman (1982)[68] and Huntington (1998)[67] provide a thorough introduction to application of latin hypercube sampling for computer experiments. See Helton and Davis (2002)[61] for a review of latin hypercube sampling applied to a variety of test cases.

2.1.3 Quasi-Monte Carlo Sampling

Quasi-Monte Carlo sampling is similar in nature to the general class of stratified sampling techniques in that it also aims to provide a uniform selection of points from across the design space. However quasi-Monte Carlo differs from traditional Monte Carlo in that the algorithms to select the points are completely deterministic. Halton (1960)[53] and Hammersley (1960)[54] were the first researchers to develop the quasi-Monte Carlo methods. The quasi-Monte Carlo methods are characterized by being considerably more efficient and having a faster rate of convergence as compared to traditional Monte Carlo [130]. In addition, several of the quasi-Monte Carlo methods offer the ability to perform sequential sampling² which is a distinct advantage over latin hypercube sampling. For a good review of the efficiency and accuracy of traditional Monte Carlo methods versus latin hypercubes and quasi-Monte Carlo methods (Halton Sequences only) see Robinson (1999)[130].

2.1.4 Adaptive-Importance Sampling

Adaptive-importance sampling achieves overall efficiency improvements as compared to traditional Monte Carlo by concentrating the distribution of sample points to the region of the input space associated with the occurrence of a specific event. This is in stark contrast to the stratified sampling methods which provide an even distribution of points across the input space. Adaptive-importance sampling is implemented by defining an importance sampling distribution, $f^*(\mathbf{x})$, to focus the sampling of the input to the desired region. Equation 2 can then be updated to calculate the probability

²Sequential sampling describes any method that allows the experimenter to iteratively collect a few sample points, check the result and repeat until a satisfactory solution converges or the allotted computation budget is expended.

of a specific event occurring.

$$P_{event} = \frac{1}{N} \sum_{i=1}^N I_{event}(\mathbf{x}_i) \frac{f(\mathbf{x}_i)}{f^*(\mathbf{x}_i)} \quad (5)$$

where, $f(\mathbf{x})$ is the multivariate distribution of the input parameters. See Melchers (1989)[102] for a thorough introduction to and development of importance sampling methods. Wu (1994)[192] further develops the importance sampling method by combining the notion of the most probable point (MPP) of failure with importance sampling. See Section 2.2.1 for a discussion of the most probable point of failure.

2.2 *Analytic Reliability Methods*

Analytic reliability methods were originally developed to account for risk and uncertainty when developing structural building codes to estimate acceptable levels of load and resistance for structural design. Historically all uncertainty in structural design was accounted for by applying conservative factors of safety to the applied loads and/or the estimated resistance of a structure; it was assumed that these factors of safety were sufficient to keep the structure from failing [94]. To address the inherent uncertainties associated with structural design, various analytic reliability methods were developed to aid the structural designer. Analytic reliability methods provide the designer with a means to rigorously account for uncertainty and produce a resultant number, β , which represents the safety or reliability of the system. β is typically referred to as the safety index or reliability index of the structure [194].

Fundamental to all analytic reliability methods is the notion of a limit state function. A limit state function is a function that relates a response of a system to a specified limiting value [156]. The response function can be any measure of performance of the system, such as stress, life, cost ,etc. The standard definition of the limit state function is:

$$g(\mathbf{X}) = Z(X_1, X_2, \dots, X_n) - z_0 \quad (6)$$

where,

$g(\mathbf{X})$ = the limit state function

$Z(\mathbf{X})$ = the response of the system

z_0 = a predefined limit on the response

The limit state divides the response space into two regions, which are referred to as the safe and unsafe regions for the response. By convention, the failure region, Ω , is defined when $g(\mathbf{X}) < 0$. Using this definition, the fundamental equation for the probability of failure, p_f , as used by all probabilistic design methods can then be defined by the integral:

$$p_f = \int \dots \int_{\Omega} f_X(x_1, x_2, \dots, x_n) dx_1 dx_2 \dots dx_n \quad (7)$$

Where f_X is the joint probability distribution of the random variables X_1, X_2, \dots, X_n and the integral is evaluated over the failure region Ω [131]. Note however that while this equation is the fundamental equation stated in the development of all analytic reliability methods, evaluation of the integral is almost never performed except in the simplest of cases. Generally the joint probability function, f_X , is impossible to obtain and even if it were available, evaluating a complex multidimensional integral like f_X is exceeding difficult, if not impossible.

To integrate Equation 7, several different analytic reliability methods have been developed. These methods include FORM (first order reliability method) [124], SORM (second order reliability method) [125] and AMV (advanced mean value) [193].

2.2.1 First Order Reliability Method

The initial development of the analytic reliability methods can be traced to back to the 1920s. During this time the research centered around questioning the established

structural design methods and the development of the concept of random structural events [94]. This new line of thinking was a significant departure from the traditional notions of structural design and analysis. Unfortunately, the majority of this development was largely ignored. It was not until a paper by Cornell (1969) [24], that the field of analytic reliability methods started to gain traction.

Cornell's initial formulation was based on a definition of a the limit state function comprised of only the estimated load and resistance of the structure, S and R respectively. R and S were assumed to be statistically independent, normal random variables. A failure event was defined when $R < S$ or equivalently, $R - S < 0$. The reliability index was then easily defined as the inverse of the coefficient of variation (COV) for a simple sum of normally distributed random variables. Following suit, the probability of failure was then the value of the standard normal cumulative distribution function (CDF) evaluated at the reliability index [24].

$$\beta = \frac{\mu_g}{\sigma_g} = \frac{\mu_R - \mu_S}{\sqrt{\sigma_R^2 + \sigma_S^2}} \quad (8)$$

$$p_f = \Phi(-\beta) = 1 - \Phi(\beta) \quad (9)$$

where

$$\begin{aligned} \mu_g &= \text{mean of the limit state function, } g \\ \sigma_g &= \text{standard deviation of limit state function, } g \\ \Phi &= \text{standard normal CDF} \end{aligned}$$

The Cornell formulation can easily be extended to limit state functions that contains more than two random variables. The limit state can be linearized by creating a first order approximation of the limit state function about a point in the design space. The linearized limit state function, \hat{g} , and the corresponding reliability index are shown in the following equations [156]:

$$\hat{g}(\mathbf{u}) = a_0 + \sum_{i=1}^n a_i (u_i - u^*) \quad (10)$$

$$\beta = \frac{\hat{g}(\mu)}{\sqrt{\sum_{i=1}^n a_i^2 \sigma_i^2}} \quad (11)$$

The primary limitation to Cornell's formulation was that it was not invariant to different yet equivalent forms of the limit state. For instance, the limit state defined above as $R < S$ can equivalently be defined as $1 - S/R < 0$. Under the Cornell formulation these two mathematically equivalent limit state functions will not yield the same reliability index as defined in Equation 8 [47].

To address this limitation, Hasofer and Lind (1974)[58] applied a simple coordinate transformation to scale the original variables into standard normal variables. The transformation re-centers the original coordinate system around the origin of the space. In doing so, the reliability index then becomes the minimum distance from the origin to the limit state surface. The Hasofer-Lind transformation and corresponding reliability index are:

$$u_i = \frac{x_i - \bar{x}_i}{\sigma_i} \quad (12)$$

$$\beta_{HL} = \sqrt{(\mathbf{u}^*)^t (\mathbf{u}^*)} \quad (13)$$

where,

- u_i = standard normal version of variable, x_i
- x_i = non-normal random variable
- \bar{x}_i = mean of random variable, x_i
- σ_i = standard deviation of random variable, x_i
- β_{HL} = Hasofer-Lind reliability index
- \mathbf{u}^* = vector of standard normal random variables
at minimum distance from the origin

The transformed vector, \mathbf{u}^* , is the point on the limit state function that is closest to the origin of the transformed space. This point is often called the design point or the most probable point (MPP) of failure [48]. Once the most probable point of failure is found in the transformed space, applying the inverse of Equation 12 will give the MPP in terms of the original random variables.

Up to this point in the development of what will eventually be called the first order reliability method (FORM), both the Cornell and Hasofer-Lind methods have relied on the assumption that the limit state function was a linear combination of normally distributed random variables. However in actuality, this is almost never the case. As the random variables become more non-normally distributed, the accuracy of each of these formulations is severely degraded. To address this limitation, there are several available means of transforming the original non-normal distribution to equivalent normal distributions. The most common methods are the classic Rosenblatt transformations, the Rackwitz-Fiessler two-parameter equivalent normal and the Chen-Lind three-parameter equivalent normal transformations.

In its standard form, the Rosenblatt transformations [139] can be used to transform any continuous, multivariate distribution, $F(x_1, x_2, \dots, x_n)$, into a uniform distribution on the n -dimensional hypercube. The Rosenblatt transformations can also

be used to transform any non-normal distribution to a normal distribution. Using the non-normal to normal transformation the Hasofer-Lind method can then be used to find the MPP of failure. The primary detractor to using the Rosenblatt transformations to simplify the mathematics is that the limit state function can become quite distorted. A originally smooth limit state becomes highly curved and may now have multiple most probable points of failure [156].

Rackwitz and Fiessler (1978)[126] extended the Hasofer-Lind formulation to approximate each non-normal random variable by an equivalent normal random variable. The Rackwitz-Fiessler approximation method creates a two-parameter equivalent normal by matching the CDFs and PDFs of the original distribution with an equivalent normal distribution at the most probable point of failure. This approximation gives a reasonably accurate estimate for the probability of failure for most applications.

Chen and Lind (1983)[22] further extended the Rackwitz-Fiessler formulation by approximating each non-normal random variable with a three-parameter equivalent normal random variable. Although the Rackwitz-Fiessler algorithm is accurate and efficient for most applications, it loses accuracy as the input distributions become highly skewed. The Chen-Lind formulation addresses this limitation by matching the CDFs and PDFs of the two distributions at the MPP (like the Rackwitz-Fiessler method) and adding the additional constraint that the slope of the PDFs of the two distributions must also match.

The iteration scheme developed for the Hasofer-Lind formulation forms the primary method used to find the MPP of failure in the first order reliability method. The following steps outline the fixed-point iteration scheme used and indicate where the appropriate Rackwitz-Fiessler and Chen-Lind modifications are used to extend the original Hasofer-Lind formulation [47]:

1. Define an appropriate limit state function for the problem.
2. Assume a most probable point of failure in the design space, x^* .
3. Apply the Rosenblatt, Rackwitz-Fiessler or Chen-Lind approximations at x^* to convert any non-normal random variables into equivalent normal random variable.
4. Construct the linear approximation of the limit state function at x^* .
5. Based on the equivalent normal distributions, calculate the minimum distance from the origin to the linearized limit state function.
6. Update x^* with the new MPP of failure.
7. Repeat steps 2 through 6 until the MPP converges.

2.2.2 Second Order Reliability Method

The second order reliability method (SORM) extends FORM by creating a second order or quadratic approximation to the limit state around the assumed MPP of failure. The second order approximation to the limit state function is [33]:

$$\hat{g}(\mathbf{u}) = a_0 + \sum_{i=1}^n a_i (u_i - u_i^*) + \sum_{i=1}^n b_i (u_i - u_i^*)^2 + \sum_{i=1}^n \sum_{j=1}^{i-1} c_{ij} (u_i - u_i^*) (u_j - u_j^*) \quad (14)$$

Even though the SORM creates a more accurate approximation to the limit state at the MPP of failure it does so at the cost of requiring roughly n^2 more function evaluations. Rackwitz (2000)[124] points out that for 90% of all applications FORM provides plenty accurate estimations of the probability of failure, thus the added cost of SORM is generally not justified. One particular case where SORM may be required is when the Rosenblatt transformations are used to transform non-normal random variables to equivalent-normal random variables. As was stated previously,

the Rosenblatt transformations may distort the shape of the limit state function which would require better approximations of the limit state function at the MPP of failure.

2.3 Indirect Monte Carlo Simulation

So far in this chapter we have examined, in roughly chronological order, the developments of two of the three primary methods for assessing the probabilistic nature of a system. These methods started out with a discussion of traditional Monte Carlo simulation which assess the probabilistic nature of a system by using what is sometimes referred to as “the full probabilistics” and “the true analysis” [16]. The term “the full probabilistics” is in contrast to the analytic reliability methods which use approximations to the probabilistics to determine the probabilistic nature of the system. In both cases, these two classes of methods use the true analysis in conjunction with their probabilistic models.

An alternative approach would be to use a full probabilistic model with an approximation to the true analysis. This is the approach taken by the indirect Monte Carlo methods. The approximation to the true analysis is often called a metamodel. A metamodel is a closed form approximation to the original analysis that is created by observing the relationship between an analyses’ inputs and its outputs. The creation of metamodels in this manner falls under the general categories of Systems Identification [85] when the designer does not have control of the inputs and Design and Analysis of Experiments [106] when the designer can control the inputs and directly observe the outputs.

Although it is not exactly clear when researchers began using metamodels in conjunction with probabilistic analysis, one of the first uses was discussed in a papers by Schueller et al (1989)[147] and Bucher et al (1990)[16]. These papers predominantly advocate the use of advanced Monte Carlo methods for assessing the probabilistic nature of a response. Advanced Monte Carlo methods refer to stratified sampling and

adaptive-importance sampling. In the cases where the response was particularly costly to evaluate (e.g. in the case of a finite element code), Response Surface Methodology (RSM) [109] was used to estimate the limit state function around the means of the input variables and find the most probable point of failure in with an adaptive-importance sampling scheme.

The response surface equation (RSE) that Schueller and Bucher used was a simplified version of the traditional polynomial equation typically associated with RSM. Traditionally, RSM uses a quadratic approximation to the response based on the following equation:

$$\hat{g}(\mathbf{x}) = \beta_0 + \sum_{i=1}^n \beta_i x_i + \sum_{i=1}^n \beta_{ii} x_i^2 + \sum_{i=1}^{n-1} \sum_{j=i+1}^n \beta_{ij} x_i x_j \quad (15)$$

where,

$$\begin{aligned} \hat{g}(\mathbf{x}) &= \text{approximation to the limit state equation} \\ \beta &= \text{least squares regression coefficient} \\ x_i &= \text{input variable, } i \end{aligned}$$

The RSE that Schueller and Bucher used was chosen to minimize the number of function evaluations as much as possible yet still approximate the curved nature of the limit state function. The simplified function neglected all of the interaction terms, $\beta_{ij} x_i x_j$, to leave a pure quadratic interpolating function.

$$\hat{g}(\mathbf{x}) = \beta_0 + \sum_{i=1}^n \beta_i x_i + \sum_{i=1}^n \beta_{ii} x_i^2 \quad (16)$$

The Schueller-Bucher method creates an initial approximation to the limit state function around the mean values of the input variables, finds the MPP using this approximation, and then recreates the simplified RSE around the MPP. Importance sampling is then used with the MPP-based RSE to estimate the probability of failure.

Schueller and Bucher stated that their method had a function evaluation cost of $2n+1$ runs per iteration, with a total expected cost of $4n+3$ for the complete analysis.

The Schueller-Bucher method was further refined by Rajashekhar and Ellingwood (1993)[127] by adding a means to adaptively add in those interaction terms that were missing from Equation 16. Rajashekhar and Ellingwood noticed that for certain analyses, basing the importance sampling scheme only on an RSE with no interaction terms often resulted in significant error in the estimated probability of failure. In addition they noted that in many cases more than two iterations were needed to home in the MPP of failure. Consequently they concluded that the Schueller-Bucher method would often require two or three times more runs than the stated $4n+3$ total runs.

In order to keep total number of runs as low as possible but also account for the important interaction terms to keep the accuracy high, the Rajashekhar-Ellingwood method applies the following relative importance criterion to each term in the initial RSE.

$$\left(\frac{g_i - g_c}{g_c} \right) < \epsilon \quad (17)$$

where,

$$\begin{aligned} g_i &= \text{limit state computed at the } i^{th} \text{ experimental point} \\ g_c &= \text{limit state computed at the center point} \\ \epsilon &= \text{minimum relative importance} \end{aligned}$$

In the Rajashekhar-Ellingwood method, the initial RSE, based on Equation 15, is created around the means of the input variables. Equation 17 is then applied to each experimental run to separate important terms from unimportant ones. Only the important terms were then included in subsequent iterations. Each subsequent iteration is conducted in a similar manner as the Schueller-Bucher method.

Thus far in the discussion of indirect Monte Carlo simulation, the presented methods are really nothing more than a merging of the key ideas from the analytic reliability methods and the adaptive-importance sampling method from the direct Monte Carlo methods. The key ideas from the analytic reliability being the most probable point of failure and approximations to the limit states. Essentially the Schueller-Bucher and Rajashekhar-Ellingwood methods side-step the problems often associated estimating the numerical derivatives when creating Equations 10 and 14 by using a standard least-squares approach to create the approximate limit state.

In both of the two previous classes of methods, analytic reliability methods and RSE based importance sampling, the probability of failure for a given response is estimated by iteratively homing in on the MPP of failure. This iterative process can essentially be classified as an optimization procedure. Meaning that the next step or setting of the input variables is based on the current observed behavior in the response. The fundamental limitation to optimization schemes is that they can only be applied to one response at a time. Primarily because the setting of input variables that is “optimal” for one response is not guaranteed to be “optimal” for the next. Optimal in this case implies that the response has reached some sort of limit, for example, the maximum probability of failure.

For the development of a probabilistic design system for the Pratt & Whitney engine company, Fox (1994)[36] took a completely different approach to indirect Monte Carlo as compared to the Schueller-Bucher and Rajashekhar-Ellingwood methods. Fox noted that in engineering design the designer is typically interested in knowing the probabilistic nature of more than one response for a system as predicted by a particular analysis code [35]. For instance in addition to estimating the life of a part, a designer is also often interested in having probabilistic estimates of quantities such as the plastic growth, allowable stresses, burst margin, etc. Consequently, rather than run an iterative probabilistic analysis method for each desired response, Fox proposed

using statistical Design of Experiments (DoE) [190] to explore the relationships between the inputs and outputs for a code. A DoE approach allows the designer to systematically vary the inputs to a design code and examine as many of the outputs as desired. The experimentally obtained data could then be used to create RSEs for each of the desired responses.

With these response surface equations, the probabilistic nature of each response is then assessed using the traditional Monte Carlo simulation method. Traditional Monte Carlo can be used because each RSE only takes a fraction of a second to evaluate, as such, it becomes possible to collect millions of sample points from the response space in a relatively short amount of time.

The primary question that always arises when one implements a probabilistic design method with metamodels and traditional Monte Carlo is the question of accuracy of the metamodels. Unfortunately there is not a simple answer to this question. Several researchers have made cases for [37, 134, 170] and against [97, 165] the use of metamodels and traditional Monte Carlo for assessing the probabilistic nature of an analysis. To address many of these concerns, the method developed by Romero and Bankston (1998)[135] can be used to incrementally increase the accuracy of the necessary metamodels. The Romero-Bankston method is a sequential sampling method that allows the user to progressively increase the accuracy of the metamodel to match any complex function. The Romero-Bankston method has the additional and highly favorable characteristic of increasing efficiency as the size of the analysis increases [135].

2.4 Discussion of Probabilistic Design Methods

The goal of this chapter was to address the first research hypothesis posed in Section 1.6, that hypothesis was: Given the computational nature of the nozzle model, it is

hypothesized that an indirect Monte Carlo simulation technique is most likely the best probabilistic design method to use. To validate this hypothesis, we will need to compare the characteristics of the finite element nozzle model with the characteristics of the probabilistic design methods outlined in Sections 2.1, 2.2, and 2.3.

Section 1.5 outlined the proposed methodology that would be used to assess the probabilistic life of the nozzle. To summarize, the proposed methodology was comprised of a zonally partitioned finite element model and an indirect Monte Carlo simulation method. The zonally partitioned finite element model was created to mimic the complex, highly variable operating environment imposed on the nozzle. By zonally partitioning the model, the various thermal boundary conditions and mechanical loads could be varied at different spatial locations across the nozzle. This resulted in a parametric model with 52 independent input parameters.

In a similar fashion, the outputs of the model were also formulated with a zonal approach. Responses of life, temperature and stress were tracked at 85 historically low life regions on the nozzle. This resulted in 255 separate responses from the model.

In addition to having 52 design variables and 255 responses, the spatially partitioned finite element model is characterized by having a relatively long run time. The total wall clock time for one run is approximately 3.5 to 4 hours. This run time is primarily due to the large number of elements that make up the finite element model.

With these three model characteristics in mind, we can then extrapolate and come up with a list of criterion by which each of the probabilistic design methods can be judged. First off, there are two characteristics that we would like all of the methods to exhibit: efficiency and accuracy. If the method that we are using takes an unreasonably long time to reach an answer, then the method is of little use to us. Likewise, if the method will not reliably come to the correct answer then that method is of no use to us.

Since we are dealing with a large scale problem³, we would like a method that has either been directly developed to address large scale problems or one that can be easily adapted. Easily adapted implies that neither the accuracy nor the efficiency of the method break down as the problem size increases. Essentially this is a measure of how well the method scales with the number of input parameters and responses.

As was stated previously, the spatially partitioned finite element model estimated the life, temperature and stress at 85 different locations. Consequently we will want a method that is able to handle multiple responses simultaneously. Any method, regardless of how efficient it is, that needs to be re-executed for each response in this system would clearly violate our desired efficiency requirement.

One characteristic that finite element models often exhibit is that they are noisy. Noisy in this context means that small perturbations in the input parameters do not necessarily produce small variations in the responses. This is directly related to the smoothness or continuity of the response function [133]. Noisy models often pose significant problems for methods that rely on numerical derivatives.

One of the most difficult problems associated with using probabilistic design methods is determining the appropriate probability distributions for the inputs variables. It is not inconceivable that a designer will assume one set of distributions, only later to acquire some new information that shows the initial set of assumed distributions was wrong. As such we would like our probabilistic method to be relatively insensitive to the input distributions. Or as an alternative, if the input distributions are found to be wrong, then the method can quickly be reapplied to assess the impact of the new distribution set.

The next few paragraphs will qualitatively rank each of the previously discussed probabilistic design methods on their applicability to the outlined characteristics. Table 1 is a summary of the down-selection process.

³See Section 3.1 for a definition of a large scale problem.

Table 1: Ranking Matrix for Probabilistic Design Methods

		Efficiency	Accuracy	Large Scale Adaptability	Multiple Responses	Insensitive to Noisy Models	Insensitive to Input Distributions
Direct Monte Carlo Methods	Traditional Monte Carlo						
	Stratified Sampling						
	Importance Sampling						
Analytic Reliability Methods	FORM						
	SORM						
Indirect Monte Carlo Methods	Importance Sampling with Metamodels						
	Traditional Monte Carlo with Metamodels						
		Excellent: Good: Fair: Bad:					

Efficiency

Starting with our first requirement, efficiency, which by all accounts could be the most important criterion for our selected method due to the long run time of our model, we can immediately rule out two of the direct Monte Carlo methods. As was stated in Section 2.1.1, the number of runs required to observe an event occurring is inversely proportional to the probability of the even occurring. Consequently, for small probabilities of failure, a traditional Monte Carlo simulation is far too costly to use with our model. Thus stratified sampling methods can also be eliminated from consideration. Even though they often times require several orders of magnitude less data than traditional Monte Carlo, they too are deemed to expensive for this problem.

As far at the other classes of methods are concerned, all of the literature on the use of these methods suggests that they were created or adopted because of the inefficient nature of the traditional Monte Carlo methods. Therefore, any one of them is a potential candidate to use with the nozzle model.

Accuracy

It is widely stated in the literature that the direct Monte Carlo methods are the “gold standard” for accuracy [130]. In addition, the other two classes of methods (analytic

reliability and indirect Monte Carlo) have also been shown to be accurate across a wide range of problems. Consequently, any one of the probabilistic design methods could be used if accuracy was the only requirement, with a preference given to the direct Monte Carlo methods.

Large Scale Adaptability

Given the large scale nature of our nozzle model, we will need a technique that is applicable to large problems. The selection of the technique should be based on whether it has been successfully demonstrated on a similar large scale problem; or is most likely scalable to meet the needs of the analysis of large scale problems. Scalable in this context refers to how the method increases in the amount of data required as the number of input variables increases.

In terms of scalability, it is not exactly clear which methods will scale the best. Rackwitz (2000)[124] states that FORM and SORM have been demonstrated on problems up to 50 variables with favorable results. However some researchers have expressed concerns as to the scalability and accuracy of these two methods [148].

As far as the metamodeling based approaches are concerned, applicability to large scale problems has not really been addressed in the literature much. In fact as we will see in Chapter 3, there is only a hand full of potential methods available for large scale metamodeling.

Multiple Responses

The requirement that the probabilistic design method be able to simultaneously assess multiple responses immediately removes any iteration or optimization based method. Consequently, the adaptive-importance sampling techniques as well as the analytic reliability methods can not be used with the nozzle model to conduct a probabilistic analysis.

The traditional Monte Carlo, stratified sampling and the Monte Carlo via meta-modeling all are potential candidates based on this requirement because they are all purely independent, sampling schemes. Meaning, there is no feedback between the previous sample and the selection of the next sample.

Insensitive to Noisy Models

Insensitivity to noisy models is of concern primarily because we are using a finite element model to predict the life of the nozzle. Finite element models are often considered to be computationally noisy [133]. Computationally noisy in this context means that small perturbations in the input parameters do not necessarily produce small variations in the responses.

Noisy models often pose significant problems for methods that rely on numerical derivatives. Consequently any probabilistic design method that uses numerical derivatives to find the probability of failure will not work reliably with models that are computationally noisy. This then implies that the analytic reliability methods, potentially, are not good candidates for use with the nozzle model. The other classes of techniques do not need derivative information, thus this requirement does not preclude their use to solve the problem.

Insensitive to Input Distributions

Insensitivity to input distributions is of a more practical concern than a technical concern, and is related to the fact that it is often difficult to know ahead of time what the actual input distributions for each of the design variables should be. Thus any method that will require re-execution of the nozzle model in light of new input distribution information is not as attractive as a method that does not. Consequently, only the methods that rely on metamodels will meet this need. The reason being is that the data required to create a metamodel is not contingent on the design variables' input distributions.

Of the two metamodel based methods, only the traditional Monte Carlo simulation via metamodels meets this requirement. The adaptive-importance sampling method does actually require input distribution information to estimate the probability of failure.

Method Down Select

Having now compared each of the three main types of probabilistic design methods against the characteristics of the nozzle model, we have enough information to support or reject the research hypothesis that initiated the need for this chapter. Recall that research hypothesis was: Given the nature of the nozzle model, an indirect Monte Carlo simulation technique is most likely the best probabilistic design method to use. Based on the previous comparison, there is definite support for that statement.

This research hypothesis is supported by the results of two of the assessments: the requirement for multiple responses, and efficiency. First the requirement that the probabilistic method be applicable to systems with multiple responses immediately rules out all the methods other than the pure sampling methods. The pure sampling methods are traditional Monte Carlo, stratified sampling and Monte Carlo via metamodels.

Second, one of the stated requirements for finding or developing a method to assess the probabilistic life of nozzle was that it had to be very computationally efficient. This is primarily driven by the long run time of the nozzle model. Of the three methods left, traditional Monte Carlo, stratified sampling and Monte Carlo via metamodels, only the metamodeling approach meets the efficiency need.

Thus based on this assessment, the hypothesis that an indirect Monte Carlo method simulation method is indeed that best choice of probabilistic design method to use with the nozzle model is supported.

Assertion: A spatially partitioned finite element model was created to address the complex operating environment that a nozzle is exposed to. This model has 52 variables, 255 responses and takes 3.5 hours to run.

Assertion: To capture the probabilistic nature of the life of nozzle, a probabilistic design method is needed. The primary methods are direct Monte Carlo, indirect Monte Carlo and analytic reliability methods.

Research Question 1: Which of the three primary probabilistic design methods is most applicable for the nozzle problem?

Research Hypothesis 1: Given the computational nature of the nozzle model, it is hypothesized that an indirect Monte Carlo simulation technique is most likely the best probabilistic design method to use.

Validity of Hypothesis 1: Based on the results of probabilistic method comparison, two requirements drove the down selection process, the need to simultaneously handle multiple responses and method efficiency. Of all the methods only traditional Monte Carlo simulation used in conjunction with metamodeling concurrently addressed both of the requirements. Consequently, the hypothesis that an indirect Monte Carlo method based on traditional Monte Carlo and metamodels is indeed a valid statement.

2.5 Summary

The goal of this chapter was to support or refute the first research hypothesis stated in Section 1.6, that hypothesis was: Given the nature of the finite element model created to predict the life of the nozzle, an indirect Monte Carlo technique the is most appropriate method to realize the probabilistic nature of the nozzle life. This hypothesis was made based on the characteristics of the nozzle model as well as the experience of the author with applying probabilistic design methods to other similar

problems. The pertinent nozzle model characteristics that drove this hypothesis were the large numbers of input variables (52), the large number of responses (255) and the long run time of the model (3.5 hours).

To support or refute this hypothesis, this chapter then reviewed the three primary classes of probabilistic design methods. The review comprised of several direct Monte Carlo simulation methods, the two primary analytic reliability methods and several indirect Monte Carlo simulation methods. The review of each of the methods followed a roughly chronological order, starting with a basic history of the methods (i.e. what field did they originate in and what problems were they initially intended to solve), and finishing with an overview of the fundamental equations for each method.

After reviewing each of these probabilistic design methods, the characteristics of these methods were then compared to a suite of requirements. The selected requirements are based on the needs of the overall analysis, as well as the constraints that the nozzle model imposes on the analysis. These requirements were overall method efficiency, accuracy, large scale adaptability, applicability to multiple responses, insensitivity to noisy models and insensitive to changes in input distributions. Based on the results of this comparison, two requirements drove the down selection process, the need to simultaneously handle multiple responses and efficiency. Consequently, the hypothesis that an indirect Monte Carlo method based on traditional Monte Carlo and metamodels is indeed a valid statement.

Chapter III

REVIEW OF LARGE SCALE METAMODELING METHODS

The goal of this chapter is to support or refute the second research hypothesis stated in Section 1.6, that hypothesis was: Given the spatially partitioned nature of the finite element model, it is hypothesized that a partitioned metamodeling scheme seems like a logical choice for reducing to the total number of runs required to create an accurate metamodel.

To support or refute this hypothesis, this chapter will introduce and outline all of the general classes of currently existing large scale metamodeling techniques. However, before getting into the review of the actual methods, a definition of just what of a large scale system is will be formulated. This definition will help to orient the subsequent sections on large scale metamodeling techniques.

As we will see from the literature, there are two general classes of approaches that are readily apparent for minimizing the required data to build accurate metamodels for large scale systems: those that partition the design variables into groups and those that take an all-at-once approach. Figure 7 will be used as a guide to summarize a general taxonomy of the various large scale metamodeling techniques and how they relate to each other.

Shown at the bottom left hand corner of the taxonomy is an empty space where no methods currently exist. As we will see, that empty spot is a place holder for a method that will be hypothesized to be the most applicable method to support the research hypothesis that is motivating this chapter.

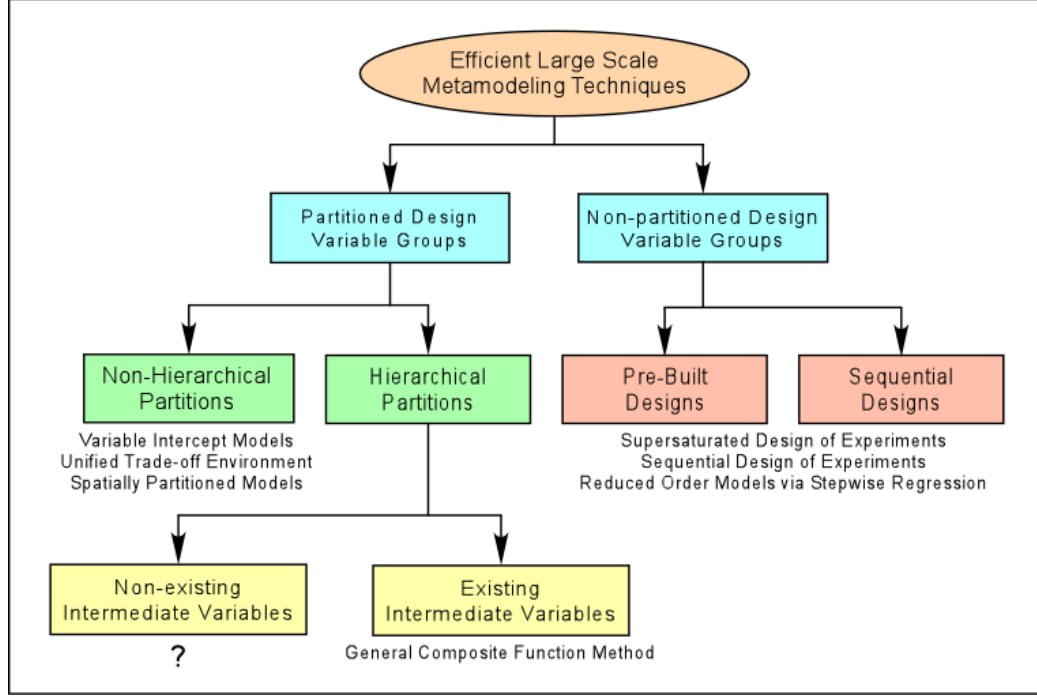


Figure 7: Taxonomy for Large Scale Metamodeling Techniques

3.1 *Definition of the Large Scale Problem*

Since this thesis is primarily devoted to developing a means of enabling probabilistic design for the first stage nozzle model introduced in Section 1.5, which can consider a large scale problem, a definition of exactly what is meant by a large scale problem is in order. The term “large scale” is a rather ambiguous description and means different things to different engineers. However, the scope of its meaning can be narrowed by first identifying the goal and process of many engineering design exercises.

When one designs something, whether it be an airplane or a bracket on that airplane, the objective of that process is to create an adequate design that meets the required specifications for that system. To make the best possible design decision, the designer would like to evaluate many alternative designs and pick the one that best meets the requirements [29].

The evaluation of alternative designs can be classified as design space exploration.

Design space exploration (DSE) simply means a process of gathering information about the design space of a system. The design space of a system is the collection of all possible values of the design metrics for that system. A design metric in this case implies a characteristic or response of the system that a designer uses to assess the quality of the design. The exploration of the design space therefore entails varying the system design variables and observing the resultant design metric values. A design variable is simply a parameter that a designer can independently set or vary to see the impact on a design metric.

Optimization is one particular example of a design space exploration procedure. There are many different types of optimization techniques that one can use to explore a design space, however optimization in and of itself is not the focus of this thesis. Another common design space exploration procedure is the exploration for the generation of metamodels. Design space exploration in the context of metamodeling is synonymous with cartographic exploration. Cartographic exploration is exploration of a land mass with the goal of creating a representation or map of that land mass. With respect to design, a metamodel is therefore a map of the design space for a system. This map is created from multiple representative samples from the design space. In modern engineering design, the design process typically involves the use of an engineering design code to simulate the system (i.e. produce samples from the design space). Often times these codes take a non-trivial amount of time to analyze one combination of input parameters. To make good design decisions one would like to analyze as many combinations of input parameters as possible where each combination represents a different design from the design space. However the designer is always faced with time constraints which will limit the total amount of possible design exploration. Metamodels can often aid the designer in analyzing large numbers of alternative designs because they are a simpler representation of the complex design code. These simpler models typically have significantly faster execution times

as compared to their originating engineering code.

The thoroughness with which one needs to explore the design space is dependent on how much initial information is known (or can be assumed to be known) about the system. The less that is known a priori, the more exploration that is required. For example, in relation to metamodeling, one typically assumes a general topology for the underlying design space (e.g. the design space is quadratic), as such one only needs to collect enough information to create a metamodel to address that topology. If on the other hand, one does not have any inkling as to the nature of the design space topology, one will need to collect data until the design space is sufficiently resolved.

Prior knowledge is one factor that influences the amount of data that must be collected from the system to create a representative metamodel of the design space. Another and perhaps more important effect on the amount of data required is directly linked to the number of input parameters that one is varying to resolve the design space. Intuitively, the more design parameters that a system has, the more combinations of parameter settings one needs to examine to create an accurate map. As an example, in standard Response Surface Methodology (RSM)[109] the minimum amount of data required to create a 2^{nd} order response surface equations (RSE) grows quadratically with the number of input parameters. Thus as one increases the number of independent parameters to examine when exploring the design space, there is a rapid, non-linear increase in the required samples from that space.

At this point an adequate definition of a large scale system can now be given. To create a metamodel, data must be collected from the system in question. From a pure efficiency standpoint, it is advantageous to create this metamodel with as little data as is necessary. As was previously outlined, there are several characteristics of creating metamodels that tend to increase the amount of data required, namely the lack of prior knowledge of the topology of the design space and the number of design variables to be explored. However, neither of these two characteristics in and of

themselves would constitute a system as being large scale. The additional aspect that makes a system a large scale system is the run time of the analysis. A long runtime in conjunction with a required large sample from the design space constitutes a large scale system.

The reason for this definition can be argued from resources stance. Essentially, the more time an analysis takes the more money it costs to perform. Thus any approach that is capable of decreasing the total samples required while at the same time creating an accurate metamodel for a large scale system is worth researching.

The following sections in this chapter will introduce and outline the general classes of metamodeling techniques that have been adopted or created to address large scale problems. Following the review, each large scale metamodeling scheme will be evaluated on its applicability to the nozzle problem.

3.2 Non-partitioned Approaches

“Non-partitioned” approaches refers to what are often called “all-at-once” approaches. All-at-once in this context refers to any method that systematically evaluates all the design variables concurrently. Essentially all classical Design of Experiments (DoE) techniques adopt this approach. This is in contrast to any approach that first divides up the design variables into groups and then evaluates each of the groups separately. Partitioned approaches are examined in the next section of this chapter.

For large scale systems where decreasing the total number of experiments performed is desirable, the choice of when to choose a partitioned approach versus an all-at-once approach is system dependent. One can certainly imagine a system where the determination of the partitions is not readily obvious, and thus an all-at-once approach is the logical choice. Conversely, there are systems where the design variables are easily partitioned, thus a partitioned approach for run minimization may be the better choice.

The all-at-once approach for experimentation can roughly be divided into two camps, those methods that build a design a priori and those that build the design sequentially, that is, as the design space is being explored. In either case, to adopt these approaches to large scale systems, there is one underlying assumption that these methods all employ, the notion of *effect sparsity* [12]. Effect sparsity refers to the hypothesis that in large scale systems not all effects are important for the variability of the response. In this context, an effect implies not only variability due to a specific factor but also combinations of factors as well (i.e. the interaction terms in a standard response surface equation).

For large scale systems, effect sparsity would allow the experimenter to create a design with an acceptable number of runs and then build a reduced order model based on the data collected. A reduced order model is a model that does not contain all possible effects. Typically one would employ one of the various stepwise regression routines such as forward selection, backwards elimination, stepwise regression or all-subsets regression. Miller (1990)[104] gives a thorough review of each of the common stepwise regression methods. Note however, these regression methods should not be blindly applied, Abraham et al (1999)[1] identified several risks associated with applying these stepwise regression routines. Essentially they concluded that these methods can not be assumed to reliably separate the active effects from the inactive effects.

The reason for the breakdown of stepwise regression routines when used with supersaturated designs is due to the complex nature of the aliasing structure associated with these designs. An alias is defined when the estimate of an effect also includes the influence of one or more other effects (usually higher order interactions). For example, if the estimate of effect D in a four factor experiment actually estimates $(D + ABC)$, then the main effect D is aliased with the 3-way interaction ABC [111]. Consequently, the experimenter does not know if the actual observed variability on

the response is due to D or ABC or actually a combination of the two.

3.2.1 Pre-built Designs

The only class of designs of experiments that are truly feasible for large scale systems are supersaturated designs. A supersaturated design is technically any design consisting of k factors and N runs where $N < k + 1$. Historically, this definition was based on a linear, first order model consisting of only an intercept and the k main factors. However, it can be extended to include any assumed linear model where the total number of effects or terms in the model is greater than $N - 1$ runs.

Typically, supersaturated designs are custom built by an exchange algorithm. In recent years there has been an explosion in the number and types of exchange algorithms that have been published. These include variations on the classical Fedorov exchange algorithm [105], columnwise [60, 89] and columnwise-pairwise algorithms [87] and stochastic algorithms built around simulated annealing [4, 69, 98] and genetic algorithms [20].

All of these exchange algorithms are faced with the same general problems, which are summarized below:

- Finding a optimum design is a NP-hard optimization problem. Thus, as Holcomb (2002)[64] states, no one algorithm is clearly superior to any other.
- No exchange algorithm is guaranteed to find the optimal solution. This is due to the fact that several dissimilar designs can have the same optimality criterion value and therefore judged equal by the algorithm [20].
- As the number of factors and experiments increases the computational effort to find the best design grows non-linearly, thus large designs can often take days or months to create.
- Even though a supersaturated design may be deemed optimal it will in general

have a complex aliasing structure which could potentially make the practical interpretation of the experimental results difficult or non-unique [118].

As was previously stated, one of the primary drawbacks to using exchange algorithms to build optimal supersaturated designs is the upfront computational expense. When this expense is unacceptable, there are several alternative classes of supersaturated designs available. Lin (1993)[88] created a classes of supersaturated designs built on half fractions of Hadamard matrices. He showed that these matrices were superior to the other existing supersaturated designs (prior to 1993) and, more importantly, were relatively easy to construct. His designs allowed for the estimation of $k = N - 2$ effects in $n = N/2$ experiments. N in this case refers to the order of the Hadamard matrix.

Another computationally inexpensive route that has been explored for adapting supersaturated designs to computer simulations are latin hypercube designs. Latin hypercubes can be quickly designed to meet any computational experimentation budget. However the typical latin hypercube algorithm is not guaranteed to produce an optimal design. To address this limitation, several authors [10, 116, 195] have developed algorithms to design near optimal latin hypercubes. These algorithms produce comparatively good designs, but typically with a significant increase in computational expense as compared to the standard latin hypercube algorithm. This class of designs has also been shown to be well suited to the exploration of deterministic computer simulations where space filling designs tend to minimize the bias error between the fitted model and the actual simulation [44].

3.2.2 Sequential Designs

As was stated in the previous section, the standard application of designs of experiments is to build a complete design, collect all the data, and then build a model around

that data. A natural extension to this approach is sequential experimentation. Sequential experimentation essentially performs all three of these tasks simultaneously. A typical sequential method starts with a base set of experiments, then collection of the data, followed by the building of a cursory model, and finally, extension of the design [141]. This approach allows the experimenter to modify or adapt the subsequent designed experiments in light of the data collected [43]. The experimental process can then be stopped when a sufficiently accurate model is built or the total computational budget is met.

Sequential designs inherently imply a tight coupling between the design of the computer experiments and the generation of the metamodel. Practically speaking, for computer experiments this seems to be a logical approach to experimentation due to the relative ease with which the experimenter can programmatically link the collection and analysis of the data. Also, as Sacks et al (1989)[142] point out, sequentially designed experiments are computational cheaper to generate than pre-built designs. This is due to the fact that the creation of an optimal pre-built design is a single $n \times d$ optimization problem, while a sequentially built design is a sequence of d -dimensional optimization problems. It is important to note however, that even though sequentially designed computer experiments are a logical approach to experimentation on the computer, the process is limited to one response at a time. If the experimenter needs to build models of more than one response the sequential process will have to be performed for each response.

3.3 Partitioned Approaches

The previous section summarized several feasible all-at-once approaches for large scale systems. As was noted, to apply supersaturated designs to large scale systems, the exploitation of effect sparsity plus the use of a stepwise regression routine was necessary to keep the computational expense reasonable.

This section adopts a slightly different paradigm for addressing the problem of size in deterministic experiments. The new focus will examine several different methods that address the large scale problem by first partitioning the factors into subgroups, analyzing each of the subgroups independently and then recombining the data to build the final metamodel.

Although several different partitioning approaches will be examined, in general they have the same underling assumption. This assumption is based on the idea that from the total set of design variables for the system, one can subdivide the set into disjoint subgroups. Further, it is assumed that the interactions between individual design variables in disjoint subgroups are of negligible importance to the overall variability to the response. However, interactions between individual factors of the same subgroup are important and thus are captured. The hierarchical partitioning techniques discussed in Section 3.3.2 will expand on this assumption by allowing the metamodel to capture general partition to partition interactions.

Essentially the assumption underlying the partitioning methods is implying nothing more than effect sparsity. However, in this case the experimenter is specifying the active effects rather than using a step-wise regression routine to find them. Granted, the resultant metamodels for each of the partitioned methods tend to conceal the direct observation of effect sparsity (as compared to the all-at-once approaches), nonetheless, effect sparsity is still the underlying assumption.

On the left side of Figure 7 the partitioned approaches are divided into two general categories, hierarchical and non-hierarchical partitioning methods. For the purpose of this thesis, two basic types of partitioning can be defined as follows:

Hierarchical A multilevel subdivision of system variables and/or responses such that parent-child relationships are readily apparent

Non-hierarchical A single level subdivision of system variables such that there are

no obvious parent-child relationships between the variables and responses

A review of the literature identified significantly more techniques centered around the all-at-once type approaches. On the partitioned side, there seems to be only a few techniques that are adaptable or have been created for addressing the “curse of dimensionality” associated with large scale computer simulations.

3.3.1 Non-hierarchical Partitioning

3.3.1.1 Variable Intercept Model Methods

There are two methods that can fall under the heading “Variable Intercept Models”. Although these two methods are essentially different, they employ the same general mathematical description. The mathematical formulation is as follows:

Responses of partition 1:

$$\hat{y}_1 = \beta_{0,1} + \sum_{i=1}^k \beta_i x_i + \sum_{i=1}^k \beta_{ii} x_i^2 + \sum_{i < j} \beta_{ij} x_i x_j \quad (18)$$

Responses of partition 2:

$$\hat{y}_2 = \beta_{0,2} + \sum_{i=k+1}^r \beta_i x_i + \sum_{i=k+1}^r \beta_{ii} x_i^2 + \sum_{i < j} \beta_{ij} x_i x_j \quad (19)$$

Intercept for partition 1:

$$\beta_{0,1} = \alpha_{0,1} + \sum_{i=k+1}^r \alpha_i x_i + \sum_{i=k+1}^r \alpha_{ii} x_{ii}^2 + \sum_{i < j} \alpha_{ij} x_i x_j \quad (20)$$

Intercept for partition 2:

$$\beta_{0,2} = \alpha_{0,2} + \sum_{i=1}^k \alpha_i x_i + \sum_{i=1}^k \alpha_{ii} x_{ii}^2 + \sum_{i < j} \alpha_{ij} x_i x_j \quad (21)$$

As can be seen from this set of equations, the set of responses and design variables is divided into two partitions (although both references [74] and [119] state that, in theory, more partitions are possible). The equations linking the responses and their

corresponding design variables are standard second order response surface equations. The interesting deviation from standard Response Surface Methodology is in Equations 20 and 21. In this case the intercept terms are formulated to be functions of the variables in the other partition. One interpretation on the functional form of the intercepts, as Koch et al state (2000)[74], is to allow the mean term of the equations to slide along the y-axis as the intercept factors change.

As stated earlier, both methods share the same mathematical descriptions; however, their methods and goals are different. The primary goal of Koch's formulation was for large scale system run minimization for computer simulations [73]. The focus for Perry et al (2001, 2002)[118, 119] was on developing a technique to examine the effect of multiple factors on sequential processes. Given this, the primary method differences are associated with how the variable intercept is actually used. Koch's use was stated earlier, namely as a true variable intercept. Perry, on the other had, uses the intercept to estimate additional significant effects on the response due to factors in the other partition. These effects are then included in the final response surface which employs a static intercept.

Another interesting deviation between the two methods is in the design of experiments used to explore the relationship between the design variables and the responses. Prior to choosing the type of design used to explore each partition, both methods sanction one of several methods for determining the partitions:

- A screening test to quantitatively separate responses and factors into partitions
- Engineering intuition/domain knowledge
- Process or engineering code boundaries

Once separated, Koch then uses two separate DoEs to examine each partition independently. In general these can be standard classical DoEs or DoEs designed explicitly for computer experiments.

Perry's DoE is a little more involved. First rather than examining each partition independently, a custom DoE is built that, although partitioned, examines the influence of all factors concurrently. The partitioned DoE first starts with a small composite design (SCD) for the partition with the largest number of factors. This DoE essentially defines the total number of runs that the overall partitioned DoE will contain. Then the experimenter generates another SCD design for the other partition. For the fractional factorial portion of the SCD, the design is built such that it is nearly orthogonal, has near-equal occurrence of highs and lows for each factor and minimizes the complexity of the aliasing structure. The axial points, of the second partition SCD and the overall centroid for the total partitioned design are constructed in the same manner as in a standard central composite design.

Returning for a moment to the previously stated three methods for partitioning the problem, its important to note that these three methods are applicable to all the partitioning methods outlined in this thesis. However, to the author's knowledge, only the the last two methods (engineering intuition and process boundaries) have been demonstrated in the literature. Although Perry and Koch state that a screening test can be used to quantitatively partition the factors and/or responses, neither author provided a description of how to solve the partitioning problem in a quantitative fashion. A discussion of possible means to quantitatively partition a large scale problem will be taken up in Chapter 4.

3.3.1.2 Unified Trade-off Environment

The Unified Trade-off Environment (UTE)[7] stemmed from research aimed at developing a method to simultaneously model the impacts of mission requirements, vehicle characteristics and technologies in aircraft systems design. In his thesis, Baker (2002)[6] proposed two alternative environments for exploring these effects, an integrated environment and a multi-space environment. The integrated environment

essentially takes the all-at-once approach for combining the various factors/design variables associated with describing mission requirements, vehicle characteristics and the desired technologies. Baker found that an all-at-once approach could not satisfactorily be adopted to his needs, thus he devised a scheme that logically partitioned the factors into the corresponding three design spaces.

The basic UTE method starts with a baseline vehicle and then formulates the impacts of the various spaces into the form of “deltas” around that baseline. As was stated earlier, Baker’s formulation was built around only three partitions, namely the mission requirements, vehicle characteristics and the technologies. However, in theory this could be expanded to any number of partitions and the generic functional relationship would be as follows:

$$Y_m = \beta_{0,overall} + \sum_{i=1}^n \Delta Y_{m,i} \quad (22)$$

where,

$\beta_{0,overall}$ = the baseline contribution to Y_m plus the β_0 from each partition

In the development of his method, Baker realized that the accuracy of the the UTE method was hampered by neglecting some important interactions that may occur between variables in disjoint groups. Thus to improve the accuracy of the method, Baker suggested using a screening test to determine important interactions that may be occurring but were not being accounted for and then used variable redundancies between partitions to directly capture important interactions.

3.3.1.3 Spatially Partitioned Metamodels

For certain classes of problems, which involve the of building metamodels of finite element models, it is possible to exploit the spatial nature of the system to create accurate metamodels with fewer runs [171]. The notion of effect sparsity can be

enforced by assuming that the influence of a variable (i.e. its interactions with other variables) is spatially localized. Volovoi et al (2003)[171] justify this approach by using the St-Venant principle which characterizes the decay rate of disturbances in the stress field by $\sqrt{G/E}$ where E and G are the Young's modulus and shear modulus of the material, respectively. Essentially what this implies is that only the interactions between neighbors need to be considered. In the general case, it is important to evaluate the decay rate and the spatial distance among zones in order to determine what constitutes the proper neighborhood for each variable.

The general spatially partitioned DoE creation process is as follows:

1. Determine the interconnections between the spatial partitions or zones. Each adjacent zone will represent an interaction to be captured in the model.
2. For a given interaction (i.e. a pair of adjacent zones), apply a fractional factorial DoE that explores the interactions between all design variables that will be varied across those two zones. All other design variables are kept at their nominal values.
3. Apply the same fractional factorial approach for all the other pairs of zones by appending the new runs to the end of the spatially partitioned DoE.
4. Execute the partitioned DoE as a standard DoE method.
5. Build the spatially partitioned metamodel by manually including only the interactions defined in Step 1; all first and second order terms are handled normally.

3.3.2 Hierarchical Partitioning

Up to this point all the partitioning techniques have, for the most part, been of a non-hierarchical nature. Koch's variable intercept method could perhaps be argued to be hierarchical since the intercept terms are themselves functions, thus masking the direct influence of factors in the other partition on the response. However, given its

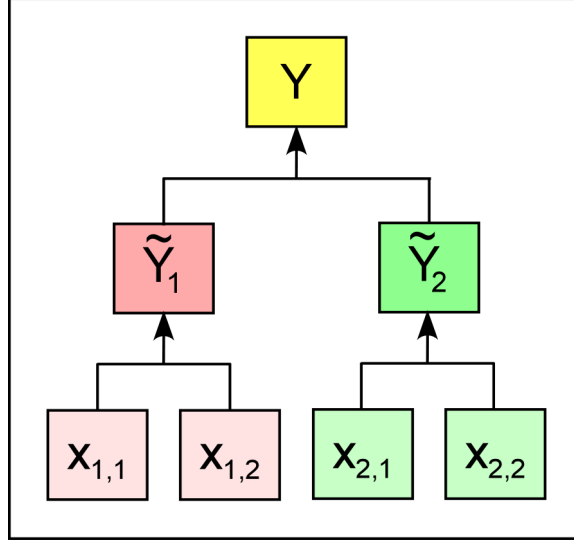


Figure 8: Generic Hierarchical Representation

mathematical similarity with Perry’s variable intercept method, which was formulated to be non-hierarchical, it was included as a non-hierarchical technique.

In general, both of the hierarchical techniques to be discussed in this section can be notionally depicted by Figure 8. This figure shows a generic system of four design variables denoted by the $x_{i,j}$ ’s divided into two subdivisions or partitions. These partitions are linked to the general response, Y , via the intermediate responses, \tilde{Y}_i ’s.

The hierarchical techniques examined in this paper can be distinguished by the nature of the intermediate responses, \tilde{Y}_i ’s, shown in Figure 8. As shown in Figure 7, in general these intermediate responses can be categorized by either being existing or non-existing. Existing in this case implies that the intermediate response is an actual response or intermediate variable that exists in the code and can be directly manipulated by the experimenter.

3.3.2.1 Existing Intermediate Responses

The basic mathematical relationship describing the mapping from the low-level design variables, x ’s, to the top level response, Y , via the intermediate responses, \tilde{Y}_i ’s, can

be generically described as a composite function (i.e. $f \circ g \equiv f(g(x))$). This is the most intuitive type of hierarchical relationship, and for certain classes of problems can be easily adapted for use as a large scale partitioning technique.

The General Composite Function (GCF) method can be outlined with a simple example. Let us assume we are trying to develop a metamodel of the range of the aircraft from an aircraft synthesis and sizing code. Based on the the Breguet range equation, the range of the aircraft is among other things a function of the lift to drag ratio (L/D) and the specific fuel consumption (SFC) of the airplane.

$$R = \frac{V (L/D)}{g \cdot SFC} \ln \left(\frac{W_i}{W_f} \right) \quad (23)$$

where,

- R = range of the aircraft
- V = flight velocity
- L/D = lift to drag ratio
- g = acceleration due to gravity
- SFC = specific fuel consumption
- W_i = initial aircraft weight
- W_f = final aircraft weight

It follows from basic aerodynamics that the L/D is itself a function of more primitive design variables directly associated with the wing of the airplane. These primitive design variables include aspect ratio, sweep, the various airfoil parameters, etc. The SFC on the other hand is a function of engine parameters such as overall pressure ratio, turbine inlet temperature, etc. Thus applying the basic assumption of all partitioning techniques, we can surmise that interactions between the wing parameters are important but interactions between the various wing and engine parameters are not as important, and thus can be neglected. Using this basic logic we can then

easily justify subdividing of the design variables into two partitions, a wing specific partition and an engine specific partition. Each partition can now be independently explored and the functional mappings between the partitioned design variables and their corresponding intermediate responses can be determined.

To complete the method, the mapping between the intermediate responses and the top-level responses needs to be explored. In the synthesis and sizing example, the influence of the lower level design variables on the range response was lumped into the L/D and SFC intermediate responses. In many synthesis and sizing codes to calculate the range of the aircraft, the user can either supply a specific L/D or specify settings for the more primitive design variables and use the codes internal aerodynamic routines to calculate the L/D. In the GCF method, both of these analysis avenues are exploited. To complete the sublevel analyses, the internal aerodynamic routines are used to explore the mappings between the wing specific design variables and the L/D intermediate response¹. To determine the mapping between the intermediate responses and the top-level response, the user specified values for the intermediate responses (or in this case, what could be called intermediate design variables) are used. Figure 9 shows the Breguet Range example mapped to the general hierarchical representation.

To apply the GCF method there are two basic caveats that must be observed:

1. The ranges of the intermediate responses used to examine the intermediate to top-level mapping must fall within the range of the intermediate response as predicted by the low level to intermediate level mapping.
2. The code must allow the user to transparently examine and set the intermediate responses.

¹Note - it is not necessary to require the experimenter use the sizing and synthesis code's internal aerodynamic routines. In fact one could instead use a separate, higher fidelity aerodynamics code to model the mappings between the low-level wing parameters and the L/D intermediate response. The same approach could be used for the engine parameters as well.

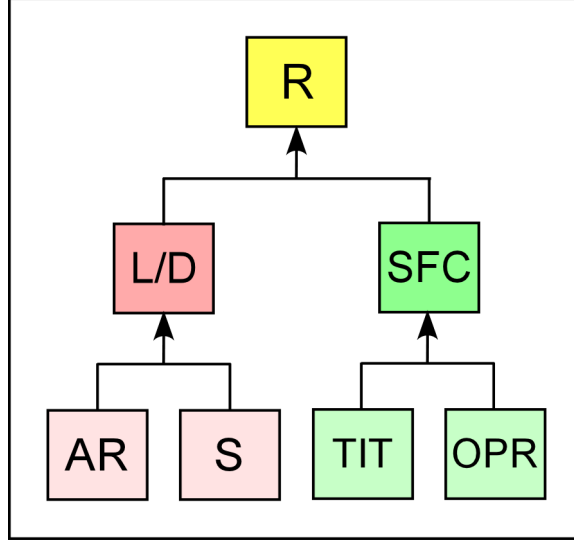


Figure 9: Breguet Range Hierarchy

Caveat one is a direct result of one of the primary limitations to metamodels, namely extrapolation. Metamodels are generated by varying the input parameters over a prescribed range of influence. The resultant response then has its own region of validity as dictated by the ranges on the input parameters. As long as the user of the metamodel stays within the chosen ranges of the input parameters, the metamodel is acting as an interpolation engine and thus can be assumed to be accurate. However, once the user steps outside of the bounds of the input parameters, the extrapolated response as predicted by the metamodel can not, in general, be assumed to be an accurate representation of the analysis that it is modeling.

This interpolation versus extrapolation characteristic has a direct impact on the accuracy of the hierarchical metamodel generated by the General Composite Function method. It is important that the range of the intermediate response/variable falls within the interpolation bounds as predicted by the sublevel design variables to intermediate response mapping.

Caveat two identifies the inherent difference between the two types of hierarchical partitioning techniques. In the aircraft range example, the L/D and the SFC are

readily identifiable intermediate responses and if the analysis code supports it, could themselves be independently varied to find the intermediate to top-level mapping. Unfortunately not all codes support this level of transparency/flexibility, in addition there might not be any readily identifiable intermediate responses for the analysis in question. It is this very problem that lead to the development of the of the hierarchical metamodeling technique that comprises one of the primary contributions of this these.

3.3.2.2 Non-Existing Intermediate Responses

As was stated in the the previous section, analysis codes that do not exhibit the ability to collect and manipulate intermediate responses pose a road block to the GCF method for large scale partitioning problems. In addition, one can readily imagine systems in which readily identifiable intermediate responses do not exist. It is these systems that will be addressed by the hierarchical metamodeling method developed for this thesis.

As an example, a typical finite element analysis (FEA) is such a system that does not contain any directly identifiable partitioning intermediaries. In standard FEA codes, the user gets a direct mapping from the inputs to the outputs. Thus, there are no intermediate responses around which to partition the lower level design variables.

On the surface, this would seem to negate the possible use of a hierarchical partitioning technique, and would require that the experimenter adopt one of the other schemes outlined in this section to attack a large scale FEA problem. Fortunately, it is possible to artificially create intermediate responses that are germane to the problem at hand. Chapter 6 introduces and outlines the steps for applying a new hierarchical metamodeling method that was developed to address large scale systems comprised of multiple responses and that do not have readily identifiable intermediaries.

3.4 Review and Discussion of Current LSM Methods

Recall that the primary goal of this chapter is to support or refute the second research hypothesis stated in Section 1.6, that hypothesis was: Given the spatially partitioned nature of the finite element model, it is hypothesized that a partitioned metamodeling scheme seems like a logical choice for reducing to the total number of runs required to create an accurate metamodel.

To investigate this hypothesis, this chapter outlined two general classes of techniques applicable for run mitigation for large scale systems. Both avenues, either directly or indirectly, exploit the notion of effect sparsity to justify the generation of reduced order models. Effect sparsity refers to the notion that in large scale systems not all effects are important for the variability of the response. Employing this assumption allows the experimenter to create reduced order models which require less data than complete models. The distinction being that complete models contain all potential effects on the variability of the response, while the reduced order models do not.

The first set of approaches outlined revolve around the use of supersaturated designs of experiments in conjunction with a stepwise regression routine to build an accurate metamodel. The second set of approaches partition the total set of design variables into smaller, disjoint subgroups. The subgroups are independently analyzed and the resultant data combined to create the final metamodel. For the partitioned approaches, effect sparsity is directly enforced by the experimenter. Conversely, the all-at-once approaches use a stepwise regression routine to determine the level of effect sparsity inherent to the system.

With this review in mind, we can now support or refute the research hypothesis that drove the literature search in this chapter. First, yes it does appear that a partitioned metamodeling approach does make sense for the analysis. The primary

reason being, the alternative approach for building metamodels of large scale systems, supersaturated designs with stepwise regression, may not reliably create metamodels that represent the true nature of the system. The reason for this is due to the complex aliasing structure of supersaturated designs. The consequence of the aliasing problem is that even though the stepwise regression routine may produce a good metamodel (i.e. one that accurately represents the shape of the modeled function), that model will most likely not be an accurate representation of the system. This implies that one should not use that model to make any statements as to the statistical nature of the system.

Assertion: Given the constraints imposed by the nozzle model’s complexity, traditional statistical linear metamodeling methods are not efficient enough to be used with the model. Consequently a large scale metamodeling method is necessary to reduce the total computational expense as much as possible.

Research Question 2: Which large scale metamodeling procedure is best to address the characteristics of the nozzle model?

Hypothesis 2: Given the spatially partitioned nature of the finite element model, it is hypothesized that a partitioned metamodeling scheme seems like a logical choice for reducing to the total number of runs required to create an accurate metamodel.

Validity of Hypothesis 2: For the most part, it does appear that a partitioned metamodeling approach does make the most sense for the analysis. The primary reason being, the alternative approach for building metamodels of large scale systems, supersaturated designs with stepwise regression, may not reliably create metamodels that represent the true nature of the system. The reason for this is due to the complex aliasing structure of supersaturated designs.

However, none of the current large scale metamodeling methods meet all of the needs of the nozzle problem. Consequently, a new partitioning method is needed to be able to fully leverage the partitioned nature of the nozzle model.

If a supersaturated method for large scale metamodeling is not a feasible approach, which then of the partitioned approaches is best? Answering this question is a little less straight forward. A review of the literature revealed three potential non-hierarchical partitioning methods and one hierarchical partitioning method. The first two non-hierarchical methods, the Variable Intercept Model and Unified Trade Off Environment, can quickly be neglected.

The VIM methods were created primarily for partitioning along process or code boundaries, that is the responses for one process and their important factors were grouped together and the responses for a subsequent process and its important factors formed the second group. For the model that we are trying to partition, there is only one process and it is not at all clear how we should group the responses. Additionally, its even less clear how we would determine which factors are most important for a given set of grouped responses. Consequently the VIM methods do not appear to be a good candidate for creating partitioned metamodels for the nozzle problem.

The UTE method can also be bypassed by closely examining Baker's stated accuracy and run results. In Baker's initial formulation of the UTE method, each of the partitions were completely disjoint from one another. To assess the accuracy of the UTE method, Baker ran two types of validation tests. One set of tests was on the predictive accuracy of each of the individual partitions, the other assessed the predictive accuracy of the combined set of partitions as depicted in Equation 22. Baker's validation tests consisted of running several hundred off-design points through the analysis code and then comparing the predictive accuracy of the metamodels with the results from the code.

The accuracy results for the individual partitions showed that each partition had

a mean error of less than 1% and a standard deviation around that mean of approximately 0.25 to 0.75%. For typical metamodeling purposes, relative errors in this range are usually quite acceptable. However when Baker analyzed the predictive accuracy of the complete UTE formulation he found the error rates considerably higher. The mean error rates for the combined equation were on the order of 6 to 7% with maximum errors as high as 43%.

Baker hypothesized that the increase in error was due to important interactions between variables in disjoint groups that he was not capturing. To address this problem, Baker used a screening test to determine these important, neglected interactions. Using these results, he then included the necessary factors in other groups and re-ran the analysis. In doing so he was able to decrease the mean error by roughly 2 to 4 times. Although Baker's new approach was successful in reducing the error, he did so at the cost of requiring more data than if he had simply analyzed all of his factors at the same time! The total data required for the new approach increased by more than 50%. Since one of the primary reasons for investigating the UTE method was to minimize the curse of dimensionality for large scale systems, and since it has been shown to be less efficient than a standard all-at-once approach, the UTE method is not a candidate to use with the nozzle model.

Ruling out the use of the spatially partition metamodeling (SPM) method for the problem is a little less straight forward. The SPM method was originally developed to create metamodels of finite element models by exploiting the spatial nature of the system. By all accounts this is exactly what we are looking for, we have a spatially partitioned finite element model and we need to create metamodels of it. Although this thesis is motivated by the analysis of one particular large scale system, the nozzle model, ideally we would like the developed method(s) to be as generic as possible. Meaning, the process that is developed should be potentially applicable to all kinds of large scale systems, not just large finite element models. Unfortunately the SPM

method does not really meet this level of generality. The argument behind that statement can be linked to the guiding idea that lead to the formulation of the SPM model.

The SPM method was developed around the idea that influence of a variable (i.e. its interactions with other variables) is spatially localized. Volovoi justify this approach by using the St-Venant principle which characterizes the decay rate of disturbances in the stress field by $\sqrt{G/E}$ where E and G are the Young's modulus and shear modulus of the material, respectively. Essentially what this implies is that only the interactions between neighboring spatial zones would need to be considered. These neighbors could be individual variables or groups of variables.

The problem with extrapolating this method to other large scale systems becomes, how does one decide if two variables or groups of variables are neighbors when the problem is not or can not be formulated in a spatial fashion? This concern limits the applicability of the SPM method to other classes of problems.

Having now reviewed and eliminated the all-at-once approaches based on super-saturated designs and stepwise regression routines, as well as each of the three non-hierarchical methods, the only currently available method remaining is the hierarchically partitioned General Composite Function method. Unfortunately, the GCF method is also not applicable to the nozzle problem. Primarily because it is only applicable to systems with readily identifiable intermediaries. However the GCF method should not be dismissed quite yet as it does have some nice properties that potentially could drive the development of a new method.

3.5 Formulation of a New LSM Method

The previous section reviewed each of the currently available large scale metamodeling (LSM) methods and concluded that none of the methods met all of the needs for the nozzle problem. However, of all the methods reviewed, the UTE method is the most

generic and could be applied to the nozzle problem. Unfortunately it seems to exhibit problems with efficiency and accuracy, although not necessarily at the same time. To address the accuracy problem Baker, used a screening test to determine potentially important interactions and then added the necessary factors to each partition. The adding of factors increased the accuracy but negated any potential efficiency gains that using a partitioning approach would gain him.

Based on the results from Baker’s accuracy analysis, we can conclude that by using a partitioning approach we are neglecting some interactions which may have negative consequences on the accuracy of the method. However we do not want to resort to variable redundancy between the partitions because it will degrade the efficiency of the method. Now the question becomes, is there a way to have the efficiency of a partitioning approach while at the same time accounting for potential interactions between partition variables that may need to be accounted for? In an indirect fashion, the GCF method does just that.

Referring to Figures 8 and 9, each arrow represents a transfer function that maps changes in lower level variables to the response in the next highest level. Therefore, even though the lowest level design variables (the x ’s) are in disjoint groups, that does not mean that they do not have interactions. In fact they do, albeit indirectly via the functional mapping between intermediate responses (the \tilde{Y} ’s) and the top-level response (Y). Therefore it is hypothesized that the intermediate to top-level mapping will indirectly account for any neglected, yet potentially necessary interactions between low-level design variables in disjoint groups.

Assertion: Baker showed that important, neglected interactions do have a negative impact on the overall accuracy of a partitioned metamodel.

Assertion: Adding variable redundancies between the partitions improved the accuracy of the UTE metamodel but did so at the cost of degraded efficiency of

the method.

Research Question 2.1: Is there a way to account for potentially important interactions without resorting to variable redundancy between the partitions?

Research Hypothesis 2.1: Using the GCF method as a prototype, it is hypothesized that the intermediate to top-level mapping will indirectly account for any neglected interactions between low-level design variables in disjoint groups.

Now that we have a good argument as to why the GCF method is probably the best, general method to use for partitioned metamodeling, we are now faced with the primary limitation of the GCF method. Namely, the GCF method is not applicable to analyses without readily identifiable intermediaries. If you do not have intermediate variables/responses in your code that can be directly manipulated, how can you do the design space exploration to create the intermediate to top-level functional mapping?

To answer this question, perhaps we can again take cues from Baker's improved method. In his improved method, he used a screening test to determine the important missing interactions. In the new, hypothesized method, these interactions will be indirectly accounted for via interactions between the intermediate responses in the intermediate to top-level mapping.

Based on this hypothesis we now have a new question to answer, is there a way to use a screening-like test with the new method, to determine the new method's means of accounting for neglected interactions? Or to put it more succinctly, can we use a screening-like test to determine the intermediate top-level mappings?

Before attempting to answer this question, a quick estimation of the potential run requirements (or savings) for the new method is in order. If the potential run savings just are not there, answering the previous question becomes a less than useful exercise.

To estimate the number of runs, the run results from the UTE method can be

Table 2: Summary of UTE Variables per Partition

	Partition 1	Partition 2	Partition 3
All-at-Once	16	-	-
UTE Orig. Formulation	5	5	6
UTE Upgrade #1	5	7	8
UTE Upgrade #2	7	7	10
Hypothesized Method	5	5	6

Table 3: Benchmarked Run Estimates for New Hypothesized Method

	Partition 1	Partition 2	Partition 3	Screening	Totals	Savings
All-at-Once	289	-	-	-	289	0%
UTE Orig. Formulation	27	27	45	0	99	66%
UTE Upgrade #1	27	79	81	137	324	-12%
UTE Upgrade #2	79	79	149	137	444	-54%
Hypothesized Method	27	27	45	34	133	54%

used as a benchmark. Table 2 summarizes the number of variables assigned to each of the three partitions for the UTE method. In Table 2, the UTE original formulation represents the original number of variables per partition before Baker used a screening test to determine important interactions that he was missing. UTE upgrades #1 and #2 represent the number of variables assigned to each partition after Baker determined which missing interactions were negatively affecting the accuracy of the method. The all-at-once row is there to represent a standard DoE approach to modeling the system. The hypothesized method row states the variable counts per partition for the new, hypothesized method.

Using the number of variables assigned to each partition as stated in Table 2, Table 3 summarizes the run requirements for the UTE methods and the new, hypothesized method as compared to a similarly sized all-at-once approach. The UTE method run requirements for each partition as well as the screening test are taken directly from Baker’s thesis [6]. In addition, the runs required for the all-at-once analysis are also taken from Baker’s thesis.

Before commenting on the projected run requirements for the new, hypothesized

method, it is interesting to note how adding the additional factors to each partition quickly eliminates any potential run savings for a partitioning method. In fact, completely neglecting the additional runs expended for the screening test, the runs required for just the partitions for the UTE upgrade #2 exceeds the expenditure for the all-at-once analysis by 6%. Obviously, having redundancies between partitions is not an advisable solution for solving the problem of poor accuracy due to neglected interactions in partitioning approaches, particularly if one of your primary objectives is efficiency.

To keep the comparison as close as possible, the projected run requirements for the new, hypothesized approach uses the variable partitions as stated for the original UTE formulation. Primarily this is because the original UTE formulation exhibits excellent run savings. A run saving of 66% as compared to a standard metamodeling approach would be fantastic. Secondly, the original UTE formulation is comprised of completely disjoint groups which is a characteristic that the new, hypothesized method will exhibit.

The main deviation between Baker's formulation and the proposed formulation is in the size of the screening test. By all accounts Baker's use of a 137 run screening test for 16 factors seems uncommonly large. Unfortunately the actual design of experiments that Baker used to perform the screening test was not stated in his thesis.²

A more common size of design for screening a 16 variable problem is on the order of 34 runs. A screening test of this size would be a resolution IV fraction factorial. A resolution IV design allows the estimation of all main effects and some

²However a little research shows that Baker's 137 run DoE is roughly the same size as a minimum run resolution V fractional factorial design (138 runs) [112] . His use of a screening test of this size makes sense because of what he was trying to accomplish, his screening test was intended to identify potentially important neglected interactions. A resolution V fraction factorial allows the experimenter to estimate all main effects and all second order interactions are only aliased with higher order effects. Consequently this would be a good design for clearly identifying important interactions.

two factor interactions are aliased with other two factor interactions [106]. Although some two factor interactions are aliased, applying engineering intuition and logic to the assignment of factors to the design columns can go along way to alleviate this potential problem. Thus if we can use a more traditionally sized screening design with the new formulation to determine intermediate to top-level mapping, we will have a projected run savings of 54% as compared to a standard all-at-once approach. Even if we have to resort to using a screening test twice this size, we will still have a projected run saving of 42%.

It is not clear at this time how we can use a screening test to determine the intermediate to top-level mappings. However, it is hypothesized that a screening can be used to determine these mappings.

Assertion: Baker used a screening test to determine important, neglected interactions and then used variable redundancy to account for these interactions.

Assertion: In the hypothesized method built on the GCF model, these interactions will be indirectly accounted for via interactions between the intermediate responses in the intermediate to top-level mapping.

Research Question 2.2: Can the data from screening test be used to determine the intermediate to top-level mappings for systems without identifiable intermediaries?

Research Hypothesis 2.2: It is hypothesized that since the data from a screening test data inherently contains the connectivity of the system, this data can be used to create the intermediate to top-level mappings for systems that do not have readily identifiable intermediaries.

3.6 *Summary*

The goal of this chapter was to support or refute the second research hypothesis stated in Section 1.6, that hypothesis was: Given the spatially partitioned nature of the finite element model, it is hypothesized that a partitioned metamodeling scheme seems like a logical choice for reducing to the total number of runs required to create an accurate metamodel.

To investigate this hypothesis, this chapter outlined two general classes of techniques applicable for run mitigation for large scale systems. Both avenues, either directly or indirectly, exploit the notion of effect sparsity to justify the generation of reduced order models. Effect sparsity refers to the notion that in large scale systems not all effects are important for the variability of the response. Employing this assumption allows the experimenter to create reduced order models which require less data than complete models. The distinction being that complete models contain all potential effects on the variability of the response, while the reduced order models do not.

The first set of approaches outlined revolve around the use of supersaturated designs of experiments in conjunction with a stepwise regression routine to build an accurate metamodel. The second set of approaches partition the total set of design variables into smaller, disjoint subgroups. The subgroups are analyzed and the resultant data combined to create the final metamodel. For the partitioned approaches, effect sparsity is directly enforced by the experimenter. Conversely, the all-at-once approaches use a stepwise regression routine to determine the level of effect sparsity inherent to the system.

Each large scale metamodeling method was then reviewed with the intent to support or refute the guiding research hypothesis for this chapter. In this review it was concluded that, yes, it does appear that a partitioned metamodeling approach does make sense for the analysis. The primary reason being, the alternative approach,

supersaturated designs with stepwise regression, may not reliably create metamodels that represent the true nature of the system. The reason for this is due to the complex aliasing structure of the designs.

Having rejected the use of a supersaturated approach for large scale metamodeling, the review of the currently available partitioned metamodeling schemes revealed that none of these methods met all of the criterion for the problem at hand. However two of the methods did have several desirable characteristics that could potentially be used to create a new large scale metamodeling method. These two methods were the General Composite Function (GCF) approach and the Unified Trade-off Environment (UTE). The desirable characteristics were the means of indirectly accounting for missing yet important interactions via the intermediate to top-level mappings (the GCF method) and the use of screening test to create the intermediate to top-level mapping and thus account for the missing interactions (the UTE method).

In the review of the current large scale metamodeling (LSM) methods and subsequent formulation of a new LSM method, two new research hypotheses were identified. One, it is hypothesized that the intermediate to top-level mapping will indirectly account for any neglected, yet potentially necessary interactions between low-level design variables in disjoint groups. Two, it is hypothesized that a screening can be used to determine these mappings.

Chapter IV

REVIEW OF SYSTEM PARTITIONING METHODS

The goal of this chapter is to support or refute the third of the initial research hypotheses posed in Section 1.6. That hypothesis was: It is hypothesized that a method from the electrical engineering community for partitioning VLSI circuits could potentially be adopted for finding an optimal set of partitions for use with partitioned metamodeling.

That research hypothesis addresses the very practical question of how should a large scale system be partitioned for use with a partitioned metamodeling scheme? In Section 3.3.1, we stated the three methods quoted from the literature. These methods were quantitative partitioning based on results from a screening test, partitioning based on engineering intuition, and partitioning along code or process boundaries. However, as was stated in Section 3.3.1, only the last two methods had actually been demonstrated for use with partitioned metamodeling.

The lack of a means to quantitatively determine a good set of partitions or to quantitatively support or reject a chosen set of partitions is seen a serious short coming for the use of partitioned metamodeling schemes. Fortunately, the quantitative partitioning of large scale systems has largely been solved by the electrical engineering community. However as we will see in this chapter, their methods are not 100% adoptable for the problem at hand.

This chapter will follow the development of partitioning techniques for large scale circuit analysis and identify the missing elements necessary to solve the partitioning

problem for large scale metamodeling. The adaptation and modification of the large scale circuit partitioning methods for quantitatively partitioning large scale metamodeling problems will comprise another of the primary contributions for this thesis.

4.1 Diakoptics

The initial, rigorous work on the analysis of large scale systems can be traced back to Kron and his work on the development of diakoptics [14]. Diakoptics is a method for finding the solution to large scale systems in a piecewise fashion [82]. The word “diakoptics” is derived the Greek word “kopto” which means “to tear” and “dia” which can be interpreted as “system” [82]. Hence diakoptics is the method of “system tearing”.

The methods of diakoptics were an outgrowth of the work that Kron conducted as electrical engineer for the General Electric Company in the 1930s and 1940s. During that time, as Kron put it, his primary job was “given some highly complicated and extensive physical or engineering structure, whose building-blocks may represent a wide range of physical phenomena; the purpose of his study was to establish the equations of state of the structure, in small installments, in a piecewise and orderly manner” [82]. Over time, Kron became disenchanted with the rather haphazard “bag of tricks” used to solve and analyze the equations of state for large scale electro-mechanical structures.

To address the disjointedness of the then “current” solution methods, Kron began looking for a unified approach that would allow him to systematically describe, simplify and solve complex engineering problems. Eventually he realized that the tools used by modern physicists to describe the complex mathematics associated with non-euclidean, n-dimensional space could be used to describe the high dimensionality problems common to many engineering disciplines. Kron first began by using tensor notation to summarize the equations of state for the individual components of his complex system into one compact form [76]. He then characterized interconnectedness

of the system by creating what he called the “connection matrix” [82]. The connection matrix linked the flow of information between each of the individual subsystems via their connectivity in the complete system.

Although Kron’s initial formulation was primarily applied to the analysis of electrical networks, he extended his system tearing method to other complex systems of equations. Examples include electro-magnetic field equations [77], the Schrodinger wave equation [79], ordinary and partial differential equations [81], the equations of elasticity [78], and the compressible and incompressible fluid flow equations [80]. In each case, Kron transformed the basic algebraic and differential equations from each of these fields into equivalent circuits analogies and applied the diakoptics method to analyze each system.

The diakoptics method worked well for Kron, however as Steward (1965)[160] put it “In the hand of someone such as Kron who has an understanding of the methods and good insight to the problem, this technique has had outstanding success. The widespread use of Kron’s technique has perhaps been limited severely for lack of practitioners skilled in the art of determining effective ways of tearing specific problems”. Essentially what this is implying is that non-electrical engineers would have problems reformulating their systems into an equivalent circuit analogy and then tearing the system based on an understanding of electrical networks. Consequently, although Kron’s methods were widely referenced, their ultimate adoption was limited due to a lack of a systematic procedure to tear the system [55]. Steward addressed this limitation by making the connection between Kron’s tensor representations of the system and sparse matrix representations of systems [159]. In doing so, Steward opened up the field of system tearing to all of the computer based methods that had been developed to solve sparse matrix systems via systems partitioning.

4.2 *Overview of Sparse Matrix Methods*

Kron's method of diakoptics or systems tearing is essentially a subset of a much larger set of methods that had been developed over time to manipulate and solve large, sparse matrices. Before getting into the pertinent sparse matrix methods that potentially could be applied to the research objectives of this thesis, a brief overview of what sparse matrices are and how they often arise in engineering analyses is in order.

In the design and analysis of engineering systems, rarely is the behaviour of the system governed by one mathematical equation, but rather engineering systems are typically characterized by systems of equations. The most prevalent method of solving systems of equations can be summarized with the following equation [178]:

$$Ax = b \tag{24}$$

where,

A = the matrix of coefficients for each variable in the system of equations

x = a vector of unknown variable values

b = the solution vector for the system of equations

In fact this equation shows up so much in science and engineering that it is probably the most studied equation in all of mathematics. For small systems, the most direct method of solution is simply to invert the coefficient matrix and multiply it by the solution vector, $x = A^{-1}b$. However, for large systems this solution approach is almost never used, primarily because inverting the coefficient matrix can be particularly time consuming and is very sensitive to ill-conditioning in the coefficient matrix. The far more common approach is to use Gaussian elimination to solve the problem [177]. When the matrix A is dense, that is, it is comprised of primarily non-zero values, Gaussian elimination works by transforming the matrix A into upper triangular

form. Upper triangular form meaning that all elements below the diagonal are zero. Once in upper triangular form, the system of equations can be easily solved by using a simple back substitution scheme. In general, Gaussian elimination performs some operation on every element in A , and when A is a very large matrix (say on the order of tens of millions of elements) this process can be quite time consuming. Fortunately for sparse matrices, the sparse nature of the system can be leveraged to minimize the number of actual elements that need to be manipulated.

As compared to a dense matrix which contains primarily non-zero elements, a sparse matrix is defined as a matrix that is populated mainly by zeros [185]. Sparse matrices arise quite frequently in the analysis of large systems of equations in science and engineering because quite often most equations do not contain every variable. As an example, consider the follow set of simple equations:

$$\begin{aligned}
 x_1 + 5x_3 &= b_1 \\
 6x_2 + x_4 + 2x_5 &= b_2 \\
 7x_3 + x_6 &= b_3 \\
 x_2 + 3x_4 + x_5 &= b_4 \\
 x_2 + 2x_5 &= b_5 \\
 x_1 + x_3 + x_6 &= b_6
 \end{aligned} \tag{25}$$

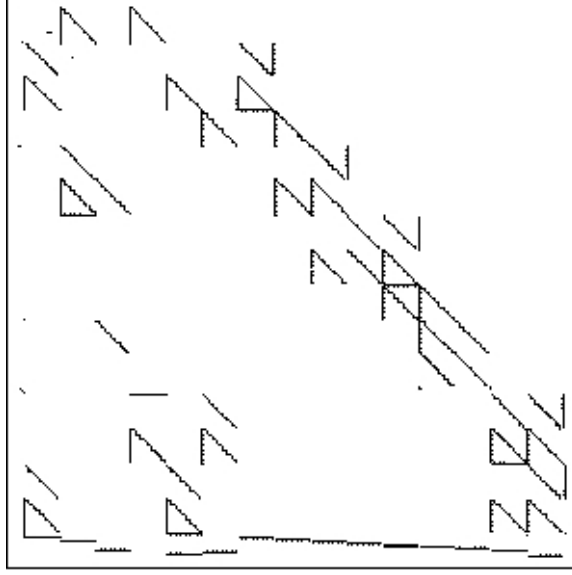


Figure 10: Example Sparse Matrix [186]

This set can be expressed in matrix notation as:

$$\begin{pmatrix} 1 & 0 & 5 & 0 & 0 & 0 \\ 0 & 6 & 0 & 1 & 2 & 0 \\ 0 & 0 & 7 & 0 & 0 & 1 \\ 0 & 1 & 0 & 3 & 1 & 0 \\ 0 & 1 & 0 & 0 & 2 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_5 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_5 \end{pmatrix} \quad (26)$$

As we can see, the coefficient matrix A is primarily comprised of zero elements. Granted this is not a large matrix and in general much larger sparse systems are characterized by a significantly larger ratio of zeros to non-zeros. So much so that it is often advantageous to not even pay attention to the actual values of the coefficients but rather to visually examine the structure of the matrix as a whole [3]. For instance, Figure 10 shows a plot of the incidence matrix for a system comprised of 380 equations and 380 variables.

An incidence matrix is a simplified version of the coefficient matrix, A , where

	x1	x2	x3	x4	x5	x6
eqn1	1	0	1	0	0	0
eqn2	0	1	0	1	1	0
eqn3	0	0	1	0	0	1
eqn4	0	1	0	1	1	0
eqn5	0	1	0	0	1	0
eqn6	1	0	1	0	0	1

Figure 11: Incidence Matrix of Equation Set 25

all of the non-zero coefficients are replaced with a “1”. The incidence matrix of the previous set of equations is shown in Figure 11. An incidence matrix is a simple way to summarize the relationship between a set of equations and the set of variables that are incident to each equation [182]. As an example, the first row in Figure 11 is associated with equation one. In the previous set of equations, equation one is only comprised of variables x_1 and x_3 , consequently there is a “1” in the first and third columns and the rest of the columns contain zeros. A “1” indicates incidence or a relation and a “0” represents no incidence or relation. The incidence matrix is also commonly called the occurrence matrix [63].

As was previously stated, in solving the $Ax = b$ problem for large sparse systems, the sparsity of the system can often be used to significantly reduce the computational expense associated with solving the system. A popular technique to is to reorder the columns and rows of the sparse matrix as to put it into block diagonal form. Reordering a sparse matrix in this fashion is referred to as “system partitioning” [136].

As an example of system partitioning, consider the incidence matrix shown in Figure 11 for the Equation set 25. A set of two independent subsystems or blocks can be created by reordering the row and columns. Figure 12 shows the reordered incidence matrix.

In the reorder matrix it is easy to see the two independent, dense sub-matrices in Figure 12. The new block structure of the sparse matrix allows the Gaussian

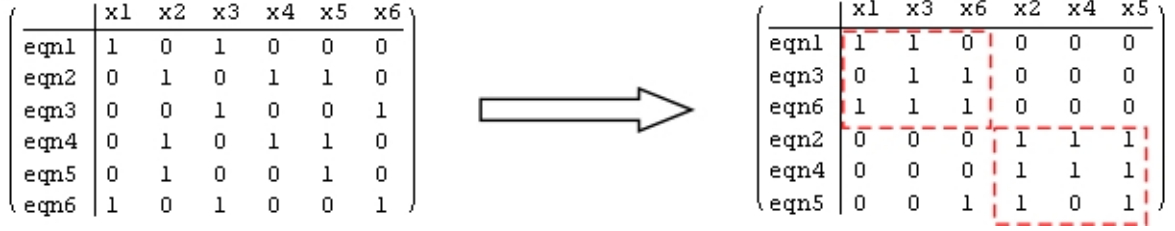


Figure 12: Reordered Incidence Matrix of Equation Set 25

elimination procedure to be applied to each of the independent blocks rather than to the entire matrix. This partitioned system will typically require less computational expense to solve than the complete problem [136].

The shape of the system partitioned incidence matrix shown in Figure 12 is typically referred to as block diagonal form (BDF). However not all sparse matrices can be reordered into a BDF. Examples of other diagonal forms include block triangular form (BTF), single and double banded block triangular form (SBBTF and DBBTF), banded form (BF) and single and double bordered banded form (SBBF and DDBF) [163]. In addition, each of these diagonal forms have specific variations on the Gaussian elimination scheme to take advantage of their specific characteristics [164].

The two independent blocks in Figure 12 are commonly referred to as disjoint subsystems. Disjoint implies that each of the subsets of functions do not contain any common variables [63]. If there were any common variables between the two sets of functions, there would be a non-zero element(s) in the upper right hand quadrant or the lower left hand quadrant in Figure 12.

When there are off block-diagonal elements in the system partitioned matrix, they are often referred to as “feedbacks”. The existence of feedbacks prevents the independent solution of each of the connected sub-matrices and will in general require the entire system to be solved as a whole [30].

As an example of how a feedback can occur, consider if the fifth equation in

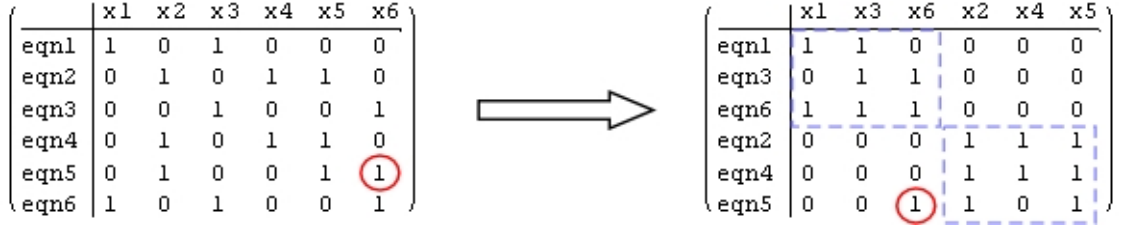


Figure 13: Reordered Incidence Matrix with Feedback

Equation set 25 was instead:

$$x_2 + 2x_5 + x_6 = b_5 \quad (27)$$

The new incidence matrix with the updated fifth equation and its system partitioned version are shown in Figure 13. If you compare Figures 12 and 13, you will see that the only difference in the re-ordered matrix is the “1” in the last row of the third column. This one, out of place element is feedback between the two sets of partitions and consequently would require Gaussian elimination on the entire system.

In situations such as this, the primary means of dealing with feedback variables is actually to remove or “tear” them from the system. The solution to the system is then obtained by iteratively solving the new torn system. The use of the word “tear” in this example is no coincidence. Tearing in this context is essentially what Kron’s method did although Kron never seemed to state it this simply. Also Kron did not use a Gaussian elimination scheme to solve each subsystem, but rather relied on the basic matrix inversion method to find their solutions [46].

4.3 *Generalized Method of Systems Decomposition*

The previous section introduced a general class of sparse matrix methods that allows one to partition a very large system into smaller subsystems, solve each of the subsystems independently and combine their solutions to create the solution for the

entire system. This description is essentially a synopsis of what is called systems decomposition [63].

Using the ideas from the previous section (systems partitioning and tearing) and the previous description of the systems decomposition, we can outline a general method for solving large scale systems [63].

1. Partition or re-order the large sparse matrix into a set of the smallest possible, irreducible subsystems.
2. Remove or tear any non-zero, off-diagonal elements from the system. Off-diagonal implies elements that are not contained in the subsystem blocks.
3. Re-partition each of the subsystems to potentially further simplify the subsystems into smaller subsystems.
4. Assume an initial value for each of the torn values. The assumed value is determined by physical principles or a best guess.
5. Apply the appropriate solution technique to each of the subsystems and update the torn values with the results from the subsystem analyses.
6. Iterate on Step 5 until the solution converges.

The previous systems decomposition process is essentially Kron's diakoptics method for the solution of large scale systems and has been reformulated and approved upon by numerous authors [23, 84, 144, 191, 196]. In mathematical notion, the solution to $Ax = b$ by the systems partitioning and tearing method can be summarized in essentially two steps [45].

1. Find a matrix, C , such that:
 - (a) the non-zero entries of C are equal to the corresponding ones in A .

(b) $B \triangleq A - C$ is a non-singular and generally makes the solution to $Bx = b$ a relatively simple task.

2. Solve $Bx = b$ by taking into account the “perturbation” caused by the non-zero entries in the “cut matrix”, C .

Although in mathematical terms, the process of solving large scale systems by systems partitioning and tearing seems quite simple, in actually it is not so straight forward. The determination of the cut matrix, C , falls into the general class of NP-complete problems [183]. The problem is considered NP-complete because for a large sparse matrices, there are billions ways to re-order the columns and rows. Actually finding the “true” optimal ordering is almost impossible without examining every possible combination. Consequently, the practitioners of diakoptics tend to assume that the decomposition of the problem is given or is in general known [113], primarily by means of using the structure of the physical system to partition the problem.

Fortunately as we will see in the next section there is a direct connection between sparse matrices and graph theory. Using graphs of the system, one can often tell where to partition the problem simply by looking the graph and separating out the strongly connected sub-graphs [137].

4.4 Graph Theoretic Methods for Partitioning

To make the connection between graph theory and systems partitioning of large sparse matrices, a brief introduction to graph theory will be useful. Then with this information into hand, we can continue with the simple example from Section 4.2 to show how graph theory can allow us to partition or reorder a sparse system.

A graph is a simple, visual way to represent the interconnectedness of a system. In mathematical terms a graph, G , is defined as $G = (V, E)$ where V is a non-empty

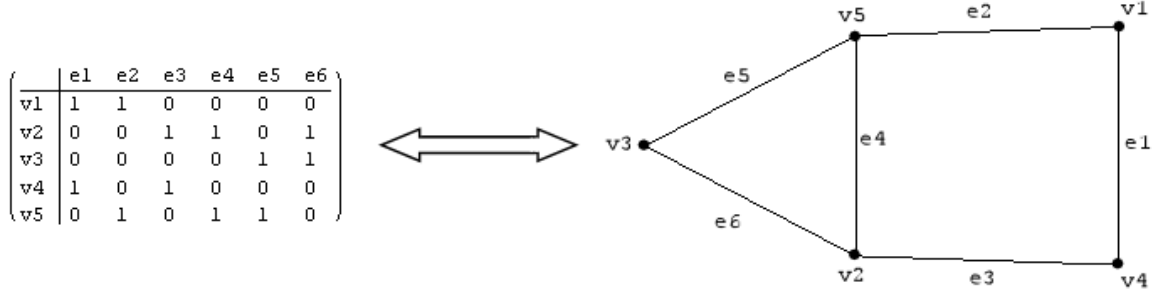


Figure 14: Incidence Matrix and Its Graph [138]

set of vertices and E is set of edges¹. Each edge is comprised of an unordered pair of vertices from the set V [56]. A common way to represent a graph in matrix form is through an incidence matrix. An incidence matrix was introduced in Section 4.2 and was used to summarize which variables were incident to which equation. In a more general fashion, an incidence matrix is a means to summarize which vertices are incident with a given edge. For instance, a graph and its corresponding incidence matrix is shown in Figure 14.

Now that we have a basic understanding of what a graph is, we can now show how a graph can be used to partition a large sparse system. The easiest way to see graph-based partitioning in action is to consider the graph of the incidence matrix shown in Figure 11. Recall that the incidence matrix in Figure 11 is based on the coefficient matrix created from the small Equation set 25. We already know that this incidence matrix can be reordered or partitioned into two completely disjoint sets of equations and variables. Figure 15 shows the graph of the incidence matrix from Figure 11. Actually, the incidence matrix that is graphed is the transpose of the matrix shown in Figure 15. The incidence matrix is shown in transposed form to place emphasis on the variables by placing them at the vertices, that is, associating them with the

¹Note - In the thesis we are only considering a very small subset of graph theory. All the graphs in this thesis are considered simple, non-directed-edge graphs. See Harary (1972)[56] for a good graph theory reference.

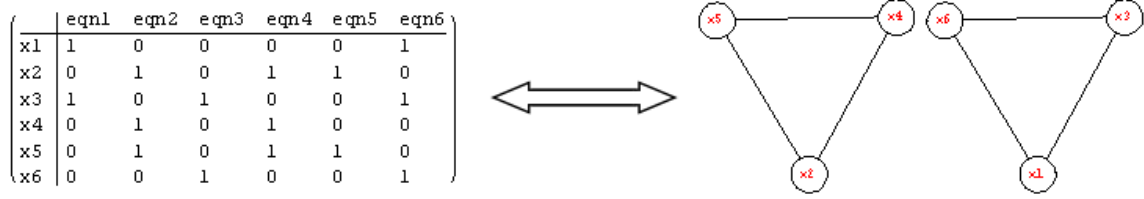


Figure 15: Graph of Incidence Matrix in Figure 11

rows.² The graph could just as easily have been created with the equations on the vertices, in either case the graph looks exactly the same.

Looking at Figure 15 we can instantly see that the system of equations is actually comprised of two completely disjoint sets. Compare this with a visual inspection of the original matrix, it is not immediately obvious that the system can be easily partitioned into two subgroups by simply looking at the matrix. The important thing to note is that this graph was created with the original, non-reordered incidence matrix. In fact, this graph would look the same regardless of how the incidence matrix is ordered. This brings us to a very important fact about graphs, graphs remain invariant under classes of transformations such as row and column exchanges [117].

Now let us extend our example to the same system of equations but now consider the case with feedback. Recall that feedback occurred when we added a relationship between variable six and the fifth equation. In doing so, when the system of equations was partitioned into two block diagonal subgroups, a lone “1” showed up in an off block diagonal position. The existence of this off block diagonal element typically precludes the application of Gaussian elimination to solve each of the subgroups independently, and consequently would require that the entire system be solved all at once. One potential solution to this problem was to apply tearing to the system to remove that

²The author chose to associate the variables with the vertices in these examples because this thesis is primarily concerned with means to partition the variables. By placing the variables on the vertices now, the reader will be more familiar with the graph-based partitioning method developed in Chapter 6.

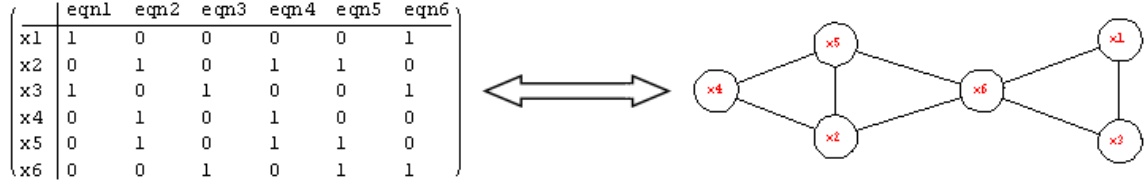


Figure 16: Graph of Incidence Matrix with Feedback

feedback.

So how does tearing manifest itself via graph theory? Figure 16 shows the incidence matrix with feedback and its graph. What is interesting to note about this graph is that the graph is now much more connected than the previous graph. That is exactly what feedbacks do, they add connectivity to the graph of the system.

Comparing the graph in Figure 15 with the more connected graph in Figure 16, we can see that the main difference between the two are the edges connecting vertices (x_5, x_6) and (x_2, x_6) . Why did these two new edges show up in the new graph? They showed up because of what the edges in this graph represent. Two variables share an edge when they are incident in the same equation. Consequently by adding variable six to the fifth equation we made a connection between variables, x_5 and x_6 as well as x_2 and x_6 .

Now we can describe what tearing means in graph theory. If we were to cut or remove these two new vertices, the graph in Figure 16 would become the disjointed graph in Figure 15. Consequently, tearing is a way to break a connected graph into disjointed sub-graphs by removing edges [137]. In addition, the disconnected sub-graphs that result from the graph tearing operation automatically gives you the optimal ordering of the incidence matrix. Each sub-graph in this case would represent a group of variables.

For the graph in Figure 16 the edges to cut are quite obvious, however for much larger graphs, where to cut is not always so straight forward. Fortunately there has been considerable work in the field of optimal graph partitioning. Unfortunately

however, since this problem is equivalent to finding the optimal orderings of the incidence matrix, it is also an NP-complete problem. Consequently, most of the algorithms that have been created rely on heuristics to find the best answer.

Harary (1971)[55] states that he believes that the best way to tear a matrix or a graph is to remove a minimal set of points or lines. This statement accurately describes the function that all current state-of-the-art min-cut algorithms perform. A min-cut algorithm is the common terminology used to describe a function that partitions a graph by finding a minimum set of cut-edges or cut-points such that the graph is partitioned into n equally sized sub-graphs.

The current state-of-the-art min-cut algorithms can all essentially be traced to a paper in 1970 by Kernighan and Lin [71]. Their algorithm is formulated to solve the problem: “Given a graph, G , with costs on its edges, partition the nodes of G into subsets no larger than a *given maximum size*, so as to minimize the total cost of the edges cut” [71]. Closely following the Kernighan-Lin paper in importance in the development of min-cut algorithms, is the paper by Fiduccia and Mattheyses (1981)[32]. Fiduccia and Mattheyses proposed an algorithm that scaled essentially linearly as the size of the problem increased. This is contrast to the Kernighan-Lin algorithm which exhibited an $n^2 \log(n)$ dependency with problem size [150]. For a good overview of the current state-of-the-art in min-cut algorithms, see the paper by Junger, Rinaldi and Thienel (2000)[70].

Examples from the literature of using graph theory with statistical modeling and estimation are very few and far between. In addition, none of the references use graph theoretic partitioning to find optimal partitions for use with metamodeling when the system structure is initially unknown. Wu and Chen (1992)[189] used graph theory to plan two-level fractional factorial experiments when interactions are important. The Wu-Chen method is essentially an extension of Taguchi’s linear graphs. Camplolongo and Braddock (1999)[19] and Cropp and Braddock (2002)[25] used graph theory to

create new screening designs that were extensions to the Morris method[143] for creating of one-at-a-time screening designs.

The closest method from the literature is by Bates et al (1996)[9]. Bates used tearing to partition a VLSI circuit into subgroups and created metamodels of each of the subgroups. However, as will be discussed in the next section, because Bates started with a known system configuration, his method is not applicable to the general problem of partitioning a system with an unknown initial structure.

4.5 Extending Graph Theory to Metamodeling

Thus far in this chapter we have investigated the common methods that electrical engineers use to partition and analyze very large-scale integrated (VLSI) circuits. This investigation was driven by the research hypothesis stated at the beginning of the chapter. That hypothesis was: It is hypothesized that a method from the electrical engineering community for partitioning VLSI circuits could potentially be adopted for finding an optimal set of partitions for use with partitioned metamodeling.

In general, it seems that this hypothesis is true. These methods have all the characteristics necessary that a partitioning scheme for large scale metamodeling would need. Namely, given some complex interconnectedness between the variables, a partitioning scheme is used to subdivide them into disjoint subsets and then any missing interactions from the torn system are captured via the hypothesized metamodeling scheme discussed in Section 3.5.

However support for this hypothesis is made with some reservation. The reservation arises because of differences between the two types of systems being compared. In the electrical engineering community, system decomposition methods are usually applied to a system where the structure is known a priori. Meaning, an electrical engineer starts off the analysis with an existing system of equations or an existing circuit design. Thus he already knows the interconnectedness of the system, so applying

a systems decomposition scheme is rather straight forward.

Assertion: A partitioned metamodeling approach is proposed to reduce to total number of runs required to create metamodels of the nozzle model.

Assertion: The electrical engineering community has been using partitioning techniques for many years to analyze VLSI circuits.

Research Question 3: Is there a way to quantitatively find an good set of partitions to use with the partitioned metamodeling process?

Research Hypothesis 3: It is hypothesized that a method from the electrical engineering community for partitioning VLSI circuits could potentially be adopted for finding an optimal set of partitions for use with partitioned metamodeling.

Validity Hypothesis 3: In general, this research hypothesis seems to be true. The sparse matrix and graph theoretic methods have all the necessary characteristics for a partitioning scheme for large scale metamodeling.

However, the lack of initial knowledge about the sparse structure of the black-box systems used with metamodeling seems to preclude their use.

In the case metamodeling however, we do not start off with a known system structure. In fact determining that structure is the fundamental goal of metamodeling. All we essentially have is a black box with a known set of inputs and outputs and we want to make metamodels of this system. Thus if we don't know the structure, how can we decompose it for use with partitioned metamodeling?

To answer that question, perhaps we can again borrow from Baker's improved method. Recall that in Section 3.4, it was stated that Baker used a screening test to determine important interactions that he was neglecting. He then used these results to add factors to the current set of partitions. In effect, he was using the screening

test to gain further insight into the structure of his system. Thus, perhaps we too can use a screening to determine the structure of our system.

Before we can address this idea, it will be useful to review what a screening test is and what it does. A screening test is a method that an experimenter uses to gain some initial insight into his problem. Often when an experimenter starts the metamodeling process, he has a rather large list of design variables that he wants to investigate the impact of on a response(s). From the notion of effect sparsity [12], he suspects that not all of those design variables are actually going to be important to the variability of the response. Thus before he commits to expending the resources to create the final metamodels, he would like to first pair down the large set of design variables to a more manageable size. This is where a screening test comes in.

A screening test typically uses a low resolution design of experiments (DoE) to perform an initial investigation of the system. Low resolution meaning that generally, only main effects and perhaps some two factor interactions will be investigated. Consequently screening designs generally do not require a lot of runs of the analysis.

With the results of the screening test, the experimenter can then use an analysis of variants (ANOVA) method to quantify the results. ANOVA is a common statistical technique that partitions the total observed variability of a response into the variability associated with each effect [106]. Then based on the magnitude of each effect's observed variability, those effects with small observed variabilities can be neglected for further consideration.

In essence then, a screening test is a statistical means of experimentally determining the structure of the system. It determines the structure of the system by reducing the set of all possible design variables to only those that are important for a given response. Essentially then, a screening test is realizing the sparse nature of the system! Thus, the results of the ANOVA analysis could be used with a graph based min-cut algorithm to partition the system into disjoint sets of variables.

Assertion: A systems decomposition method for sparse systems is a logical means to partition the inputs for large scale metamodeling, however there is not enough initially known about the sparsity of the system.

Research Question 3.1: Given the black box nature of analyses used in metamodeling, how can their underlying sparse structure be determined?

Research Hypothesis 3.1: It is hypothesized that a screening test using a low resolution DoE with ANOVA can be used to determine the underlying sparse structure of the black box system, which then can be partitioned using a graph-based min-cut algorithm.

Now that we have hypothesized a means for realizing the sparse structure of our system via a screening test, we can actually take this idea one step further. Recall that in Section 3.5 we proposed using a screening-like test to determine the intermediate to top-level mappings. Perhaps we can use the screening test that was proposed in hypothesis 2.1 as the same screening test needed to create these mapping. If so, we would be maximizing the efficiency of our method. No data collected would ever be wasted. The same screening test data that we use to determine our partitions could also be used to create the top-level mappings. This would be a significant improvement over Baker's method which used the screening data to determine factor redundancies and then discarded that data. We would be getting 100% data utilization out of the new method.

Assertion: The use of the data from a screening test is proposed to perform two different operations in the new large scale metamodeling method: one for partitioning or tearing the system and another for determining the intermediate to top-level mappings.

Research Question 3.2: Is it possible to use the same screening test data, that

was collected for use with the lumped parameter method, to also determine the sparse nature of the system for partitioning?

Research Hypothesis 3.2: It is hypothesized that since the data from a screening test, which is to be used to tear the system into disjoint subgroups, inherently contains the connectivity information; this data can also be used to create the intermediate to top-level mappings which are designed to account for the impact of system tearing.

4.6 Summary

The goal of this chapter was to support or refute the third of the initial research hypotheses posed in Section 1.6. That hypothesis was: It is hypothesized that a method from the electrical engineering community for partitioning VLSI circuits could potentially be adopted for finding an optimal set of partitions for use with partitioned metamodeling.

That research hypothesis addressed the very practical question of how should a large scale system be partitioned for use with a partitioned metamodeling scheme? Section 3.3.1 stated the three methods quoted from the literature, these methods were quantitative partitioning based on results from a screening test, partitioning based on engineering intuition, and partitioning along code or process boundaries. However, as was also stated in Section 3.3.1, only the last two methods had actually been demonstrated for use with partitioned metamodeling. The lack of a means to quantitatively determine a good set of partitions or to quantitatively support or reject a chosen set of partitions is seen as a serious short coming for the use of partitioned metamodeling schemes.

To support the research hypothesis, we investigated how the electrical engineering community's methods could be potentially applied to our problem. We reviewed the development of systems decomposition methods for VLSI circuits. In this review we

showed how the sparse nature of large scale systems promoted the use of systems decomposition methods such as tearing and graph-based partitioning to simplify the system. Unfortunately we realized that these methods relied on initial knowledge of the system. Consequently we were not able to directly apply these partitioning methods to our problem because we did not initially know the structure of our system.

To determine the initial structure of our system, we hypothesized that the results from an ANOVA analysis based on a screening test could be used to realize the sparse nature of our system. In addition it was also hypothesized that perhaps the same screening data used to partition the system could also be used to create the intermediate to top-level mapping for the hypothesized new method from Section 3.5.

Chapter V

INTERIM SUMMARY

As the title suggests, this chapter is intended to summarize the main points of the last four chapters to give a clear picture for the work to be discussed in the rest of the thesis. The summary will include the main research questions and hypotheses developed during the review of the background literature. In addition a summary will be given of the methods formulated to fill specific gaps and shortcomings of the current large scale metamodeling techniques as identified in the literature.

A secondary goal of this chapter is to give the “results only interested” reader a quick primer to the previous background chapters so that the development of the methods in Chapter 6 and the validation of these methods in Chapter 8 will have the appropriate context.

5.1 Review of the Previous Four Chapters

Chapter 1 was essentially a systematic narrowing of the scope for the motivation for this thesis. The motivation started with the design and development of a specific system and was subsequently whittled down to a specific engineering task. The high level motivator was the development of GE’s new H machine gas turbine engine. Clearly, the design of a new gas turbine power plant is far too broad a topic for an engineering thesis, thus Chapter 1 identified a specific part on the engine that is of strategic importance to the economics of operating the engine. The part was the first stage nozzle and the engineering task was to find or develop a method to accurately assess the life of the nozzle. The work for the thesis was ultimately motivated by limitations of the current methods that are typically used to execute this engineering

task.

The life of a first stage nozzle is primarily influenced by the operating conditions that the nozzle is exposed to. It follows then that to accurately predict the life of a nozzle, a good understanding of what creates these conditions is quite important. Thus, Chapter 1 reviewed how the three key technologies used on the H machine had a direct negative impact on the life of the nozzle. The three key technologies, closed-loop steam-cooling, single crystal materials and dry, low NOx combustors, were enablers that facilitated the H machine reaching its primary design goal of 60% efficiency.

Given that the nozzle life is adversely affected by the operating conditions imposed by enabling technologies and poor life has a direct negative impact on the bottom line of gas turbine operators, there becomes a definite need to design a nozzle that is robust to these operating conditions. To do so, Chapter 1 outlined a common methodology used for assessing the life of gas turbine parts. The methodology used a complex finite element analysis in conjunction with a probabilistic design method. The finite element model created for this system was a spatially partitioned model comprised of 52 independent design variables and 255 independent responses. The model took approximately 3.5 hours to compute one set of loading and boundary conditions. Figure 4 shows the meshed finite element model of the nozzle and Figure 5 shows the basic layout of the spatially partitioned model.

Based on the nature of this analysis, a common probabilistic design method used to estimate the probabilistic nature of the life is a indirect Monte Carlo simulation. Although indirect Monte Carlo methods are typically considered efficient probabilistic analysis methods and have been successfully applied to many complex systems, the large scale, partitioned nature of the system identified the need for a new meta-modeling method. Thus, it was hypothesized that a new procedure could be created that was able to exploit the partitioned nature of model to create metamodels more

efficiently but just as accurately as the currently available statistical linear model based metamodeling methods.

Stemming from this hypothesis, were three initial research questions that needed to be answered before it could be justified to spend the resources to develop the new hypothesized partitioned metamodel.

- First, even though an indirect Monte Carlo was chosen as the appropriate probabilistic design method, is this probabilistic analysis method actually the most appropriate for the system?
- Second, assuming that an indirect Monte Carlo method is the appropriate probabilistic design method, does the proposed use of a partitioned metamodeling scheme make sense?
 - Alternatively, is there another large scale metamodeling scheme that is more appropriate for the system.
- Third, assuming that a partitioned metamodeling approach is the best approach for the system, how does one actually determine how to partition the system?

Chapter 2 reviewed each of the common probabilistic design methods and compared their characteristics against the needs of the probabilistic lifeing analysis. Three general classes of probabilistic methods there were reviewed: direct Monte Carlo simulation, analytic reliability methods and indirect Monte Carlo simulation. The specific direct Monte Carlo simulation methods were traditional Monte Carlo, stratified sampling, quasi-Monte Carlo and adaptive-importance sampling. The specific analytic reliability methods were first order reliability method (FORM) and second order reliability method (SORM). The specific indirect Monte Carlo methods were adaptive-importance sampling with metamodels and traditional Monte Carlo simulation with metamodels.

A comparison of the characteristics of each of these methods with the needs and characteristics of the nozzle lifing analysis, essentially concluded at the original choice of using indirect Monte Carlo based on metamodels and traditional Monte Carlo sampling was justified. The justification was based on the fact that of all the methods reviewed, only the Monte Carlo/metamodel combination was able to simultaneously analyze multiple responses and exhibited the necessary efficiency to enable the probabilistic analysis. Table 1 shows a summary of each of the methods ranked on their applicability to each of the important characteristics and requirements of the nozzle model.

With Chapter 2 lending credence to the proposed probabilistic design method, Chapter 3 addressed the second of the primary research questions, namely, what large scale metamodeling technique is most appropriate to use with the system? To answer this question two general classes of large scale metamodeling techniques were reviewed, non-partitioned methods and partitioned methods. The fundamental difference between the two classes is that non-partitioned approaches analyzes all of the design variables at one time and use a supersaturated design of experiments in conjunction with a stepwise regression routine to create an accurate metamodel. The partitioned methods take the alternative approach of partitioning the total set of design variables into smaller, disjoint subgroups. The subgroups are analyzed and the resultant data combined to create the final metamodel.

Both of these classes of methods are essentially exploiting the notion of effect sparsity to create their reduced order models. Effect sparsity refers to the notion that in large scale systems not all effects are important for the variability of the response. Employing this assumption allows the experimenter to create reduced order models which require less data than complete models. The distinction being that complete models contain all potential effects on the variability of the response, while the reduced order models do not. For the partitioned approaches, effect sparsity is

directly enforced by the experimenter. Conversely, the all-at-once approaches use a stepwise regression routine to determine the level of effect sparsity inherent to the system.

Each large scale metamodeling method was then reviewed with the intent of supporting or refuting the guiding research hypothesis for Chapter 3. In this review it was concluded a partitioned metamodeling approach makes the most sense for the analysis. The primary reason being, the alternative approach, supersaturated designs with stepwise regression, may not reliably create metamodels that represent the true nature of the system. The reason for this is due to the complex aliasing structure of the supersaturated designs.

Having rejected the use of a supersaturated approach for large scale metamodeling, the review of the currently available partitioned metamodeling schemes revealed that none of these methods met all of the criteria for the problem at hand. However two of the methods did have several desirable characteristics that could potentially be used to create a new large scale metamodeling method. These two methods were the General Composite Function (GCF) approach and the Unified Trade-off Environment (UTE). The desirable characteristics were the means of indirectly accounting for missing yet important interactions via the intermediate to top-level mappings (the GCF method) and the use of screening test to create the intermediate to top-level mapping and thus account for the missing interactions (the UTE method). These two characteristics will be combined in Chapter 6 to create the Lumped Parameter Modeling process.

With Chapter 3 ending with the formulation of a new hierarchically partitioned metamodeling technique for large scale systems, the third of the initial research questions was now of key importance. The third research question was the very practical question how to best partition a system for use with a partitioned metamodeling approach. The review in Chapter 3 of the currently available partitioned metamodeling

methods identified three means for determining the partitions of a system: quantitatively partitioning using a screening test, partitioning using engineering intuition, and partitioning along code or process boundaries. Although these three methods were stated in the literature, only the last two have actually been demonstrated. The third method, code or process boundaries, was not applicable to the nozzle problem because there was only one process. The second method, engineering intuition, while feasible, requires a good understanding of the system to determine a good set of partitions. Ideally a quantitative method that could be used to support engineering intuition would be best.

To find a quantitative means of partitioning a large scale system, Chapter 4 reviewed the system decomposition methods that electrical engineers use to partition and analyze very large-scale integrated (VLSI) circuits. These system decomposition methods allow the electrical engineer to tear the VLSI circuit into several smaller, disjoint subsets, analyze each of the subsets independently and combine the results back together to get estimates of the total system response. The primary mathematical tools used for systems decomposition are sparse matrix and graph theoretic partitioning methods.

Unfortunately these systems decomposition methods are not directly applicable to the problem of determining a good set of partitions for use with metamodeling. The reason is due to what is initially known about the two different systems. In the VLSI circuit problem, the engineer always starts out the decomposition process with a given circuit diagram or set of equations for the system. Thus the sparse nature of the system is known upfront. In the case of large scale metamodeling, the model builder starts with a black box and in general knows nothing about the underlying structure of the system. Consequently, using sparse matrix methods to decompose the problem is not initially possible.

To determine the sparse nature of the black box, a screening test used in conjunction with a linear model ANOVA analysis is proposed. Based on the data collected from the initial screening test, an ANOVA analysis can be used to separate the significant variables from the insignificant ones for each the responses. The results of the ANOVA analysis can then be combined to create a sparse incidence matrix for the system. A min-cut graph partitioning algorithm can then be used to partition the system into disjoint subgroups.

With two new methods having been formulated in the previous four chapters (a new hierarchically partitioned metamodel and a graph-based partitioning method for large scale metamodeling), the remaining chapters will first start by formally developing these two new methods, and propose a validation plan to assess the goodness of the new methods. Based on the needs of the validation plan, a new computational framework will be discussed that will enable the successful testing of the new methods. Two test cases will then be outlined and analyzed using the new methods and the computational framework. Finally, a discussion of the results of the test case analyses and conclusions about the research will finish out the main body of the thesis.

5.2 Summary of Research Questions and Hypotheses

This section outlines a map of the thesis built around the research questions and hypotheses identified in the previous four chapters. Each research question will be stated in context to the section that it was initially posed in. Following that, each research hypothesis will then been linked to the section that either addressed or will address the validity of the specific research question/hypothesis.

The following research question and hypothesis were posed in Section 1.5 and subsequently addressed in Section 2.4. The two assertions that spurred the subsequent research question and hypothesis were based on the need to combine a multi-physics

finite element analysis with an appropriate probabilistic design method to adequately assess the probabilistic nature of the life of the nozzle. The statement of the hypothesis was made based on the author's experience of using probabilistic design methods used with finite element models.

Research Question and Hypothesis:

Assertion: A spatially partitioned finite element model was created to address the complex operating environment that a nozzle is exposed to. This model has 52 variables, 255 responses and takes 3.5 hours to run.

Assertion: To capture the probabilistic nature of the life of nozzle, a probabilistic design method is needed. The primary methods are direct Monte Carlo, indirect Monte Carlo and analytic reliability methods.

Research Question 1: Which of the three primary probabilistic design methods is most applicable for the nozzle problem?

Research Hypothesis 1: Given the computational nature of the nozzle model, it is hypothesized that an indirect Monte Carlo simulation technique is most likely the best probabilistic design method to use.

Validity of Hypothesis 1: Based on the results of probabilistic method comparison, two requirements drove the down selection process, the need to simultaneously handle multiple responses and method efficiency. Of all the methods only traditional Monte Carlo simulation used in conjunction with metamodeling concurrently addressed both of the requirements. Consequently, the hypothesis that an indirect Monte Carlo method based on traditional Monte Carlo and metamodels is indeed a valid statement.

The following research question and hypothesis were posed in Section 1.5 and subsequently addressed in Section 3.4. The assertion that drove the subsequent research

question and hypothesis were based on a simple estimation of the projected computational expense required to create metamodels of the nozzle problem. A fairly conservative estimate was on the order of 5,000 hours. Thus, it was felt that any additional information about the system that could be used to create the metamodels would reduce the total computational expense.

Research Question and Hypothesis:

Assertion: Given the constraints imposed by the nozzle model's complexity, traditional statistical linear metamodeling methods are not efficient enough to be used with the model. Consequently a large scale metamodeling method is necessary to reduce the total computational expense as much as possible.

Research Question 2: Which large scale metamodeling procedure is best to address the characteristics of the nozzle model?

Hypothesis 2: Given the spatially partitioned nature of the finite element model, it is hypothesized that a partitioned metamodeling scheme seems like a logical choice for reducing to the total number of runs required to create an accurate metamodel.

Validity of Hypothesis 2: For the most part, it does appear that a partitioned metamodeling approach does make the most sense for the analysis. The primary reason being, the alternative approach for building metamodels of large scale systems, supersaturated designs with stepwise regression, may not reliably create metamodels that represent the true nature of the system. The reason for this is due to the complex aliasing structure of supersaturated designs.

However, none of the current large scale metamodeling methods meet all of the needs of the nozzle problem. Consequently, a new partitioning method

is needed to be able to fully leverage the partitioned nature of the nozzle model.

The following research question and hypothesis were posed in Section 3.5 and will be addressed in Section 8.3.1. The assertions that spurred the research question and hypothesis were a result of reviewing the accuracy analysis that Baker used to validate the UTE method. In that analysis he concluded that missing interactions due to partitioning were negatively impacting the accuracy of the UTE metamodels.

Research Question and Hypothesis:

Assertion: Baker showed that important, neglected interactions do have a negative impact on the overall accuracy of a partitioned metamodel.

Assertion: Adding variable redundancies between the partitions improved the accuracy of the UTE metamodel but did so at the cost of degraded efficiency of the method.

Research Question 2.1: Is there a way to account for potentially important interactions without resorting to variable redundancy between the partitions?

Research Hypothesis 2.1: Using the GCF method as a prototype, it is hypothesized that the intermediate to top-level mapping will indirectly account for any neglected interactions between low-level design variables in disjoint groups.

The following research question and hypothesis were posed in Section 3.5 and will be addressed in Section 6.1. The assertion that drove this research question and hypothesis was based on the primary limitation of using the GCF method. The limitation being the lack of a means to create metamodels of analyses that do not have

any readily identifiable intermediaries around which to create the intermediate to top-level mappings. This research question and hypothesis were the primary motivation for the research that lead to the development of the new large scale metamodeling method in Section 6.1.

Research Question and Hypothesis:

Assertion: Baker used a screening test to determine important, neglected interactions and then used variable redundancy to account for these interactions.

Assertion: In the hypothesized method built on the GCF model, these interactions will be indirectly accounted for via interactions between the intermediate responses in the intermediate to top-level mapping.

Research Question 2.2: Can the data from screening test be used to determine the intermediate to top-level mappings for systems without identifiable intermediaries?

Research Hypothesis 2.2: It is hypothesized that since the data from a screening test data inherently contains the connectivity of the system, this data can be used to create the intermediate to top-level mappings for systems that do not have readily identifiable intermediaries.

The following research question and hypothesis were posed in Section 1.5 and subsequently addressed in Section 4.5. These two assertions were the result of proposing to use a partitioned metamodeling approach to create metamodels of the nozzle problem and realizing that there is a need for a method to help determine a good set of partitions.

Research Question and Hypothesis:

Assertion: A partitioned metamodeling approach is proposed to reduce to total number of runs required to create metamodels of the nozzle model.

Assertion: The electrical engineering community has been using partitioning techniques for many years to analyze VLSI circuits.

Research Question 3: Is there a way to quantitatively find an good set of partitions to use with the partitioned metamodeling process?

Research Hypothesis 3: It is hypothesized that a method from the electrical engineering community for partitioning VLSI circuits could potentially be adopted for finding an optimal set of partitions for use with partitioned metamodeling.

Validity Hypothesis 3: In general, this research hypothesis seems to be true.

The sparse matrix and graph theoretic methods have all the necessary characteristics for a partitioning scheme for large scale metamodeling.

However, the lack of initial knowledge about the sparse structure of the black-box systems used with metamodeling seems to preclude their use.

The follow research question and hypothesis were posed in Section 4.5 and will be addressed in Section 8.3.2. The assertion that spurred the subsequent research question and hypothesis was made after reviewing the common sparse matrix methods used for decomposing large scale systems into smaller disjoint subsets.

Research Question and Hypothesis:

Assertion: A systems decomposition method for sparse systems is a logical means to partition the inputs for large scale metamodeling, however there is not enough initially known about the sparsity of the system.

Research Question 3.1: Given the black box nature of analyses used in meta-modeling, how can their underlying sparse structure be determined?

Research Hypothesis 3.1: It is hypothesized that a screening test using a low resolution DoE with ANOVA can be used to determine the underlying sparse structure of the black box system, which then can be partitioned using a graph-based min-cut algorithm.

The follow research question and hypothesis were posed in Section 4.5 and will be addressed in Section 8.3.2. The following assertion was made after realizing that to maximize the efficiency of the new large scale metamodeling method, all the data collected from the system needs to be used to create the metamodels of the system.

Research Question and Hypothesis:

Assertion: The use of the data from a screening test is proposed to perform two different operations in the new large scale metamodeling method: one for partitioning or tearing the system and another for determining the intermediate to top-level mappings.

Research Question 3.2: Is it possible to use the same screening test data, that was collected for use with the lumped parameter method, to also determine the sparse nature of the system for partitioning?

Research Hypothesis 3.2: It is hypothesized that since the data from a screening test, which is to be used to tear the system into disjoint subgroups, inherently contains the connectivity information; this data can also be used to create the intermediate to top-level mappings which are designed to account for the impact of system tearing.

Chapter VI

THE HIERARCHICAL DESIGN SPACE EXPLORATION PROCESS

This chapter provides the formal development of the two new methods hypothesized and formulated in Sections 3.5 and 4.5. The first method to be developed is a new hierarchically partitioned metamodel for use with large scale systems. The second method to be developed is a graph-based partitioning method based on experimental data collected from the system. The actual application of these two new methods will be combined into a step-by-step process for performing design space exploration and metamodel building for large scale, multi-objective computer simulations. The chapter will conclude with a discussion of the validation plan for the new design space exploration process.

6.1 Development of the Lumped Parameter Model

This section presents the development of the new large scale metamodeling method that comprises one of the primary contributions to this thesis. Before getting into the formal development of the method, it will be useful to first recap what spurred its formulation.

Recall that Section 3.4 concluded that none of the currently available large scale metamodeling methods met all of the requirements for creating accurate metamodels for the nozzle problem. Metamodels based on supersaturated DoEs and stepwise

regression were ruled out because of concerns with the complex aliasing structures associated with these designs. The VIM methods were ruled out because of concerns of how to partition the problem around the responses. The UTE method was ruled out because of accuracy and efficiency problems. The SPM method was ruled out because it was not general enough. Finally, the GCF method was ruled out because it could not be applied to problems without identifiable intermediaries.

Having ruled out all of the available methods, Section 3.5 hypothesized that even though none of the current methods individually met all of needs of the nozzle problem, perhaps combining the desirable characteristics and processes from several methods to create a new method would meet the necessary requirements. Recall, Baker (2002)[6] concluded that the original formulation of the UTE method produced meta-models were not sufficiently accurate. Baker hypothesized that UTE method's poor accuracy was due to a few important, missing interactions between variables in disjoint subgroups. Thus to solve the problem, Baker added variable redundancies between the subgroups to directly account for these interactions. These redundancies fixed the accuracy problem but significantly degraded the efficiency of the method.

Based on Baker's finding, it was concluded that a partitioned approach will most likely result in the missing of potentially important interactions which could degrade the accuracy of the partitioned metamodel. However, it was also concluded that variable redundancy was not the best solution to the problem. A connection was then made that the General Composite Function method was indirectly accounting for potentially important interactions; the interactions were indirectly captured via the intermediate to top-level mappings. Thus the hierarchical structure of the GCF method constitutes the first borrowed characteristic that will be used to create the new partitioned metamodeling method.

Figure 17 is a notional representation of a new hierarchically partitioned metamodel based on the GCF method. Figure 17 shows a generic system of four design

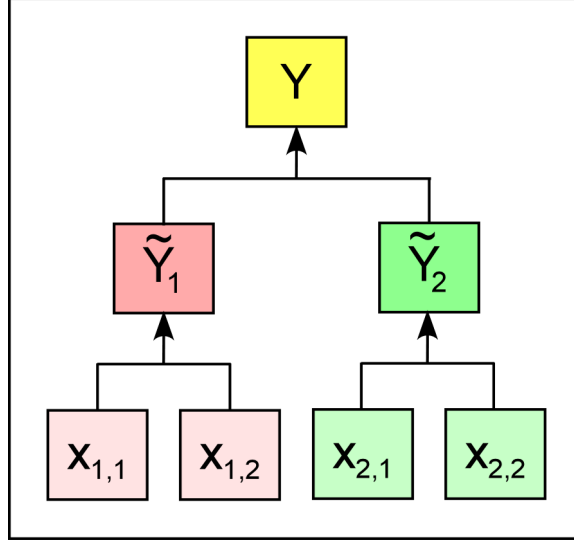
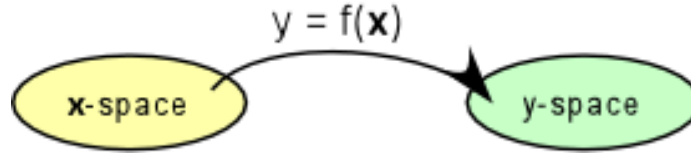


Figure 17: Generic Hierarchical Representation

variables denoted by the $x_{i,j}$'s divided into two subgroups or partitions. These partitions are linked to the top-level response, Y , via the intermediate responses, \tilde{Y}_i 's. The intermediate to top-level mapping will indirectly account for missing interactions between the design variable in disjoint subgroups.

The second borrowed idea is based on how Baker determined the missing interactions. Baker used a screening test to find and account for the missing interactions. Consequently, it was hypothesized that perhaps a screening could also be used to find and account for the potential missing interactions with the new method. The screening test would be used to somehow create the intermediate to top-level mappings. The “somehow” in the previous sentence is important because it is not intuitively obvious how a screening test can be used in this manner. The reason being is because the structure of the new hierarchical metamodel is based off of the GCF model and a GCF model is created by directly manipulating existing intermediate responses/variables to create the intermediate to top-level mappings. To make link between using the data from a screening test and the creation of the intermediate to top-level mapping for codes that do not have existing intermediate responses/variables, consider the



where,

- \mathbf{x} = a vector of input variables from the input space
- y = a response value from the output space
- \mathbf{x} -space = a mathematical set of vectors of design variables, \mathbf{x}
- y -space = a mathematical set of response values
- $y = f(\mathbf{x})$ = a generic function that maps from \mathbf{x} -space to y -space

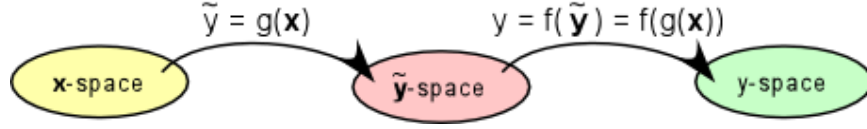
Figure 18: Generic Functional Mapping [138]

following development:

In abstract terms, a functional mapping is a relation that uniquely associates the members of one mathematical set (the inputs) with the members of another mathematical set (the outputs). Functions often have a many-to-one relation, meaning that the object from the input set is comprised of several sub-objects and the output set is one single object [176]. In the context of metamodeling, the metamodel itself is the functional mapping that links a vector of variable settings from the input space to a single value of a response in the output space. In general, any generic functional mapping can be represented as shown in Figure 18.

Notionally, any functional mapping of inputs to outputs can be depicted as shown in Figure 18. This figure can also be applied to metamodels, as a metamodel is typically a closed-form equation that links inputs to outputs. Note, in all the figures in this section, a bold symbol represents a vector of values. Thus any abstract space that is labeled using a bold symbol is a vector space. For example the mathematical set labeled “ \mathbf{x} -space” is a vector space because the “ \mathbf{x} ” is set in boldface.

The notion of a basic functional mapping can now be linked with the hierarchical



where,
 \tilde{y} -space = a mathematical set of intermediate response/variable values
 $\tilde{y} = g(\mathbf{x})$ = a generic function that maps from \mathbf{x} -space to \tilde{y} -space

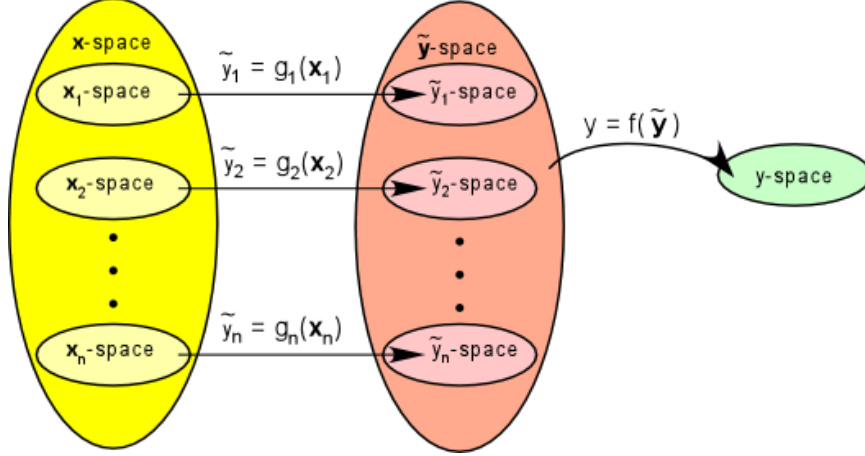
Figure 19: Generic Composite Function Mapping

mapping depicted in Figure 17. Recall that Section 3.3.2 stated that the basic mathematical relationship describing the mapping from the low-level design variables, x 's to the top level response, Y , via the intermediate responses, \tilde{Y}_i 's can be generically described as a composite function (i.e. $f \circ g \equiv f(g(x))$). Figure 19 shows a composite function notionally depicted using these terms.

Essentially, a composite function adds additional layers of mappings between the direct input to output mappings shown in Figure 18. However, Figure 19 is not quite descriptive enough to depict the general hierarchical metamodeling problem. Figure 20 is a more complete representation of the general hierarchical mapping depicted in Figure 18.

The primary difference between the generic composite function mapping shown in Figure 19 and the hierarchical mapping extended to a composite function form in Figure 20 is the partitioning of the \mathbf{x} -space and $\tilde{\mathbf{y}}$ -space into disjoint subspaces. But what do these disjoint subspaces actually mean?

In terms of the complete \mathbf{x} -space, the explanation of the subspaces is pretty straight forward. Let us assume that an analysis that has fifteen inputs, or x 's. Then, the total \mathbf{x} -space represents the set of all possible vectors of input values made up of values from each of individual x 's. If we were to partition this set of fifteen x 's into three disjoint sets, say (x_1, \dots, x_5) , (x_6, \dots, x_{10}) , and (x_{11}, \dots, x_{15}) , then, the subspace labeled, \mathbf{x}_1 -space, as shown in Figure 20, contains all of the vectors



where,
 $\mathbf{x}_i\text{-space}$ = a disjoint subset of the complete \mathbf{x} -space
 $\tilde{\mathbf{y}}_i\text{-space}$ = a disjoint subset of the complete $\tilde{\mathbf{y}}$ -space
 $\tilde{y}_i = g_i(\mathbf{x}_i)$ = a generic function that maps from $\mathbf{x}_i\text{-space}$ to $\tilde{\mathbf{y}}_i\text{-space}$

Figure 20: Hierarchical Mapping in Composite Function Terms

comprised of values only from the subset (x_1, \dots, x_5) . Likewise the subspace labeled, $\mathbf{x}_2\text{-space}$ contains all of the vectors comprised of values from the subset (x_6, \dots, x_{10}) , and so forth and so on. Thus, a general vector of values from the \mathbf{x} -space is comprised of a union of sub-vectors from each of the subspaces.

Moving on to the $\tilde{\mathbf{y}}$ -space, Figure 20 shows that each of the $\tilde{\mathbf{y}}$ subspaces are created by using some function, $g_i(\mathbf{x}_i)$, that maps values from the \mathbf{x} -space subsets into the $\tilde{\mathbf{y}}$ -space subsets. Thus if the function $g_i(\mathbf{x}_i)$ that maps from one space to the next is known, a set of defined values from the \mathbf{x}_i subspace will produce a sets of defined values from the $\tilde{\mathbf{y}}_i$ subspace.

Consider for a moment the functional mapping between the $\tilde{\mathbf{y}}$ -space and the top-level \mathbf{y} -space. The \mathbf{y} -space in this case is a generic place holder for a response from some analysis. The function $y = f(\tilde{\mathbf{y}})$ is a multivariate function that takes in a vector from the $\tilde{\mathbf{y}}$ -space and produces a single response value, y . The $\tilde{\mathbf{y}}$ vector is the union of individual values, one from each of the $\tilde{\mathbf{y}}_i$ subspaces. Thus if the function $f(\tilde{\mathbf{y}})$

is known, a set of defined values from the \tilde{y}_i subspaces will produce a set of defined values from the y -space.

With all of the previous three paragraphs in mind, if we were to take a complete vector from the \mathbf{x} -space, break it down into its constitutive sub-vectors, and map each of these sub-vectors through the appropriate $g_i(\mathbf{x}_i)$ functions, we would get several sets of \tilde{y}_i values. If we then combine individual values from the sets of \tilde{y}_i values into a vector, $\tilde{\mathbf{y}}$ and map $\tilde{\mathbf{y}}$ through the $f(\tilde{\mathbf{y}})$ function we then would have a value y . Therefore, in a two-stage fashion we have mappings of complete vectors in the \mathbf{x} -space to a values in the y -space.

Returning to the previous example, let us assume again that a certain analysis that is comprised of fifteen inputs, the x 's, and one output, y . Let us further assume that the set of inputs is partitioned into three disjoint sets, (x_1, \dots, x_5) , (x_6, \dots, x_{10}) , and (x_{11}, \dots, x_{15}) .

If we then independently assessed the impact of each of these subsets of x 's on the response, y , we could create three separate metamodels such as $y = g_1(\mathbf{x}_1)$, $y = g_2(\mathbf{x}_2)$, $y = g_3(\mathbf{x}_3)$, where each y is only a function of one the set of x 's. So in a sense, each of the previous y 's could be considered partitioned or “localized” versions of the total response y .

The redundant use of the symbol “ y ” to denote the mathematical quantities represented by the \tilde{y}_i 's and the response y is not sloppiness on the part of the author. But rather the redundant use of “ y ” is to denote that the \tilde{y} 's are *localized* versions of the top-level response, y . Thus the general hierarchical formulation can be summarized as: the top-level response, y , is a function of the localized versions of the response, \tilde{y} 's, which are in turn functions of the partitioned subsets of the variables. This formulation is called the Lumped Parameter Model (LPM).

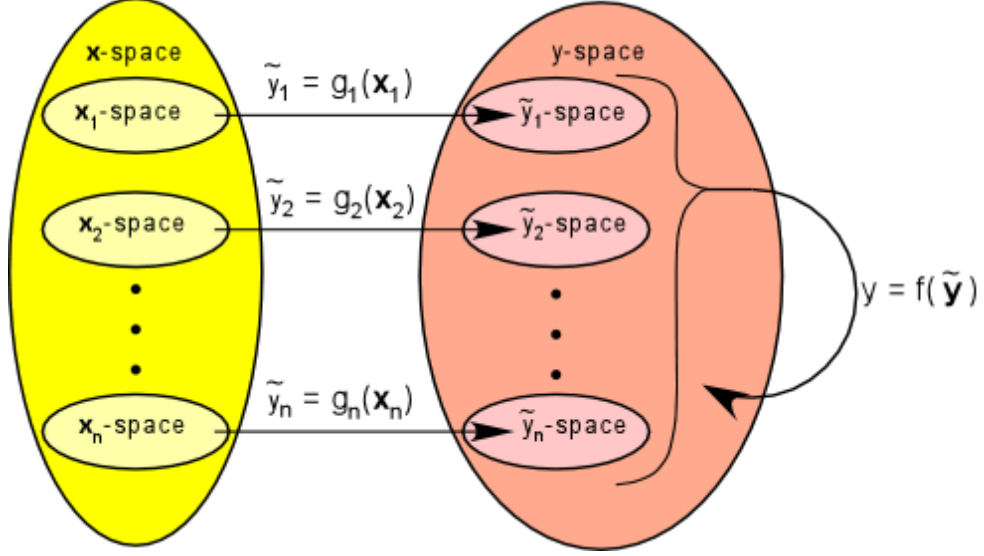


Figure 21: Functional Mappings for the Lumped Parameter Model

The name “Lumped Parameter Model” was chosen because the effects of the individual design variables on the top-level response are *lumped* together via the intermediate, localized versions of the top-level response. Since the intermediate responses, the \tilde{y} ’s are actually versions of the top-level response, then the \tilde{y} subspaces shown in Figure 20 actually are subsets of the actual y-space. Thus, a more appropriate explanation of the $y = f(\tilde{\mathbf{y}})$ function is that the function $f(\tilde{\mathbf{y}})$ is a function that takes a set of values from subsets of the the y-space and maps these subsets back onto the originating space. The complete depiction of the functional mappings that occur in the lumped parameter model is shown in Figure 21.

Even with this formulation it is still not exactly clear how to use the LPM formulation in Figure 21 with a screening test to determine the functional mapping. Recall, if we were to take a complete vector from the \mathbf{x} -space, break it down into its constitutive sub-vectors, and map each of these sub-vectors through the appropriate $g_i(\mathbf{x}_i)$ functions, we would get several sets of \tilde{y}_i values. If we then combine individual values from the sets of \tilde{y}_i values into a vector, $\tilde{\mathbf{y}}$, and map this vector through the $f(\tilde{\mathbf{y}})$ function we then will then have a resultant value y . Therefore, in a two-stage fashion,

there is a mapping of complete vectors in the x -space to a values in the y -space.

Let us assume for a second that we have partitioned up our analysis inputs and executed a design of experiments on each of the sets of partitioned inputs. With the data from each of the sublevel or input partition analyses, we can create each of the $\tilde{y}_i = g_i(\mathbf{x}_i)$ functional mappings. Now lets assume we preformed some type of screening test on our analysis, what type of information would we now have? First, we would have the \mathbf{x}_i to \tilde{y}_i mappings from the individual subgroup analyses and from the screening test, we would have a set of data that represents a mappings between complete vectors of x values and a set of specific y values. The function that produced that mapping of course was the original analysis.

If we then were to break each of the vectors of x 's from the screening test into sub-vectors and map these sub-vectors through $\tilde{y}_i = g_i(\mathbf{x}_i)$ functions, we would now have a set of \tilde{y}_i values that are directly related to the set of y values from the results of the screening test. The set of \tilde{y}_i values mapped to y values can now be use to create the $y = f(\tilde{\mathbf{y}})$ function. That is how a screening test can be used to create the intermediate to top-level to mappings.

In a more visual fashion, Figure 22 summarizes the results of breaking up the runs of screening test and mapping the sub-vectors through the $\tilde{y}_i = g_i(\mathbf{x}_i)$ functions to get the an array of \tilde{y}_i values. The array shown on the left hand side of Figure 22 is a standard screening array with the resultant y values from the screening analysis. The array shown on the right hand side of Figure 22 is the regression array that will be used to determine the intermediate to top-level mapping, that is the $y = f(\tilde{\mathbf{y}})$ function.

Thus far in the development of the lumped parameter model and the process used to create a LPM, the discussion has been limited to the case of only one top-level response. Obviously to use the LPM with the nozzle problem, which consists of 255 responses, it would be preferable to not have to run 255 separate screening tests.

Run	Subvector 1		Subvector 2		Subvector 3		Y				
	x1	x2	x3	x4	x5	x6		\tilde{Y}_1	\tilde{Y}_2	\tilde{Y}_3	Y
1	1	-1	-1	1	1	-1	109	105	93	109	109
2	-1	1	-1	1	1	-1	93	91	100	97	93
3	-1	-1	1	1	-1	-1	98	101	106	108	98
4	1	-1	1	1	-1	-1	109	103	108	106	109
5	-1	1	1	-1	-1	-1	105		99	95	105
6	1	-1	1				96		101	91	96
7	1	1	1				95		90	92	95
8	-1	1	1				99		110	107	99
9	-1	-1	1				95		106	109	95
10	-1	1	-1				92		108	93	92
11	1	1	-1	-1	-1	1	90	107	93	95	90
12	1	1	1	1	1	1	108	97	109	101	108
13	1	1	-1	1	-1	-1	92	95	105	92	92
14	-1	-1	-1	-1	-1	-1	96	106	100	106	96
15	1	-1	-1	1	1	1	102	101	100	106	102
16	-1	-1	-1	1	-1	1	109	96	104	100	109

Figure 22: Creation of \tilde{Y} Regression Array

That would surely negate any potential run savings associated with the method. Fortunately, the method laid out so far only needs one minor change to address the multi-objective problem. A simple example will illuminate the solution.

Consider the differences between applying a stratified Monte Carlo sampling scheme to a multi-objective problem and applying an adaptive-importance sampling scheme to the same problem. Recall that a stratified sampling method attempts to quantify the probabilistic nature of a set of responses by spreading out the sampling points as evenly as possible over the whole input space.¹ Each input is evaluated at multiple levels over its defined range. Thus every region of the input space is explored. This then implies that the entire output space is also explored, and not just for one response but for every response at the same time.

Contrast this to an adaptive importance sampling scheme which only chooses sampling points that are “optimal” for one particular response. These optimal points are most likely not the optimal points for another response. Thus while the output space for one response is adequately explored, the output space for another is woefully under explored. This essentially describes the problem of using a standard fractional

¹See Section 2.1 for a discussion of stratified and adaptive-importance sampling.

factorial design to create the intermediate to top-level mapping.

It has been the author's experience that using the results of a fraction factorial design to create the intermediate to top-level mappings results in \tilde{y} -spaces that are very disjoint and not even explored. Consequently the metamodels that result from regressing the y -values against the \tilde{y} regression arrays are of a particularly poor quality. To solve this problem the author essentially used a stratified sampling approach to explore the \mathbf{x} -space. Space filling type designs such as latin hypercubes seem to work very well for creating the intermediate to top-level mappings.

By using a space filling design, the resultant \tilde{y} regression arrays seem to have a nice even distribution of points throughout each of the \tilde{y} -spaces. In addition, any multicollinearity [179] that exists between the regression array columns does not seem to negatively impact on the results of the regression.

Based on the previous development of the lumped parameter model, hypothesis 2.2 can now be addressed. Hypothesis 2.2 was concerned with how to extend the GCF method to systems without identifiable intermediaries. The GCF method relied on existing intermediate responses/variables to perform the necessary design space exploration to create the intermediate to top-level mappings. Obviously then, for systems that do not have these available intermediaries, the GCF method is not a viable method to use to create hierarchically partitioned metamodels.

First, recall that we wanted to use the GCF model as a prototype for the LPM because of the GCF's ability to indirectly account for missing interactions between the disjoint subgroups. The idea to use the intermediate to top-level mappings to indirectly account for the missing interactions was in direct response to the accuracy problems that the UTE method exhibited. Baker solved the UTE accuracy problem by first using a screening test to determine the important, missing interactions and then used variable redundancies between the subgroups to directly create the

necessary interactions. However, by resorting to variable redundancies, Baker significantly degraded the efficiency of the UTE method. Thus, to use the GCF model as a prototype for the lumped parameter model, the problem of how to do the design exploration to create the intermediate to top-level mappings needed to be solved.

The proposed method follows Bakers lead and uses a screening test to get a direct mapping between the unpartitioned \mathbf{x} -space and the top-level y -space. With this mapping, the inputs to this design space exploration can then be broken into their constitutive sub-vectors and individually evaluated with the $\tilde{y}_i = g_i(\mathbf{x}_i)$ functions to create the \tilde{y} regression arrays. The individual \tilde{y} regression arrays can then be combined with the response data from the screening test to create the $y = f(\tilde{\mathbf{y}})$ the intermediate to top-level mappings.

Assertion: Baker used a screening test to determine important, neglected interactions and then used variable redundancy to account for these interactions.

Assertion: In the hypothesized method built on the GCF model, these interactions will be indirectly accounted for via interactions between the intermediate responses in the intermediate to top-level mapping.

Research Question 2.2: Can the data from screening test be used to determine the intermediate to top-level mappings for systems without identifiable intermediaries?

Research Hypothesis 2.2: It is hypothesized that since the data from a screening test data inherently contains the connectivity of the system, this data can be used to create the intermediate to top-level mappings for systems that do not have readily identifiable intermediaries.

Validity of Hypothesis 2.2: The logical development of the lumped parameter model (LPM) demonstrates how data from a screening test can be used to

create the intermediate to top-level mappings for systems that do not have readily identifiable intermediaries.

Partitioning the inputs to a screening test into their constitutive sub-vectors and individually evaluating these sub-vectors with the $\tilde{y}_i = g_i(\mathbf{x}_i)$ functions will create the \tilde{y} regression arrays. The \tilde{y} regression arrays can then be combined with the response data from the screening test to create the $y = f(\tilde{\mathbf{y}})$ intermediate to top-level mappings.

6.2 Development of the Graph-based Partitioning Method

This section presents the formal development of the graph-based partitioning method that comprises the other primary contribution to this thesis. Before getting into the development of the method, it will be useful to first recap what spurred its formulation.

Recall in Section 4.5 that the systems decomposition schemes used by the electrical engineering community to simplify the analysis of VLSI circuits were not directly applicable to the problem of partitioning design variables for use with large scale metamodeling. The primary reason being is that in the electrical engineering community, system decomposition methods are usually applied to a systems where the structure is known a priori. Meaning, an electrical engineer starts off the analysis with an existing system of equations or an existing circuit diagram. In the case of large scale metamodeling, the model builder starts with a black box and in general knows nothing about the underlying structure of the system. Consequently, how can the sparse matrix methods be used to decompose the problem when little is initially known about the system.

To answer that question, Section 3.4 proposed again borrowing from Baker's improved method. Recall that Section 3.4 stated that Baker used a screening test to

determine important interactions that he was neglecting. He then used these results to add factors to the current set of partitions. In effect, he was using the screening test to gain further insight into the structure of his system. Thus, it was hypothesized that a screening could also be used to determine the unknown structure of the system.

As it turns out, a screening test works very well for determining the interconnectedness of the system. Determining the underlying structure of the systems is as simple as performing a screening test and using ANOVA to separate the active or important factors from the inactive ones for each response. The results of ANOVA will produce a sparse incidence matrix that then can be used with a min-cut algorithm to partition the system into disjoint subgroups.

Before the graph-based partitioning method for large scale metamodel is formally laid out, a few important points need to be considered. First the screening test. Recall the lumped parameter model is using the results from a space filling design to create the intermediate to top-level mappings. In addition in Section 4.5 it was hypothesized that perhaps the same design that was used to determine the system structure for partitioning could also be used to create the intermediate to top-level mappings. Thus it would be ideal to double dip, that is, to reuse the initial screening data to keep the efficiency of the method as high as possible.

It has been the author's experience that using a traditional resolution IV fractional factorial design [106] works very well for determining the underlying structure for partitioning. However as was stated in Section 6.1, a fractional factorial design does not seem to work well for creating the intermediate to top-level mappings, thus a space filling design was recommended. To solve this apparent contradiction the author recommends a hybridized design. Half of the design is a resolution IV fractional factorial and the other half is a latin hypercube design. The latin hypercube should be "optimal" in the sense that the pairwise correlations between the factor columns are minimized.

One problem with using an optimal latin hypercube is that few statistical packages offer latin hypercubes as a design of experiments option. Fortunately, Matlab contains a simple function called “lhsdesign” [167] that creates fairly good designs quite quickly. In addition, a paper by Joseph and Hung (2006)[123] provides an algorithm for creating excellent latin hypercube designs. The Joseph-Hung algorithm minimizes the pairwise correlations between the factor columns while at the same time maximizes the distance between each of the design points.

The second point that needs discussing is on the use of the ANOVA results to determine the sparse nature of the system. A typical ANOVA analysis gives the user several statistical measures of the system, such as the sum of squares, F-test values, and p-values for tests of significance [109]. Unfortunately the problem with these statistical measures is that they were developed to quantify stochastic systems, not deterministic computer simulations. Thus as several authors point out [90, 142], these quantities have no statistical meaning for deterministic computer experiments and should not be used to determine which effects have any statistically significant impact on a response. However, none of these authors provide any alternative means of reducing the number of effects in deterministic systems. That is, they do not provide any means for determining the sparse nature of a deterministic system via experimentation. Fortunately, it has been the author’s experience that even though these methods were not developed for deterministic systems, they seem to be robust to this apparent misuse.²

²It should be noted that while the author appreciates the fact that using statistical measures developed for stochastic systems is not necessarily appropriate for deterministic systems, it should also be noted that this is an engineering thesis. Consequently, engineering theses are intended to address and solve problems and not necessarily be 100% mathematically pure. If a method was not created for a certain problem but that method works well to solve the problem, does that not justify its use?

This problem of extending statistical measures intended for stochastic systems to deterministic systems is an item that needs to be relegated to a future work. The author would like to have addressed this problem in this thesis, but solving that problem is most likely a thesis in itself!

6.3 The Hierarchical Design Space Exploration Process

This section will combine the methods developed in the previous two sections with the standard design and analysis of experiments method to create a step-by-step process for analyzing large scale systems. Following the set of instructions, the method will be outlined in a visual manner so that the reader can easily follow how the data that is generated and manipulated in the process is related to the general input and output spaces for the large scale system under analysis. See Figure 23 at the end of this chapter for a visual process outline.

The Hierarchical Design Space Exploration Process (Hi-DSE):

1. Determine the set of inputs and outputs to be investigated.
2. Determine the range of values over which the inputs will be varied.
3. Create the screening design of experiments that will be used to collect the data that will be used for both partitioning and to create the intermediate to top-level mappings.
 - A good choice for this design is a hybridized design where 50% of the runs are a resolution IV fractional factorial and the rest are an equally sized space filling type design such as a correlation minimized latin hypercube.
4. Using the screening design from Step 3 and the variable ranges from Step 2, collect an the initial set of response data.
5. Apply a main effects only ANOVA analysis to the data.
 - A “main effects only analysis” implies that the linear model created for the ANOVA analysis only contains main effects, no two-factor interaction terms.

6. Using the p-values from the ANOVA results, determine the “statistically” important variables for each response.
 - The standard statistical assumption that effects with p-values less than or equal to 0.01 are significant seems to work well for determining the sparse nature of the system.
7. Using the list of significant factors for each response from Step 3, create the incidence matrix for the system.
 - Assign the variables to the rows and the responses to the columns.
 - A statistically important variable for a given response is given a value of “1” in the matrix and a “0” otherwise.
 - In general, the incidence matrix of the system will not be square.
8. Create the variable adjacency matrix by multiplying the incidence matrix by the transpose of itself.
 - $C_v = A * A^t$ where A is the incidence matrix and C_v is the variable adjacency matrix.
 - The response adjacency matrix can be created by multiplying the transpose of the incidence matrix by itself. $C_r = A^t * A$
 - The adjacency matrices will no longer be made up of solely of 1’s and 0’s. An element in the variable adjacency matrix, c_{ij} , is the number of times variable i occurs with variable j in the same response.
 - The diagonal elements, c_{ii} , represent the total number of responses that variable i is significant in.

9. Create, and examine the graph of the adjacency matrix.
 - Oftentimes, a good set of partitions will be fairly obvious.
 - What is more important however is to determine if the graph *can not* be partitioned. A system with a strongly coupled graph where every variable appears connected to every other is not a good candidate for partitioning.
 - Both Mathematica and Matlab provide built-in functions for creating and manipulating graphs.
10. Choose a set number of partitions, and apply a min-cut algorithm to the graph.
 - Most min-cut algorithms require the user to specify the number of partitions to decompose the graph into. Typically they will try to create as equally sized partitions as possible.
11. Overlay the partitioned sub-graphs that result from the min-cut operation on top of the original graph.
 - The overlaying of the partitioned graph on the original graph provides intuitive, visual way to address the quality of the partitioning.
12. Repeat Steps 10 and 11 to examine several different numbers of partitions and pick the one that is best.
 - The best set of partitions is one that provides a good trade off between run savings and missing interactions.
 - As Guardabassi and Sangiovanni-Vincentilli (1976)[45] state, it is also important to pick a set that is justifiable using engineering intuition.

13. Using the set of input variable partitions from Step 13, individually assess the impact of each set of variables on the set of responses.
 - The ranges for each variable are the same ranges defined in Step 2.
 - Standard designs of experiments can be used for each of the subgroup analyses.
 - While one subgroup is being analyzed, hold the values of variables in the other groups at their nominal values.
14. Analyze each of the subgroup analyses from Step 13 to create the $\tilde{y}_i = g_i(\mathbf{x}_i)$ functions.
 - If there are n top-level responses, there will be n intermediate $\tilde{y}_i = g_i(\mathbf{x}_i)$ functions per subgroup.
15. Partition up the hybridized, screening design to create the individual \tilde{y} regression arrays.
 - If there are n top-level responses, there will be n individual \tilde{y} regression arrays.
 - There will be one column per subgroup in each regression array.
 - For n subgroups, and response, y_i , the indices of the columns for regression array i will be $(\tilde{y}_{i1}, \tilde{y}_{i2}, \dots, \tilde{y}_{in})$.
16. Analyze each of the regression arrays from Step 15 with the response data from Step 3 to create the $y_i = f(\tilde{\mathbf{y}}_i)$ functions.
17. Link the outputs of the low-level to intermediate functions, $\tilde{y}_i = g_i(\mathbf{x}_i)$, to the appropriate inputs of the intermediate to top-level functions, $y_i = f(\tilde{\mathbf{y}}_i)$, and you will then have a complete set of lumped parameter models.

Figure 23 at the end of this chapter shows the hierarchical design space exploration process in graphical form.

6.4 *Validation Plan*

The validation requirements for the contributions of this thesis are actually quite straight forward and can be broken into two separate parts. The two separate validation steps are: validation of the lumped parameter modeling method and validation of the graph-based partitioning method.

6.4.1 For the Lumped Parameter Metamodeling Process

Although the primary motivation for this work was to find or develop a method that enables accurate and efficient probabilistic lifing of first stage nozzles, actually implementing the full lifing analysis is not necessary. Recall that Section 2.4 determined that an indirect Monte Carlo method was the most appropriate probabilistic design method to use with the nozzle problem. The primary reason was that traditional Monte Carlo simulation combined with metamodels was the only method that possessed both the potential efficiency and could evaluate multiple responses simultaneously.

The metamodel/Monte Carlo combination has been successfully demonstrated numerous times in the literature [37, 134, 170], thus the accuracy of the method itself is not in question. Intuitively however, the accuracy of the combined method rests solely on the accuracy of the metamodels. If the metamodels accurately reproduce the behavior of the system, then the Monte Carlo results will be as if they were obtained directly from the original model. Thus the primary validation for the new lumped parameter model will be how well it mimics the response of the actual system.

Absolute accuracy however will not tell the whole story. Recall that Section 1.5 stated that given the long runtime of this model, traditional statistical linear model based metamodeling schemes were not deemed efficient enough and that perhaps

the partitioned nature of the nozzle model could be leveraged to reduce the total computational expense required to create the metamodels. Thus not only is the absolute accuracy of the model important but it should also be benchmarked against traditional statistical linear model based metamodeling schemes applied to the same problem. If the new method is more efficient, however not as accurate, then this will preclude its usefulness for application to the nozzle model and to the engineering community as a whole.

The only way to definitively tell if a metamodel is an accurate representation of the system is to compare its predictive accuracy at off-design point locations. In fact several authors³ state that the only way to truly determine the accuracy of computer based models is to evaluate the predictive accuracy at off-design point locations. Off-design point meaning combinations of variables that were not used to create the model.

Thus the relative accuracy of the LPM models and several traditional statistical linear model based metamodeling approaches will be compared by individually assessing each of their accuracies with a sample from the actual model at several thousand off-design points.

The comparison of accuracy between the two methods will not be based on mean error or maximum absolute error but rather based on comparing the distributions of each of the relative errors. Relative error, in this case is the percent error. As will be shown in Chapter 8 and Appendices A and B, all of the relative error assessments have an approximate mean of zero and are all symmetrical distributions around that mean, thus the primary differences between the individual error distributions are their variances. Consequently, the assessment of one metamodeling method versus another will consist of comparing the ratio of their variances referenced to a baseline metamodeling method. The idea being that a model with a lower variance of error

³See Lin (2004)[90], Sections 2.3.2 and 2.3.3 for a thorough review.

about the mean is deemed to be a more accurate model.

6.4.2 For the Graph-based Partitioning Method

The validation process for of the graph-based partitioning method is less straightforward than the validation process for a metamodeling scheme. For a metamodeling scheme, it is easy to assess the accuracy of the models it creates and to benchmark its performance versus other existing metamodeling schemes. It would be nice if the same process could be applied to the graph-based partitioning scheme. Unfortunately, to the author's knowledge this is the first time a similar graph-based partitioning method has been used to characterize and partition a large system for use with metamodeling. Consequently an objective, benchmark-based validation plan that compares the quality of the partitions created by the proposed method to an available method(s) is not an option.⁴

One simple way to validate the process is to use the graph-based partitioning method to create several different groups of partitions, and then apply the validation plan laid out in Section 6.4.1 to assess the accuracy of each of the LPMs created with these sets of partitions. In addition, one can then use phenomenological knowledge of the system to qualitatively assess the goodness of a given set of partitions.

Even though there are no other quantitative based partitions schemes to benchmark against, Section 3.3.1 did discuss two qualitative means for partitioning a large scale problem. Those methods were partitioning based on engineering intuition and partitioning along code or process boundaries. Since the partitioning method is intended for use with individual, black box codes, obviously process or code boundary partitioning is not an option. However, engineering intuition (i.e. phenomenological knowledge about the system) can still be used to partition the problem. Thus if metamodels based on the graph-based partitioning are more accurate than the metamodels

⁴See the end of Section 4.4 for a review a summary of graph theoretic methods used with metamodeling and statistical analysis.

based on engineering intuition, then that would validate the graph-based partitioning method. An even better outcome would be if the graph-based partitioning method created the same, or very similar, sets of partitions as was created using engineering intuition.

As it turns out, the graph-based partitions can in fact be benchmarked versus a previously developed engineering intuition-based partitioning scheme. Of the two methods developed for this thesis, the LPM method was formulated, developed and benchmarked before the work on the graph-based partitioning method had even been contemplated. Consequently, the author needed a set of partitions to use with the new metamodels for benchmarking. Initially it was hypothesized that there were two fairly obvious ways to partition the nozzle model: subsystem partitioning and variable-type partitioning. The subsystem partitioning grouped all of the variables pertinent to a spatial zone together, thus each geographic zone was considered a subsystem. For instance, all of the variables associated with zone 1 were grouped together: zone 1 wall thickness, zone 1 internal temperature, zone 1 external temperature, etc. The variable-type partitioning on the other hand grouped like variables together, for instance all of the wall thickness variables were grouped together, likewise all of the external temperature variables were also grouped, etc. Thus these two partitioning schemes could be used to benchmark the graph-based partitioning method.

Therefore the validation plan of the graph-based partitioning method will consist of creating several different sets of partitions and validating the metamodels created from these partitions using the validation plan from Section 6.4.1. The accuracy of the graph-based partitions will be compared with the variable-type and subsystems based partitions as well with several traditional all-at-once metamodeling schemes.

6.4.3 The Validation Test Cases

Obviously the best test case to validate the methods against would be the model that spurred the development of these methods, namely the nozzle model. However this model can not be used to assess the validity of the new large scale metamodeling scheme or the graph-based partitioning method.

The reason is purely practical. Given the very long runtime of the nozzle model, 3.5 hours per run, assessing the accuracy of the each of the methods discussed in Sections 6.4.1 and 6.4.2 would take on the order of 30,000 hours (or 1,250 days).⁵ Clearly the nozzle model is not the proper vehicle for method assessment. Thus, an alternative model that runs significantly faster yet captures the salient features of the nozzle model is needed.

The first test case is a parametric, finite element model of a hollow I-beam. The hollow I-beam is spatially partitioned in a similar manner as the nozzle model and has all of the same characteristic variables as the nozzle model. The model is partitioned into six spatial zones. In each spatial zone, the wall thicknesses, internal and external temperatures as well as the internal and external heat transfer coefficients are allowed to vary. Thus the model has 30 independent variables. The responses were also collected in a zonal fashion. The maximum temperature and stress from each zone was recorded, thus the model has 12 independent responses. The total run time for the hollow I-beam model is approximately 1.5 minutes on a vintage 2006 computer. Chapter 8 will go into much more detail on the hollow I-beam model.

System responses modeled with finite element analyses are often referred to as implicit functions [48], that is, they do not have a nice closed formed solution. Thus, the solution technique is to discretize the system and use FEA to iteratively solve the

⁵This assessment is based on assuming that each partitioned method would take on average 50% of the runs for the complete model (approximately 1430 runs) and that there would be assessments of 2 complete models and 8 partitioned models. This also assumes that each run would be executed sequentially, obviously parallel evaluation of the runs can cut the total time down significantly.

problem. One problem associated with the finite element method is that the solution can be sensitive to variations in the discretization technique used. This means that for the exact same model, two different finite element meshes can give two different answers. Therefore it would be beneficial to apply the partitioning methods to a system that does not experience this type of numerical noise.

Therefore a second test case based on a closed-form model of a torsional-vibration system (TVS) is also proposed for the validation process. The TVS model has 18 variables and three responses and the entire behavior of the system can be captured in a series of closed form equations.

The closed form nature of the TVS model actually provides a very nice tool for assessing the graph-based partitioning method as well. Since all of the equations are known a priori, the exact sparse matrix representation of the system can easily be created. The exact sparse structure can then be compared against the sparse structure predicted from the graph-based partitioning analysis. This will provide further validation of the graph-based partitioning method.

6.5 Summary

This chapter was the formal development of the two new methods hypothesized and formulated in Sections 3.5 and 4.5. The first method that was developed was a new hierarchically partitioned metamodel for use with large scale systems. The model created with this new method is referred to as the lumped parameter model (LPM).

The second method that was developed was a graph-based partitioning method based on experimental data collected from the system. The process for performing the graph-based partitioning of large scale systems for metamodeling is referred to as the graph-based partitioning (GBP) method.

The actual application of these two new methods was combined into a step-by-step process for performing design space exploration and metamodel building for

large scale, multi-objective computer simulations. This new design space exploration process is referred to as the hierarchical design space exploration (Hi-DSE) process. Figure 23 is a visual depiction of the Hi-DSE process.

The chapter concluded with a discussion of the validation plan for the new design space exploration process. The validation plan outlined how the LPM and GBP methods will be assessed as well as the two test cases that will be used in the assessment.

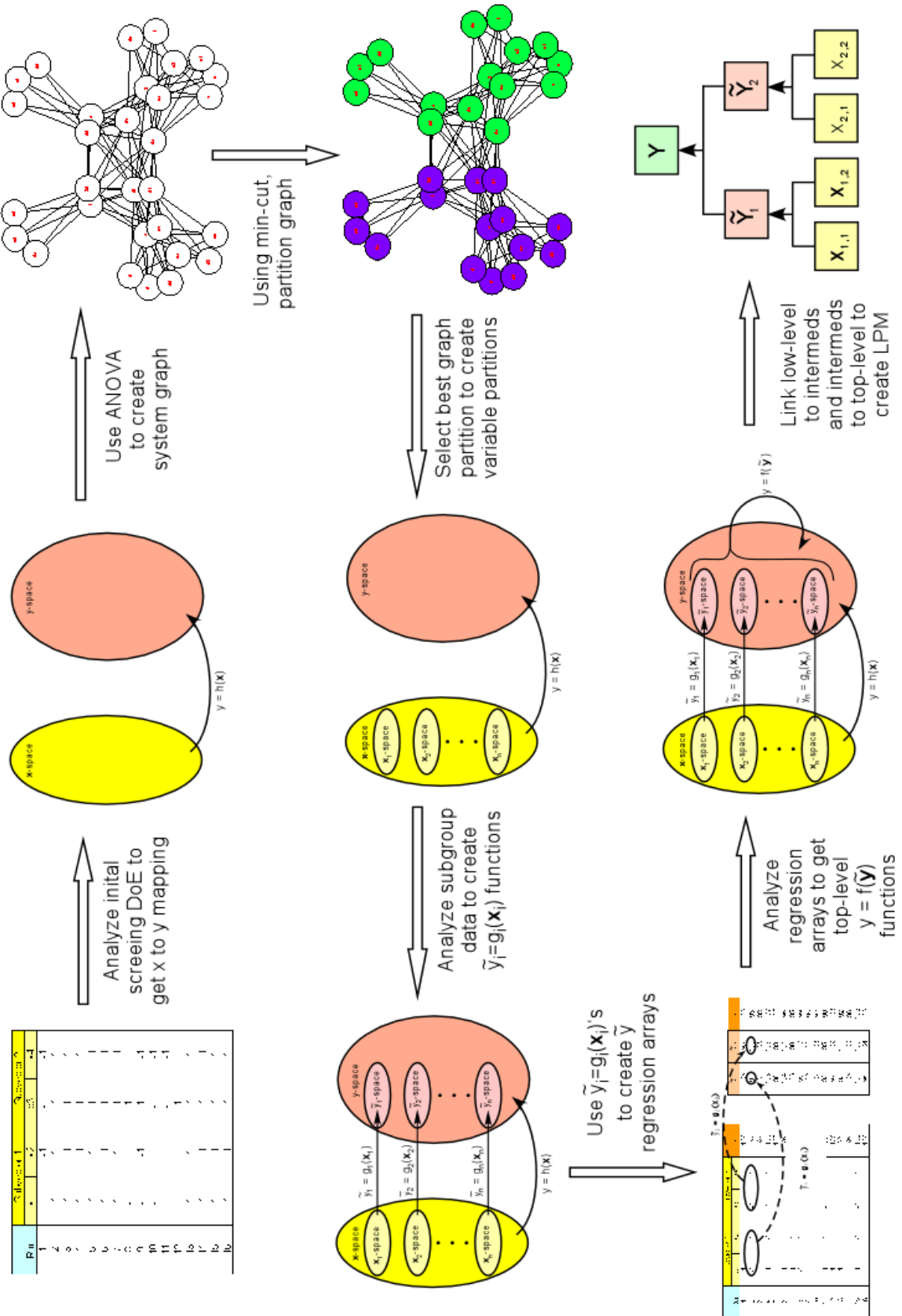


Figure 23: Hierarchical Design Space Exploration Process

Chapter VII

FORMULATION OF A PARALLEL, DISTRIBUTED COMPUTING ENVIRONMENT

This thesis has primarily been focused on methods for creating metamodels of large scale computer simulations without much discussion of the actual experimentation process. This chapter delves into practical aspects of the necessary computational frameworks needed to perform these computer experiments. The framework laid out in this chapter will form the core of the tool set used to collect and analyze the data for the validation of the hierarchical design space exploration process.

First, an overview of the history of the different alternatives to integrating computational frameworks is given. Next, a new framework will be introduced, which incorporates the latest computational techniques and more importantly a mind-set emphasizing flexibility, modularity, portability and re-usability. This introduction will include a thorough review of the fundamental design decisions that went into developing this new integrated computational framework [197].

Unfortunately, not all of the code that was developed for the new framework can be included in the thesis, doing so would add roughly 200 more pages to the appendices.

7.1 Need for Integrating Frameworks

Today's engineering designers have come to the realization that no longer can successful designs revolve around the analysis and optimization of a single discipline. But rather, successful designs are now viewed as a balance between competing disciplines. Given this, accounting for and balancing disciplines through the sharing of data between disciplines becomes a monumental task.

To date, several commercial applications and research programs [49, 108, 107] have been developed to aid in the dissemination of information between discipline analyses. Nonetheless, these tools do not always afford the designer the flexibility necessary to implement novel design space exploration techniques and data manipulation techniques.

The requirements for this research stem from the needs of MDO (Multidisciplinary Design Optimization). The term MDO was coined almost a decade ago. This relatively new field consists of the following principal conceptual components [153].

1. Design Oriented Analysis: System level designing allows the designer to answer the “what-if” questions. Designers want to know the sensitivity of the design with respect to the design variables.
2. Approximation Concepts: Metamodels allow the designer the ability to bypass the expensive direct coupling of analysis codes to the design space exploration tool. Common meta-modeling techniques such as Response Surface Methodology (RSM) and Neural Networks (NN) [145] can be used instead of these complex disciplinary analysis codes.
3. Mathematical Modeling of Systems: It is common that an engineering system is usually modeled by multiple disjoint analysis codes and not one monolithic code. Data reduction techniques may need to be applied if large amounts of data are exchanged between codes.
4. Decomposition: Given that codes analyzed on the same level are often tightly coupled, it is usually preferable (if possible) to decouple the individual codes and let the system-level take care of the coupling. Here system decomposition techniques and tools such as the Bi-level Integrated System Synthesis (BLISS [154]) are used.

5. Design Space Exploration: Exploration of the design space for optimization is the search to find the constrained minimum. Various algorithms such as Sequential Linear Programming (SLP) and Sequential Quadratic Programming (SQP) can be applied. Alternatively, where applicable, algorithms employing stochastic processes (Genetic Algorithms (GA) and Simulated Annealing (SA)) are also an option [29].
6. Optimization Procedures: System optimization is conducted at the system-level. The system-level optimizer knows which codes to execute and in what fashion. This element effectively ties together the different codes in an execution sequence.
7. Human Interface: Manual intervention in the system design process is important and is not an after-thought. In a well designed environment, the implementation should allow for straightforward, designer intervention. This intervention is needed since MDO often relies on human interaction to guide the process.

Typically each analysis code handles one disciplinary component of the overall systems engineering problem. This implies that at the top-level all these disparate codes, each with unique data formats and running on different platforms, need to communicate with each other through some system-level executor/controller. Furthermore, there is often a coupling of inputs and outputs between analyses resulting in an iterative analysis loop.

The problem is formulated by Sobieszczanski-Sobieski (1993)[153] as follows: “Engineering system analysis is expensive, time-consuming and a non-trivial managerial task”. Hence the clear need for a framework to handle distributed MDO problems. Such an integrating framework needs to capture all the enumerated MDO components if the implementation is to be successful.

7.2 *History of Integrating Frameworks*

7.2.1 Hard-Coded Algorithms

In the earliest frameworks, all disciplinary executables were brought together and execution control was given to a fixed algorithm. There are three variations of this formulation, in chronological order: monolithic codes, direct integration, and meta-modeling techniques. Most of the monolithic programs were written in FORTRAN and some examples of these efforts are still around, such as: FLOPS (Flight Optimization System)[99] and ACSYNT (Aircraft Synthesis)[2]. A general disadvantage of these systems is the relative difficulty to include higher-fidelity tools as they become available, since these approaches require a total reconfiguration of the script that controls the execution.

Until recently, there was no valid alternative to this *hard-coding* of programs. In the last decade, several commercial alternatives have been developed that address many of the short comings to the early frameworks.

7.2.2 Commercial Applications

Scott (2001)[149] gave a good overview of the commercial endeavors. The following high-level overview of three commercial packages is given here as an introduction.

Adaptive Modeling Language AML (Adaptive Modeling Language) is developed by Technosoft [162]. AML is built on the philosophy of object-oriented software design and uses LISP as its programming language, which is a fairly uncommon language. Variables are created by instances of some previously defined primitives. When defining formulas, AML automatically keeps track of which variables depend on others. AML has easy to use graphical visualization capabilities (especially for aerodynamic design). Some disadvantages from a user friendliness perspective are that a fairly good working knowledge is required of object-oriented programming. The use of object-oriented programming is

not necessarily detrimental as will be shown when discussing the open-source requirements. Unfortunately, integration with certain common tools (Excel spreadsheets etc.) is not yet functional.

iSIGHT iSIGHT is produced by Engineous Software [31]. iSIGHT is based on MDOL (Multidisciplinary Optimization Language), its own language. Pre-made building blocks are accessible through GUIs so to avoid direct interaction with the underlying language. Logic-based control and optimization boxes are readily available from the GUI environment. Options for parsing input and output files are very extensive. The linking between codes occurs simply by using the same variable names. Unfortunately, cross-platform integration of different codes and front-end is not straightforward.

ModelCenter ModelCenter is made by Phoenix Integration [120]. The front-end interface is called ModelCenter, while in the background the Analysis Server services the request coming from the ModelCenter GUI [121]. Using the ModelCenter “web-browser”, it is very easy to use a resource/code once it is wrapped and placed on the Analysis Server from any location. Response Surface generators, Monte Carlo simulation and stochastic optimization toolboxes were recently added to the basic package. One of the remaining drawbacks of ModelCenter is that multiple instances of a code, also known as parallelization, is currently not supported.

Given the ease of use and polished execution that these commercial packages exhibit, these environments have certain, inherent drawbacks as well. Most notably, they are not open-source thus not allowing the designer to tailor the tool to exactly meet the needs that the analysis may require. These tools allow for the designer to link codes through the GUI (Graphical User Interface), but do not always give direct access to the underlying core of the tool. Access to source code for extreme flexibility is a

very important prerequisite in a conceptual design research environment where these new methodologies are developed and investigated. Consequently, they are not truly conceptual design tools since they do not allow the investigation of concepts. They only allow for perturbations around a user-provided baseline input file.

In that sense, these commercial packages are preliminary design tools. To use these packages at the conceptual level is possible, however requires extensive changing and use of the sometimes provided API (Applications Programmer's Interface). As a result, they become not much different from an actual programming language.

An interesting comparison can be made by looking at Microsoft Excel and The MathWorks' Matlab. Excel inherently uses a GUI, the spreadsheet, to enter equations and visualize its output. Matlab on the other hand opens up a library of functions, which can be programmatically combined in a text file (the Matlab m-file) and executed from a command input window. Over time, Excel added the capability of VBA (Visual Basic for Applications), which allowed for more powerful operations comparable to Matlab. Nonetheless, Matlab is still a more powerful and flexible tool since it was conceived as an API. The same comparison is true for integrating frameworks: most commercial applications use a GUI to interact with the user. It is this GUI which makes these tools very user-friendly and easy to use, however, a general API built on solid object-oriented programming is potentially much more powerful.

7.2.3 Open-Source Applications

An alternative solution is to develop a general, systems analysis API. This API would be a general library of functionality that would allow the designer to programmatically link and execute any number of analyses and manipulate the resulting data in any conceivable manner.

Recent advances in business-to-business data transfer as well as server-client application interfacing have made the task for the engineer to develop such an environment

significantly easier. More specifically, it is possible to create an API that incorporates object-oriented code wrappers (Agents) written in Java with uniform, standard data transport tools (XML (Extensible Markup Language) and SOAP (Simple Object Access Protocol)) to create an infrastructure which is both platform independent and flexible to the needs of the designer.

Starting such a task from scratch would entail a significant programming feat for any designer/programmer. Fortunately, readily available tool boxes for optimization [114], statistics and simulation [65], visualization [157, 42], and web-server applications [166] are pre-written, plug-and-playable, and more importantly, generally open-source.

These tools combined with the use of the agents give total flexibility and modularity. This allows the designer to concentrate on the actual design task. More details on the open-source building blocks of the framework will follow in subsequent sections.

7.3 Formulation of an Open-Source Alternative

The previous section highlighted some of the advantages and disadvantages of commercial packages. An open-source tool should and can draw on the strengths and weaknesses listed above. Below are a list of items that are worthy for incorporation and investigating in this new integrating framework.

- Use the sound basis of object-oriented programming (from AML).
- Extensive tools for parsing input and output files (from iSIGHT).
- Logic-based control boxes are pre-written and available as functions (methods in Java) and in the API (from iSIGHT).
- Wrapped codes (also called *agents*) are immediately available from distributed servers (from ModelCenter).

- Methods for multiple instances of agents for parallelization need to be provided (from a shortcoming in ModelCenter and iSIGHT).
- Allow for growth potential when incorporating statistical (from ModelCenter), optimization (from iSIGHT), and visualization toolboxes (from AML).

7.3.1 Enabling Technologies

As was stated before, in recent years there has been an explosion of technologies developed to facilitate business-to-business data transfer and server-client application interfacing which can be readily employed to aid an engineer in the development of an open source distributed integrating framework. The following is a brief overview of the key technologies chosen to form the backbone of the framework.

7.3.1.1 *Java*

Java is best explained from an excerpt from Sun's (the developers of Java) web site [161]:

Java is a simple, object-oriented, distributed, interpreted, robust, secure, architecture-neutral, portable, high performance, multi-threaded, and dynamic language.

Essentially what this is saying is that Java offers an object-oriented, non platform-specific programming language that has built into the very structure of the language the ability to develop parallelizable, distributed code. All of these attributes are essential for the development of the distributed framework. These traits will allow Java to be both the development language as well as the language that the designer will program with to create the custom systems level analyses.

7.3.1.2 *Extensible Markup Language*

XML is a meta-markup language for text documents. The data is included as strings of text marked-up by tags describing the data. There are two important features to

XML which make it very useful in data transfer [57, 101].

Firstly, portability. Just like Java, XML is non-platform specific since it is merely a text file and can be directly transferred between platforms. Java and XML produce “portable code, portable data”.

Secondly, interoperability. The XML standard [188] specifies the format and structure of an XML file but not the content of the tags, the strings, the attributes, etc. An XML data structure or file can define airplane data as easily as it can contain a conference paper, as long as it conforms to the formatting standards.

7.3.1.3 Simple Object Access Protocol

Like XML, SOAP is a standard [187]. SOAP allows for straightforward data transfer protocol using HTTP (Hyper-Text Transfer Protocol) as the transport layer. The use of HTTP helps to resolve complicated issues as firewalls, ports, sockets, etc.

There are two implementations of the SOAP standard: Apache [166] and Microsoft. The Apache implementation specifies two methods to invoke SOAP services: RPC (Remote Procedure Call) and the message-based model. The former is used in this research.

7.3.1.4 Apache Tomcat Application Server

Distributed object computing extends an object-oriented system which allows objects to interact across heterogeneous networks and inter-operate as a unified whole. The workhorse of this distributed environment is the server. Servers act as an intermediary, brokering communication channels between the client and sever applications.

The Apache Tomcat Application Server [188] is an open source application web sever that uses SOAP as a means of deploying servlets and web applications. Although any application sever could be used for this function, the Apache Tomcat server was chosen because it is open source and the May 2005 Netcraft Web Server Survey found that 56% of the web sites on the Internet are using Tomcat, thus making it more

widely used than all other web servers combined.

7.3.1.5 *Agents and Models*

Earlier work by Hale and Mavris (1999)[51] documented the history of frameworks and the evolution to the new *collaborative* environment. All these frameworks try to aid the engineer in complex design problems with solutions that require analysis from several domains. The automation of the solution process requires control and communication of domain analysis be provided by the framework [50, 52].

Within this framework, a key technology is the implementation of the *agent*. Hale and Craig (1994)[49] added a new component to the agent: the *model*, and proposed an updated definition:

An agent is a resource, which has been modeled and wrapped for inclusion in a distributed design environment. The agent design requires a designer-centered, bi-directional wrap, independent of proprietary boundaries and capable of supporting increasing fidelity models.

These agents can generate accountable design information. This includes the “what, why, when, and where” information needed as decision-support for the designer. The result is intelligent agents, which effectively conceal the proprietary codes and data formats from the end user. The schematic in Figure 24 depicts the breakdown of these agents.

The wrapper is a bi-directional information exchange layer which shields the resource and model from the computing backbone. The main role of the wrapper is data exchange and conditioning.

The second element, the model, adds context to the information provided by the agent. Models include behavior and implementation information. The former is a mathematical formulation, engineering principle, or geometrical construction describing explicitly what the resource does. The latter captures execution characteristics:

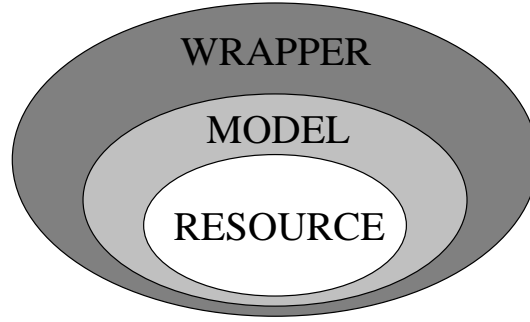


Figure 24: Notional Agent Configuration

variable definition, file descriptions, units, executions characteristics, and platform dependencies.

Using Java, this model is easily generated by the Javadoc utility. Javadoc is the tool for generating API documentation in HTML format from “doc” comments in source code. These “doc” comments include, but are not necessarily limited to, identification of parameters and methods provided by the wrapper and resource.

Lastly, the resource is the computer program. Typically these are off-the-shelf analysis codes. Examples include ASTROS, ANSYS etc. Basically any command line executable code can be wrapped with an agent and exposed for distributed computing.

7.4 Framework Design and Components

As in any object-oriented coding project, the extensibility and ease of reuse of the code hinges on the initial design of the class structure. For stand-alone programs, the designer must decide how much extensibility is necessary or desired. Non-extensible, one-time use code can often be a quick and dirty design just to satisfy current needs. On the other hand, building extensibility into the class structure requires a higher level of elegance. The design of a systems analysis API must be formulated with extensibility as the primary driver of the design because the designer will want to be able to easily incorporate new systems analysis methods and agents into the API as

they are developed. Also, the framework is designed to be a toolbox of functionality, thus reuse is inherently implied.

There are many published references on object-oriented design patterns and heuristics [38, 72, 95, 129] available to the designer to aid in the development of extensible class structures. Ultimately though, all of these patterns and heuristics employ one or both of the fundamental reuse mechanisms that form the basis of all true object-oriented programming languages: composition and inheritance. The level of reuse and extensibility exhibited by a design are directly related to the intelligent application of inheritance and efficacious use of composition.

In the design of the class structure for the distributed framework, there were two important heuristics on the use of inheritance and composition that were employed:

1. Dependencies or associations between classes should be linked through interfaces or abstract classes not concrete classes.
2. Build complex functionality into a class through the aggregation of simpler, functionality focused classes.

Granted, these rules of thumb can not be blindly followed. For instance, a major caveat associated with item one is related to the depth of the inheritance hierarchy used to define the interface and functionality of the actual concrete subclasses. The deeper the hierarchy, the more difficult it is for the user to navigate the class structure and determine at what level of abstraction to program to.

Item two on the other hand has the drawback that developing too focused of class functionality can result in an explosion of the number of classes. Thus it is necessary to decide the level of functionality that a class contains before it is advantageous to spawn some of that functionality off into a new class. Once spawned, composition can then be used to include the new class functionality into the original class, thus completely hiding the details of the implementation to the user class.

Figures 25 and 26 show UML (Unified Modeling Language) [95] diagrams outlining the basic class structures for the client and the sever side APIs in the distributed framework. Examining these two figures will show clear examples of how these two object-oriented design heuristics where used.

7.4.1 The Client Side API

As can be seen in Figures 25 and 26, the core functionality of the distributed framework can be easily divided into two separate collections of classes, the client and server APIs. The client API is the library of classes which the user directly programs with when performing a systems analysis. The main functionality of this API is primarily centered around one type of class, the design space explorer (DSE) classes. The Code classes and the Metamodel classes are simpler in nature and designed to work in conjunction with the execution of the design space explorer classes. *Note - Figure 25 is actually a subset of the total client side API. This subset was chosen as a representative portion of the actual API because it demonstrates all of the fundamental design issues that went into the creation of the client side API.*

The main focus of Figure 25 is the AbstractDSE class. The AbstractDSE is an abstract class that provides all of the core functionality that all DSE subclasses will need. This includes data management tasks such as persistence and data storage, serial and parallel execution of code and metamodels, and creation of the XML input files for the Code classes. The only functionality that the AbstractDSE class is lacking is the intelligence to perform a specific design space exploration. DoeDSE is an example of a class that inherits from the AbstractDSE class and adds the ability to conduct a design of experiments (DoE) analysis around any Code or Metamodel object.

Other DSE type classes that are not shown on Figure 25 but are included in the actual client side API include a Monte Carlo class, various types of optimizer classes

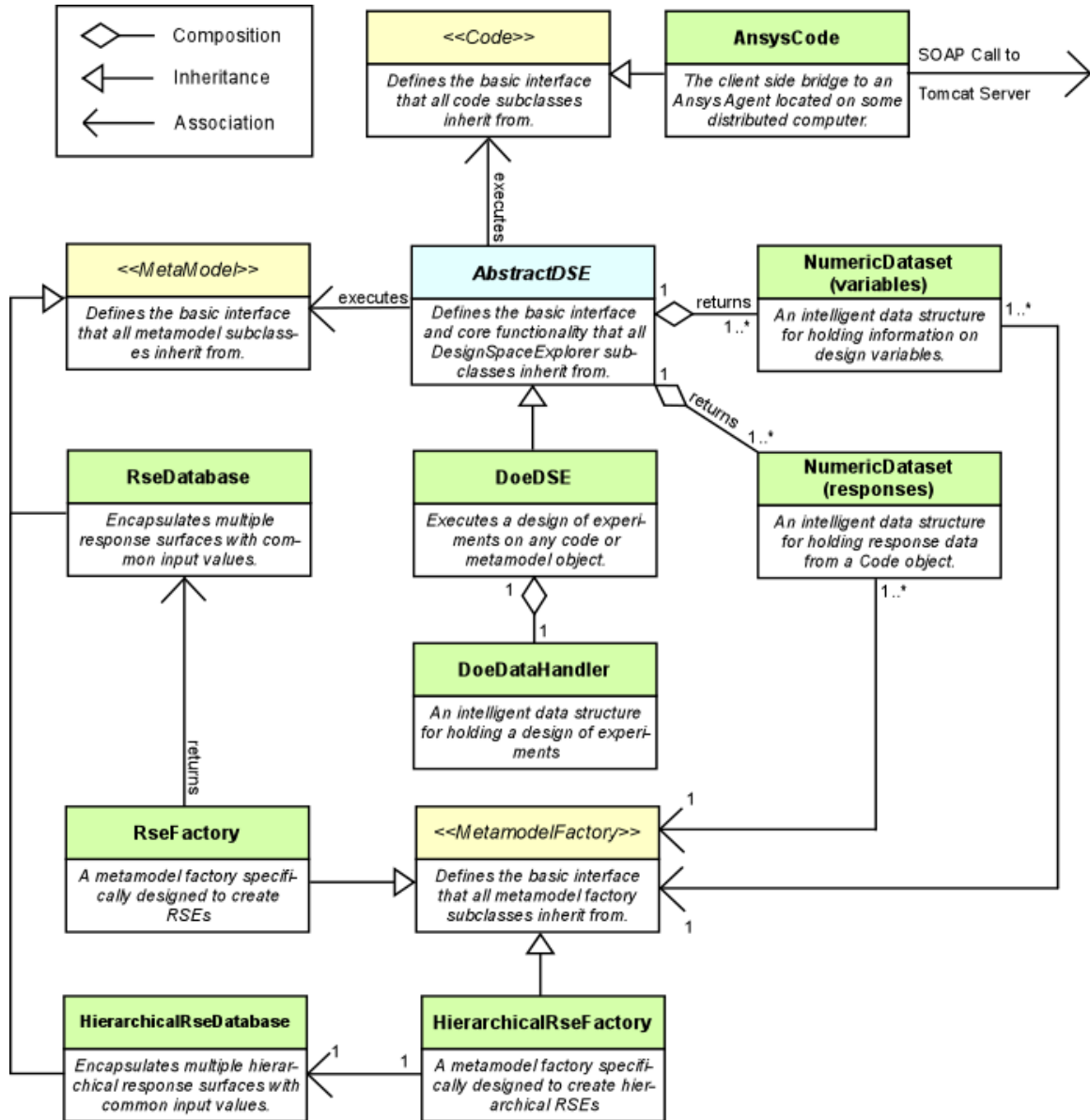


Figure 25: Client Side UML Diagram

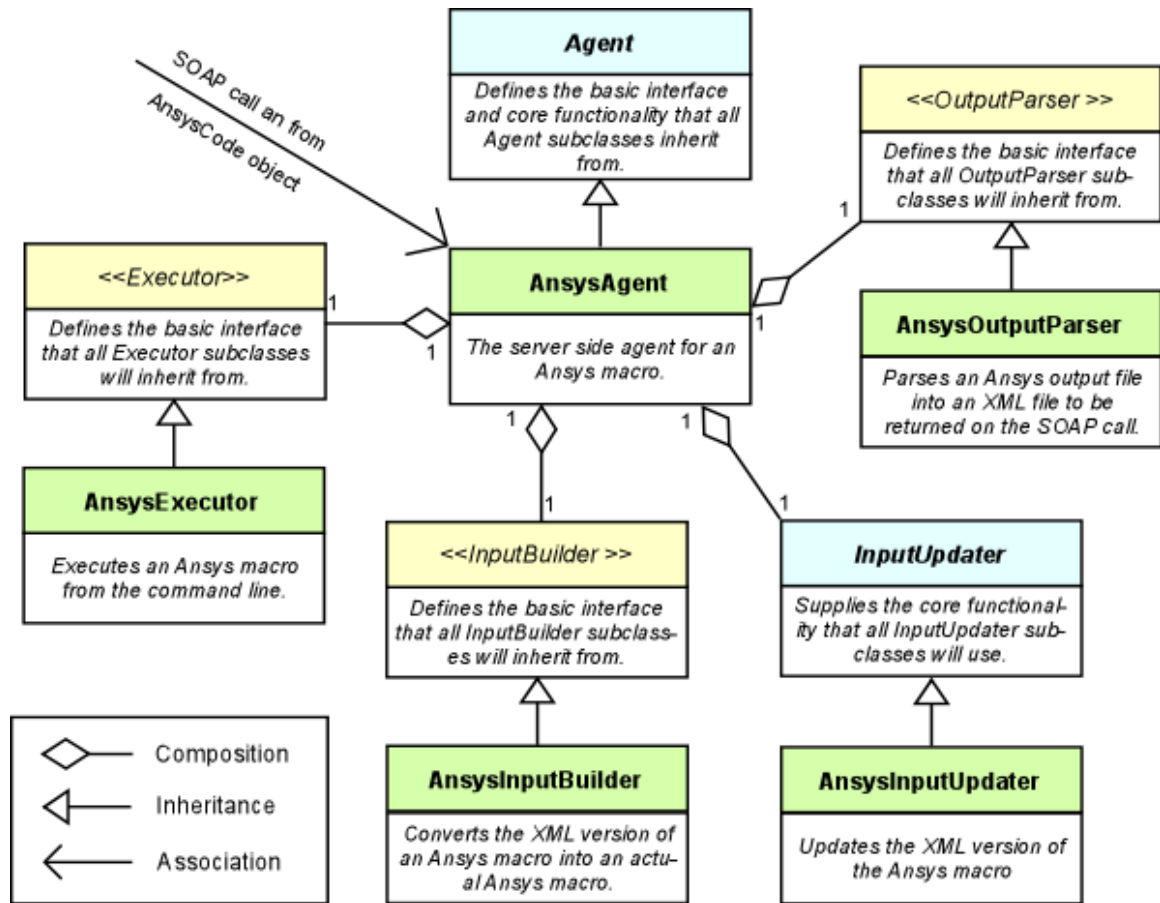


Figure 26: Server Side UML Diagram

(i.e. Genetic Algorithm and Sequential Quadratic Programming optimizer classes) and a one shot code executor class that performs a single execution of a code (this is handy for debugging a code class and its distributed agent counterpart). And as was stated earlier, all these classes inherit from a similar parent class, namely the AbstractDSE class.

The reason for this inheritance hierarchy is due to the nature of how one would perform many of the operations encapsulated in the design space explorer classes. Fundamentally all design space exploration methods revolve around the same three tasks:

1. Selection of settings for the design variables.
2. Generation of an XML input file for a Code object (based on the selected design variable values).
3. Collection (and manipulation if necessary) of the resultant responses from the execution of the Code object.

For instance, the selection of the design variable settings for a design of experiments is done a priori and stored in an array (or in this case in a DoeDataHandler object), and each run of the DoE represents a new unique setting to execute the Code object with. A Monte Carlo simulation and an optimization process on the other hand both generate the settings of the design variables on the fly as they are interacting with the code object. Regardless, all of these design space exploration techniques still need to generate design variable settings, create XML Code object input files and collect/interpret the resultant response data.

Based on this abstraction of the core functionality, one can quickly see two primary benefits of this approach:

- Any error correction due to bugs and improvements to this code only need to be

done in one place and will automatically propagate to the inheriting subclasses (after recompilation that is).

- Any future class that codes to the interface of the AbstractDSE parent class can substitute one DSE object for another without changing any code, this promotes flexibility in the API.

In addition to supplying its subclasses with core functionality, the AbstractDSE class also defines several overloaded, abstract methods¹ that every class that inherits from the AbstractDSE class must implement. These abstract methods are the methods that define how each DSE subclass creates its next setting of design variables. Since each of these execution methods will most likely be different for each type of DSE subclass, requiring that each inheriting DSE subclass provide its own underlying implementation is all that can be initially defined.

As was stated before, the Code and Metamodel classes are simpler in nature and designed to work in conjunction with the execution of the design space explorer classes. The concrete Code subclasses (i.e. the AnsysCode class in Figure 25) act as the front end bridge between the design space explorer objects residing on the client computer and the Agent objects residing on the server. To the user, a Code object is merely a local implementation of their favorite code to be used in conjunction with any DSE object. They need not have any idea as to how or where the actual code is implemented.

7.4.2 The Server Side API

The primary functionality of the client side API was centered around the design space explorer classes. On the server side, the only classes that are exposed by the Tomcat sever are the Agents. Thus designing an agent interface and class structure that can

¹It is these method definitions, lacking any defined or underlying implementation, that make the AbstractDSE class an abstract class.

be extended to wrap any engineering code is particularly important.

Creating an extensible class structure that can be wrapped around any command line executable engineering code really comes down to two primary tasks:

- Determination of the common tasks that all agents will need to perform to execute an engineering code
- Creation of a collection of classes that minimizes the amount of new code that needs to be generated for each new agent to be added

Identification of the common tasks that all agents will need to perform, again can be broken down into several subparts. This list of functionality defines much of the basic internal structure that all agents will be built upon. Figure 26 visually echoes this list.

1. Update the agent's internal "XML'ized"² engineering code input file via an InputUpdater type class.
2. Convert the updated, "XML'ized" input file into a native code input file via an InputBuilder type class.
3. Execute the engineering code with the newly created, native input file via an Executor type class.
4. Extract the desired data from the code output text file and summarize the collected data into an XML file to be returned on the SOAP call via an OutputParser type class.

Since each agent will have to perform these basic tasks in one form or another and each task is wholly different in nature, it is advantageous to separate each task into

²An "XML'ized" input file, refers to an engineering code input file that has been manually converted into an XML file, with XML tags surrounding and identifying the input variables that are to be updated.

its own class, thus using composition to build these functionalities into the agent. Using composition in this way is a direct adherence to heuristic two, defined at the beginning of this section.

Now that the basic functionality and structure of the agent has been laid out, agent design task two comes into play - minimize the amount of new code that needs to be generated to create a new agent class. This design task is not so much related to the class structure of the agent, but rather at the actual underlying implementation of the agent and how that implementation can be extended to work for all agents.

At this point it is important to discuss the common paradigm used to update and/or create a text input file for an engineering code and how this new agent structure deviates from that paradigm. Typically one of two approaches are used, either the programmer parses the input file into an in-memory data structure that can be updated and then written out as an updated input file or a complicated text file parsing and updating scheme is created that can be applied to any input text file (this is the approach that ModelCenter uses). Approach one inherently violates agent design task two as a new parsing and data structure would need to be created for every type of agent subclass. Approach two, on the other hand, is much more in-line with agent design task two because it can be applied to just about any engineering text input file. However, this still requires that the engineer create a fully robust text file parsing and updating scheme. This is the approach that the author initially examined but abandoned due to implementation difficulties.

An alternative approach to the text file parsing and updating scheme was alluded to earlier, namely create an XML'ized input file and transform that file into a native text input file. Initially, this new approach would actually seem to be more difficult because now two classes are needed rather than one. One class for the updating of the XML'ized input (an InputUpdater type class) and another class to convert the XML'ized input file into the native input file (an InputBuilder type class). This

however does not actually turn out to be the case. As it turns out, each class in and of themselves are relatively easy to create. The primary reason for this is due to the use of the XML data structure employed for the new input file.

Creating a robust text file parser is not a particularly easy task, however by using XML as the base data structure for the XML'ized input file, the framework can use readily available, professionally developed XML parsers and manipulation APIs to update and convert the XML'ized input file into a native text input file. This is the primary reason that XML was chosen as the native data format for the entire framework architecture.

An additional benefit to using XML as the base data structure is related to the type of information that can be included in an XML file. An XML file contains not only the desired XML'ized input file information but it can also include metadata on that information. This metadata can take the form of information on converting the XML file to the native text file, it can also be used to embed information into the file to add intelligence to the Agent. Traditionally metadata for a native text input file would take the form of comments and the like (that is if the engineering code supports comments in its input files), however this XML metadata can be much more in depth and allow the developer/user to categorize and use it in any conceivable fashion. Plus, the beauty of this is that the XML parser and InputUpdater class will totally ignore any additional metadata that it does not know what to do with. Thus, the metadata can just be for the user or creator of the XML'ized input file.

Earlier, it was stated that two new classes would need to be created to facilitate this new input file updating and creation paradigm. However, due to the rather generic nature of the new XML'ized input file (which can be applied to any text file based engineering code input file) the InputUpdater class can be used to update any XML'ized input file. Thus, the user will only need to create the InputBuilder class for each agent, but even this class can be abstracted to work for any Agent of a particular

type (i.e. ANSYS, ASTROS, etc.).

7.5 *Summary*

An overview was given of the different alternatives to an integrating computational framework. The seven conceptual elements of a good MDO environment were identified and it was illustrated that these elements can be captured with a design tool made of open-source elements.

In this chapter a distributed computing framework was introduced which incorporates the latest computational techniques and more importantly, a mind-set emphasizing flexibility, modularity, portability, and re-usability. With the described object-oriented tools, such a framework can now truly be built for the first time by non-computer scientists.

The chapter illustrates that usage of open-source tools are a valid alternative to commercial packages. Added advantages are the access to source code which is extremely useful when developing new design space exploration processes such as the ones developed for this thesis. It is important to stress however that for out-of-the-box performance the commercial applications can not be beat. They perform very well and for most problems are the optimal solution.

Growth potential of the open-source framework was allowed for from the outset. Future work will include adding new design space explorer functionality, focusing more on the optimization and statistical capabilities of the framework. These improvements will add significant computing and analytical force and make the comparison with the commercial alternatives much more competitive.

The framework laid out in this chapter will form the core of the tool set used to collect and analyze the data for the validation of the hierarchical design space exploration process. Unfortunately, not all of the code that was developed for the new framework can be included in the thesis, doing so would add roughly 200 more

pages to the appendices.

Chapter VIII

VALIDATING THE HIERARCHICAL DESIGN SPACE EXPLORATION PROCESS

This chapter applies the hierarchical design space exploration (Hi-DSE) process developed in Section 6.3 to the two test cases discussed in Section 6.4.3. For each test case a thorough description of the model used with the Hi-DSE process will be given, followed by a thorough review of actually applying the Hi-DSE process to the model and the results of the validation process.

8.1 Test Case 1

8.1.1 The Hollow I-beam Test Model

The hollow I-beam test model is a parametric, finite element model that is a stand-in for the actual nozzle model. Its primary development goal was to facilitate testing and development of the lumped parameter modeling (LPM) and graph-based partitioning (GBP) methods.

The hollow I-beam is spatially partitioned in a similar fashion as the nozzle model and has all of the same characteristic variables as the nozzle model. The model is partitioned into six spatial zones. In each spatial zone, the wall thicknesses, internal and external temperatures, and the internal and external heat transfer coefficients are allowed to vary. Thus the model has 30 independent variables. The responses are also collected in a zonal fashion. The maximum temperature and stress from each zone is collected, giving the model 12 independent responses. The total run time for

the hollow I-beam model is approximately 1.5 minutes on a vintage 2006 computer.

Section 6.4.3 stated that finite element models can exhibit numerical noise simply by modifying the finite element mesh applied to model. The fully parametric formulation of the model helps to alleviate this problem. A parametric definition enables the use of a predefined, mapped mesh. A mapped mesh is a finite element grid that is uniquely defined based on the geometric definition of the model. Thus as the geometry of the model changes (e.g. by changing the wall thickness of a zone), the essential layout of the grid does not change. Granted the size and spacing of the elements and nodes do change but their relative relationship to one another and to the model does not. A mapped mesh is in contrast to a free-form mesh, which are typically generated automatically by the finite element code and will change from one run to the next. The numerical noise associated with variations in the finite element grid discussed in Section 6.4.3 is primarily associated with free-form meshes.

Figure 27 is a rendering of the hollow I-beam test model with the various geographic zones labeled. The complete ANSYS APDL [5] macro for creating and analyzing the model can be found in Appendix C.

8.1.2 Applying the Hi-DSE Process to the I-beam Problem

This section will outline the main points of applying the hierarchical design space exploration process to the hollow I-beam test case. Not all of the steps of the process will be explicitly outlined, only those that are either significant deviations from standard design and analysis of experiments or are important for independent recreation of the validation process.

The first step in any design space exploration process is to define the desired variables and responses to investigate and the ranges over which those variables will be varying. Table 4 summarizes the I-beam input variables and ranges. Each of the sub-analyses all use these basic definitions.

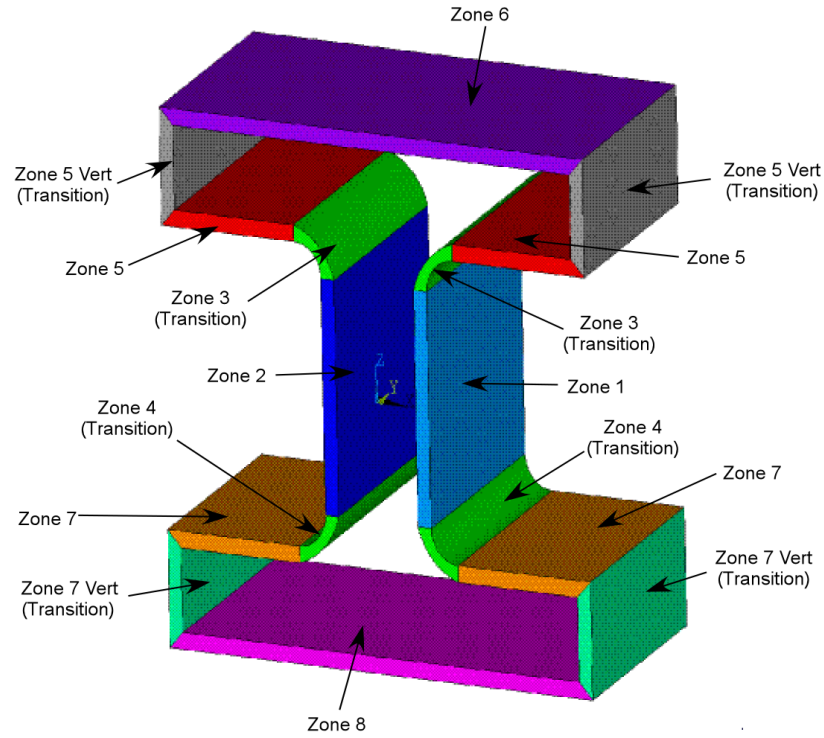


Figure 27: The Hollow I-beam Test Model

Table 4: I-beam Variable Definitions and Ranges

Name	Ansys Name	Range	
		Minimum	Maximum
External Temp Zone 1	TEXT1	1350	1650
External Temp Zone 2	TEXT2	1350	1650
External Temp Zone 5	TEXT5	1350	1650
External Temp Zone 6	TEXT6	1350	1650
External Temp Zone 7	TEXT7	1350	1650
External Temp Zone 8	TEXT8	1350	1650
External Heat Trans Coeff Zone 1	HEXT1	700	900
External Heat Trans Coeff Zone 2	HEXT2	700	900
External Heat Trans Coeff Zone 5	HEXT5	700	900
External Heat Trans Coeff Zone 6	HEXT6	700	900
External Heat Trans Coeff Zone 7	HEXT7	700	900
External Heat Trans Coeff Zone 8	HEXT8	700	900
Internal Temp Zone 1	TINT1	800	1000
Internal Temp Zone 2	TINT2	800	1000
Internal Temp Zone 5	TINT5	800	1000
Internal Temp Zone 6	TINT6	800	1000
Internal Temp Zone 7	TINT7	800	1000
Internal Temp Zone 8	TINT8	800	1000
Internal Heat Trans Coeff Zone 1	HINT1	700	900
Internal Heat Trans Coeff Zone 2	HINT2	700	900
Internal Heat Trans Coeff Zone 5	HINT5	700	900
Internal Heat Trans Coeff Zone 6	HINT6	700	900
Internal Heat Trans Coeff Zone 7	HINT7	700	900
Internal Heat Trans Coeff Zone 8	HINT8	700	900
Wall Thickness Zone 1	TH1	0.1875	0.3125
Wall Thickness Zone 2	TH2	0.1875	0.3125
Wall Thickness Zone 5	TH5	0.1875	0.3125
Wall Thickness Zone 6	TH6	0.1875	0.3125
Wall Thickness Zone 7	TH7	0.1875	0.3125
Wall Thickness Zone 8	TH8	0.1875	0.3125

The next step in the Hi-DSE process is to create the design of experiments to use as the initial screening test to determine the system partitions. For the I-beam model, the screening test was a hybridized design consisting of a resolution IV fractional factorial and an “optimal” latin hypercube of the same size. The fractional factorial design contained 30 factors and 63 runs. The design was created using Design Expert 7.0 [158]. The optimal latin hypercube was “optimal” in the sense that the design was created to minimize the pair-wise correlations between the columns. Matlab 7.0.1’s “lhsdesign” function was used to create the design. It too has 30 factors and 63 runs. Thus the hybridized screening design contained a total of 126 runs (the first 63 were the fractional factorial and the last 63 the optimal latin hypercube).

The data collected from the screening test was used to determine the sparsity of the I-beam system. As described in the Hi-DSE process in Section 6.3, an ANOVA analysis was used with a main effects only linear model to determine the significant effects for each response. The non-partitioned graph of the system can be seen in Figure 28.

The picture on the right hand side of Figure 28 is a sparse matrix plot of the system. A sparse matrix plot is a square plot where the vertices of a graph are enumerated along both the rows and the columns. Each grey box represents a connection or edge between two vertices. For instance, row one has a grey box in the 19th column, thus there is an edge connecting vertex 1 and vertex 19 in the graph on the left hand side of Figure 28. The graph and sparse matrix plot contain exactly the same data but provide two different ways to view it. The usefulness of the sparse matrix plot will be shown when the partitions of the system are created.

What is interesting about the graph in Figure 28 is the four non-connected vertices shown in the upper right hand corner of the graph. Those four vertices are variables that do not have a significant impact on any of the 12 responses of the system. Those variables are: external heat transfer coefficient for zone 1 (vertex #7), external heat

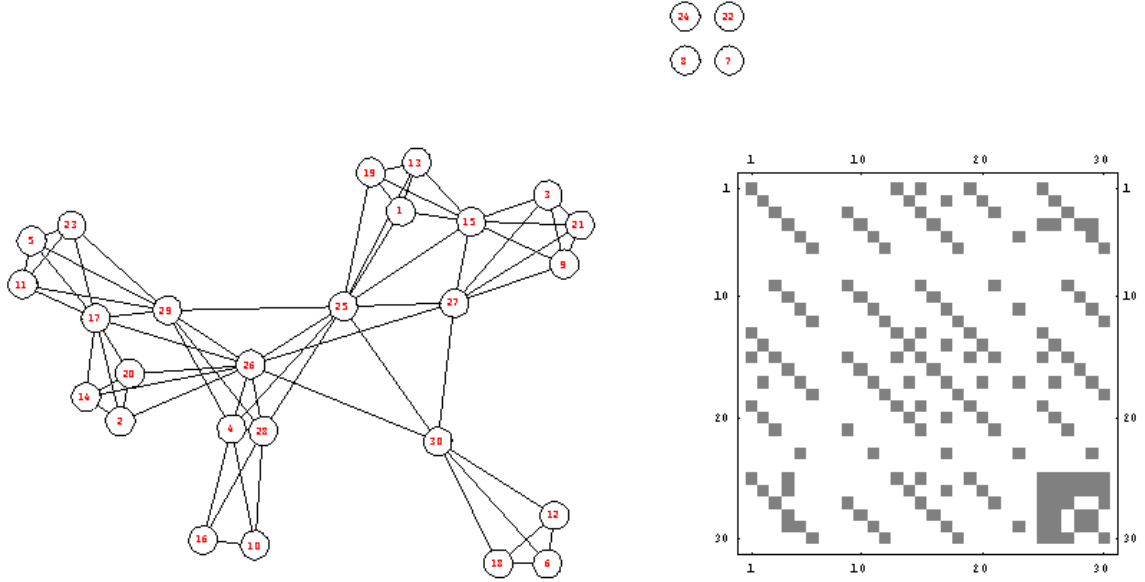


Figure 28: I-beam System Graph and Matrix Plot

transfer coefficient for zone 2 (vertex #8), internal heat transfer coefficient for zone 6 (vertex #6) and internal heat transfer coefficient for zone 8 (vertex #24).

Typically in a screening analysis, any variables that are not significant across all responses are left out of design of experiments for the subsequent data collection and model building process. Thus the initial screening data must be discarded. Alternatively, if all of the variables were found to be significant, then the initial screening data could simply be augmented with new data to create the full model. This would be 100% data utilization.

Recall that one of the primary goals of the development of the Hi-DSE process was to create models as efficiently as possible, thus any data that is collected should be used to create the lumped parameter model. Consequently all the variables will need to be assigned to partitions in the subsequent analyses. The existence of insignificant variables possess some problems for the min-cut algorithms used to partition the system graph. Because the insignificant variables do not contain any connections to any other variables, they are not part of any sub-graphs of the system. Thus, the

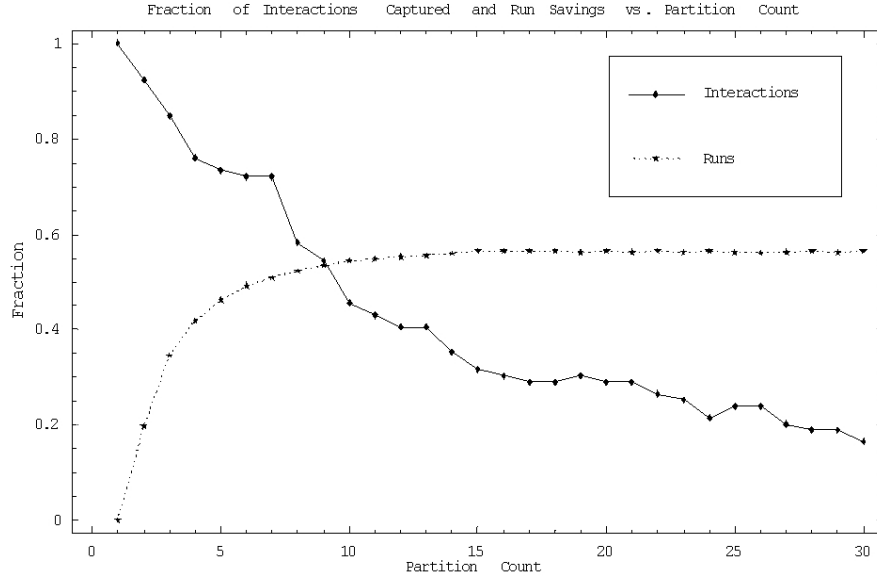


Figure 29: Interactions Missed and Runs Saved per Partition Count

min-cut algorithms will randomly assign unconnected variables to a sub-graph. When this occurs, it is in the best interest of the model builder to “intelligently” re-assign an unconnected variable to an appropriate subgroup. *Note - a sub-graph is synonymous with partition or variable subgroup.*

With the sparse nature of the system determined, the system partitioning process can occur. Since it only takes a second to create a potential set of partitions, essentially all of the partitions can be created. The question becomes, which set of partitions should be chosen? Figure 29 is a plot of projected runs saved and potential interactions captured versus partition count for the I-beam system.

In Figure 29, the projected runs saved is estimated by comparing the current number of runs from the screening test plus the additional runs required for a set of partitions to the projected total number of runs for the complete system. For a given number of variables, the runs required can be roughly estimated using an estimate of the runs required of a quadratic model. In Figure 29, the fraction of runs saved for a given number of partitions is estimated with Equation 28.

$$f = 1 - \frac{N_{screening} + \sum_{i=1}^m \frac{(n_i+1)(n_i+2)}{2}}{\frac{(n_{total}+1)(n_{total}+2)}{2}} \quad (28)$$

where,

- f = fraction of projected runs saved for a given partition set
- $N_{screening}$ = current number of runs for the screening DoE
- n_i = variable count for subgroup, i
- m = the number of subgroups in a set of partitions
- n_{total} = total number of variables for the system

In Figure 29, the fraction of interactions captured is estimated by Equation 29. The number of “potential missed interaction” is essentially the number of edges cut by the min-cut algorithm.

$$f = \frac{I_{total} - I_{missed}}{I_{total}} \quad (29)$$

where,

- f = fraction of interactions captured for a given set of partitions
- I_{total} = total number of potential interactions in the system
- I_{missed} = number of potential interactions missed due to partitioning

Using Figure 29 as a guide, after seven subgroups of variables, the fraction of interactions captured starts to drop off quickly and the potential run savings does not increase much. Based on Baker’s analysis of the impact of missed interactions on the accuracy of the UTE method¹, it is logical to assume that the more interactions that are directly neglected, the more the accuracy of the model will be negatively impacted. Thus it seems from Figure 29 that seven is the maximum number to consider. Now the question is, which of these to choose?

¹See Sections 3.4 and 3.5

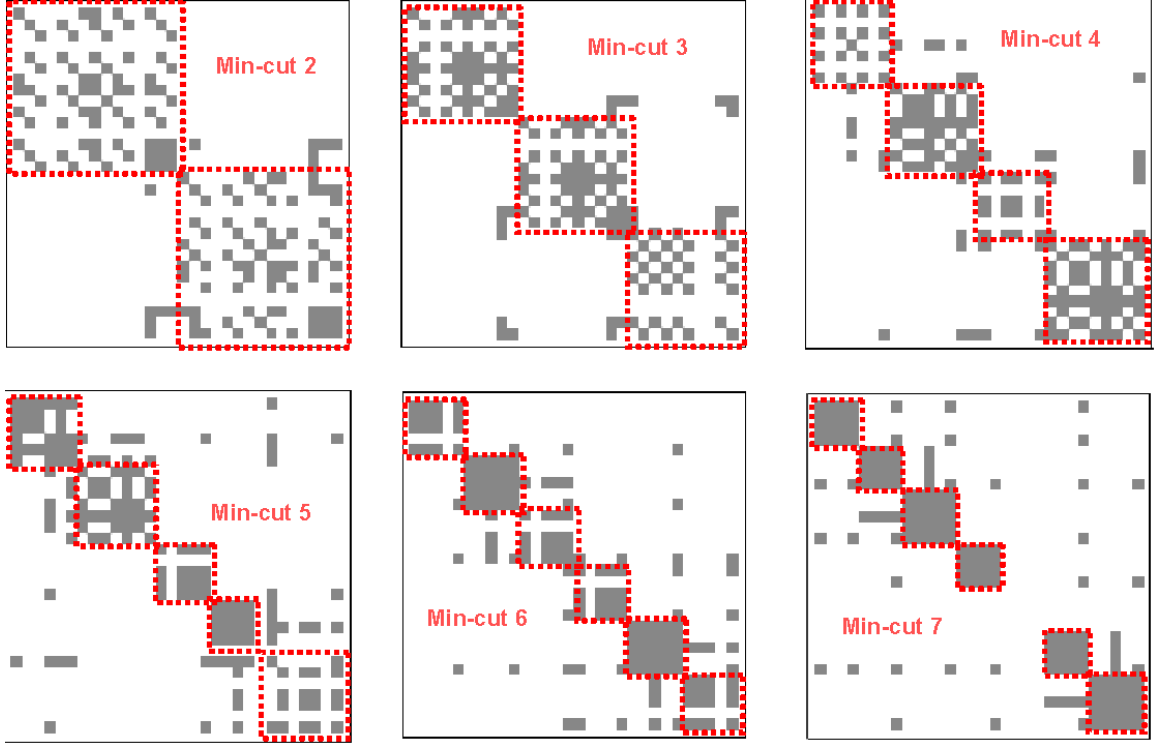


Figure 30: Partitioned Sparse Matrix Plots

This is where engineering intuition and considerations of the total computational budget for the analysis come into play. Logically, fewer numbers of subgroups equates to more directly captured interactions but also implies a larger computational expenditure. To aid in the decision making process, Figure 30 shows the matrix plots of the I-beam system for the first six partition sets. From the upper left to lower right, the number of subgroups per partition set is two, three, four, five, six and seven subgroups. In Figure 30, any grey square that is not included in one of the block diagonal structures is a potential interaction that will not be captured by the proposed partition set.

Recall in Section 6.4.2, one of the proposed means of validating the graph-based partitioning method was to compare the accuracy of the partitions determined via the GBP method with partitions based on engineering intuition. The two engineering intuition partitioning schemes were variable-type partitions and subsystem type

partitioning. As it turns out, the min-cut six and min-cut seven partitions are almost exactly the same as the subsystem type partitioning scheme created using engineering intuition!

There are a couple of minor differences between the subsystem partitions and the min-cut partitions but the differences are primarily due to how the min-cut algorithm assigned the unconnected variables to subgroups. Using a little logic and re-assigning the unconnected variables to more appropriate subgroups, the mincut-6 partitions produced very close to the subsystem partitioning scheme. The mincut-7 partitioning scheme also produced a subsystem partitions, but only amongst the connected variables, the unconnected variables were actually grouped together in their own separate subgroup. For a complete summary of the min-cut partitioning results see Appendix A. The summary in Appendix A includes the variables assigned to each partition, the partitioned graphs as well as the sparse matrix array for each set of partitions. The variables per partition in Appendix A are the exact same grouping that the author used to assess the accuracy of the min-cut partitions.

Having collected the initial screening data and determined the sets of partitions to use for the validation process, the individual subgroup analyses can now be performed. Each subgroup of variables identified by the graph-based partitioning were evaluated using an appropriately sized minimum resolution V CCD² from Design Expert 7.0. While one subgroup was being analyzed the variables assigned to the other groups were held constant at their nominal values. After the data for each of the subgroups was collected, Design Expert 7.0 was then used to create standard quadratic response surface equations (RSE). These RSEs represent the $\tilde{y}_i = g_i(\mathbf{x}_i)$ functions discussed in Section 6.1.

With the $\tilde{y}_i = g_i(\mathbf{x}_i)$ functions created for each subgroup, the \tilde{y} regression arrays

²CCD means central composite design. See Myers (2002) [109] for a good discussion on central composite designs.

Table 5: Variable-type and Subsystem Partitions for I-beam Model

Variable-Type Partitioning Scheme Variables	Subsystem Partitioning Scheme Variables
External Temp Zone 1	External Temp Zone 1
External Temp Zone 2	External Heat Trans Coeff Zone 1
External Temp Zone 5	Internal Temp Zone 1
External Temp Zone 6	Internal Heat Trans Coeff Zone 1
External Temp Zone 7	Wall Thickness Zone 1
External Temp Zone 8	External Temp Zone 2
External Heat Trans Coeff Zone 1	External Heat Trans Coeff Zone 2
External Heat Trans Coeff Zone 2	Internal Temp Zone 2
External Heat Trans Coeff Zone 5	Internal Heat Trans Coeff Zone 2
External Heat Trans Coeff Zone 6	Wall Thickness Zone 2
External Heat Trans Coeff Zone 7	External Temp Zone 5
External Heat Trans Coeff Zone 8	External Heat Trans Coeff Zone 5
Internal Temp Zone 1	Internal Temp Zone 5
Internal Temp Zone 2	Internal Heat Trans Coeff Zone 5
Internal Temp Zone 5	Wall Thickness Zone 5
Internal Temp Zone 6	External Temp Zone 6
Internal Temp Zone 7	External Heat Trans Coeff Zone 6
Internal Temp Zone 8	Internal Temp Zone 6
Internal Heat Trans Coeff Zone 1	Internal Heat Trans Coeff Zone 6
Internal Heat Trans Coeff Zone 2	Wall Thickness Zone 6
Internal Heat Trans Coeff Zone 5	External Temp Zone 7
Internal Heat Trans Coeff Zone 6	External Heat Trans Coeff Zone 7
Internal Heat Trans Coeff Zone 7	Internal Temp Zone 7
Internal Heat Trans Coeff Zone 8	Internal Heat Trans Coeff Zone 7
Wall Thickness Zone 1	Wall Thickness Zone 7
Wall Thickness Zone 2	External Temp Zone 8
Wall Thickness Zone 5	External Heat Trans Coeff Zone 8
Wall Thickness Zone 6	Internal Temp Zone 8
Wall Thickness Zone 7	Internal Heat Trans Coeff Zone 8
Wall Thickness Zone 8	Wall Thickness Zone 8

can be created using the initial screening data. The resultant regression arrays were then analyzed in Design Expert 7.0 to create the $y = f(\tilde{y})$ functions for each of the twelve responses of the system.

As a comparison for validation purposes, the min-cut based lumped parameter models were compared to two full scale all-at-once analyses as well as the LPMs created using the variable-type and subsystem partitioning schemes. Table 5 shows the partitioning of each of the variables for the variable-type and subsystem partitioning methods.

The two full scale metamodels were created using a 527 run minimum run resolution V CCD from Design Expert 7.0 and a 527 run optimal latin hypercube from Matlab's "lhsdesign". These two full scale models were used for comparison because they represent models created using more traditional response surface methodology (RSM)[109]. In fact, the minimum run resolution V CCD model is considered to be

the baseline that all the other models will be compared to.

To keep the comparison of one method to another as objective as possible, the exact same model building process was used in every case. The metamodel creation process used Design Expert 7.0 as the analysis engine and all metamodels were created using a standard quadratic response surface equation. Each RSE was created without using Design Expert's stepwise regression, thus each metamodel is a full rank, quadratic RSE. Full rank meaning that every term in a standard quadratic RSE is estimated.

The actual numerical accuracy of the traditional RSEs and the LPM based metamodels was conducted using 5,000 random, off-design points from the actual I-beam model. For each of the 5,000 points, the predicted value from the metamodel was compared to the actual value from the I-beam test model and the relative difference between the two estimates was used to characterize the error.

$$\epsilon_i = \frac{A_i - P_i}{A_i} \quad (30)$$

where,

- ϵ_i = relative error for off-design point, i
- A_i = actual value for off-design point, i
- P_i = predicted value for off-design point, i

Using the relative error rather than the absolute error, $\epsilon_i = A_i - P_i$, provided a nice means to compare errors between metamodels but it also allowed for error comparisons to be conducted between responses for the same metamodel type because the relative error value is non-dimensional. In addition, the non-dimensional relative error also allowed for the accuracy of a given metamodel to be assessed for all the responses at one time. The response combined error or cumulative error measurement provided a clear means of deducing which partitioning scheme was the overall best. The reason

that this is important is that a set of partitions may be optimal for one response and not another, and in general we want a set of partitions that is “good” across all responses at the same time.

Table 6 is a complete summary of the cumulative error of the results for each of the metamodels compared in this assessment. The first or top sub-table is the total cumulative error summary for across all twelve responses at the same time, sub-table two is the cumulative error for the stress responses and likewise sub-table three is the cumulative error for the temperature responses. For the complete error results, Appendix A summarizes results for each metamodeling method applied to the I-beam problem broken down by individual response. In addition, actual versus predicted charts and the histograms of the error for each response are shown for every metamodel type.

In Table 6 the primary means of comparing one method to another is via the “variance ratio” column. In every sub-table, the variance of one metamodeling method is referenced to the variance of the baseline method, the minimum run resolution V CCD analysis. The variance ratio is calculated as the ratio of the variance of the baseline to the variance of the alternative method. This means that a variance ratio greater than one implies that the baseline method has a wider distribution of error than the alternative method. For two error distributions with equals means, a wider error distribution is not as good as a narrower error distribution. The wide versus narrow implies that the wider distribution is more likely to have higher error rates for a random off-design point than the narrower distribution.

Examining the histograms of the error values in Appendix A for each response and metamodel type will show that each relative error distribution is essentially centered around zero, is symmetrical and roughly normally distributed. Consequently the 95% confidence bounds for the variance ratios of the error shown in Table 6 can be used to determine if one method is statistically more accurate than another [17]. Statistical

Table 6: I-beam Model Error Summary

Total Response Relative Error Comparisons:										
Response	Analysis Type	Runs Required	Runs Savings	% Error Mean	% Error Std. Dev.	% Error Variance	Variance Ratio	95% Confidence L.B.	95% Confidence U.B.	
All Responses	All-at-once Min Run Res. V CCD	527	0%	0.078%	1.271%	1.616E-04	1	n/a	n/a	
	All-at-once Optimal Latin Hypercube	527	0%	0.017%	1.777%	3.158E-04	0.512	0.503	0.520	
	Variable-Type Partitioning Scheme	301	43%	-0.085%	1.041%	1.085E-04	1.490	1.465	1.515	
	Subsystem Partitioning Scheme	289	45%	-0.011%	1.239%	1.535E-04	1.053	1.035	1.070	
	Mincut 2 Partitioning Scheme	432	18%	-0.057%	1.022%	1.044E-04	1.547	1.522	1.573	
	Mincut 3 Partitioning Scheme	357	32%	-0.080%	0.919%	8.441E-05	1.914	1.883	1.946	
	Mincut 4 Partitioning Scheme	326	38%	-0.034%	0.986%	9.714E-05	1.663	1.636	1.691	
	Mincut 5 Partitioning Scheme	303	43%	0.001%	1.114%	1.241E-04	1.302	1.280	1.323	
	Mincut 6 Partitioning Scheme	288	45%	-0.022%	1.093%	1.195E-04	1.352	1.330	1.375	
	Mincut 7 Partitioning Scheme	305	42%	-0.011%	1.291%	1.666E-04	0.970	0.954	0.986	
Total Stress Response Relative Error Comparisons:										
Response	Analysis Type	Runs Required	Runs Savings	% Error Mean	% Error Std. Dev.	% Error Variance	Variance Ratio	95% Confidence L.B.	95% Confidence U.B.	
Stress Responses	All-at-once Min Run Res. V CCD	527	0%	0.154%	1.793%	3.216E-04	1	n/a	n/a	
	All-at-once Optimal Latin Hypercube	527	0%	0.035%	2.512%	6.310E-04	0.510	0.498	0.522	
	Variable-Type Partitioning Scheme	301	43%	-0.170%	1.468%	2.154E-04	1.493	1.459	1.529	
	Subsystem Partitioning Scheme	289	45%	-0.023%	1.751%	3.068E-04	1.048	1.024	1.073	
	Mincut 2 Partitioning Scheme	432	18%	-0.116%	1.442%	2.078E-04	1.547	1.512	1.584	
	Mincut 3 Partitioning Scheme	357	32%	-0.160%	1.294%	1.673E-04	1.922	1.878	1.968	
	Mincut 4 Partitioning Scheme	326	38%	-0.074%	1.389%	1.929E-04	1.668	1.629	1.707	
	Mincut 5 Partitioning Scheme	303	43%	-0.001%	1.570%	2.465E-04	1.305	1.275	1.336	
	Mincut 6 Partitioning Scheme	288	45%	-0.045%	1.544%	2.384E-04	1.349	1.318	1.381	
	Mincut 7 Partitioning Scheme	305	42%	-0.025%	1.824%	3.328E-04	0.966	0.944	0.989	
Total Temp. Response Relative Error Comparisons:										
Response	Analysis Type	Runs Required	Runs Savings	% Error Mean	% Error Std. Dev.	% Error Variance	Variance Ratio	95% Confidence L.B.	95% Confidence U.B.	
Temp. Responses	All-at-once Min Run Res. V CCD	527	0%	0.002%	0.058%	3.394E-07	1	n/a	n/a	
	All-at-once Optimal Latin Hypercube	527	0%	-0.001%	0.076%	5.840E-07	0.581	0.568	0.595	
	Variable-Type Partitioning Scheme	301	43%	0.001%	0.025%	6.383E-08	5.317	5.194	5.443	
	Subsystem Partitioning Scheme	289	45%	0.001%	0.047%	2.230E-07	1.522	1.487	1.558	
	Mincut 2 Partitioning Scheme	432	18%	0.002%	0.054%	2.881E-07	1.178	1.151	1.206	
	Mincut 3 Partitioning Scheme	357	32%	-0.001%	0.047%	2.192E-07	1.548	1.513	1.585	
	Mincut 4 Partitioning Scheme	326	38%	0.006%	0.105%	1.093E-06	0.310	0.303	0.318	
	Mincut 5 Partitioning Scheme	303	43%	0.003%	0.133%	1.762E-06	0.193	0.188	0.197	
	Mincut 6 Partitioning Scheme	288	45%	0.002%	0.063%	4.020E-07	0.844	0.825	0.864	
	Mincut 7 Partitioning Scheme	305	42%	0.003%	0.060%	3.613E-07	0.939	0.918	0.961	

difference implies that the confidence intervals do not overlap, if they do then one can not say that the error rates are statistically different [59]. Note however, the very large data set that was used to create the error data results in considerable statistical power for the variance ratio test. With enough data, two essentially equal distributions can be shown to be statistically different, consequently one needs to compare the confidence intervals on the variance ratios as being “essentially different” and not necessarily statistically different [59].

8.2 Test Case 2

8.2.1 The Torsional Vibration System Test Problem

A closed form model based on a torsional vibration system (TVS) is also proposed for the validation process. The TVS model is a doubly cantilevered system comprised of two disks and three shafts. The two disks are made of steel and aluminum and the shafts are made of steel, titanium and aluminum. The TVS model has 18 variables and 3 responses (high and low natural frequency and system weight) and the entire behavior of the system can be captured in a series of closed form equations. Figure 31 is a basic sketch of the TVS system.

The closed form nature of the TVS model actually provides a nice test case for assessing the graph-based partitioning method. Since all of the equations are known a priori, the exact sparse matrix representation of the system can easily be created. The exact sparse structure can then be compared against the sparse structure predicted from the graph-based partitioning analysis. This will provide further validation of the graph-based partitioning method.

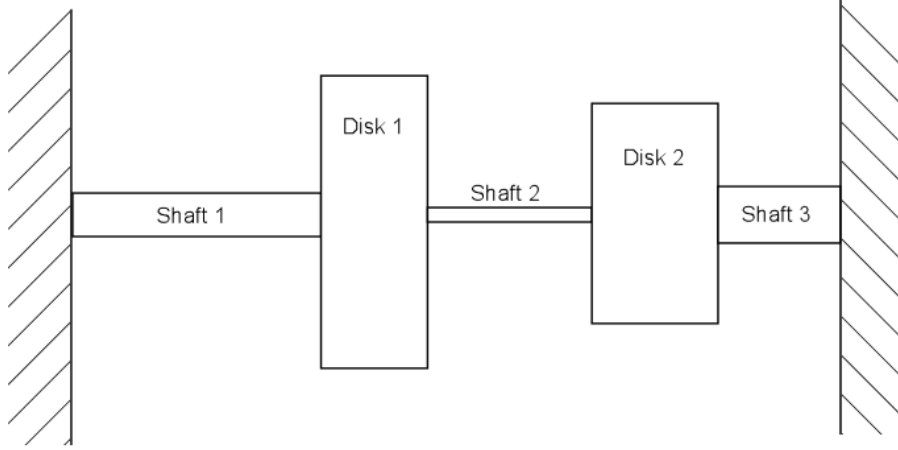


Figure 31: Torsional Vibration System Test Model

The following are all of the equations that define the torsional vibration system.

$$k_i = \frac{\pi G_i d_i}{32 l_i} \quad i = 1 \dots 3 \quad (31)$$

where,

$$k_i = \text{shaft } i \text{ spring constant, [lbs/in}^2\text{]}$$

$$G_i = \text{shaft } i \text{ modulus of rigidity, [lbs/in}^2\text{]}$$

$$d_i = \text{shaft } i \text{ diameter, [in]}$$

$$l_i = \text{shaft } i \text{ length, [in]}$$

$$M_j = \frac{\pi \rho_j t_j}{g} \left(\frac{D_j}{2} \right)^2 \quad j = 1 \dots 2 \quad (32)$$

where,

$$\begin{aligned}
M_j &= \text{disk } j \text{ mass , } [lbs \cdot sec^2/in] \text{ or } [slug] \\
\rho_j &= \text{disk } j \text{ density, } [lbs/in^3] \\
t_j &= \text{disk } j \text{ thickness, } [in] \\
D_j &= \text{disk } j \text{ diameter, } [in] \\
g &= \text{acceleration due to gravity, } [386.4 in/sec^2]
\end{aligned}$$

$$J_j = \frac{1}{2} M_j \left(\frac{D_j}{2} \right)^2 \quad j = 1 \dots 2 \quad (33)$$

where,

$$J_i = \text{disk } j \text{ mass moment of inertia, } [slug \cdot in^2]$$

$$\begin{aligned}
a &= 1 \\
b &= \frac{k_1 + k_2}{J_1} + \frac{k_2 + k_3}{J_2} \\
c &= \frac{k_1 k_2 + k_2 k_3 + k_3 k_1}{J_1 J_2}
\end{aligned} \quad (34)$$

where,

$$\begin{aligned}
a &= \text{intermediate constant} \\
b &= \text{intermediate variable, } [1/sec^2] \\
c &= \text{intermediate variable, } [1/sec^2]
\end{aligned}$$

$$\begin{aligned}
\omega_{low} &= \left(\frac{-b - \sqrt{b^2 + 4ac}}{2a} \right)^{1/2} \\
f_{low} &= \frac{2\pi}{\omega_{low}}
\end{aligned} \quad (35)$$

where,

ω_{low} = low frequency period, $[sec]$

f_{low} = low natural frequency, $[radians/sec]$

$$\begin{aligned}\omega_{high} &= \left(\frac{-b + \sqrt{b^2 + 4ac}}{2a} \right)^{1/2} \\ f_{high} &= \frac{2\pi}{\omega_{high}}\end{aligned}\tag{36}$$

where,

ω_{high} = high frequency period, $[sec]$

f_{high} = high natural frequency, $[radains/sec]$

$$W = \sum_{i=1}^3 \pi \lambda_i l_i \left(\frac{d_i}{2} \right)^2 + \sum_{j=1}^2 \pi \rho_j t_j \left(\frac{D_j}{2} \right)^2\tag{37}$$

where

W = total system weight, $[lbs]$

λ_i = shaft i density, $[lbs/in^3]$

8.2.2 Applying the Hi-DSE Process to the TVS Problem

This section will outline the main points of applying the hierarchical design space exploration process to the torsional vibration system (TVS) test case. Not all of the steps of the process will be explicitly outlined, only those that are either significant deviations from standard design and analysis of experiments or are important for independent recreation of the validation process.

As we will see from this validation process, the TVS model is not a good candidate for use with the Hi-DSE process. The primary reason that it was included was not

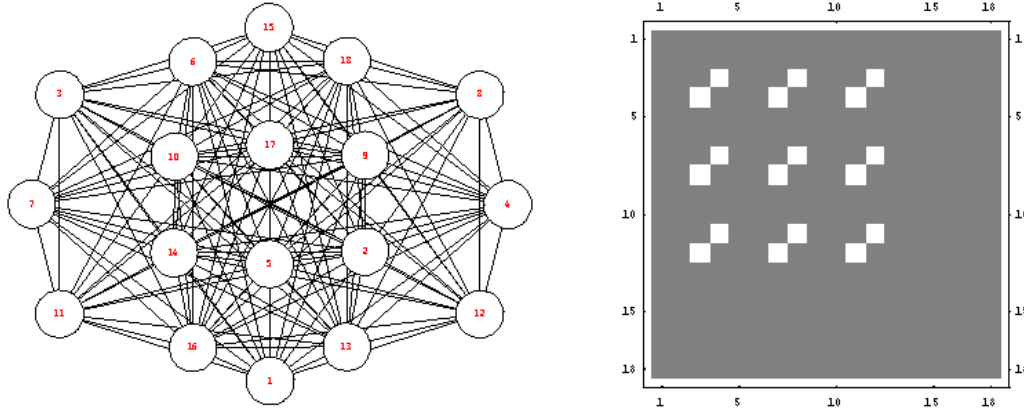


Figure 32: TVS Model True Graph and Matrix Plot

necessarily to show how good or bad of models the Hi-DSE method can create but rather to characterize a system that is not well suited for partitioning.

To show why this is true, Figure 32 is the graph and matrix plot of the system of equations for the TVS model summarized in Section 8.2.1. It is quite clear from the graph that the TVS model is not a good candidate for partitioning. Primarily because there are not any easily identifiable sub-graphs. In addition, the matrix plot is almost completely one solid block of grey which implies that every variable is connected to every other variable.

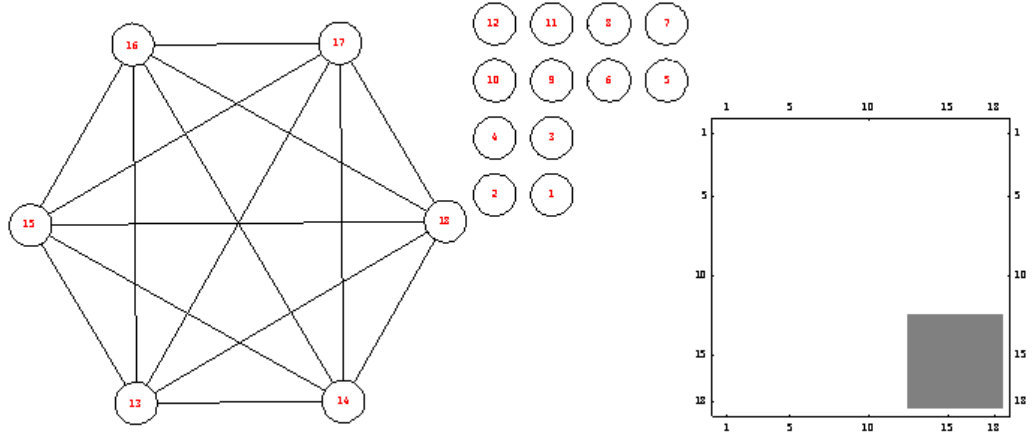
In light of this information, lumped parameter models can still be created for the system although we would expect the resultant metamodels to be of poor quality. To begin the Hi-DSE process with the TVS problem the initial screening data needs to be collected. Table 7 shows the variables and ranges that were used for the TVS model.

The initial screening design of experiments used to determine the sparsity of the TVS model was a resolution IV fractional factorial design from Design Expert 7.0 combined with an optimal latin hypercube. The design consisted of 18 factors and 78 runs, 39 runs for the fractional factorial and 39 for the latin hypercube portion.

Using the data collected from the initial screening test, an ANOVA analysis with a

Table 7: TVS Model Variable and Range Definitions

Name	Variable	Range	
		Minimum	Maximum
Shaft 1 Diameter	d_1	1.6	2.4
Shaft 1 Length	l_1	8	12
Shaft 1 Modulus of Rigidity	G_1	9.36E+06	1.46E+07
Shaft 1 Density (Steel)	λ _1	0.224	0.336
Shaft 2 Diameter	d_2	1.46	2.19
Shaft 2 Length	l_2	9.6	14.4
Shaft 2 Modulus of Rigidity	G_2	4.96E+06	7.44E+06
Shaft 2 Density (Titanium)	λ _2	0.128	0.192
Shaft 3 Diameter	d_3	1.8	2.7
Shaft 3 Length	l_3	6.4	9.6
Shaft 3 Modulus of Rigidity	G_3	3.12E+06	4.68E+06
Shaft 3 Density (Aluminum)	λ _3	0.08	0.12
Disk 1 Diameter	D_1	9.6	14.4
Disk 1 Thickness	t_1	2.4	3.6
Disk 1 Density (Steel)	ρ _1	0.224	0.336
Disk 2 Diameter	D_2	9.6	14.4
Disk 2 Thickness	t_2	2.4	3.6
Disk 2 Density (Steel)	ρ _2	0.08	0.12

**Figure 33:** TVS Model Experimentally Determined Graph and Matrix Plot

main effects only linear model was then used to determine the experimentally derived graph of the system. The results of the ANOVA analysis actually showed that of all the 18 variables, only the six variables associated with the disks had any significant impact on the variability of the the responses. Figure 33 is the graph and matrix plot of the TVS system based on the ANOVA analysis.

Although Figures 32 and 33 do not look exactly the same, they both do have one common characteristic, namely, the complete lack of any sub-graphs. The primary similarity is the tight interconnectedness that both graphs exhibit.

Table 8: TVS Model Variable-type and Subsystem Partitions

Subsystem Partitions	Variable-type Partitions
Shaft 1 Diameter	Shaft 1 Diameter
Shaft 1 Length	Shaft 2 Diameter
Shaft 1 Modulus of Rigidity	Shaft 3 Diameter
Shaft 1 Density (Steel)	Disk 1 Diameter
Shaft 2 Diameter	Disk 2 Diameter
Shaft 2 Length	Shaft 1 Length
Shaft 2 Modulus of Rigidity	Shaft 2 Length
Shaft 2 Density (Titanium)	Shaft 3 Length
Shaft 3 Diameter	Disk 1 Thickness
Shaft 3 Length	Disk 2 Thickness
Shaft 3 Modulus of Rigidity	Shaft 1 Density
Shaft 3 Density (Aluminum)	Shaft 2 Density
Disk 1 Diameter	Shaft 3 Density
Disk 1 Thickness	Disk 1 Density
Disk 1 Density (Steel)	Disk 2 Density
Disk 2 Diameter	Shaft 1 Modulus of Rigidity
Disk 2 Thickness	Shaft 2 Modulus of Rigidity
Disk 2 Density (Steel)	Shaft 3 Modulus of Rigidity

Clearly, by looking at the graph in Figure 33 one would expect the min-cut algorithms to have troubles creating any meaningful partitions. For one, there are only six interconnected variables. Secondly, the non-connected variables will be randomly assigned to a subgroup. Thus the graph-based partitioning portion of the Hi-DSE process is useless for this problem.

Earlier it was stated that lumped parameter models can still be created for the system even given its strong interconnectedness. However, the partitions used with the LPM process will be solely based on engineering intuition. To create lumped parameter models of the TVS problem, the same variable-type and subsystem partitioning schemes discussed for the I-beam problem will be used. Table 8 shows how the variables are partitioned for each of theses partitioning schemes.

The accuracy analysis method used for the TVS problem was exactly the same as the one that was used for the I-beam test case. 5,000 off-design points were created from the true model and the relative error of each of the metamodels was assessed. Table 9 is the complete error summary for the TVS model analysis. The actual versus predicted charts and the histograms of the error for each response are shown

in Appendix B for the TVS model analysis.

8.3 Discussion of the Test Case Results

This section will use the results of the applying the Hi-DSE process to the two test cases to qualify the last of the un-addressed research hypotheses. First, Section 8.3.1 will address the following research hypothesis:

- Using the GCF method as a prototype, it is hypothesized that the intermediate to top-level mapping will indirectly account for any neglected interactions between low-level design variables in disjoint groups.

Next, Section 8.3.2 will address the last two hypotheses. Those hypotheses are:

- It is hypothesized that a screening test using a low resolution DoE with ANOVA can be used to determine the underlying sparse structure of the black box system, which then can be partitioned using a graph-based min-cut algorithm.
- It is hypothesized that since the data from a screening test, which is to be used to tear the system into disjoint subgroups, inherently contains the connectivity information; this data can also be used to create the intermediate to top-level mappings which are designed to account for the impact of system tearing.

8.3.1 Validating Hypothesis 2.1

The primary motivation behind hypothesis 2.1 was to find a means to account for important interactions without resorting to variable redundancies between the partitions. The reason being that while variable redundancies improve the accuracy of the partitioned metamodel, they quickly negate any potential efficiency improvements that the method offers over traditional metamodeling approaches. Thus the lumped parameter model was developed to indirectly account for missing interactions via

Table 9: TVS Model Error Analysis Summary

Response	Analysis Type	Runs Required	Runs Savings	% Error Mean	% Error Std. Dev.	% Error Variance	Variance Ratio	95% Confidence	
								L.B.	U.B.
All Responses	All-at-once Min Run Res. V CCD	209	0%	0.117%	3.085%	9.518E-04	1	n/a	n/a
	All-at-once Optimal Latin Hypercube	209	0%	0.370%	3.358%	1.128E-03	0.844	0.817	0.871
	Variable-Type Partitioning Scheme	144	31%	0.353%	3.514%	1.235E-03	0.770957	0.747	0.796
	Subsystem Partitioning Scheme	153	27%	-0.310%	3.895%	1.517E-03	0.627	0.608	0.648

Response	Analysis Type	% Error Mean	% Error Std. Dev.	% Error Variance	Variance Ratio	95% Confidence	
						L.B.	U.B.
Natural Freq - low	All-at-once Min Run Res. V CCD	0.543%	4.491%	0.002017	1.000	n/a	n/a
	All-at-once Optimal Latin Hypercube	0.934%	4.600%	0.002116	0.953316	0.901896	1.007669
	Variable-Type Partitioning Scheme	-0.016%	4.220%	0.001781	1.132677	1.071581	1.197255
	Subsystem Partitioning Scheme	-0.269%	6.080%	0.003697	0.545585	0.516157	0.576691

Response	Analysis Type	% Error Mean	% Error Std. Dev.	% Error Variance	Variance Ratio	95% Confidence	
						L.B.	U.B.
Natural Freq - high	All-at-once Min Run Res. V CCD	-0.164%	2.781%	0.000773	1.000	n/a	n/a
	All-at-once Optimal Latin Hypercube	0.033%	3.450%	0.00119	0.649608	0.614569	0.686645
	Variable-Type Partitioning Scheme	0.778%	4.115%	0.001693	0.456582	0.431955	0.482614
	Subsystem Partitioning Scheme	-0.631%	2.860%	0.000818	0.945378	0.894386	0.999278

Response	Analysis Type	% Error Mean	% Error Std. Dev.	% Error Variance	Variance Ratio	95% Confidence	
						L.B.	U.B.
Weight	All-at-once Min Run Res. V CCD	-0.028%	0.612%	3.74E-05	1.000	n/a	n/a
	All-at-once Optimal Latin Hypercube	0.142%	0.545%	2.97E-05	1.258394	1.190517	1.33014
	Variable-Type Partitioning Scheme	0.299%	1.408%	0.000198	0.188832	0.178647	0.199598
	Subsystem Partitioning Scheme	-0.030%	0.437%	1.91E-05	1.96291	1.857033	2.074824

the intermediate to top-level mappings while at the same time exhibiting excellent efficiency improvements over traditional methods.

Based on the results in Table 6, the lumped parameter model's means of accounting for the important yet missing interactions appears to be a good approach. For each of the various partitioning schemes, the variance of the error of the resultant LPMs across all of the responses is as good or better than that of the baseline method, the all-at-once minimum run resolution V CCD. In addition, all of the LPMs perform much better than the metamodels created using the other all-at-once approach, the optimal latin hypercube based metamodels.

One point needs to be made concerning the differences in the results in the second sub-table (the middle table) and the third sub-table (the bottom table) in Table 6. First, it seems that for the stress responses (the middle table), all of the LPMs are sufficiently better than the baseline method with first place going to the mincut-6 partitions because it has good overall accuracy and the most run savings. Comparing this to the temperature response results (bottom table), the LPM error results are all over the board with no clear, best method.

However, the results stated in the table do not illuminate the whole picture. Examining any of the temperature response figures in appendix Section A.2.2 will show that every single metamodeling method applied to the I-beam problem produced actual versus predicted results that were excellent. Every model builder would be elated as to have actual versus predicted charts that look that good. Consequently only the stress responses are important for comparing one method versus another.

The individual stress response charts in appendix Section A.2.2 tell a different story, particularly the zone 5 and zone 7 stresses. For some of the partitioning schemes, the resultant LPM zone 5 and zone 7 stresses are quite good and for others they are not. For instance compare the zone 5 and zone 7 stresses between the mincut-4 and mincut-5 partitioning schemes. Tables 10 and 11 in appendix Section

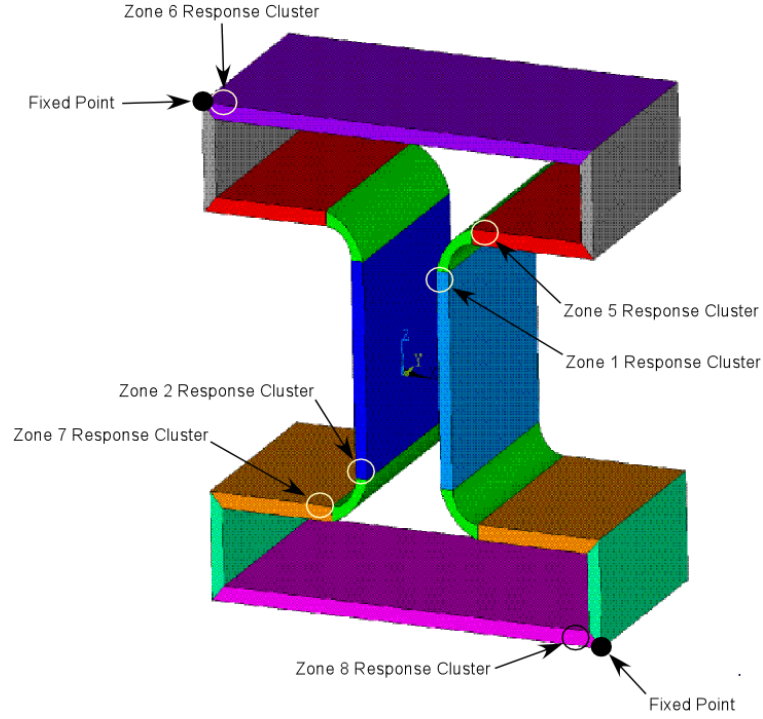


Figure 34: I-beam Model Response Clusters

A.2.1 show that the zone 5 stress for mincut-4 partitions is better than that of the baseline while the zone 5 stress for the mincut-5 partitions is worse than the baseline. For the zone 7 stresses the trend is completely the opposite, the mincut-5 performs better while the mincut-4 is worse. This seems to be a rather perplexing trend since the I-beam model was designed to be completely symmetrical. As will be shown, the variations are due to the differences between the two partition sets and which interactions each set directly accounts for.

Figure 34 shows the locations of the clusters of nodes that each response was taken from. These locations represent the maximum stress points in each zone for the baseline model. The baseline model being the model resulting from setting each of the input variables at their nominal setting. Nominal meaning the setting equal to the average of a variable's maximum and minimum values in Table 4. The points labeled "Fixed Point" in Figure 34 are the locations where the model was fixed to

constrain the deformation at those points.

Examining the variable grouping for the mincut-4 and mincut-5 partitions in Figures 38 and 39 in appendix Section A.1, we can see that in the mincut-4 set, the zone 1 and zone 5 wall thicknesses are grouped together and in the mincut-5 set they are not. This implies then that there is a direct zone 1-zone 5 wall thickness interaction for the mincut-4 set but not for the mincut-5 set.

As it turns out, the wall thickness parameters have the single largest impact on the stress responses, which makes perfect sense. Looking at Figure 34, we can see that the zone 1 response cluster is located very close to the zone 5 response cluster. Consequently it is not surprising that by not directly accounting for the zone 1 and zone 5 wall thickness interaction, the mincut-5 partition set exhibits worse relative error values for the zone 5 stress response.

Just the opposite can be shown for the zone 7 stress response. For the mincut-5 partition set, zone 2 and zone 7 wall thickness are grouped together, and in the mincut-4 partition set they are not. Thus we can see in Table 11 that the mincut-5 partition set performs much better for zone 7 stress response while the mincut-4 partitions do not. Thus the missing, direct zone 2-zone 7 wall thickness interaction is negatively affecting the accuracy of the zone 7 stress response for the mincut-4 partitions set.

Even though the relative errors for the mincut-5, zone 5 stress are higher than that of the baseline, the actual overall error values are not particularly bad. Looking at the error distribution for zone 5 stress in Figure 70 in appendix Section A.2.2, the maximum error value looks to be about 9% relative error. It also appears to be somewhat of an outlier. Examining the actual histogram data used to create that chart shows that the maximum error recorded was 9.3% and there are only four points out of 5,000 that have error values greater than 7.25%. In addition, the standard deviation of the error for the mincut-5, zone 5 stress response is only 1.3 times as

large as the baseline. The baseline zone 5 stress response standard deviation being only 1.64%, thus in general, the mincut-5, zone 5 stress is not all that bad.

While an examination of the zone 5 and zone 7 stress responses seems to indicate that missing interactions can explain why one set of partitions seems to do much better than another; the same type of analysis for the zone 1 and zone 2 stress responses do not lead to the same conclusion. It seems logical that since the zone 1 and zone 5 response clusters are so close, problems with the zone 5 stress responses should also show up in the zone 1 stress responses. However this is not the case at all. Reviewing Table 10 indicates that all of the partitioning schemes exhibit less error than the baseline method for zone 1 stress. Likewise, since the zone 2 and zone 7 response clusters are so close, the zone 2 stresses should exhibit similar error distributions as zone 7 stresses, but again they do not.

Examining the results for all the other responses in Tables 10 and 11, it appears that other than for the zone 5 and zone 7 stress responses, all of the lumped parameter based models perform better than the baseline method. In addition all the LPMs have anywhere from 18% to 45% run savings.

Moving on to the torsional vibration system test case results in Section 8.2.2. Unfortunately the results for the LPMs are not nearly as good as they are for the I-beam test case. But this was to be expected based on the graph of the system shown in Figure 32. The graph in Figure 32 is that of a very tightly coupled system, there are no obvious sub-graphs to partition the system into. Consequently both of the partitioning schemes applied to this system performed quite poorly. One can then draw that conclusion, that for tightly coupled systems, partitioned metamodels are not a good choice for reducing the data required to analyze the system. Tightly coupled systems seem to be limited to metamodeling techniques that account for all interactions directly.

In light of these examinations, it seems that reserved support can be given to

hypothesis 2.1. It appears in general, that the lumped parameter models are able to indirectly account for missing interactions and typically produce better results than traditional methods. The argument that the LPMs are indirectly accounting for the interactions is based on that fact that for most cases the LPMs perform as good or better than the all-at-once approaches, and the all-at-once approaches inherently account for these interactions. Thus the lumped parameter models must also be accounting for these interactions.

Also the fact that Baker had to resort to using variable redundancies to get (assumingly) equal accuracy with UTE method as compared with an all-at-once approach, and the lumped parameter models do not, further supports the claim that the intermediate to top-level mappings are accounting for these interactions. “Assumingly” was added in the previous sentence because Baker did not actually quote the accuracy of the all-at-once method that he was using for comparison. Thus, it is assumed that by adding the variable redundancies that he was able to match the all-at-once accuracy with the UTE method.

The “reserved” support for hypothesis 2.1 is because in some cases, the lumped parameter models do not work as well as one would like. Those cases seem to be related to instances where strong coupling exists. Examples of those cases include the entire torsional vibration system and to the apparently strong (yet not necessarily symmetrical) coupling between the zones 1 zone 5; and zones 2 and 7 in the I-beam test case.

Assertion: Baker showed that important, neglected interactions do have a negative impact on the overall accuracy of a partitioned metamodel.

Assertion: Adding variable redundancies between the partitions improved the accuracy of the UTE metamodel but did so at the cost of degraded efficiency of the method.

Research Question 2.1: Is there a way to account of potentially important interactions without resorting to variable redundancy between the partitions?

Research Hypothesis 2.1: Using the GCF method as a prototype, it is hypothesized that the intermediate to top-level mapping will indirectly account for any neglected interactions between low-level design variables in disjoint groups.

Validity of Hypothesis 2.1: Based on the accuracy results of the I-beam problem, it appears that the lumped parameter models are able to indirectly account for missing interactions and typically produce better metamodels than traditional methods. This statement is based on the fact that the traditional all-at-once methods inherently capture all important interactions.

Although for tightly coupled systems such as the TVS model, the lump parameter model is not a good candidate for creating partitioned metamodels.

8.3.2 Validating Hypotheses 3.1 and 3.2

This section will address the last two un-addressed hypotheses in this thesis. Those hypotheses are:

- It is hypothesized that a screening test using a low resolution DoE with ANOVA can be used to determine the underlying sparse structure of the black box system, which then can be partitioned using a graph-based min-cut algorithm.
- It is hypothesized that since the data from a screening test, which is to be used to tear the system into disjoint subgroups, inherently contains the connectivity information; this data can also be used to create the intermediate to top-level mappings which are designed to account for the impact of system tearing.

First hypothesis 3.1, yes it does appear that a low resolution screening test can be used to determine the underlying sparse structure of system, and actually it seems to work

pretty well. Essentially this hypothesis is concerned with using a graph to visualize the results from an ANOVA analysis across all responses at the same time. The whole goal of ANOVA is to determine which factors are important to which response. Since most factors are rarely important to all the responses, creating a sparse matrix with these results is quite straight forward. It is actually kind of surprising that no one in the literature seems to have combined ANOVA with graph theory before.

Comparing the true graph (Figure 32) and the experimentally derived graph (Figure 33) of the TVS problem shows the strength of the method. The true graph obviously shows all of the potential interactions, whether they are significant or not. The experimental graph on the other hand not only shows the interconnectedness between the important factors, but also separates the non-significant factors from the significant ones automatically. Thus model builder immediately knows how to proceed. If the graph is tightly coupled, weed out the insignificant factors and proceed with an all-at-once approach. If the graph contains several clear sub-graphs, partition the system and apply the Hi-DSE process to create a lumped parameter model.

As far as the use of the min-cut algorithm is concerned. It seems that for systems that are good candidates for partitioning, such as the I-beam problem, the resultant partitions produced will work well with the lumped parameter model. The results in Table 6 from Section 8.1.2 show that in general all of the sets of partitions created with the min-cut algorithm produce LPMs that are as good or better than metamodels based on traditional metamodeling approaches. Also the min-cut produced two sets of partitions that essentially matched those created using engineering intuition.

Assertion: A systems decomposition method for sparse systems is a logical means to partition the inputs for large scale metamodeling, however there is not enough known about the initial sparsity of the system.

Research Question 3.1: Given the black box nature of analyses used in metamodeling, how can their underlying sparse structure be determined?

Research Hypothesis 3.1: It is hypothesized that a screening test using a low resolution DoE with ANOVA can be used to determine the underlying sparse structure of the black box system, which then can be partitioned using a graph-based min-cut algorithm.

Validity of Hypothesis 3.1: Yes, a low resolution design of experiments can be used with ANOVA to determine the underlying sparse nature of the system.

Combining the results of ANOVA with graph theory provides the model builder with a clear path of how to proceed with any subsequent model building. If the graph is tightly coupled, weed out the insignificant factors and proceed with an all-at-once approach. If the graph contains several clear sub-graphs, partition the system and apply the Hi-DSE process to create a lumped parameter model. In addition, based on the accuracy results of the I-beam test case, the min-cut algorithm will find good sets of partitions to use with a LPM.

Using the discussion from Section 8.3.1 and the results of addressing hypothesis 3.1, validating hypothesis 3.2 is quite straight forward. Section 8.3.1 showed that in general the lumped parameter model's means of indirectly accounting for interactions via the intermediate to top-level mappings did create accurate metamodels. Thus it appears that the inherent connectivity information contained in the screening test data was transferred to the intermediate to top-level mappings. In addition, the same data that was used to create the partitions was also used to create the intermediate \tilde{y} regression arrays. Consequently hypothesis 3.2 is indeed a true statement.

Assertion: The use of the data from a screening test is proposed to perform two different operations in the new large scale metamodeling method: one for partitioning or tearing the system and another for determining the intermediate to

top-level mappings.

Research Question 3.2: Is it possible to use the same screening test data, that was collected for use with the lumped parameter method, to also determine the sparse nature of the system for partitioning?

Research Hypothesis 3.2: It is hypothesized that since the data from a screening test, which is to be used to tear the system into disjoint subgroups, inherently contains the connectivity information; this data can also be used to create the intermediate to top-level mappings which are designed to account for the impact of system tearing.

Validity of Hypothesis 3.2: It does appear, that based on the accuracy results of the min-cut based lumped parameter models for the I-beam test case, the same data that is used for the graph-based partitioning analysis can also be used to create the intermediate to top-level mappings without any significant impact on the accuracy of resultant the lumped parameter model.

Chapter IX

CONCLUSIONS AND OBSERVATIONS

9.1 Review of Research Questions and Hypotheses

This section outlines a map of the thesis built around the research questions and hypotheses identified throughout the entire thesis. Each research question will be stated in context to the section that it was initially posed in. Following that, each research hypothesis will then been linked to the section that addressed the validity of the specific research question/hypothesis.

The following research question and hypothesis were posed in Section 1.5 and subsequently addressed in Section 2.4. The two assertions that spurred the subsequent research question and hypothesis were based on the need to combine a multi-physics finite element analysis with an appropriate probabilistic design method to adequately assess the probabilistic nature of the life of the nozzle. The statement of the hypothesis was made based on the author's experience of using probabilistic design methods used with finite element models.

Research Question and Hypothesis:

Assertion: A spatially partitioned finite element model was created to address the complex operating environment that a nozzle is exposed to. This model has 52 variables, 255 responses and takes 3.5 hours to run.

Assertion: To capture the probabilistic nature of the life of nozzle, a probabilistic design method is needed. The primary methods are direct Monte Carlo, indirect Monte Carlo and analytic reliability methods.

Research Question 1: Which of the three primary probabilistic design methods is most applicable for the nozzle problem?

Research Hypothesis 1: Given the computational nature of the nozzle model, it is hypothesized that an indirect Monte Carlo simulation technique is most likely the best probabilistic design method to use.

Validity of Hypothesis 1: Based on the results of probabilistic method comparison, two requirements drove the down selection process, the need to simultaneously handle multiple responses and method efficiency. Of all the methods only traditional Monte Carlo simulation used in conjunction with metamodeling concurrently addressed both of the requirements. Consequently, the hypothesis that an indirect Monte Carlo method based on traditional Monte Carlo and metamodels is indeed a valid statement.

The following research question and hypothesis were posed in Section 1.5 and subsequently addressed in Section 3.4. The assertion that drove the subsequent research question and hypothesis were based on a simple estimation of the projected computational expensive required to create metamodels of the nozzle problem. A fairly conservative estimate was on the order of 5,000 hours. Thus, it was felt that any additional information about the system that could be used to create the metamodels would reduce the total computational expense.

Research Question and Hypothesis:

Assertion: Given the constraints imposed by the nozzle model's complexity, traditional statistical linear metamodeling methods are not efficient enough to be used with the model. Consequently a large scale metamodeling method is necessary to reduce the total computational expense as much as possible.

Research Question 2: Which large scale metamodeling procedure is best to address the characteristics of the nozzle model?

Hypothesis 2: Given the spatially partitioned nature of the finite element model, it is hypothesized that a partitioned metamodeling scheme seems like a logical choice for reducing to the total number of runs required to create an accurate metamodel.

Validity of Hypothesis 2: For the most part, it does appear that a partitioned metamodeling approach does make the most sense for the analysis. The primary reason being, the alternative approach for building metamodels of large scale systems, supersaturated designs with stepwise regression, may not reliably create metamodels that represent the true nature of the system. The reason for this is due to the complex aliasing structure of supersaturated designs.

However, none of the current large scale metamodeling methods meet all of the needs of the nozzle problem. Consequently, a new partitioning method is needed to be able to fully leverage the partitioned nature of the nozzle model.

The following research question and hypothesis were posed in Section 3.5 and was addressed in Section 8.3.1. The assertions that spurred the research question and hypothesis were a result of reviewing the accuracy analysis that Baker used to validate the UTE method. In that analysis he concluded that missing interactions due to partitioning were negatively impacting the accuracy of the UTE metamodels.

Research Question and Hypothesis:

Assertion: Baker showed that important, neglected interactions do have a negative impact on the overall accuracy of a partitioned metamodel.

Assertion: Adding variable redundancies between the partitions improved the accuracy of the UTE metamodel but did so at the cost of degraded efficiency of the method.

Research Question 2.1: Is there a way to account for potentially important interactions without resorting to variable redundancy between the partitions?

Research Hypothesis 2.1: Using the GCF method as a prototype, it is hypothesized that the intermediate to top-level mapping will indirectly account for any neglected interactions between low-level design variables in disjoint groups.

Validity of Hypothesis 2.1: Based on the accuracy results of the I-beam problem, it appears that the lumped parameter models are able to indirectly account for missing interactions and typically produce better metamodels than traditional methods. This statement is based on the fact that the traditional all-at-once methods inherently capture all important interactions.

Although for tightly coupled systems such as the TVS model, the lump parameter model is not a good candidate for creating partitioned metamodels.

The following research question and hypothesis were posed in Section 3.5 and was addressed in Section 6.1. The assertion that drove this research question and hypothesis was based on the primary limitation of using the GCF method. The limitation being the lack of a means to create metamodels of analyses that do not have any readily identifiable intermediaries around which to create the intermediate to top-level mappings. This research question and hypothesis were the primary motivation for the research that lead to the development of the new large scale metamodeling method

in Section 3.5.

Research Question and Hypothesis:

Assertion: Baker used a screening test to determine important, neglected interactions and then used variable redundancy to account for these interactions.

Assertion: In the hypothesized method built on the GCF model, these interactions will be indirectly accounted for via interactions between the intermediate responses in the intermediate to top-level mapping.

Research Question 2.2: Can the data from screening test be used to determine the intermediate to top-level mappings for systems without identifiable intermediaries?

Research Hypothesis 2.2: It is hypothesized that since the data from a screening test data inherently contains the connectivity of the system, this data can be used to create the intermediate to top-level mappings for systems that do not have readily identifiable intermediaries.

Validity of Hypothesis 2.2: The logical development of the lumped parameter model (LPM) demonstrates how data from a screening test can be used to create the intermediate to top-level mappings for systems that do not have readily identifiable intermediaries.

Partitioning the inputs to a screening test into their constitutive sub-vectors and individually evaluating these sub-vectors with the $\tilde{y}_i = g_i(\mathbf{x}_i)$ functions will create the \tilde{y} regression arrays. The \tilde{y} regression arrays can then be combined with the response data from the screening test to create the $y = f(\tilde{\mathbf{y}})$ intermediate to top-level mappings.

The following research question and hypothesis were posed in Section 1.5 and subsequently addressed in Section 4.5. These two assertions were the result of the proposing

to use a partitioned metamodeling approach to create metamodels of the nozzle problem and realizing that there is a need for a method to help determine a good set of partitions.

Research Question and Hypothesis:

Assertion: A partitioned metamodeling approach is proposed to reduce to total number of runs required to create metamodels of the nozzle model.

Assertion: The electrical engineering community has been using partitioning techniques for many years to analyze VLSI circuits.

Research Question 3: Is there a way to quantitatively find an good set of partitions to use with the partitioned metamodeling process?

Research Hypothesis 3: It is hypothesized that a method from the electrical engineering community for partitioning VLSI circuits could potentially be adopted for finding an optimal set of partitions for use with partitioned metamodeling.

Validity Hypothesis 3: In general, this research hypothesis seems to be true. The sparse matrix and graph theoretic methods have all the necessary characteristics for a partitioning scheme for large scale metamodeling. However, the lack of initial knowledge about the sparse structure of the black-box systems used with metamodeling seems to preclude their use.

The following research question and hypothesis were posed in Section 4.5 and was addressed in Section 8.3.2. The assertion that spurred the subsequent research question and hypothesis was made after reviewing the common sparse matrix methods used for decomposing large scale systems into smaller disjoint subsets.

Research Question and Hypothesis:

Assertion: A systems decomposition method for sparse systems is a logical means to partition the inputs for large scale metamodeling, however there is not enough initially known about the sparsity of the system.

Research Question 3.1: Given the black box nature of analyses used in metamodeling, how can their underlying sparse structure be determined?

Research Hypothesis 3.1: It is hypothesized that a screening test using a low resolution DoE with ANOVA can be used to determine the underlying sparse structure of the black box system, which then can be partitioned using a graph-based min-cut algorithm.

Validity of Hypothesis 3.1: Yes, a low resolution design of experiments can be used with ANOVA to determine the underlying sparse nature of the system.

Combining the results of ANOVA with graph theory provides the model builder with a clear path of how to proceed with any subsequent model building. If the graph is tightly coupled, weed out the insignificant factors and proceed with an all-at-once approach. If the graph contains several clear sub-graphs, partition the system and apply the Hi-DSE process to create a lumped parameter model.

In addition, based on the accuracy results of the I-beam test case, the min-cut algorithm will find good sets of partitions to use with a LPM.

The following research question and hypothesis were posed in Section 4.5 and was addressed in Section 8.3.2. The following assertion was made after realizing that to maximize the efficiency of the new large scale metamodeling method, all data collected from the system needed to be used to create the metamodels of the system.

Research Question and Hypothesis:

Assertion: The use of the data from a screening test is proposed to perform two different operations in the new large scale metamodeling method: one for partitioning or tearing the system and another for determining the intermediate to top-level mappings.

Research Question 3.2: Is it possible to use the same screening test data, that was collected for use with the lumped parameter method, to also determine the sparse nature of the system for partitioning?

Research Hypothesis 3.2: It is hypothesized that since the data from a screening test, which is to be used to tear the system into disjoint subgroups, inherently contains the connectivity information; this data can also be used to create the intermediate to top-level mappings which are designed to account for the impact of system tearing.

Validity of Hypothesis 3.2: It does appear, that based on the accuracy results of the min-cut based lumped parameter models for the I-beam test case, the same data that is used for the graph-based partitioning analysis can also be used to create the intermediate to top-level mappings without any significant impact on the accuracy of resultant the lumped parameter model.

9.2 Contributions of this Thesis

The primary contributions of this thesis are the lumped parameter model (LPM) and the graph-based partitioning (GBP) methods for black-box systems. The secondary contribution of this thesis is the formulation of the hierarchical design space exploration (Hi-DSE) procedure that provides the model builder with a step-by-step process for creating partitioned metamodels of large scale systems.

However, one could argue that the research that went into addressing the initial three research questions and hypotheses summarized in Section 9.1 would constitute a contribution. For instance, the simple (albeit tedious) task of summarizing the literature necessary to answer and validate research question 1 and hypothesis 1 can be considered a contribution. Future researchers, tackling similar problems, can use the literature search on probabilistic design methods from Chapter 2 to point themselves in the direction of the salient references. In addition, they can use the qualitative down selection in Section 2.4 to gauge their own problem. Although the conclusions shown in Table 1 of the characteristic of probabilistic methods mapped to the characteristics of the nozzle model are primarily qualitative in nature, the stated rankings are either based on the author’s experience or statements and conclusions drawn from the literature. Thus they can be used as a gauge for future problems.

Along a similar vein, the literature search that went into addressing research question 2 and hypothesis 2 which was associated with reviewing and selecting a large scale metamodeling technique for the nozzle problem is also a contribution. To the author’s knowledge, this type of information has not been summarized before and constitutes the only methods currently stated in the literature for creating metamodels of large scale systems.

The literature search associated with the development of the graph-based partitioning method for black-box systems is not especially informative for engineering community as a whole (especially for electrical engineers). However for statistical model builders, the review of where these methods came from and how they can be easily adapted for determining the structure of large scale systems is important. Thus for a small portion of the engineering community, the literature search for answering and addressing research question 3 and hypothesis 3 is a useful contribution.

Thus, the contributions of this thesis can be summarized as:

1. The development of the lumped parameter method.

2. The development of the graph-based partitioning scheme for black-box systems.
3. The formulation of the hierarchical design space exploration procedure for large scale systems.
4. A thorough summary of the primary probabilistic design method and their applicability for use with large scale systems.
5. A thorough review of the currently available large scale metamodeling methods and their weaknesses.
6. A review of the sparse matrix methods for systems decomposition and how they can be used to quantify the sparse nature of a black-box system.

9.3 Conclusions and Observations

The first conclusion or observation that needs to be made about the primary contributions for this thesis is how it relates to the problem that initiated the work. Recall the primary goal for developing these new methods was to enable the probabilistic lifting the first stage nozzle on the new H machine. The motivation being that the traditional metamodeling methods were not efficient enough and thus a new method that was just as accurate but significantly more efficient was desired. Consequently the author used the nozzle problem as a driver and created the hierarchical design space exploration process which used a graph-based partitioning scheme to create lumped parameter models of the system.

Although the nozzle problem was the motivator, the methods developed are completely generic and, in general, can be applied to any large scale system. That is the true strength of this work. The methods contained in this thesis address not only the nozzle problem but allow a model builder to systematically tackle all sorts of large scale problems.

The reason being is because the lumped parameter model was developed in a very generic manner without reference to any particular system, characteristic of a system or model for that matter. This is in contrast to say the spatially partitioned models¹ which were developed by leveraging a natural phenomena to create a partitioned model of the system. Consequently, as long as the system makes logical sense to partition, the lumped parameter model is a good candidate for creating metamodels of that system.

The generality of the lumped parameter model and its partitioned nature produce two additional characteristics that were not leveraged in this thesis but should be mentioned none-the-less. First, because the lumped parameter model was developed without any specific functional mapping in mind, it can be used with any type of metamodeling scheme. While all of the validation for the LPM was done using response surface equations, it could have just as easily been done using kriging models, neural nets, or even gaussian processes.

Even better still, each of these modeling types can be heterogeneously combined to create one specific lumped parameter model. Thus, during the lumped parameter building process, if any specific model used to create the $\tilde{y}_i = g_i(\mathbf{x}_i)$ or $y = f(\tilde{\mathbf{y}})$ functions is not meeting expectations, upgrading to a higher fidelity model is straight forward and does not require the modifications of any of the lower level functions. However, if a low-level $\tilde{y}_i = g_i(\mathbf{x}_i)$ is changed, the associated \tilde{y} regression arrays should also be recreated and analyzed. But if the process is automated, then upgrading the localized metamodeling type is not a significant undertaking.

The second important characteristic that was not directly addressed in this thesis was how the lumped parameter model supports “localized” incremental model building. In a typical incremental modeling process the modeler collects some data, fits a model, checks its quality and iterates with more data until a sufficiency accurate

¹See Section 3.3.1 for a review of the spatially partitioned model.

metamodel is created. At each iteration of the process all the variables have to be re-analyzed at the same time, thus the required data per iteration grows like the data requirements needed to fit a model as the system size grows. Contrast this to the localized, incremental modeling building enabled by the lumped parameter model.

As an example, assume that we have partitioned our system, analyzed each of the subgroups and created the $\tilde{y}_i = g_i(\mathbf{x}_i)$ functions. In the evaluation of the subgroup functions, we see that for the current data collected, one or two of the subgroups is not producing particularly good metamodels. To improve these functions all we have to do is locally add more data to just the under performing subgroups, not to all the subgroups. Thus the localized incremental model building enabled by the lumped parameter model allows for additional data to be collected in a targeted fashion, which reduces that total data required as compared to a typical incremental model building process.

While the lumped parameter models are generically applicable, the graph-based visualization and partitioning method for black-box systems is probably the most useful piece of work to come out of this thesis. Its utility can immediately been seen simply by looking at Figures 28 and 33. Figure 28 is essentially the result of twelve different Pareto analyses wrapped into one graph.

A Pareto chart is a simple bar chart that summarizes the results of an ANOVA analysis for a given response. Each bar is ordered from left to right or top to bottom based on length. The length of each bar is equates to how much variability of the response is associated with the effect assigned to that bar. Thus examining several Pareto charts back-to-back next requires the user to have to remember which effects were predominate from one chart to the next. For two or three responses, this is pretty easy but for 12 or 255 in the case of the nozzle problem, it becomes a little tedious.

Fortunately the graph-based method tells you all of that information instantly and

concisely. It tells you not only which factors have potentially important interactions via their connectedness; it also tells you which factors are not important to any of the twelve responses at the same time. Thus with this information in hand, the model builder can easily make a decision of how to proceed next in model building process. As was stated in Section 8.3.2, if the graph is tightly coupled, weed out the insignificant factors and proceed with an all-at-once approach. If the graph contains several clear sub-graphs, partition the system and apply the Hi-DSE process to create a lumped parameter model. Its that simple.

The previous comments were related to how the graph shows you the interconnect-edness between the variables. But as was shown in Section 6.3, the incidence matrix created from the ANOVA results can also be used to examine the interconnectedness between the responses. The response adjacency matrix is created by reversing the order of the matrix multiplication used to transform the incidence matrix into the variable adjacency matrix. The vertices in the graph based on the response adjacency matrix are now the responses and the edges connecting the vertices present shared variables. Therefore, two responses that are connected share at least one variable that is statistically significant to both responses. Unfortunately the author does not immediately know how to leverage this information but new things are created all the time without an immediate application. The glue on the back of sticky notes is an excellent example.

9.4 Closing Remarks

The final section in this thesis will address a very important question that should be addressed in every Ph.D. thesis. That question is: Is this work actually Ph.D. degree worthy? To answer this question first the goal of what a Ph.D. dissertation should accomplish needs to be discussed. From the Georgia Institute of Technology dissertation guidelines [41], the first bullet states: “Doctoral research should provide

a useful educational experience for the student emphasizing creativity, independent action and learning, research methodology and scholarly approach”.

So does the work presented in this thesis meet these fundamental characteristics? First, yes the work that went into completing this thesis can definitely be classified as an “education experience”, for two reasons.

One, from a purely literal interpretation of educational experience, the process of finding and reviewing the background literature necessary to successfully pose and solve problems addressed in this thesis was definitely an educational experience. Before the literature search was started, the author knew very little about the analytic reliability methods, other large scale metamodeling methods, sparse matrix methods and graph theory. Now however, the author would consider himself not necessarily an expert in any of these topics, but rather an informed researcher that can talk intelligently enough to know their realm of applicability and inherent limitations.

Second, the educational experience associated with learning how to do research was probably the most valuable aspect of the work that produced this thesis. The primary reason being is because the author executed the research process completely wrong and now knows how to do it right.

When the motivating statement of work was initially posed for doing lifing of the first stage nozzle on the H machine, the author almost immediately settled on the hierarchically partitioned metamodel solution path without actually doing much of an initial background literature search. Consequently, the better part of a six months was spent trying to support the proposed solution with references from the literature. Fortunately, the author initially made some good choices and finding support was not too difficult. However, he did learn from his mistakes and when the research to solve the problem of finding a good set of partitions was initiated, a thorough literature search found a potential solution that was easy to adapt to the problem.

As far as the other primary characteristics that a Ph.D. dissertation should possess, the author feels that yes, the work in this thesis is both creative and independent and contains the necessary ingredients to be classified as exhibiting a research methodology and scholarly approach. Meeting the characteristic of independence is easy, other than discussions with other researchers to circumvent roadblocks that always pop-up, the work in this thesis is solely that of the author. Creativity is a little more subjective, what is a creative solution for one student may be pedestrian for another. However, discussions between the author and several statistics professors at Georgia Tech indicated that the work for this thesis was completely new and had not been demonstrated in the literature, that is to their knowledge anyway.

The scholarly approach is addressed by both the thoroughness of the literature search as well as the posing of the research questions and hypotheses. Good research questions and the subsequent hypotheses should on the one hand be testable, that is can they be proved or disproved, and on the other hand, interesting enough to be non-trivial. That is not have obvious solutions but still be solvable. The research questions and hypotheses in this thesis seem to fit that requirement. When posed, they all were a logical extension of the background information that preceded their posing. In addition, they were all solvable either by literature search or by testing. Consequently the author feels that this thesis meets the requirements of a scholarly approach.

Of the other items in the Georgia Tech dissertation guidelines, two more need direct addressing. These items are: “the research should possess the major characteristics of the scientific method, namely objectivity and reproducibility”; and “the dissertation should reflect a level of competence indicative of a significant achievement beyond the masters level”.

First, the author made every attempt, when comparing the new methods developed to current standard methods, to be as objective as possible. The exact same

analysis procedure was used for every assessment and standard techniques were used whenever possible. As far as reproducibility is concerned, the steps that the author used to assess each of the methods were summarized as close a possible to the actual procedure used and all of the pertinent code and process steps necessary to reproduce the work have been included.

Second, this research completed for this thesis is definitely above the research required for a master thesis. To the author's understanding, a masters thesis is primarily associated with applying the state-of-the-art in a given field to a problem of interest, that is, a masters thesis is a research study. A Ph.D thesis on the other hand is intended to extend the state-of-the-art in a given field, to produce new knowledge or understanding that fills a gap or a lack of knowledge in the field. This thesis certainly does the latter. The development of the lumped parameter model and the extension of graph-based partitioning to metamodeling adds new knowledge to the field of large scale systems analysis. These methods enable a model builder to analyze and create metamodels of large scale systems that were previously difficult or impossible analyze. The new methods offer a step-by-step approach for creating partitioned metamodels of a system at are both more efficient to create and offer as good or better accuracy than traditional metamodeling methods. Consequently, the author feels that the research completed for this thesis does indeed extend the state-of-the-art in large scale model building and thus, is Ph.D. quality work.

Appendix A

I-BEAM TEST CASE RESULTS

This appendix contains all of the results from validation process for the the I-beam test case. Section A.1 shows the graphs, matrix array plots and variables assigned to each of the partitions for the complete graph as well as each of the partitioned graphs created using the min-cut graph partitioning algorithms.

Section A.2.1 contains the individual response error summaries for each of the metamodel types. Each table contains the mean, standard deviation and variance of the percent error for each of the individual stress and temperature responses for the I-beam test case. The cumulative error summaries for each of the metamodel types are summarized in Table 6 in Section 8.1.2.

Section A.2.2 contains the actual versus predicted charts for each response and metamodel type. In addition, the histograms of the relative error distributions for the validation data are also shown.

A.1 Min-cut Partition Summary

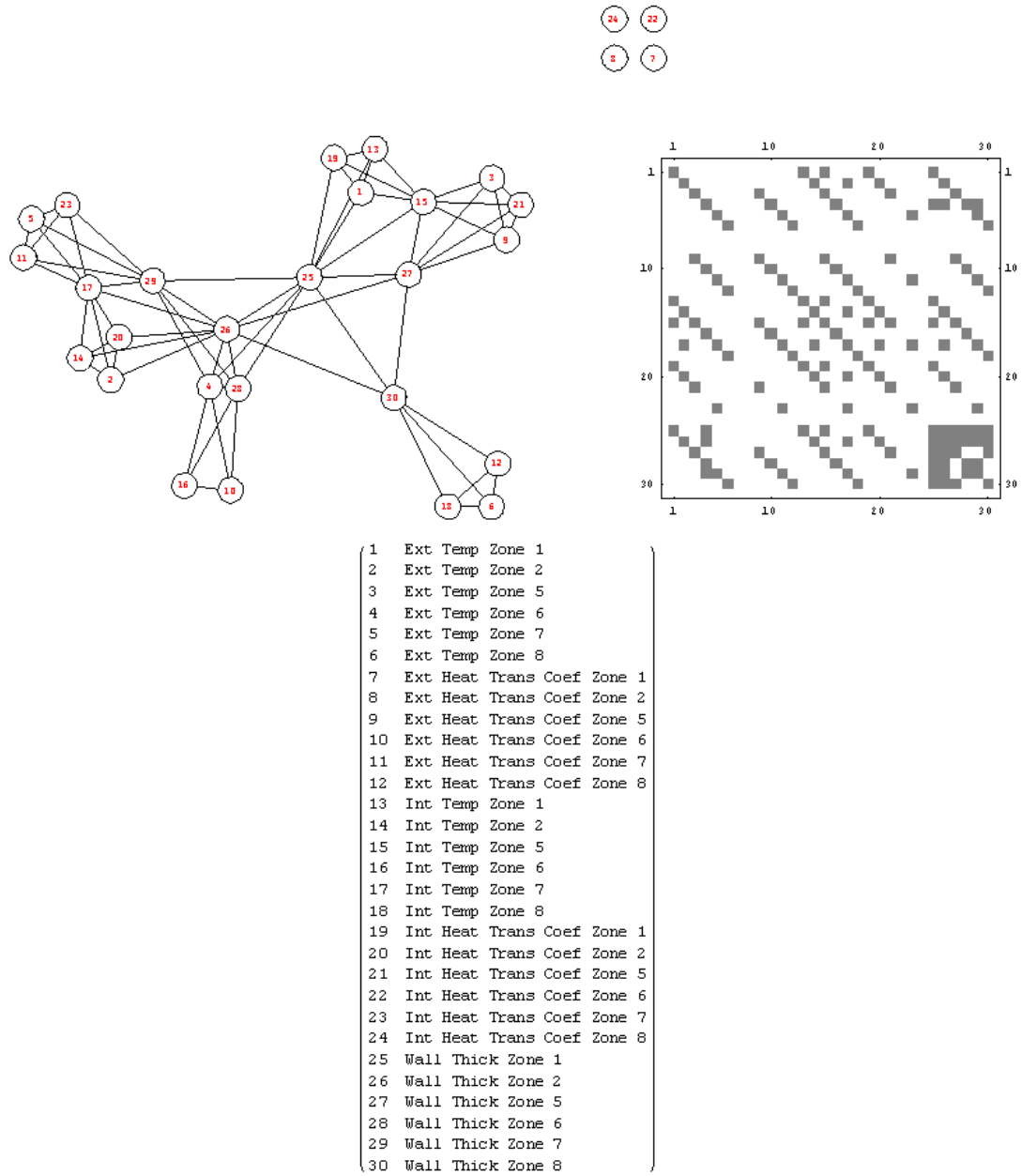


Figure 35: Complete Graph Results

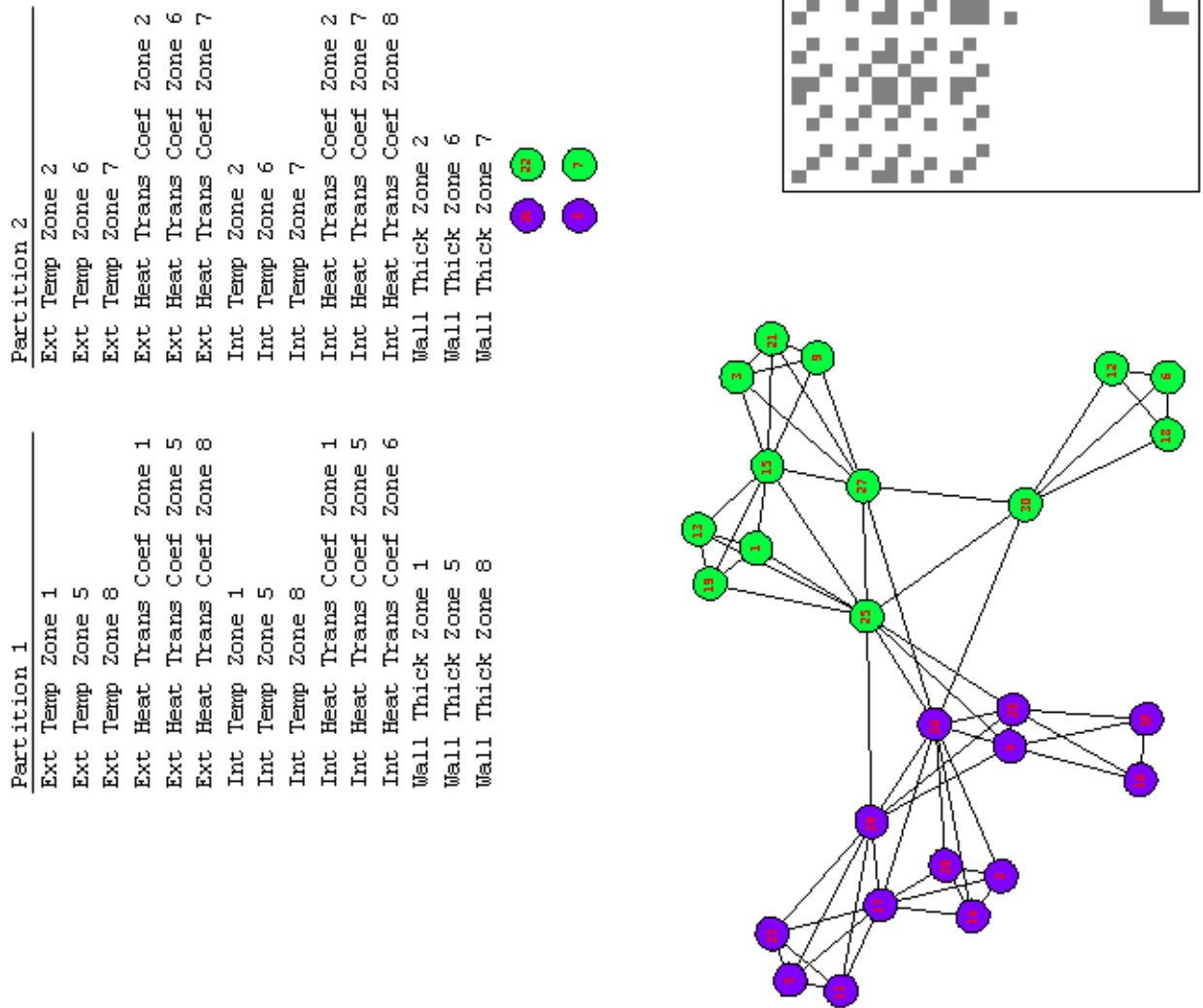


Figure 36: Mincut-2 Results

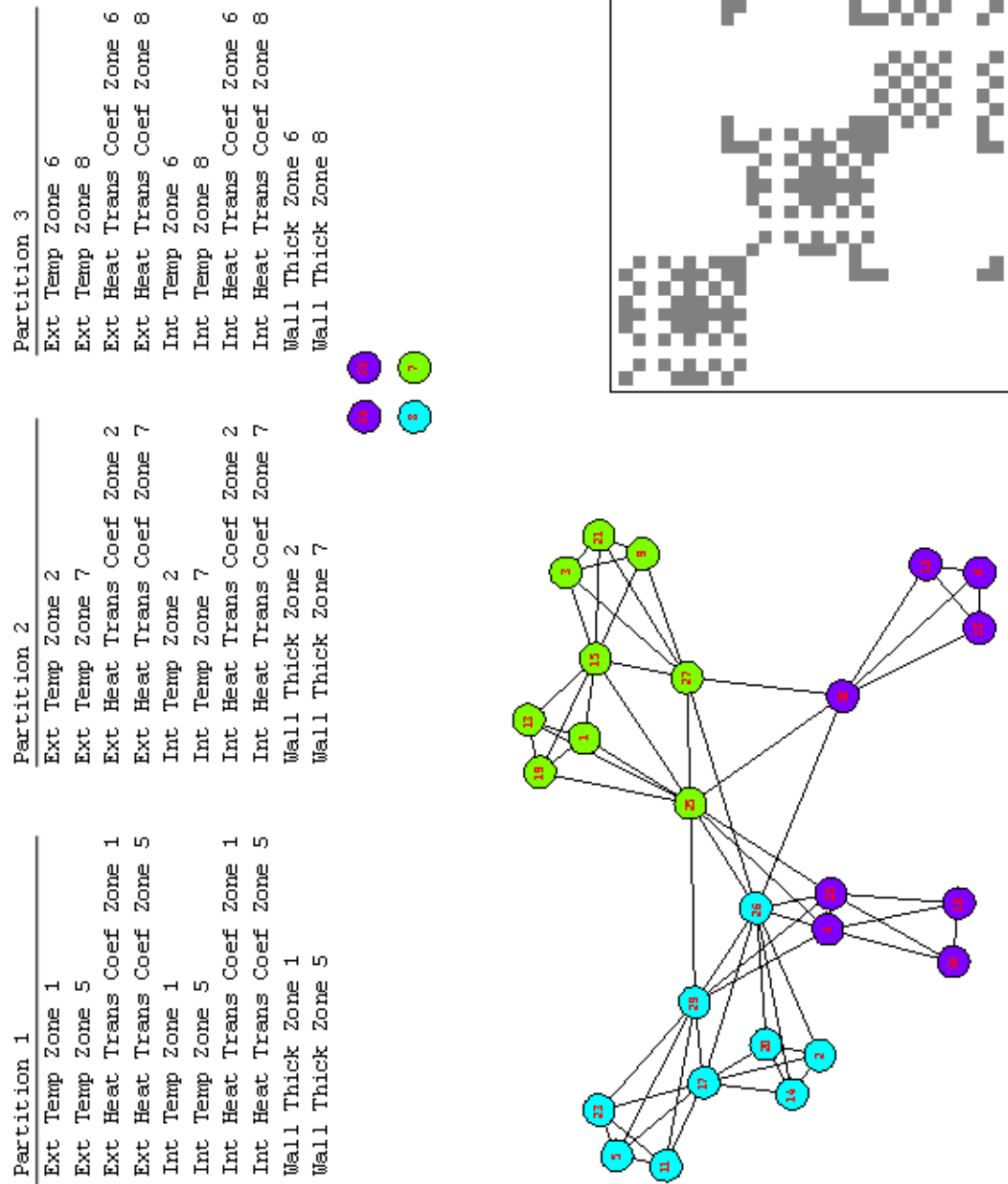


Figure 37: Mincut-3 Results

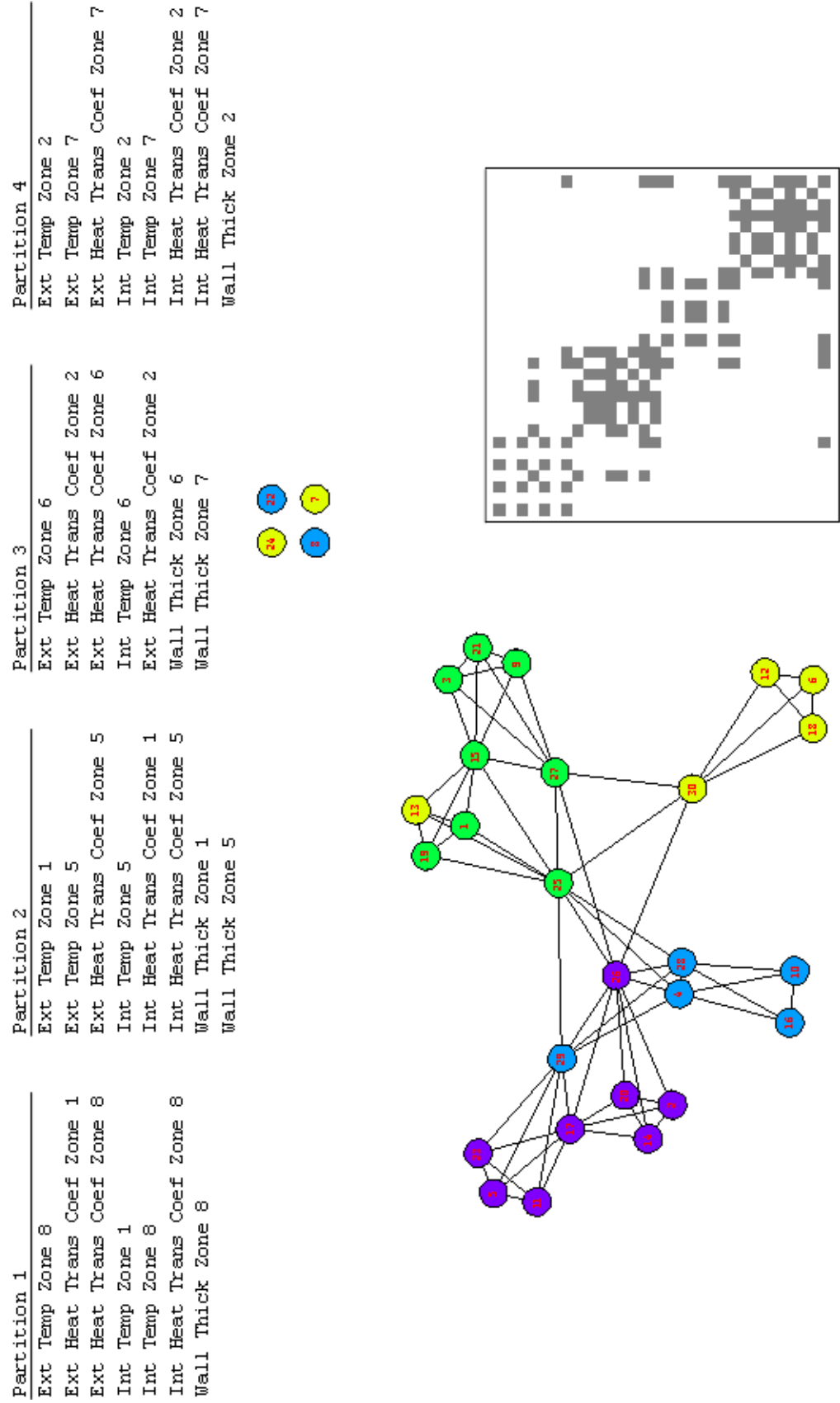


Figure 38: Min-cut-4 Results

Partition 1	Partition 2	Partition 3
Ext Temp Zone 6	Ext Temp Zone 2	Ext Temp Zone 8
Ext Heat Trans Coef Zone 6	Ext Temp Zone 7	Ext Heat Trans Coef Zone 1
Int Temp Zone 6	Ext Heat Trans Coef Zone 7	Ext Heat Trans Coef Zone 8
Wall Thick Zone 2	Int Temp Zone 2	Int Temp Zone 8
Wall Thick Zone 6	Int Temp Zone 7	Wall Thick Zone 8
Wall Thick Zone 7	Int Heat Trans Coef Zone 2	
	Int Heat Trans Coef Zone 7	
	Int Heat Trans Coef Zone 2	
	Int Heat Trans Coef Zone 7	
	Int Heat Trans Coef Zone 2	
	Int Heat Trans Coef Zone 7	
Partition 4	Partition 5	
Ext Temp Zone 1	Ext Temp Zone 5	
Int Temp Zone 1	Ext Heat Trans Coef Zone 2	
Int Temp Zone 5	Ext Heat Trans Coef Zone 5	
Int Heat Trans Coef Zone 1	Int Heat Trans Coef Zone 5	
Int Heat Trans Coef Zone 6	Int Heat Trans Coef Zone 8	
Wall Thick Zone 1	Int Heat Trans Coef Zone 5	
	Wall Thick Zone 5	

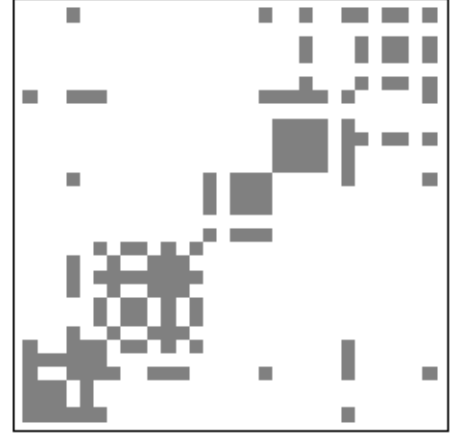
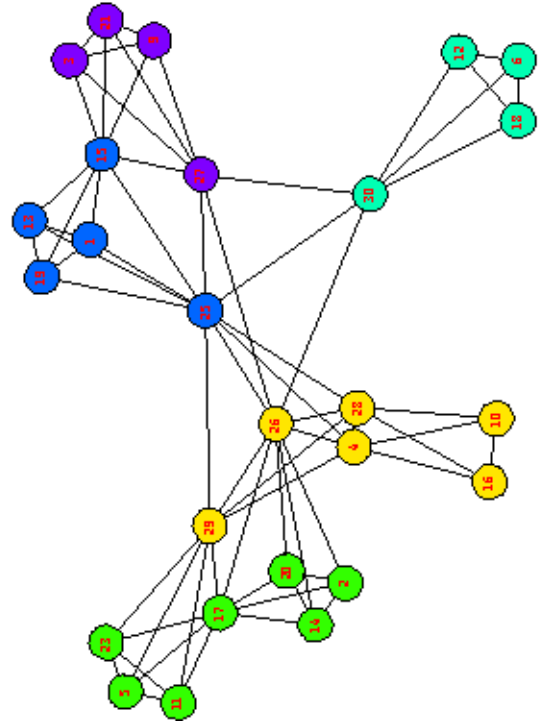


Figure 39: Mincut-5 Results

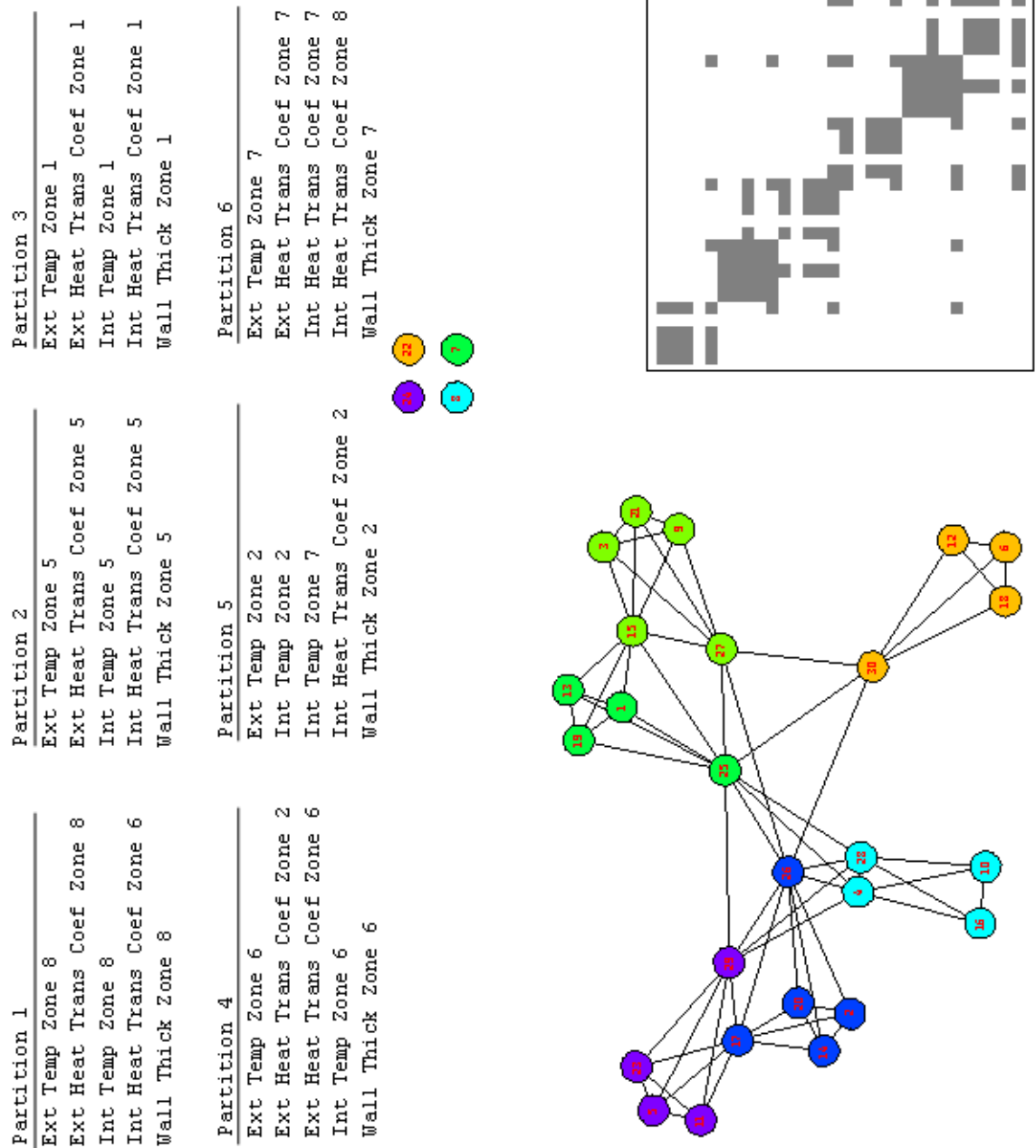


Figure 40: Mincut-6 Results

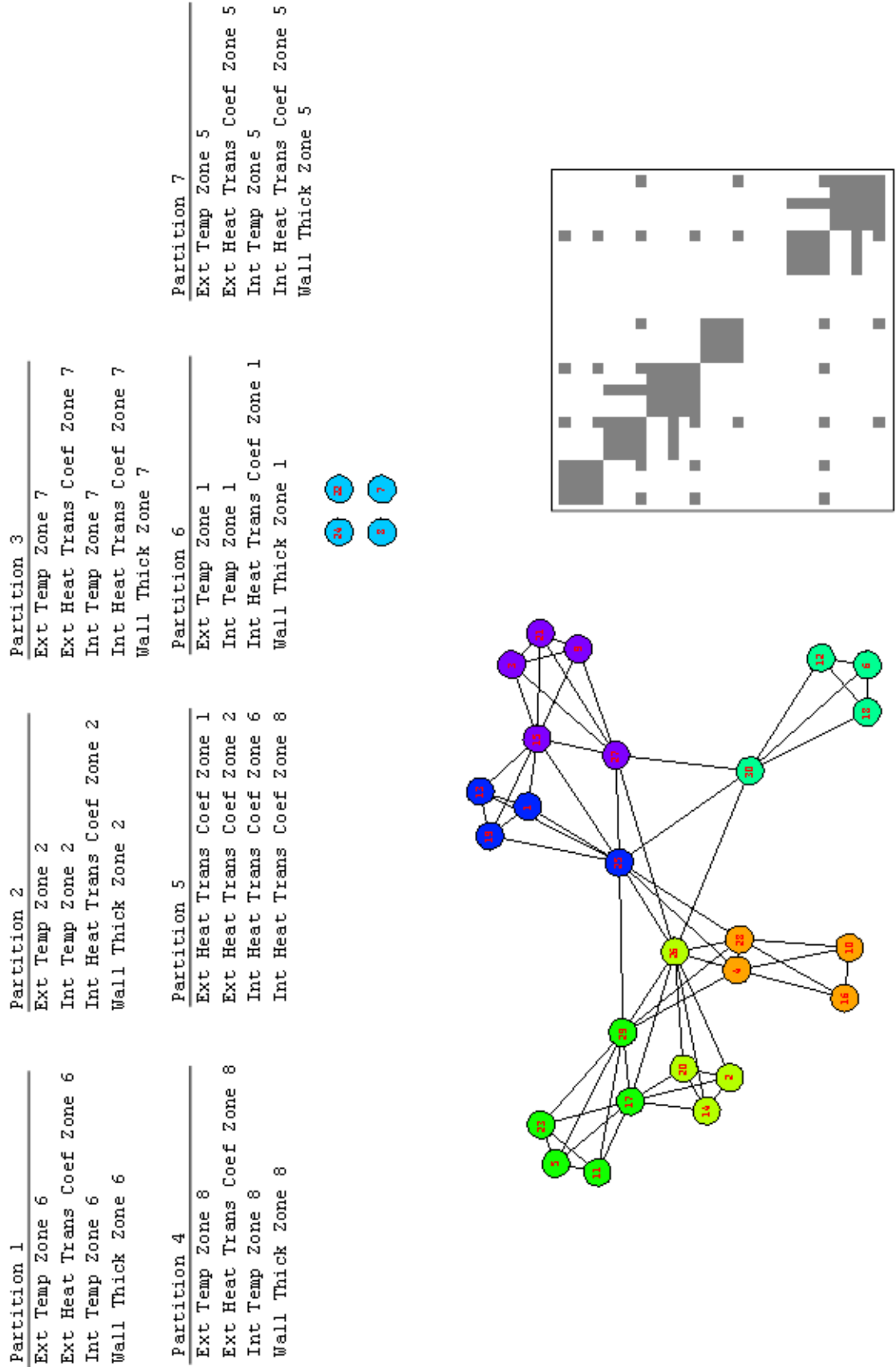


Figure 41: Partition-7 Results

A.2 Complete Error Results for each Response per Metamodel Type

Section A.2.1 contains the individual response error summaries for each of the metamodel types. Each table contains the mean, standard deviation and variance of the percent error for each of the individual stress and temperature responses for the I-beam test case. The cumulative error summaries for each of the metamodel types are summarized in Table 6 in Section 8.1.2.

Section A.2.2 contains the actual versus predicted charts for each response and metamodel type. In addition, the histograms of the relative error distributions for the validation data are also shown.

A.2.1 Tables of Individual Response Error Summaries

Table 10: Zones 1, 2 and 5 Stress Response Error Summary

Response	Analysis Type	% Error Mean	% Error Std. Dev.	% Error Variance	Variance Ratio	95% Confidence	
						L.B.	U.B.
Zone 1 Stress	All-at-once Min Run Res. V CCD	0.175%	1.779%	3.166E-04	1.000	n/a	n/a
	All-at-once Optimal Latin Hypercube	0.063%	2.379%	5.660E-04	0.559	0.528	0.592
	Variable-Type Partitioning Scheme	-0.223%	1.562%	2.440E-04	1.297	1.225	1.374
	Subsystem Partitioning Scheme	-0.147%	1.261%	1.591E-04	1.990	1.879	2.108
	Mincut 2 Partitioning Scheme	-0.112%	1.445%	2.089E-04	1.516	1.431	1.605
	Mincut 3 Partitioning Scheme	-0.057%	1.295%	1.676E-04	1.889	1.783	2.000
	Mincut 4 Partitioning Scheme	-0.160%	1.373%	1.884E-04	1.680	1.587	1.779
	Mincut 5 Partitioning Scheme	-0.036%	1.224%	1.499E-04	2.112	1.994	2.236
	Mincut 6 Partitioning Scheme	-0.163%	1.217%	1.482E-04	2.136	2.017	2.262
	Mincut 7 Partitioning Scheme	-0.123%	1.310%	1.717E-04	1.844	1.741	1.952

Response	Analysis Type	% Error Mean	% Error Std. Dev.	% Error Variance	Variance Ratio	95% Confidence	
						L.B.	U.B.
Zone 2 Stress	All-at-once Min Run Res. V CCD	0.189%	1.666%	2.775E-04	1.000	n/a	n/a
	All-at-once Optimal Latin Hypercube	0.099%	2.281%	5.201E-04	0.533	0.504	0.565
	Variable-Type Partitioning Scheme	-0.233%	1.533%	2.350E-04	1.181	1.115	1.250
	Subsystem Partitioning Scheme	0.041%	1.402%	1.966E-04	1.412	1.333	1.495
	Mincut 2 Partitioning Scheme	0.018%	1.420%	2.016E-04	1.376	1.300	1.458
	Mincut 3 Partitioning Scheme	-0.126%	1.328%	1.764E-04	1.573	1.485	1.665
	Mincut 4 Partitioning Scheme	0.087%	1.194%	1.427E-04	1.945	1.836	2.059
	Mincut 5 Partitioning Scheme	-0.042%	1.456%	2.119E-04	1.310	1.237	1.387
	Mincut 6 Partitioning Scheme	0.033%	1.362%	1.856E-04	1.495	1.411	1.583
	Mincut 7 Partitioning Scheme	0.022%	1.604%	2.572E-04	1.079	1.019	1.142

Response	Analysis Type	% Error Mean	% Error Std. Dev.	% Error Variance	Variance Ratio	95% Confidence	
						L.B.	U.B.
Zone 5 Stress	All-at-once Min Run Res. V CCD	0.207%	1.624%	2.639E-04	1.000	n/a	n/a
	All-at-once Optimal Latin Hypercube	-0.080%	2.970%	8.821E-04	0.299	0.282	0.317
	Variable-Type Partitioning Scheme	-0.310%	1.324%	1.754E-04	1.505	1.421	1.593
	Subsystem Partitioning Scheme	0.084%	2.150%	4.624E-04	0.571	0.539	0.604
	Mincut 2 Partitioning Scheme	-0.276%	1.390%	1.932E-04	1.366	1.290	1.446
	Mincut 3 Partitioning Scheme	-0.259%	1.313%	1.725E-04	1.530	1.444	1.620
	Mincut 4 Partitioning Scheme	-0.089%	1.346%	1.810E-04	1.458	1.376	1.543
	Mincut 5 Partitioning Scheme	0.284%	2.047%	4.190E-04	0.630	0.595	0.667
	Mincut 6 Partitioning Scheme	0.110%	2.132%	4.544E-04	0.581	0.548	0.615
	Mincut 7 Partitioning Scheme	0.084%	2.329%	5.425E-04	0.486	0.459	0.515

Table 11: Zones 6, 7 and 8 Stress Response Error Summary

Response	Analysis Type	% Error Mean	% Error Std. Dev.	% Error Variance	Variance Ratio	95% Confidence	
						L.B.	U.B.
Zone 6 Stress	All-at-once Min Run Res. V CCD	0.022%	2.112%	4.462E-04	1.000	n/a	n/a
	All-at-once Optimal Latin Hypercube	0.009%	2.063%	4.256E-04	1.048	0.990	1.110
	Variable-Type Partitioning Scheme	0.069%	1.519%	2.307E-04	1.934	1.827	2.049
	Subsystem Partitioning Scheme	-0.066%	1.172%	1.373E-04	3.251	3.070	3.443
	Mincut 2 Partitioning Scheme	-0.251%	1.418%	2.011E-04	2.219	2.095	2.350
	Mincut 3 Partitioning Scheme	-0.291%	1.186%	1.408E-04	3.170	2.994	3.357
	Mincut 4 Partitioning Scheme	-0.133%	1.287%	1.657E-04	2.693	2.543	2.852
	Mincut 5 Partitioning Scheme	-0.132%	1.638%	2.683E-04	1.663	1.570	1.761
	Mincut 6 Partitioning Scheme	-0.067%	1.171%	1.371E-04	3.255	3.073	3.447
	Mincut 7 Partitioning Scheme	-0.109%	1.225%	1.500E-04	2.974	2.808	3.149

Response	Analysis Type	% Error Mean	% Error Std. Dev.	% Error Variance	Variance Ratio	95% Confidence	
						L.B.	U.B.
Zone 7 Stress	All-at-once Min Run Res. V CCD	0.254%	1.529%	2.337E-04	1.000	n/a	n/a
	All-at-once Optimal Latin Hypercube	0.106%	3.244%	1.052E-03	0.222	0.210	0.235
	Variable-Type Partitioning Scheme	-0.153%	1.354%	1.833E-04	1.275	1.204	1.350
	Subsystem Partitioning Scheme	-0.043%	2.699%	7.283E-04	0.321	0.303	0.340
	Mincut 2 Partitioning Scheme	-0.003%	1.389%	1.928E-04	1.212	1.145	1.284
	Mincut 3 Partitioning Scheme	-0.061%	1.339%	1.792E-04	1.304	1.232	1.381
	Mincut 4 Partitioning Scheme	0.037%	1.833%	3.360E-04	0.696	0.657	0.737
	Mincut 5 Partitioning Scheme	-0.057%	1.502%	2.257E-04	1.036	0.978	1.097
	Mincut 6 Partitioning Scheme	-0.205%	1.845%	3.403E-04	0.687	0.649	0.727
	Mincut 7 Partitioning Scheme	0.032%	2.623%	6.878E-04	0.340	0.321	0.360

Response	Analysis Type	% Error Mean	% Error Std. Dev.	% Error Variance	Variance Ratio	95% Confidence	
						L.B.	U.B.
Zone 8 Stress	All-at-once Min Run Res. V CCD	0.078%	1.971%	3.886E-04	1.000	n/a	n/a
	All-at-once Optimal Latin Hypercube	0.013%	1.839%	3.383E-04	1.149	1.085	1.217
	Variable-Type Partitioning Scheme	-0.170%	1.469%	2.159E-04	1.800	1.700	1.906
	Subsystem Partitioning Scheme	-0.005%	1.241%	1.540E-04	2.524	2.383	2.672
	Mincut 2 Partitioning Scheme	-0.073%	1.556%	2.421E-04	1.605	1.515	1.700
	Mincut 3 Partitioning Scheme	-0.169%	1.276%	1.629E-04	2.386	2.253	2.527
	Mincut 4 Partitioning Scheme	-0.188%	1.172%	1.374E-04	2.828	2.670	2.995
	Mincut 5 Partitioning Scheme	-0.024%	1.393%	1.940E-04	2.003	1.891	2.121
	Mincut 6 Partitioning Scheme	0.021%	1.256%	1.577E-04	2.465	2.327	2.610
	Mincut 7 Partitioning Scheme	-0.055%	1.358%	1.844E-04	2.107	1.989	2.231

Table 12: Zones 1, 2 and 5 Temperature Response Error Summary

Response	Analysis Type	% Error Mean	% Error Std. Dev.	% Error Variance	Variance Ratio	95% Confidence	
						L.B.	U.B.
Zone 1 Temp	All-at-once Min Run Res. V CCD	0.002%	0.063%	4.022E-07	1.000	n/a	n/a
	All-at-once Optimal Latin Hypercube	-0.001%	0.066%	4.413E-07	0.912	0.861	0.965
	Variable-Type Partitioning Scheme	0.001%	0.034%	1.187E-07	3.388	3.199	3.588
	Subsystem Partitioning Scheme	-0.002%	0.056%	3.103E-07	1.296	1.224	1.373
	Mincut 2 Partitioning Scheme	0.002%	0.060%	3.597E-07	1.118	1.056	1.184
	Mincut 3 Partitioning Scheme	0.000%	0.052%	2.742E-07	1.467	1.385	1.553
	Mincut 4 Partitioning Scheme	0.003%	0.107%	1.142E-06	0.352	0.333	0.373
	Mincut 5 Partitioning Scheme	-0.003%	0.079%	6.290E-07	0.639	0.604	0.677
	Mincut 6 Partitioning Scheme	-0.002%	0.056%	3.103E-07	1.296	1.224	1.373
	Mincut 7 Partitioning Scheme	-0.006%	0.078%	6.158E-07	0.653	0.617	0.692

Response	Analysis Type	Mean	Standard Deviation	Variance	Variance Ratio	95% Confidence	
						L.B.	U.B.
Zone 2 Temp	All-at-once Min Run Res. V CCD	0.002%	0.062%	3.871E-07	1.000	n/a	n/a
	All-at-once Optimal Latin Hypercube	-0.003%	0.081%	6.612E-07	0.586	0.553	0.620
	Variable-Type Partitioning Scheme	0.000%	0.036%	1.314E-07	2.946	2.782	3.119
	Subsystem Partitioning Scheme	0.004%	0.054%	2.970E-07	1.304	1.231	1.380
	Mincut 2 Partitioning Scheme	0.004%	0.060%	3.637E-07	1.064	1.005	1.127
	Mincut 3 Partitioning Scheme	-0.003%	0.052%	2.719E-07	1.424	1.344	1.508
	Mincut 4 Partitioning Scheme	0.010%	0.075%	5.642E-07	0.686	0.648	0.727
	Mincut 5 Partitioning Scheme	0.011%	0.199%	3.957E-06	0.098	0.092	0.104
	Mincut 6 Partitioning Scheme	0.014%	0.074%	5.482E-07	0.706	0.667	0.748
	Mincut 7 Partitioning Scheme	0.011%	0.076%	5.844E-07	0.662	0.626	0.701

Response	Analysis Type	Mean	Standard Deviation	Variance	Variance Ratio	95% Confidence	
						L.B.	U.B.
Zone 5 Temp	All-at-once Min Run Res. V CCD	0.004%	0.063%	3.977E-07	1.000	n/a	n/a
	All-at-once Optimal Latin Hypercube	-0.001%	0.082%	6.730E-07	0.591	0.558	0.626
	Variable-Type Partitioning Scheme	0.001%	0.023%	5.146E-08	7.728	7.298	8.184
	Subsystem Partitioning Scheme	0.000%	0.050%	2.504E-07	1.588	1.500	1.682
	Mincut 2 Partitioning Scheme	0.000%	0.059%	3.460E-07	1.149	1.085	1.217
	Mincut 3 Partitioning Scheme	0.002%	0.048%	2.309E-07	1.722	1.626	1.824
	Mincut 4 Partitioning Scheme	0.003%	0.052%	2.735E-07	1.454	1.373	1.540
	Mincut 5 Partitioning Scheme	-0.014%	0.107%	1.137E-06	0.350	0.330	0.370
	Mincut 6 Partitioning Scheme	0.000%	0.050%	2.504E-07	1.588	1.500	1.682
	Mincut 7 Partitioning Scheme	0.000%	0.050%	2.511E-07	1.584	1.496	1.678

Table 13: Zones 6, 7 and 8 Temperature Response Error Summary

Response	Analysis Type	Mean	Standard Deviation	Variance	Variance Ratio	95% Confidence	
						L.B.	U.B.
Zone 6 Temp	All-at-once Min Run Res. V CCD	0.000%	0.048%	2.266E-07	1.000	n/a	n/a
	All-at-once Optimal Latin Hypercube	0.000%	0.053%	2.852E-07	0.795	0.750	0.842
	Variable-Type Partitioning Scheme	0.002%	0.012%	1.554E-08	14.580	13.768	15.440
	Subsystem Partitioning Scheme	0.002%	0.034%	1.144E-07	1.982	1.871	2.099
	Mincut 2 Partitioning Scheme	0.000%	0.040%	1.602E-07	1.414	1.336	1.498
	Mincut 3 Partitioning Scheme	-0.002%	0.035%	1.253E-07	1.809	1.708	1.916
	Mincut 4 Partitioning Scheme	0.006%	0.033%	1.079E-07	2.100	1.983	2.224
	Mincut 5 Partitioning Scheme	0.005%	0.047%	2.166E-07	1.046	0.988	1.108
	Mincut 6 Partitioning Scheme	0.002%	0.034%	1.144E-07	1.982	1.871	2.099
	Mincut 7 Partitioning Scheme	0.006%	0.047%	2.236E-07	1.014	0.957	1.073

Response	Analysis Type	Mean	Standard Deviation	Variance	Variance Ratio	95% Confidence	
						L.B.	U.B.
Zone 7 Temp	All-at-once Min Run Res. V CCD	0.002%	0.067%	4.472E-07	1.000	n/a	n/a
	All-at-once Optimal Latin Hypercube	0.002%	0.104%	1.080E-06	0.414	0.391	0.438
	Variable-Type Partitioning Scheme	0.000%	0.023%	5.153E-08	8.678	8.195	9.190
	Subsystem Partitioning Scheme	0.005%	0.050%	2.467E-07	1.813	1.712	1.919
	Mincut 2 Partitioning Scheme	0.006%	0.055%	2.997E-07	1.492	1.409	1.580
	Mincut 3 Partitioning Scheme	0.000%	0.049%	2.432E-07	1.839	1.737	1.947
	Mincut 4 Partitioning Scheme	0.017%	0.208%	4.325E-06	0.103	0.098	0.109
	Mincut 5 Partitioning Scheme	0.018%	0.208%	4.332E-06	0.103	0.097	0.109
	Mincut 6 Partitioning Scheme	-0.001%	0.103%	1.055E-06	0.424	0.400	0.449
	Mincut 7 Partitioning Scheme	0.005%	0.050%	2.515E-07	1.778	1.679	1.883

Response	Analysis Type	Mean	Standard Deviation	Variance	Variance Ratio	95% Confidence	
						L.B.	U.B.
Zone 8 Temp	All-at-once Min Run Res. V CCD	0.000%	0.042%	1.748E-07	1.000	n/a	n/a
	All-at-once Optimal Latin Hypercube	-0.003%	0.060%	3.623E-07	0.482	0.456	0.511
	Variable-Type Partitioning Scheme	-0.001%	0.012%	1.389E-08	12.587	11.886	13.329
	Subsystem Partitioning Scheme	-0.001%	0.034%	1.154E-07	1.514	1.430	1.604
	Mincut 2 Partitioning Scheme	0.001%	0.044%	1.972E-07	0.887	0.837	0.939
	Mincut 3 Partitioning Scheme	-0.001%	0.041%	1.686E-07	1.037	0.979	1.098
	Mincut 4 Partitioning Scheme	-0.004%	0.035%	1.237E-07	1.413	1.334	1.496
	Mincut 5 Partitioning Scheme	0.003%	0.049%	2.401E-07	0.728	0.687	0.771
	Mincut 6 Partitioning Scheme	-0.001%	0.034%	1.154E-07	1.514	1.430	1.604
	Mincut 7 Partitioning Scheme	0.004%	0.047%	2.250E-07	0.777	0.734	0.823

A.2.2 Charts of Individual Metamodeling Scheme Error Distributions

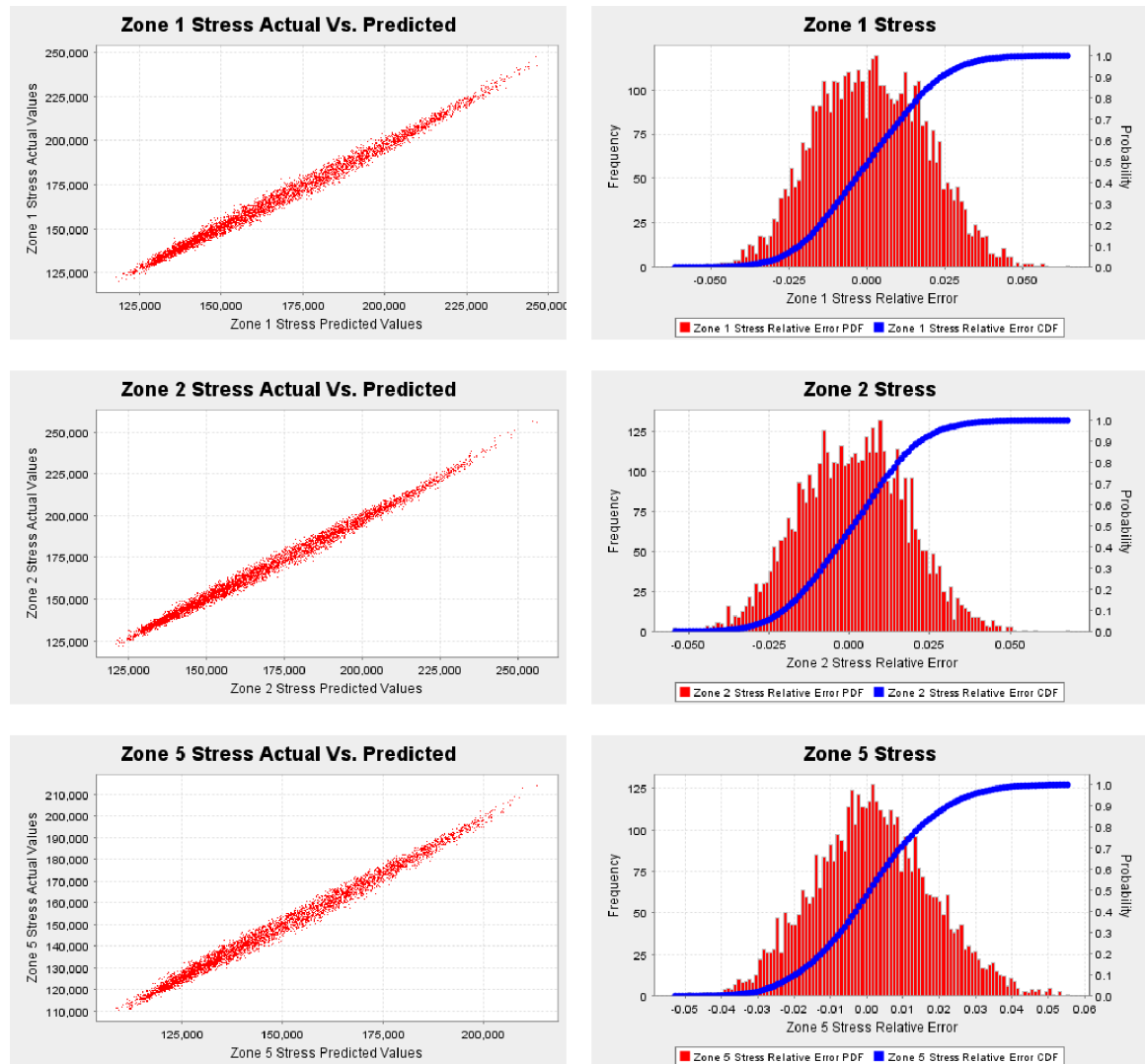


Figure 42: All-at-once Min. Run Res. V CCD Error Distributions - Stresses Zone 1, 2, 5

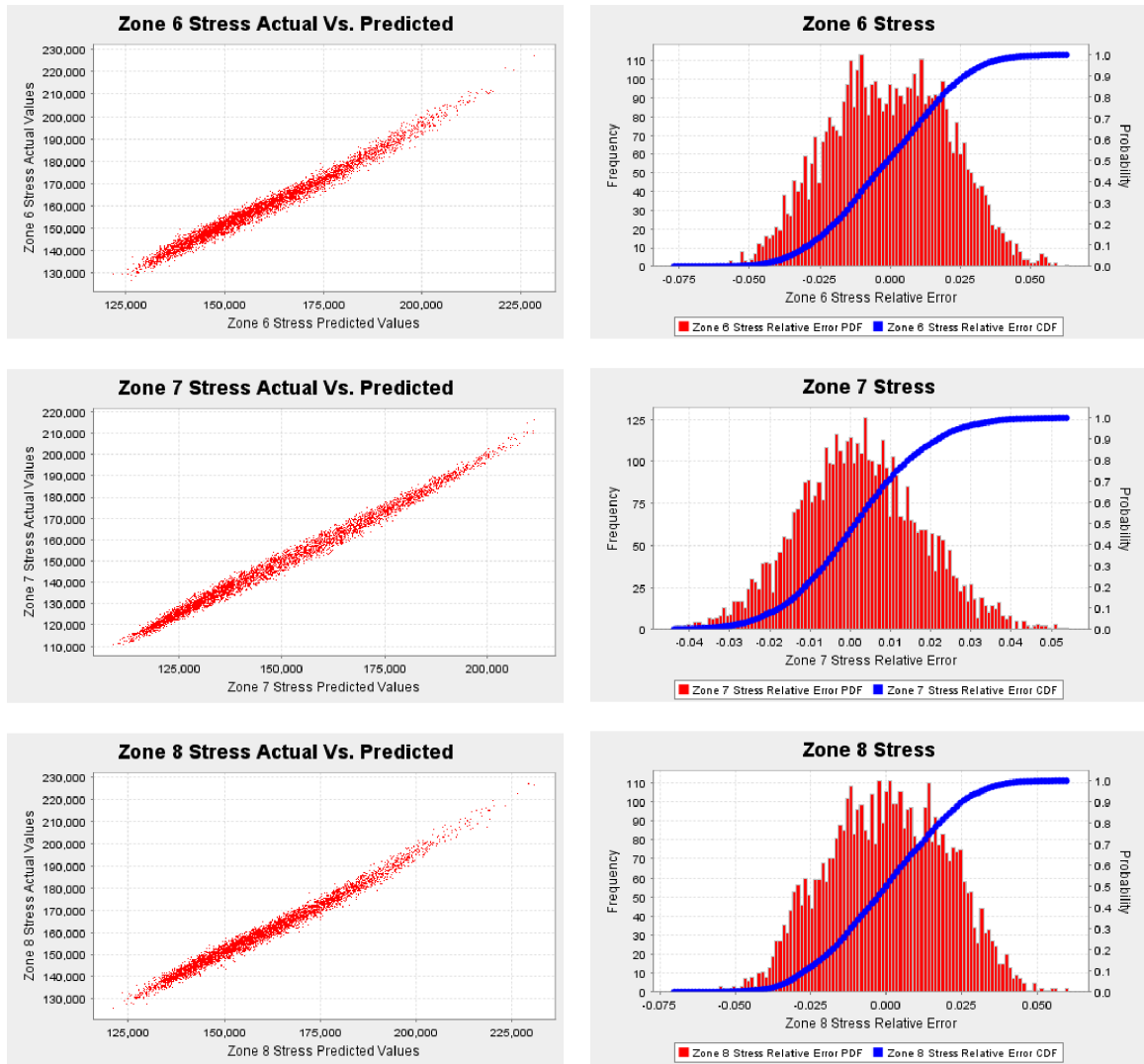


Figure 43: All-at-once Min. Run Res. V CCD Error Distributions - Stresses Zone 6, 7, 8

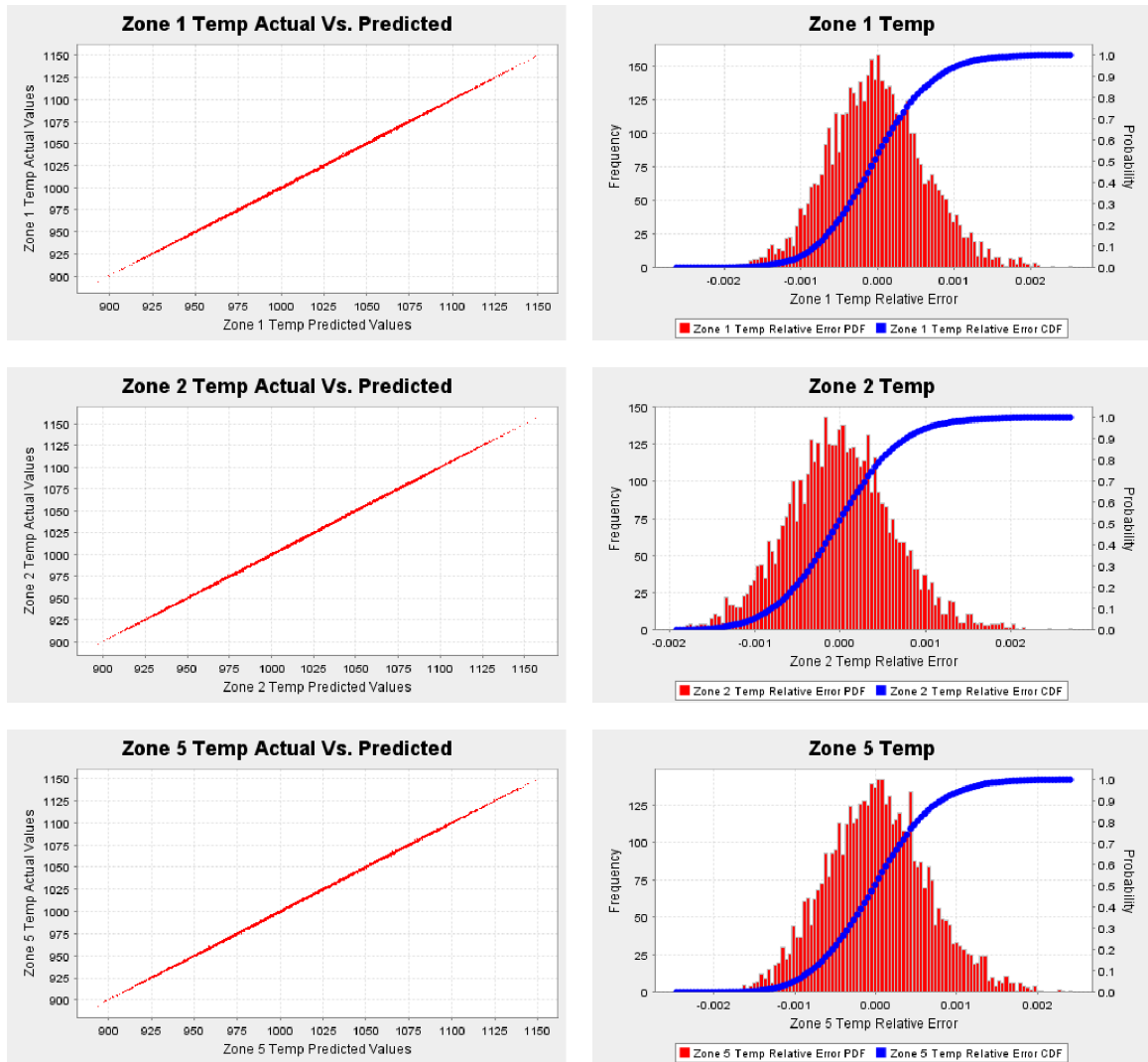


Figure 44: All-at-once Min. Run Res. V CCD Error Distributions - Temperatures
Zone 1, 2, 5

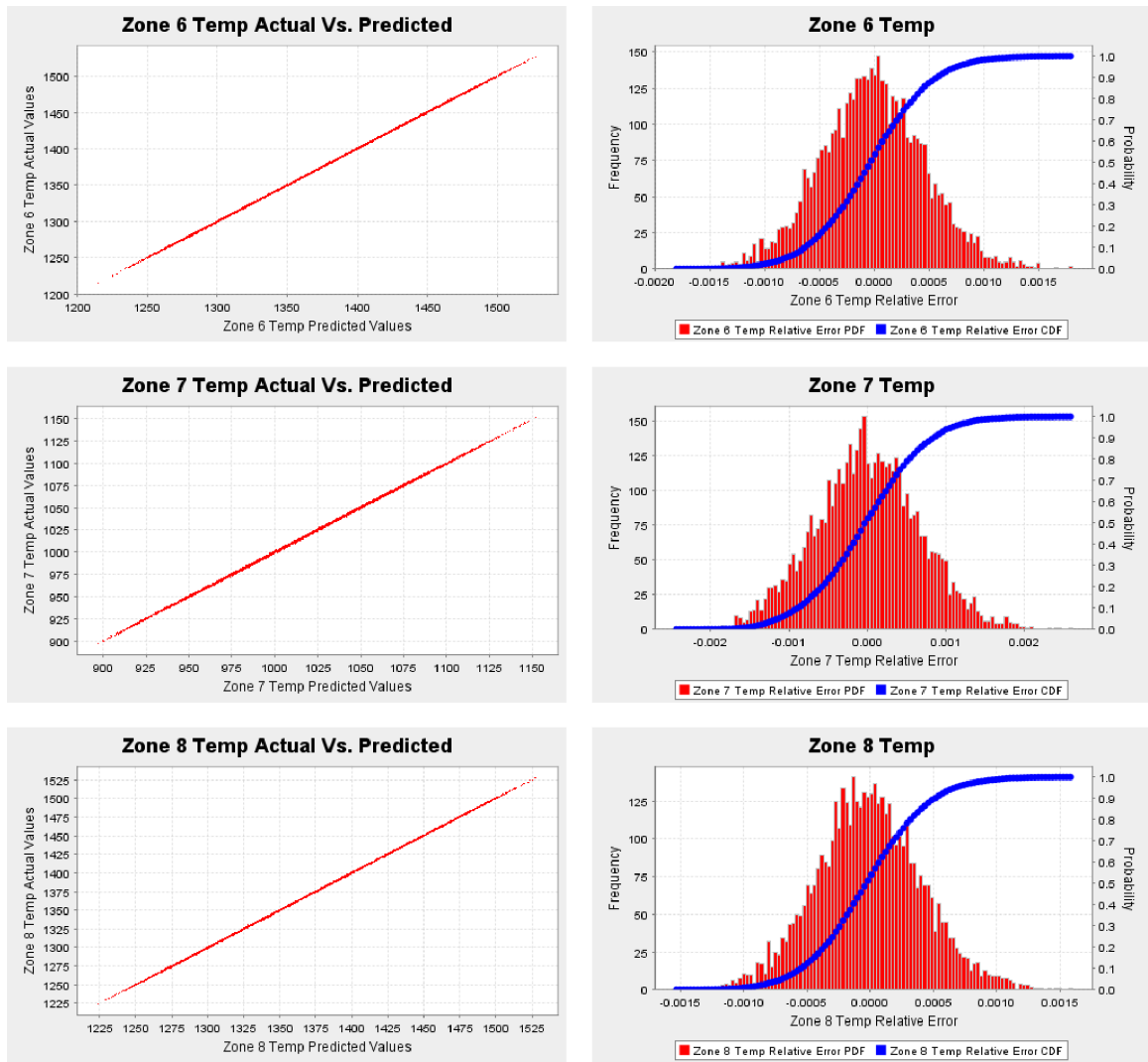


Figure 45: All-at-once Min. Run Res. V CCD Error Distributions - Temperatures
Zone 6, 7, 8

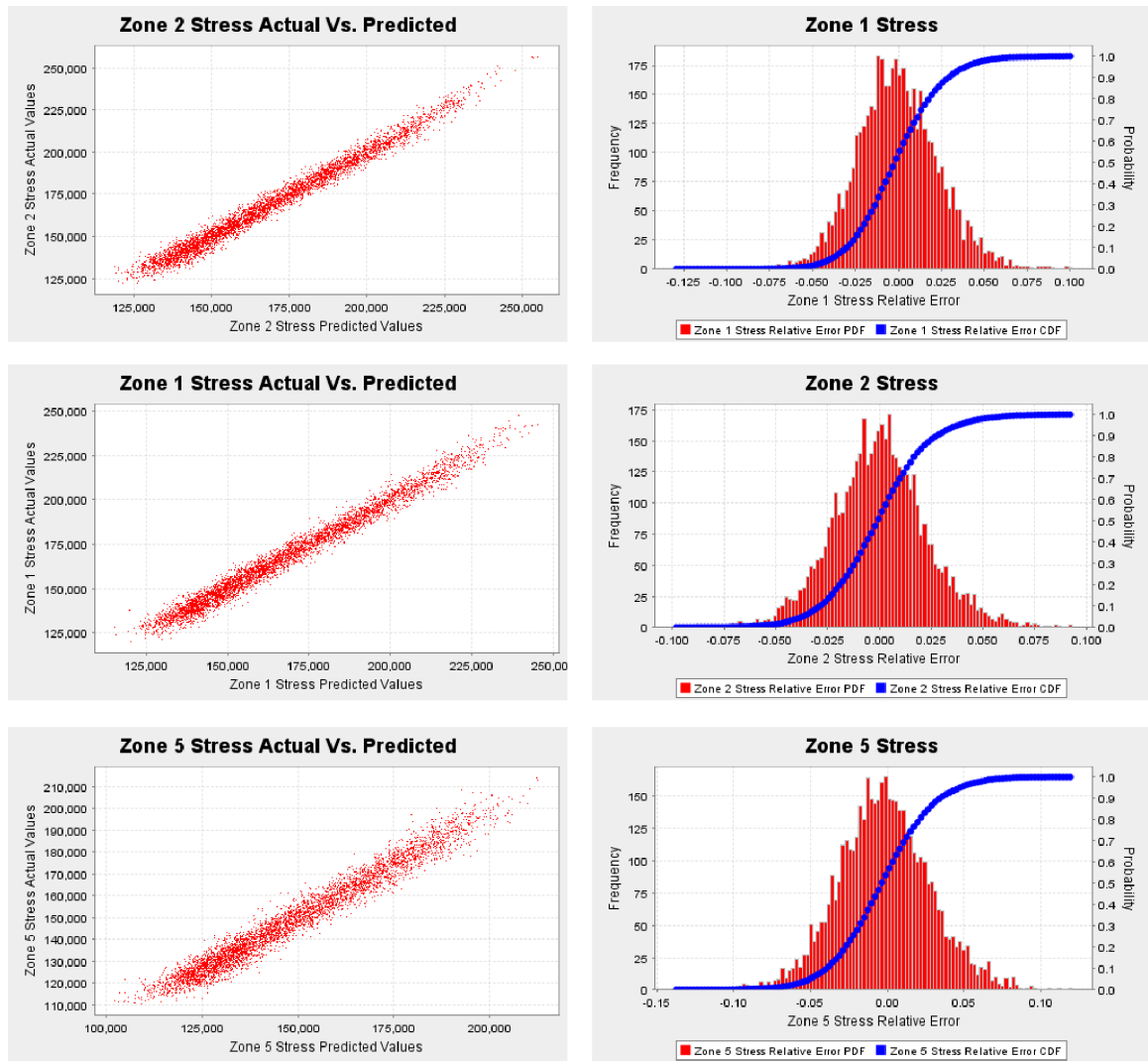


Figure 46: All-at-once Optimal Latin Hypercube Error Distributions - Stresses Zone 1, 2, 5

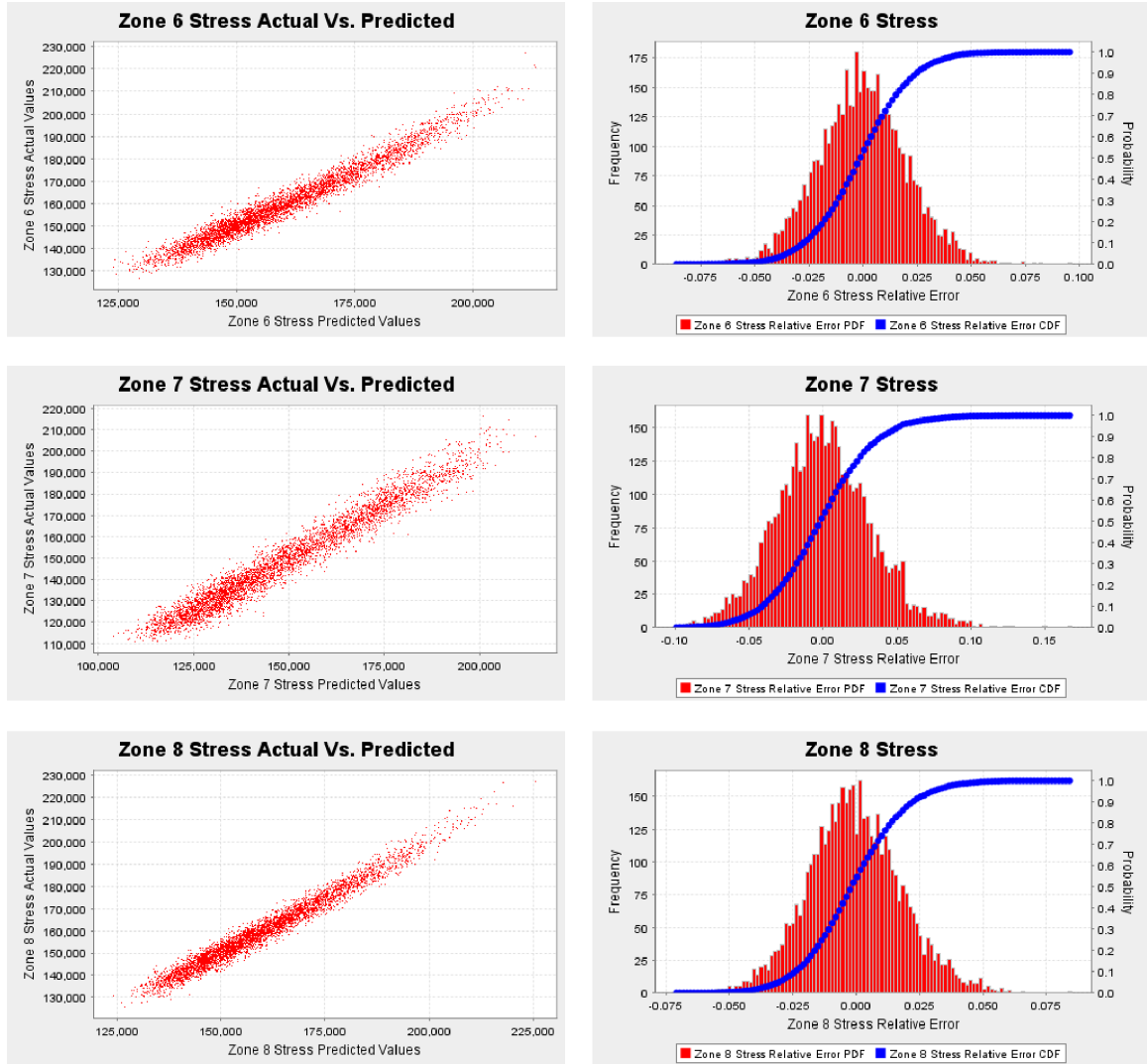


Figure 47: All-at-once Optimal Latin Hypercube Error Distributions - Stresses Zone 6, 7, 8

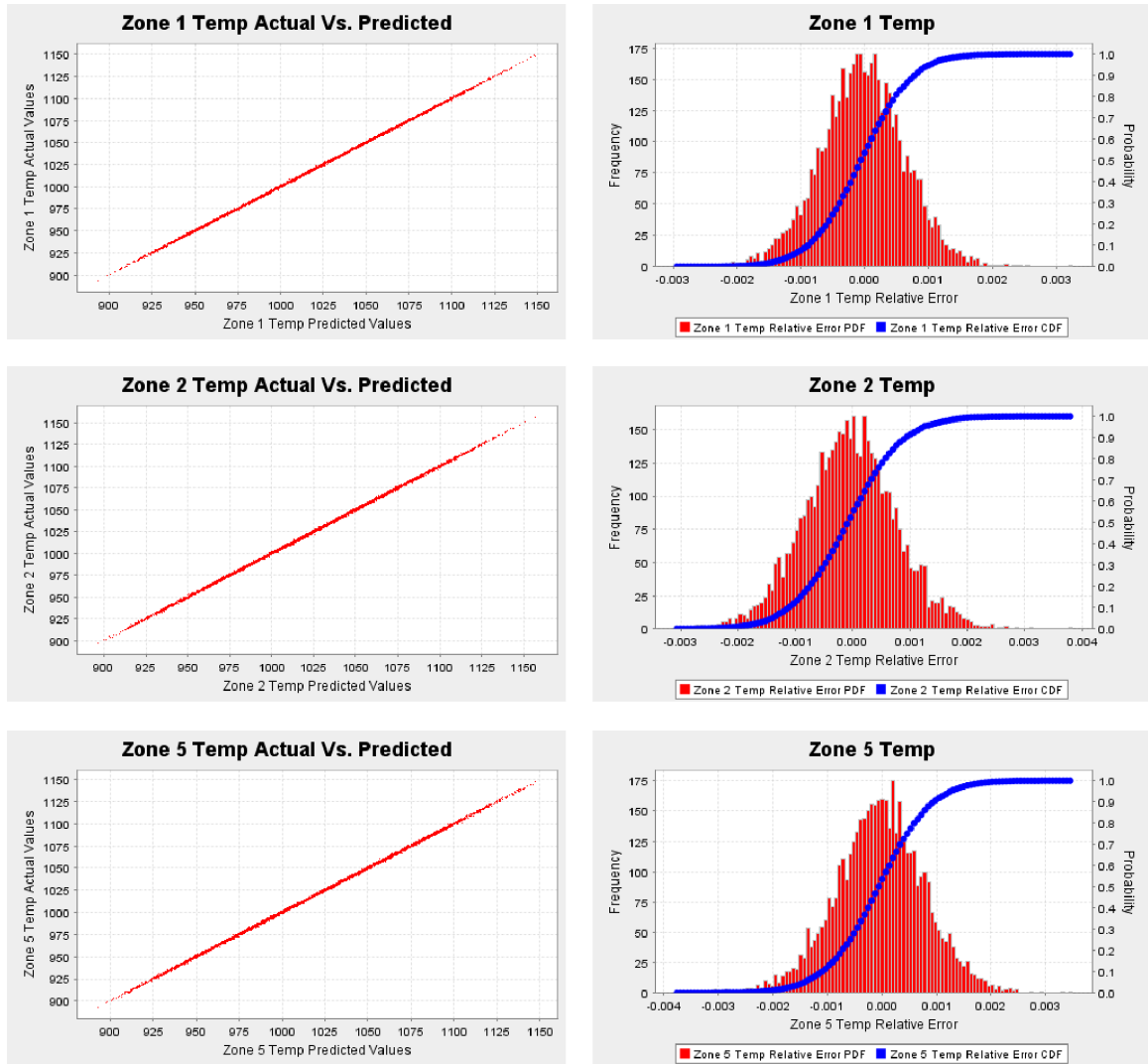


Figure 48: All-at-once Optimal Latin Hypercube Error Distributions - Temperatures
Zone 1, 2, 5

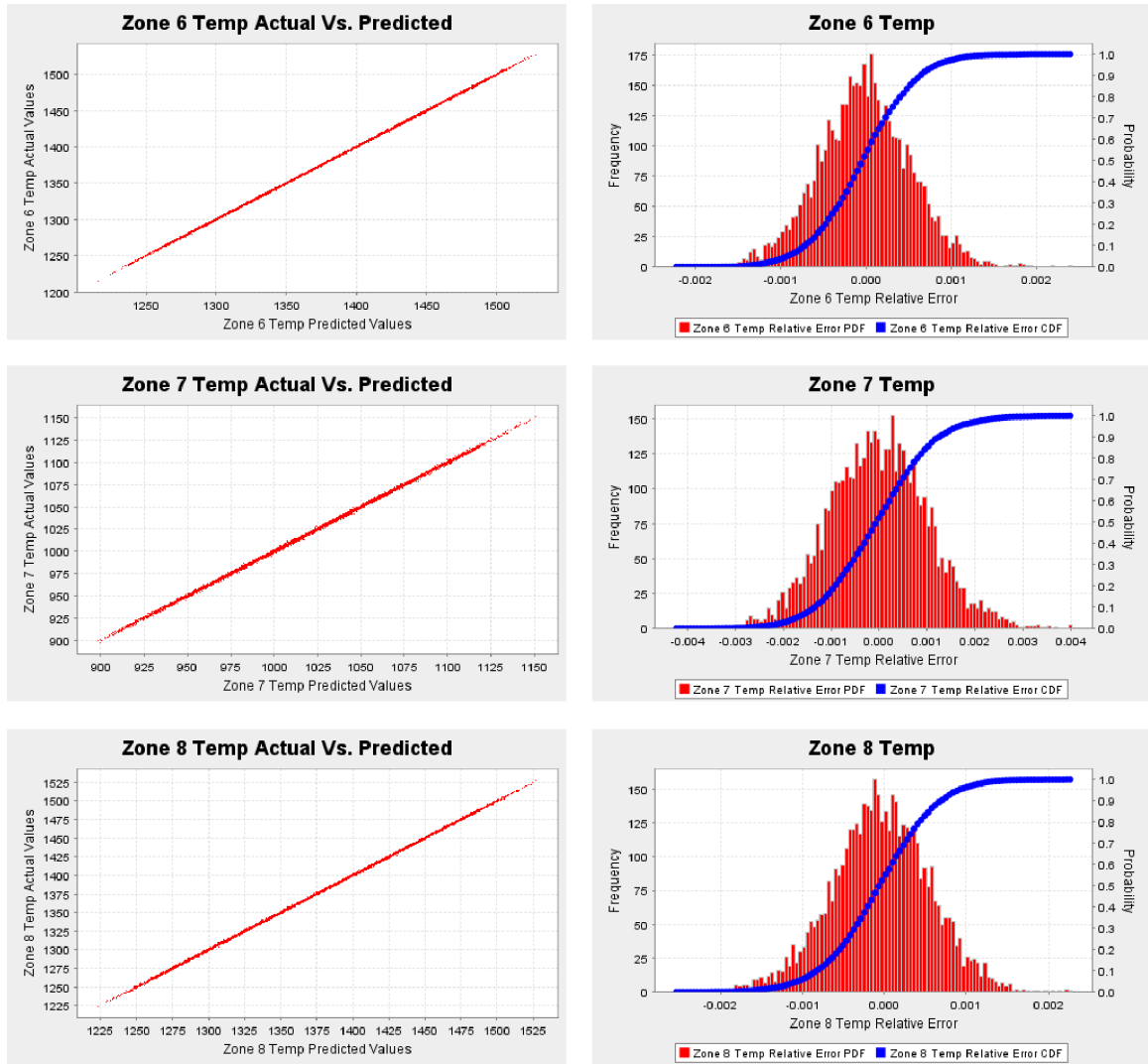


Figure 49: All-at-once Optimal Latin Hypercube Error Distributions - Temperatures
Zone 6, 7, 8

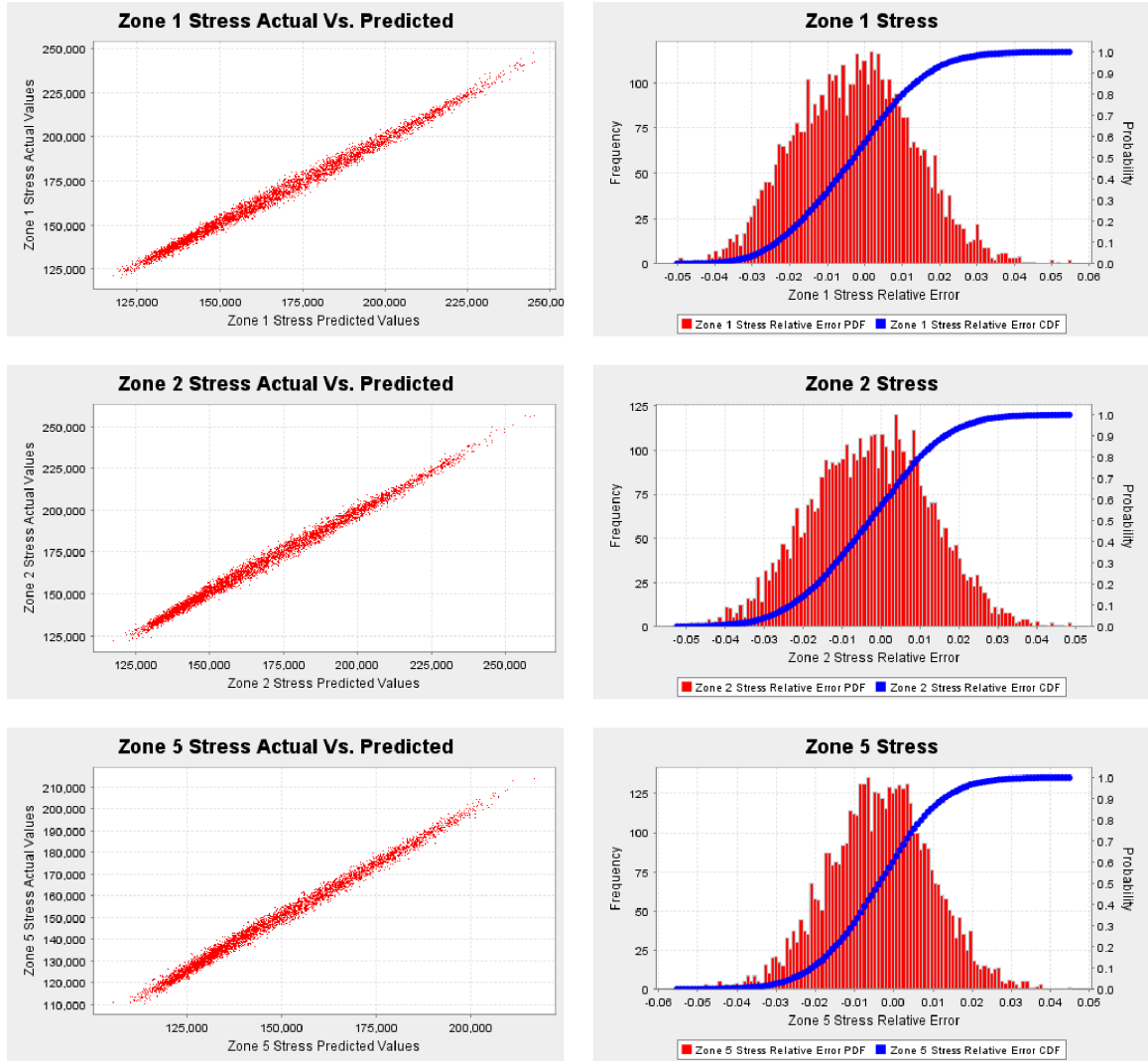


Figure 50: Variable-type Partitioning Error Distributions - Stresses Zone 1, 2, 5

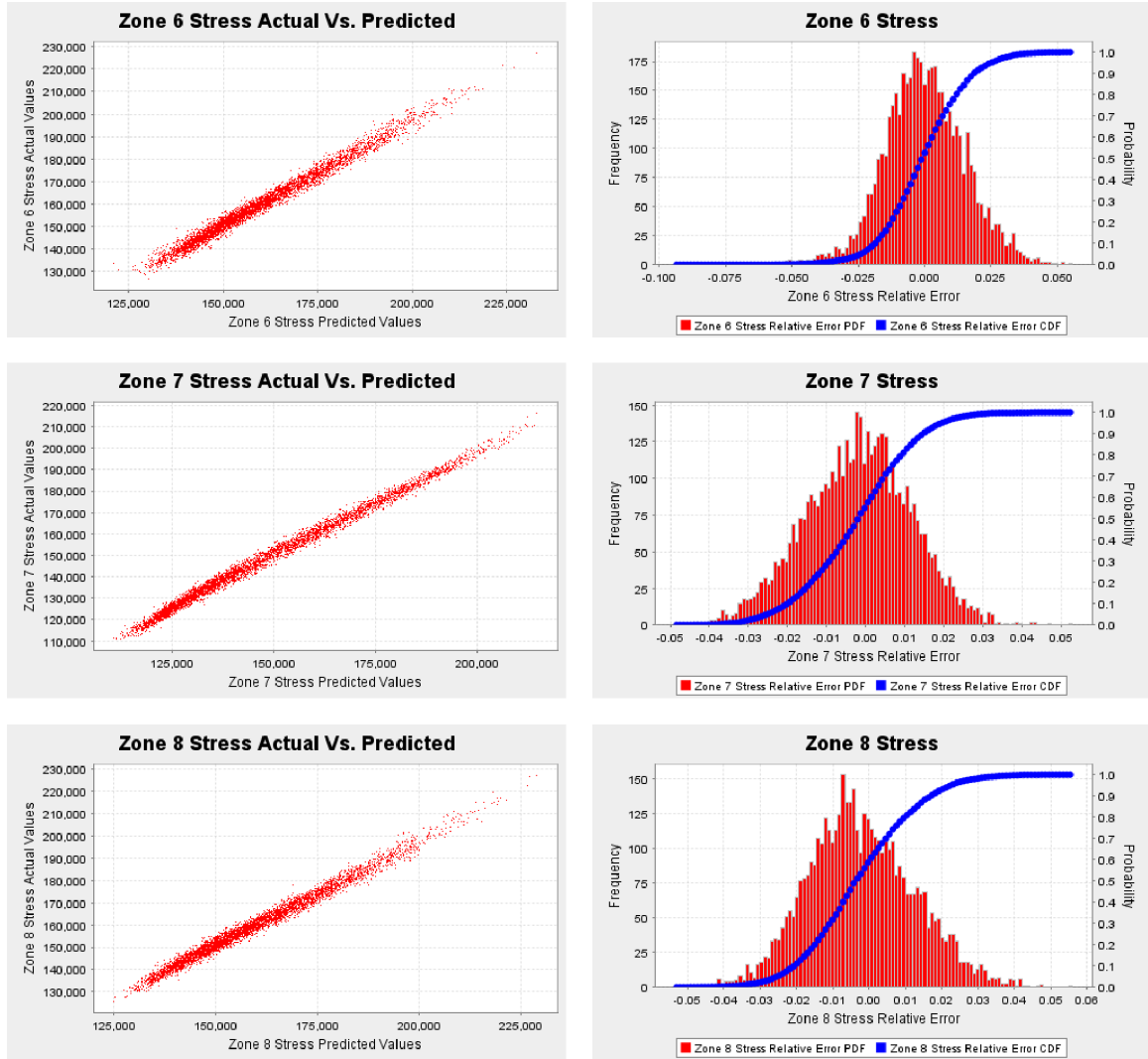


Figure 51: Variable-type Partitioning Error Distributions - Stresses Zone 6, 7, 8

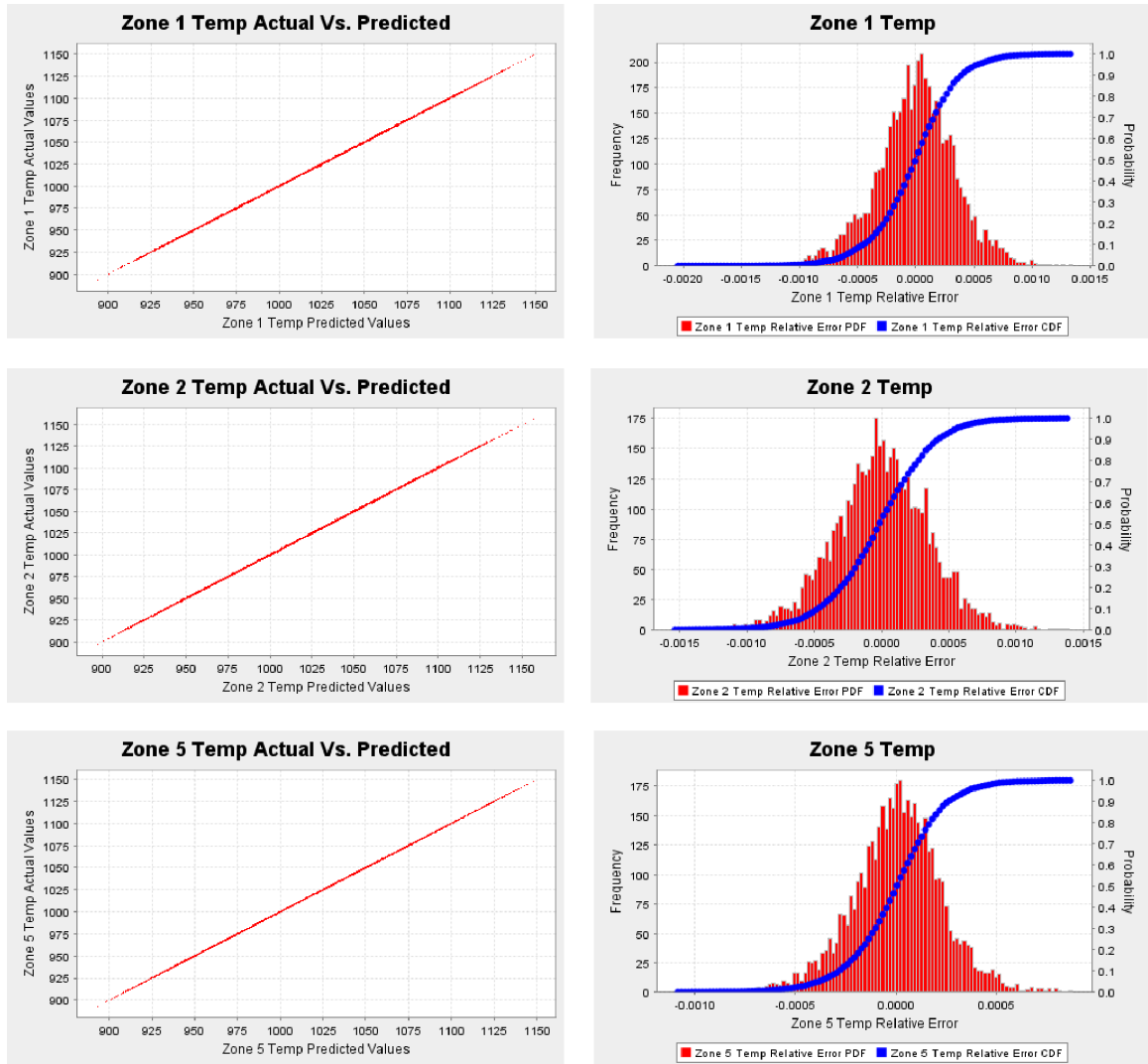


Figure 52: Variable-type Partitioning Error Distributions - Temperatures Zone 1, 2, 5

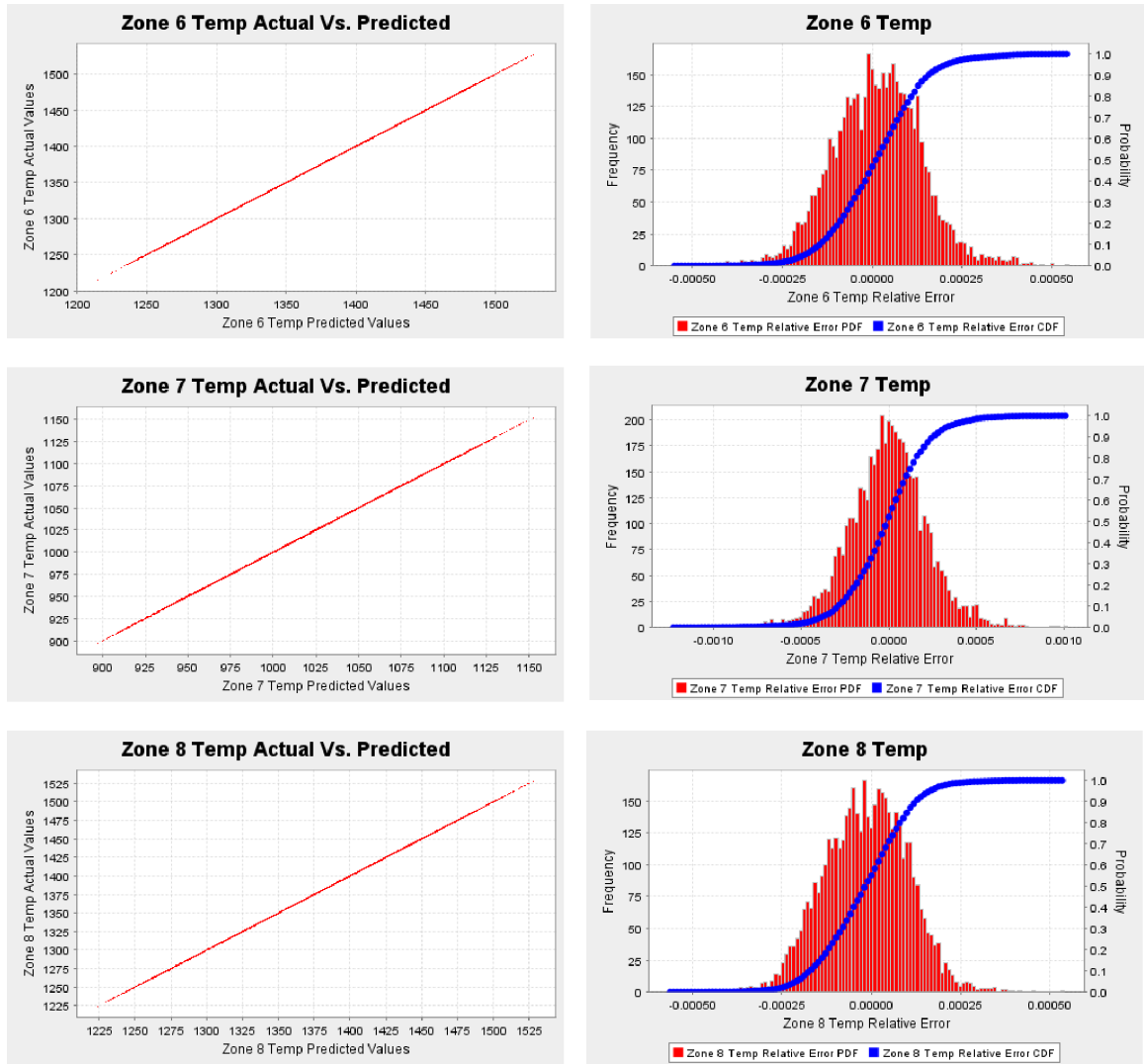


Figure 53: Variable-type Partitioning Error Distributions- Temperatures Zone 6, 7,

8

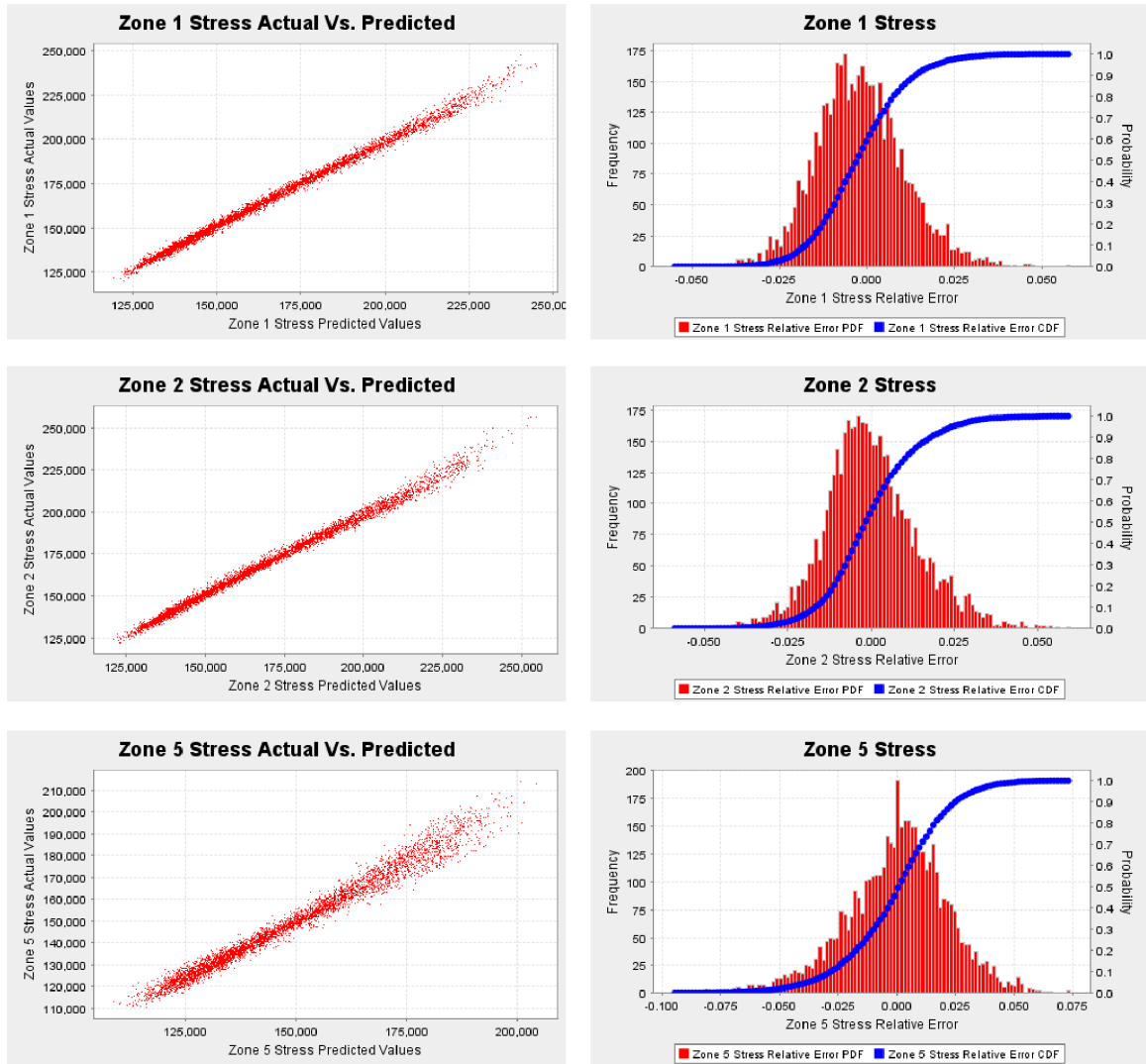


Figure 54: Subsystem Partitioning Error Distributions - Stresses Zone 1, 2, 5

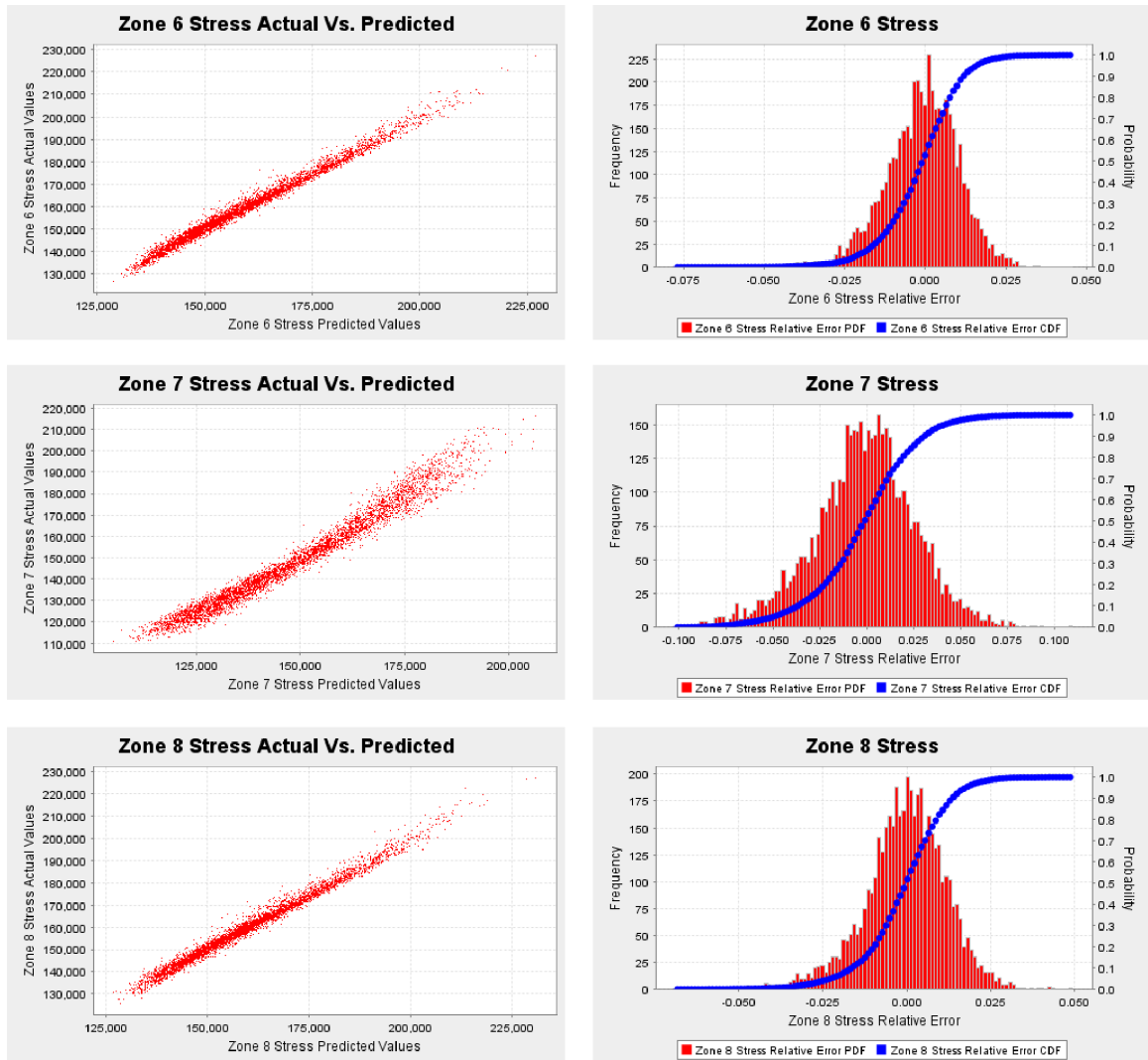


Figure 55: Subsystem Partitioning Error Distributions - Stresses Zone 6, 7, 8

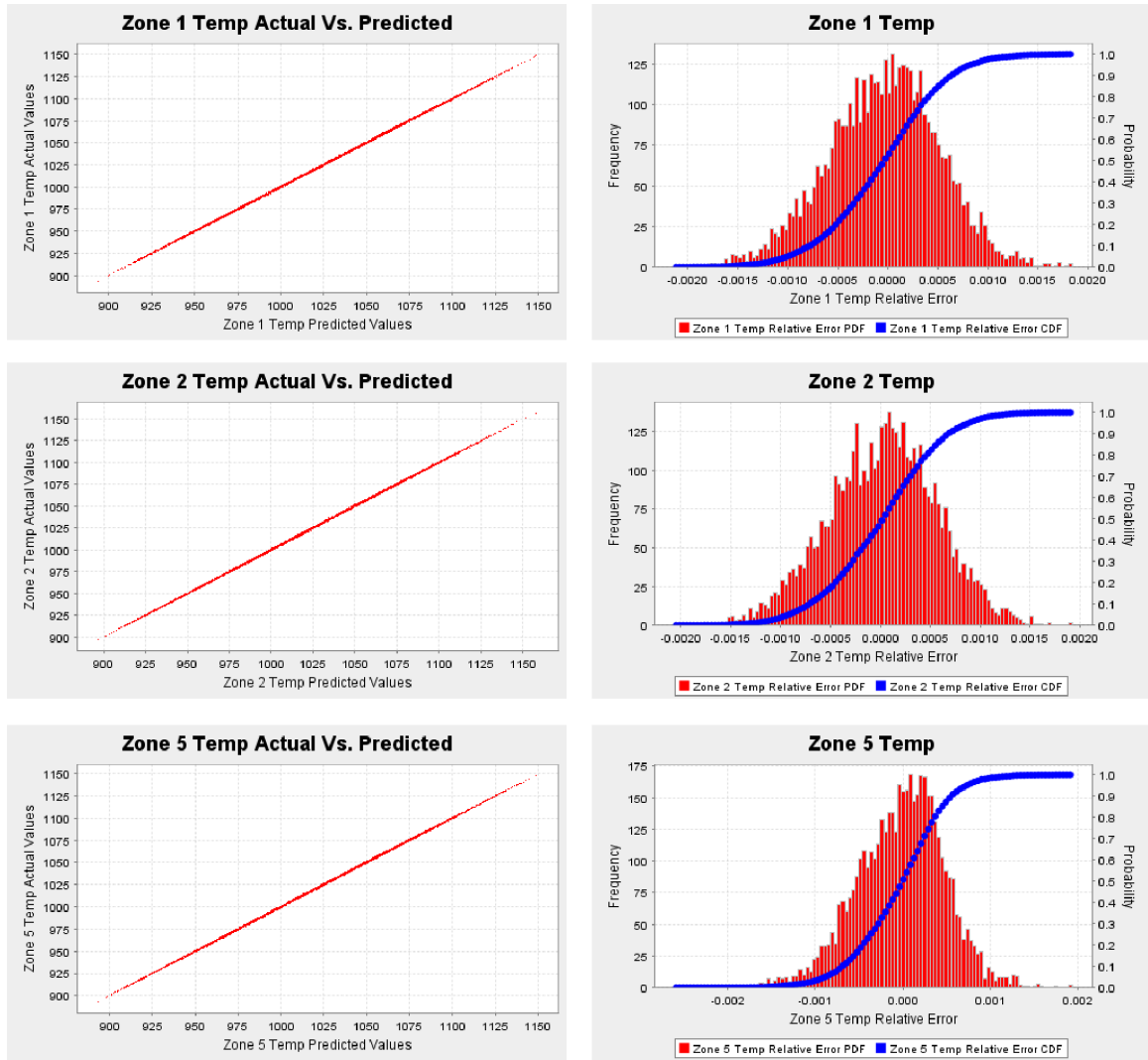


Figure 56: Subsystem Partitioning Error Distributions - Temperatures Zone 1, 2, 5

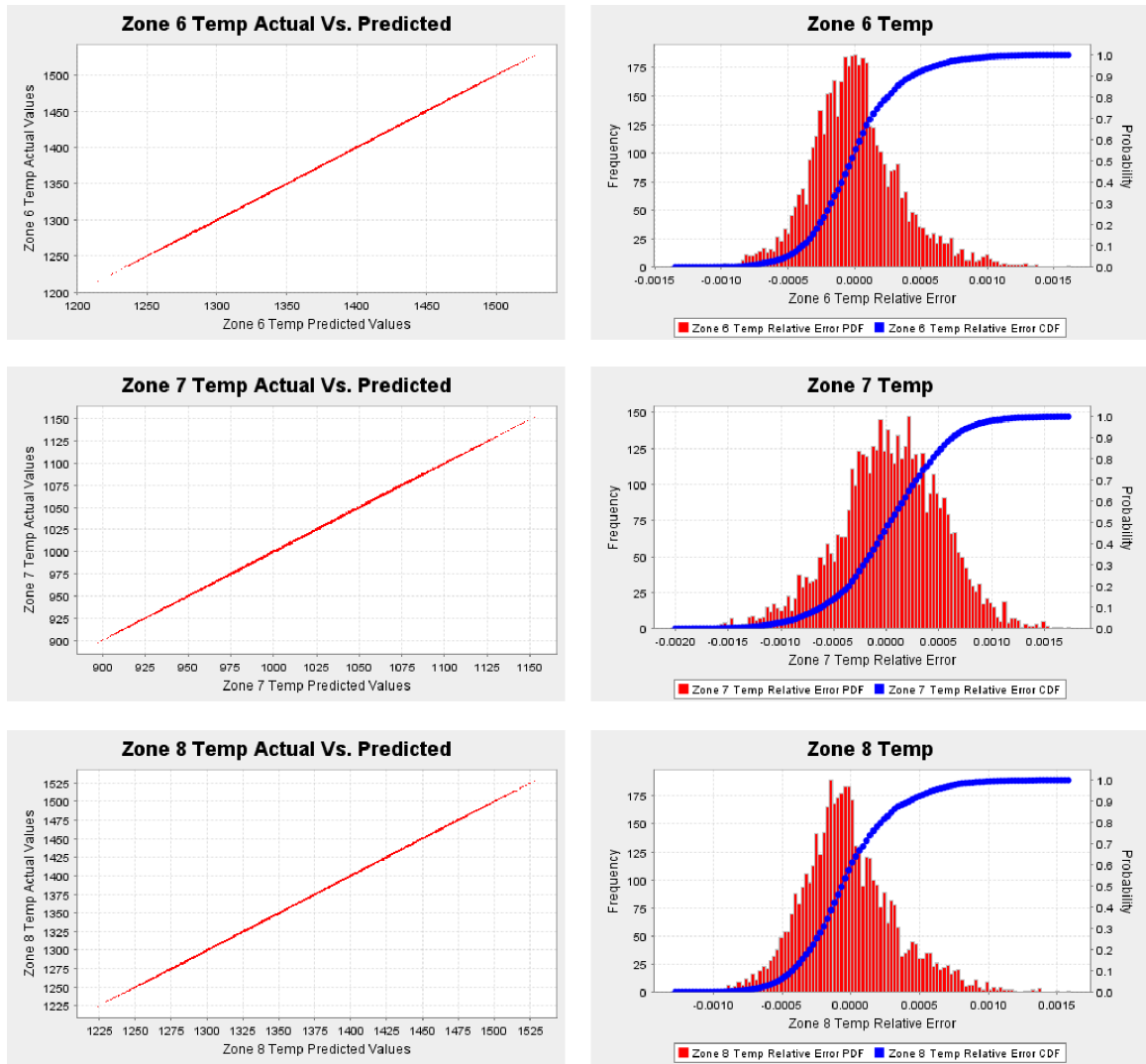


Figure 57: Subsystem Partitioning Error Distributions - Temperatures Zone 1, 2, 5

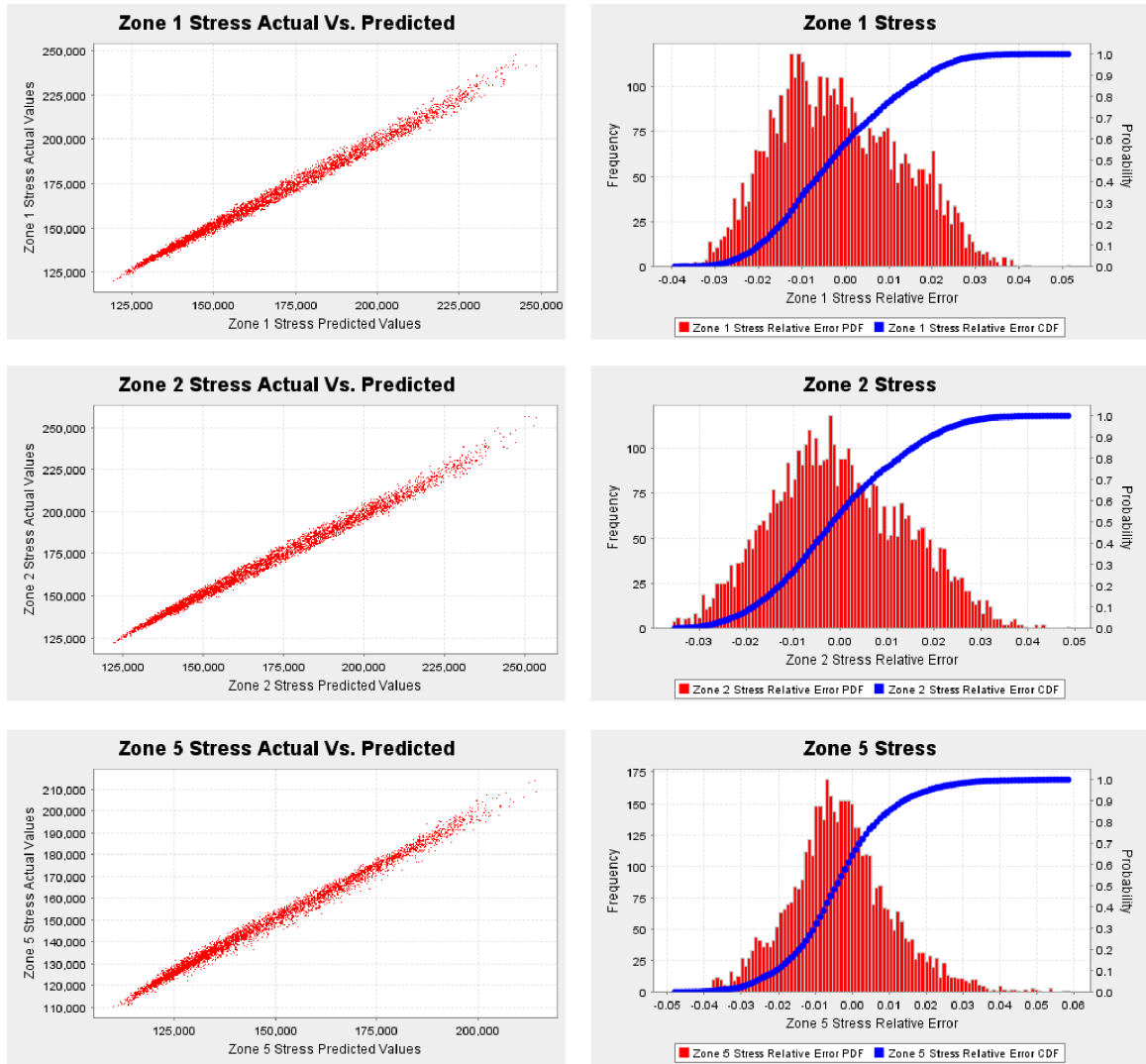


Figure 58: Mincut-2 Partitioning Error Distributions - Stresses Zone 1, 2, 5

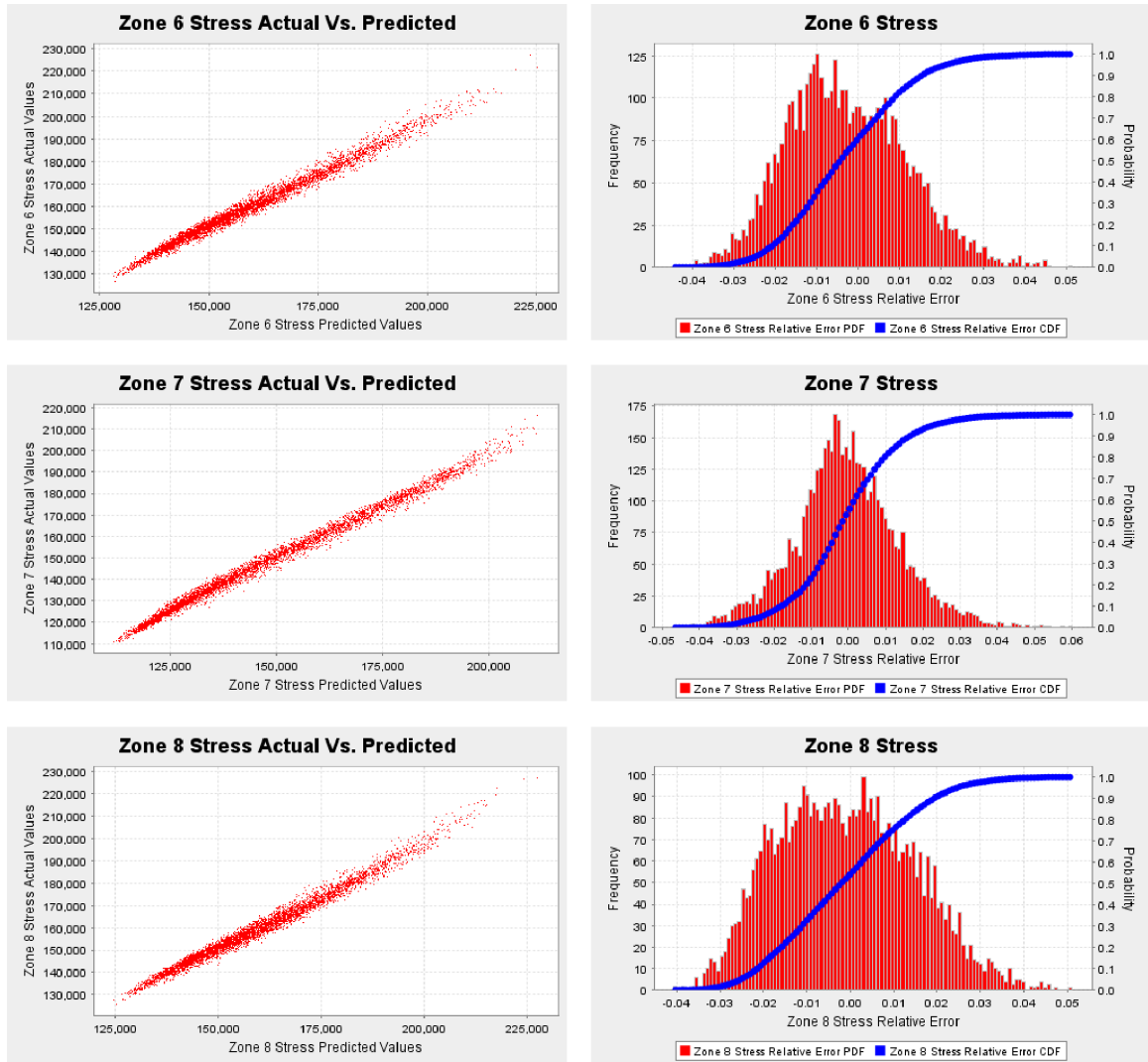


Figure 59: Mincut-2 Partitioning Error Distributions - Stresses Zone 6, 7, 8

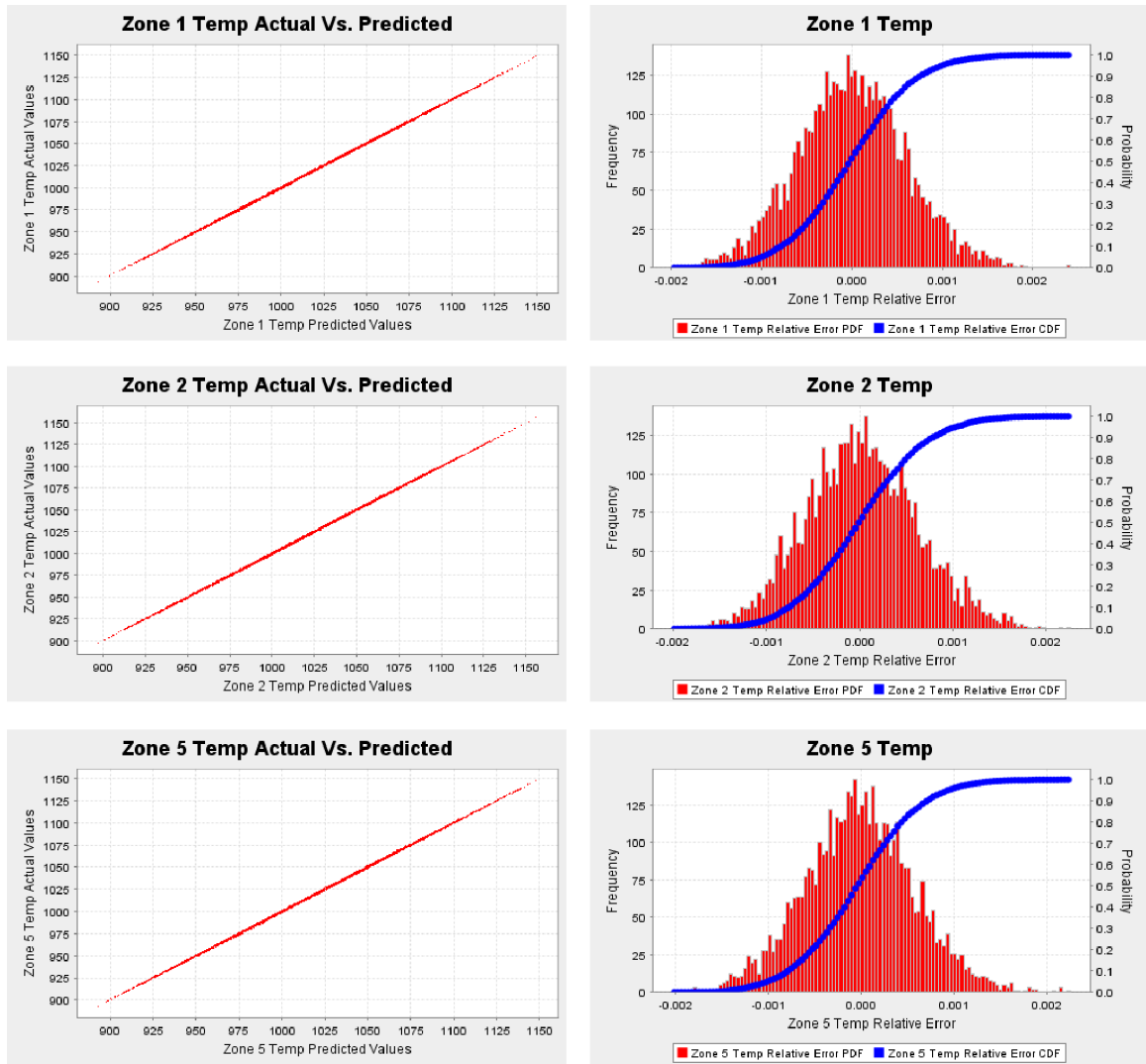


Figure 60: Mincut-2 Partitioning Error Distributions - Temperatures Zone 1, 2, 5

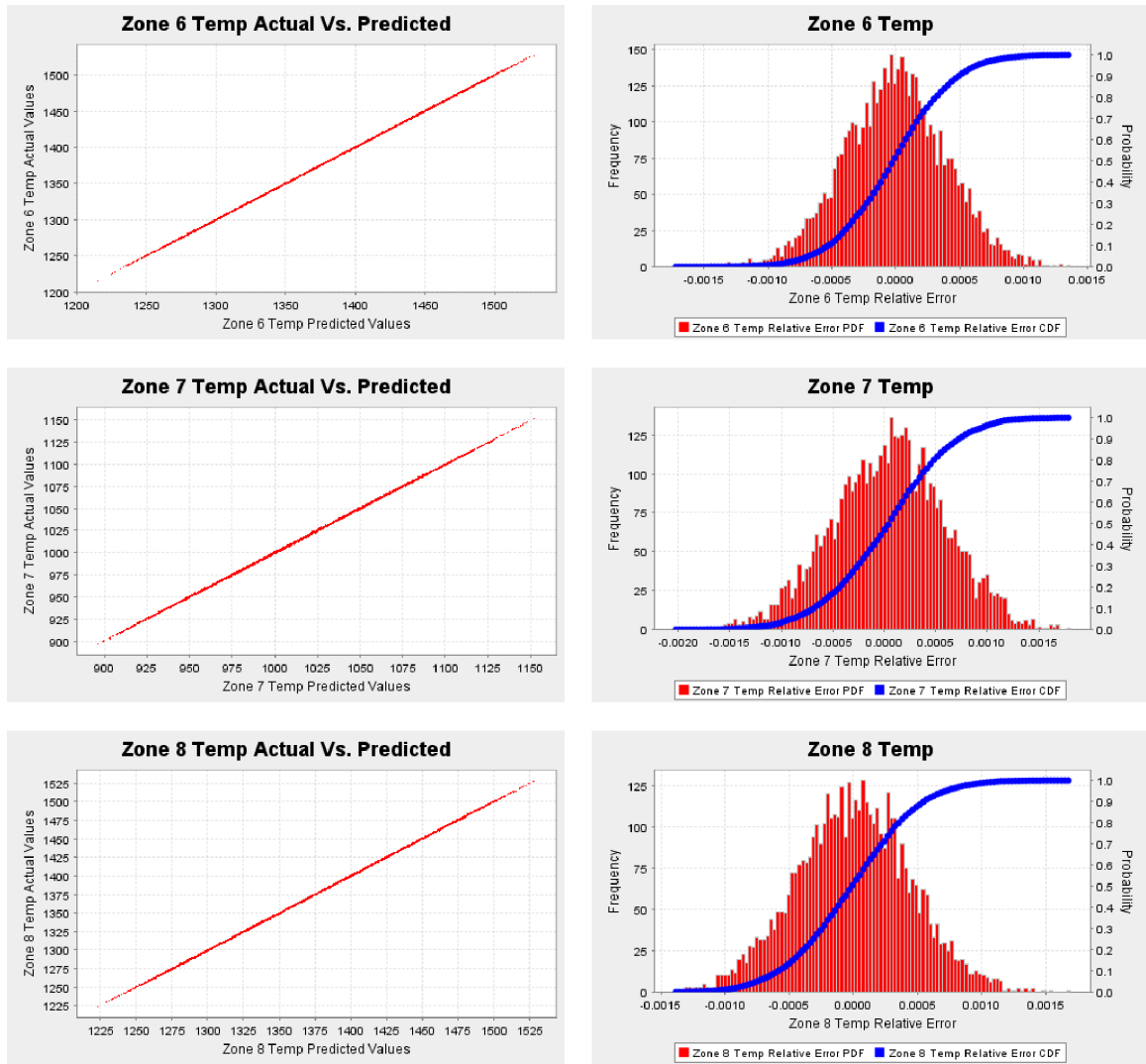


Figure 61: Mincut-2 Partitioning Error Distributions - Temperatures Zone 6, 7, 8

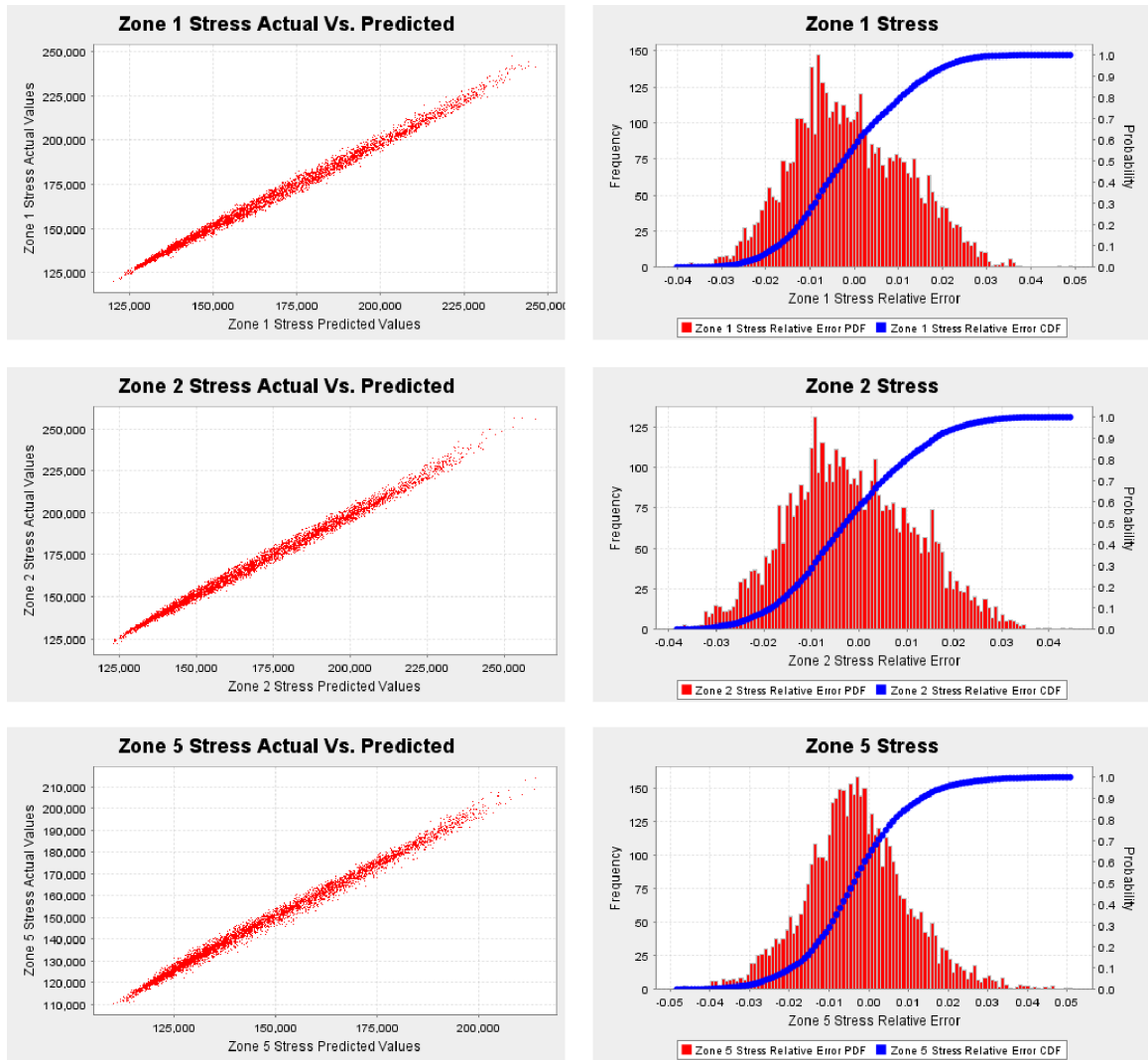


Figure 62: Mincut-3 Partitioning Error Distributions - Stresses Zone 1, 2, 5

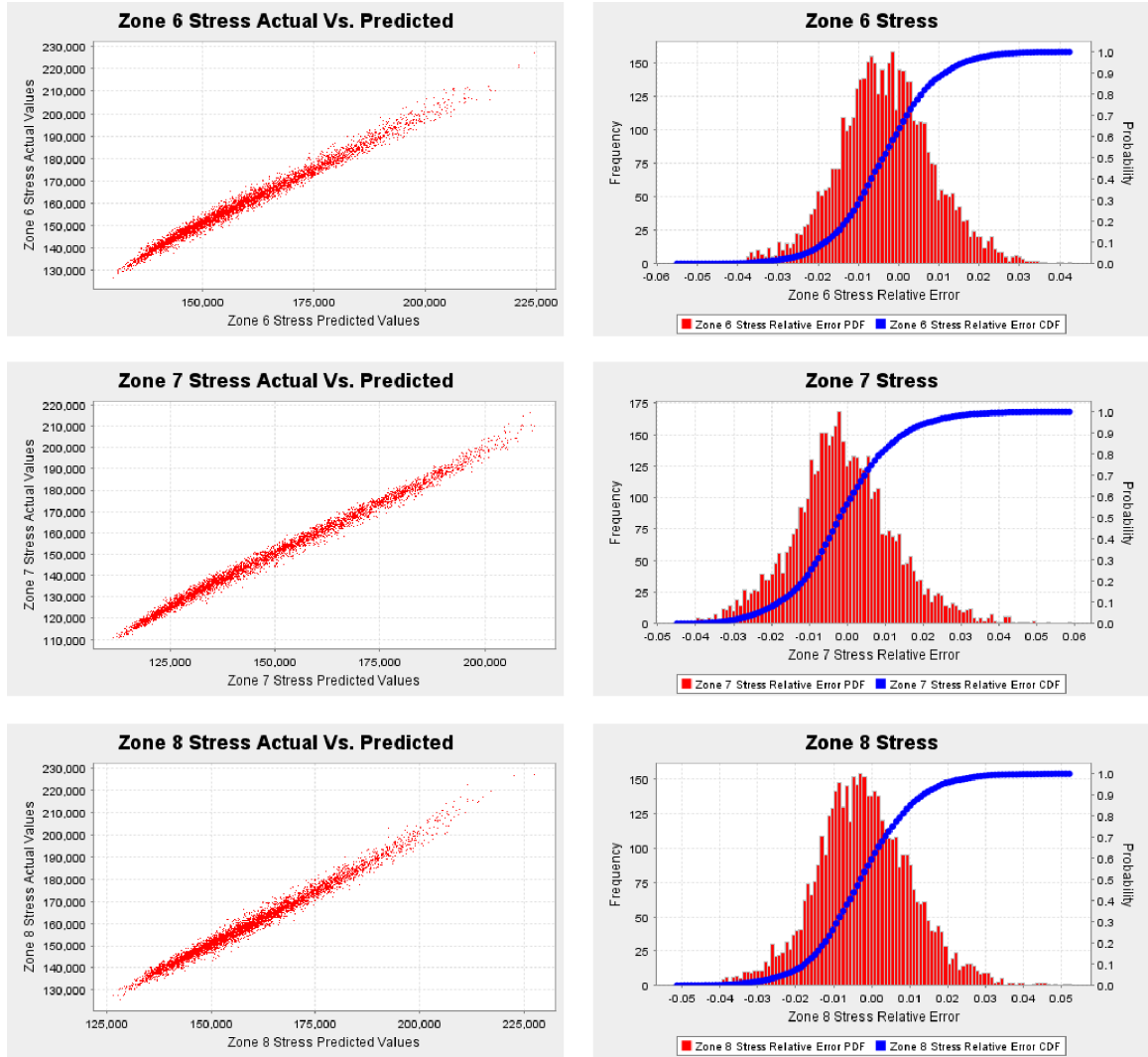


Figure 63: Mincut-3 Partitioning Error Distributions - Stresses Zone 6, 7, 8

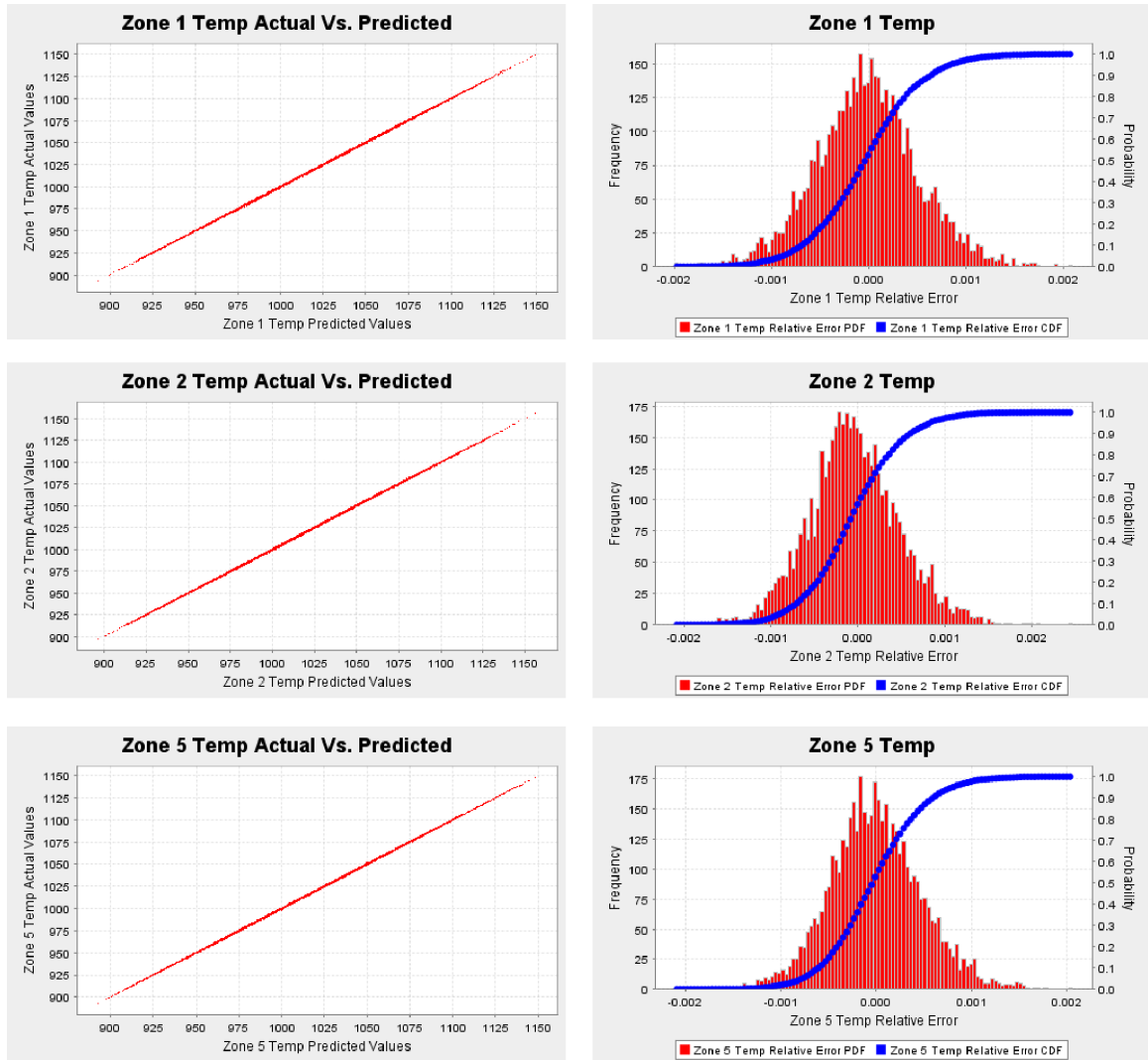


Figure 64: Mincut-3 Partitioning Error Distributions - Temperatures Zone 1, 2, 5

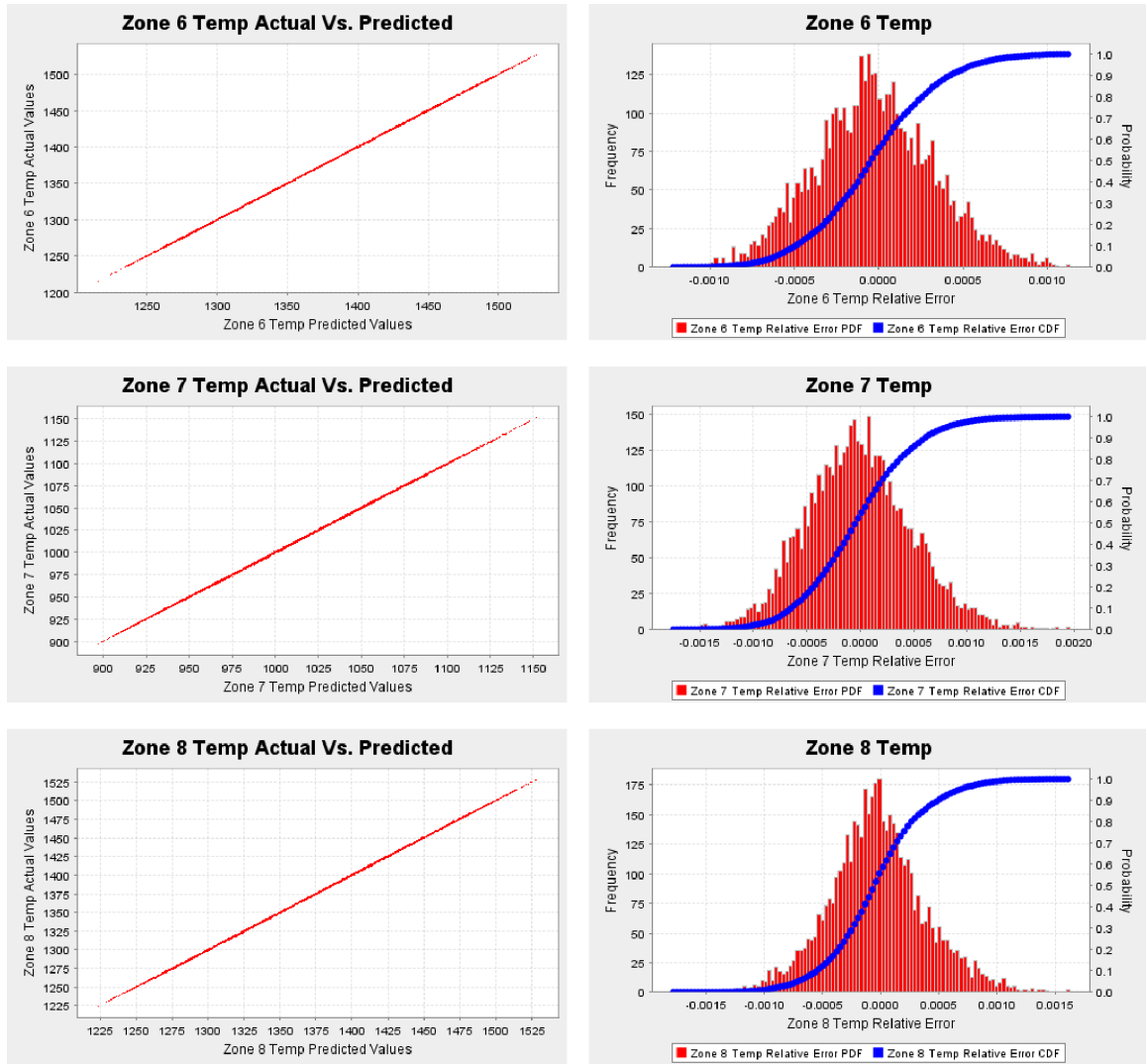


Figure 65: Mincut-3 Partitioning Error Distributions - Temperatures Zone 6, 7, 8

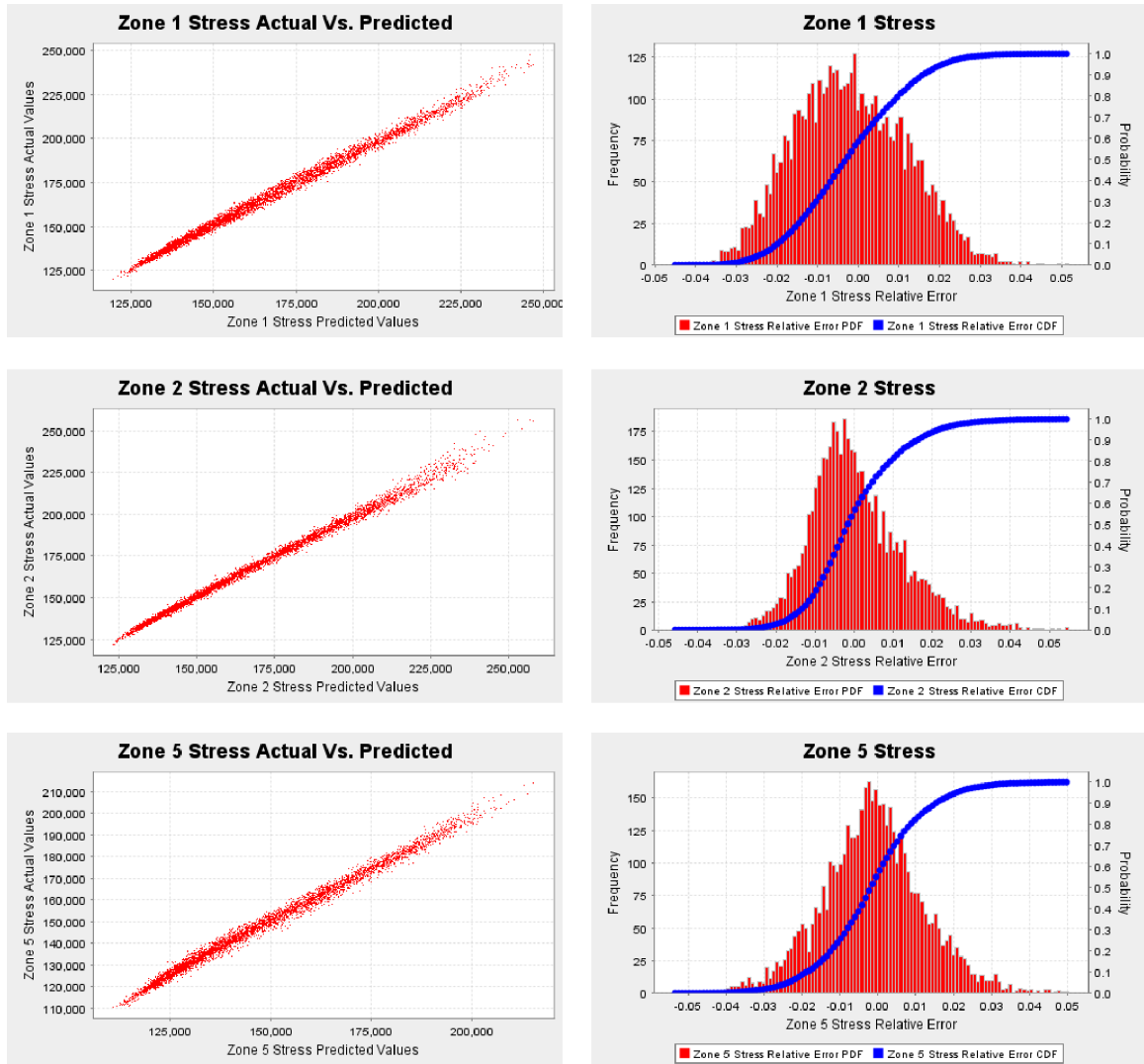


Figure 66: Mincut-4 Partitioning Error Distributions - Stresses Zone 1, 2, 5

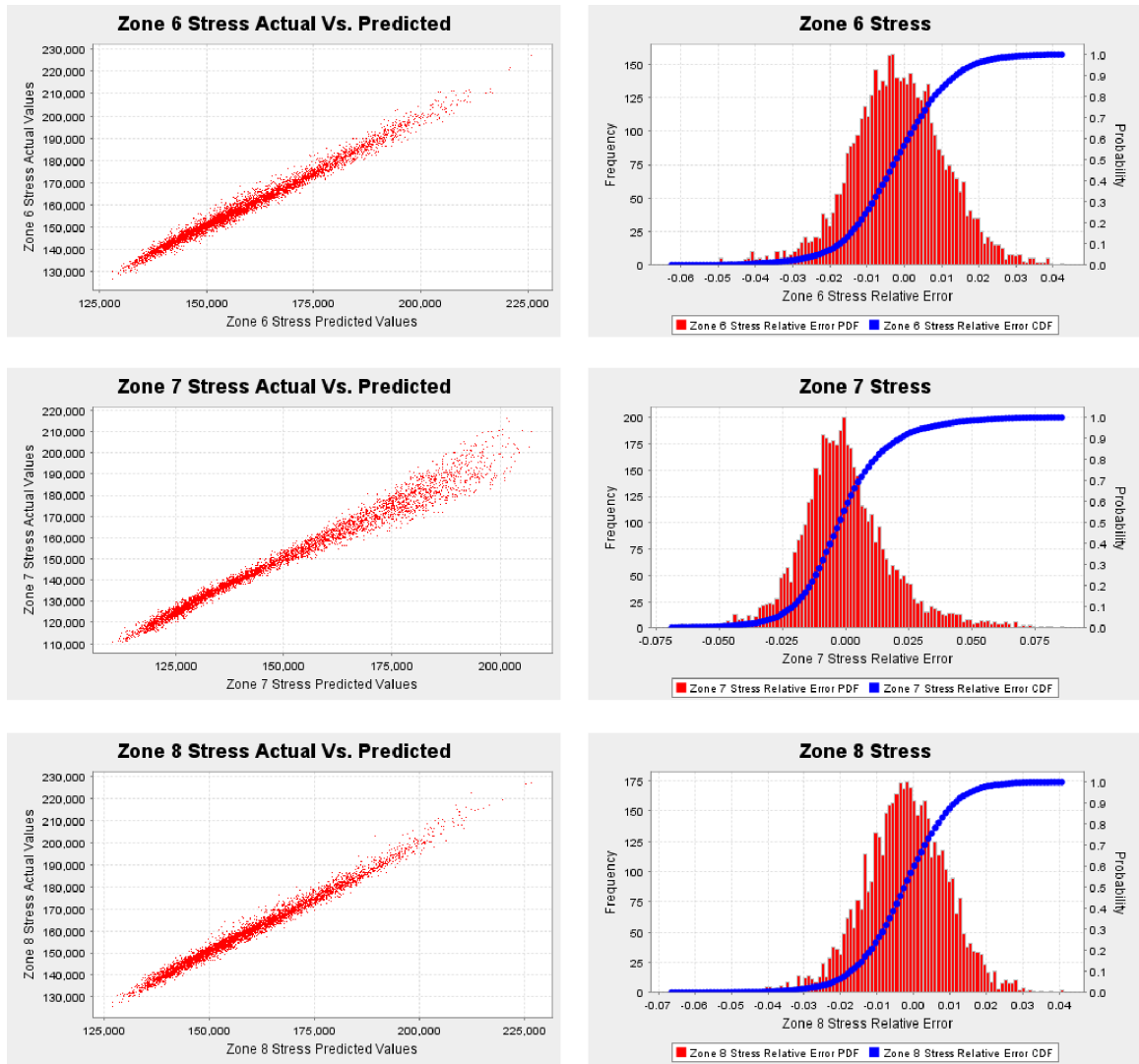


Figure 67: Mincut-4 Partitioning Error Distributions - Stresses Zone 6, 7, 8

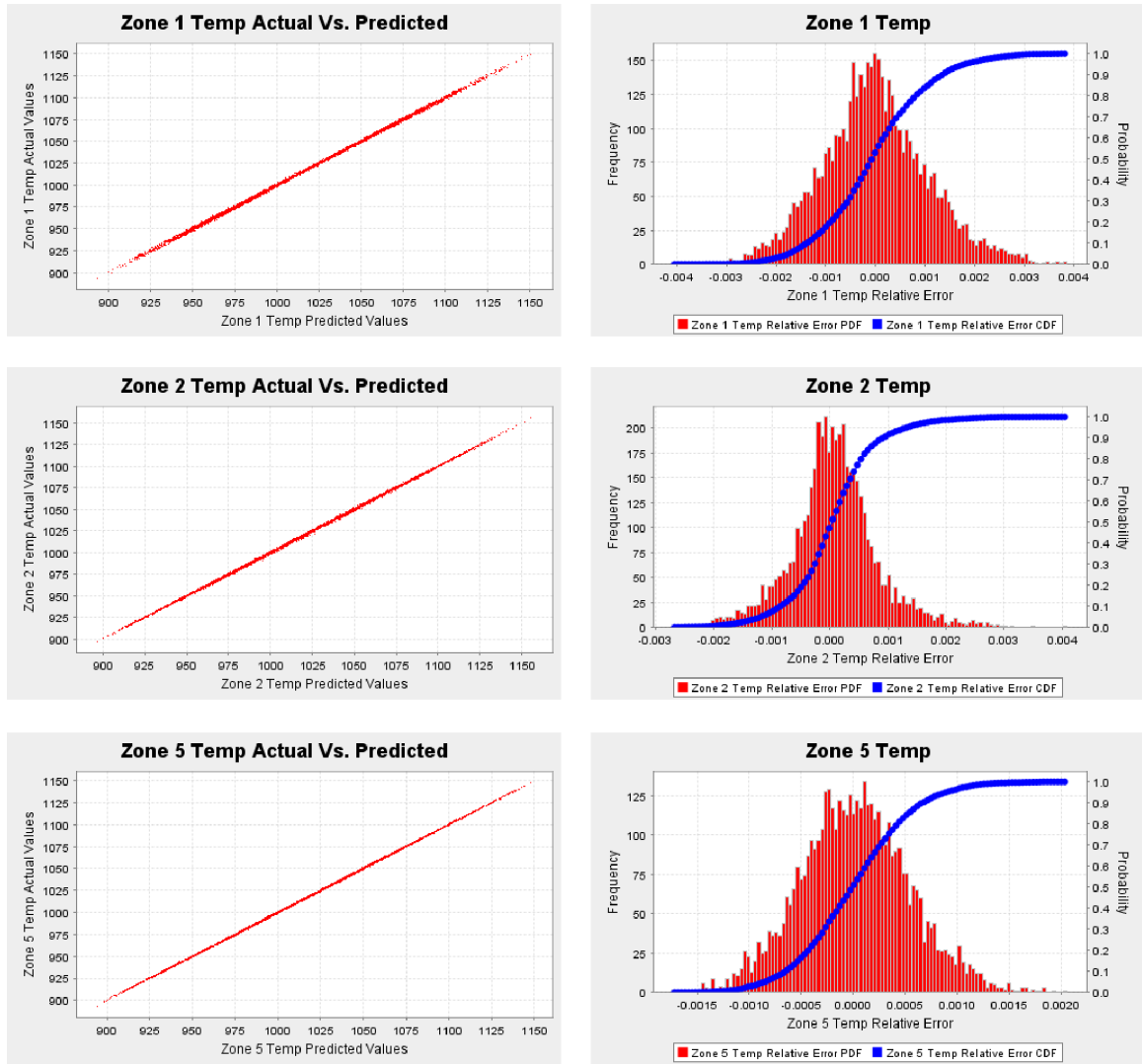


Figure 68: Mincut-4 Partitioning Error Distributions - Temperatures Zone 1, 2, 5

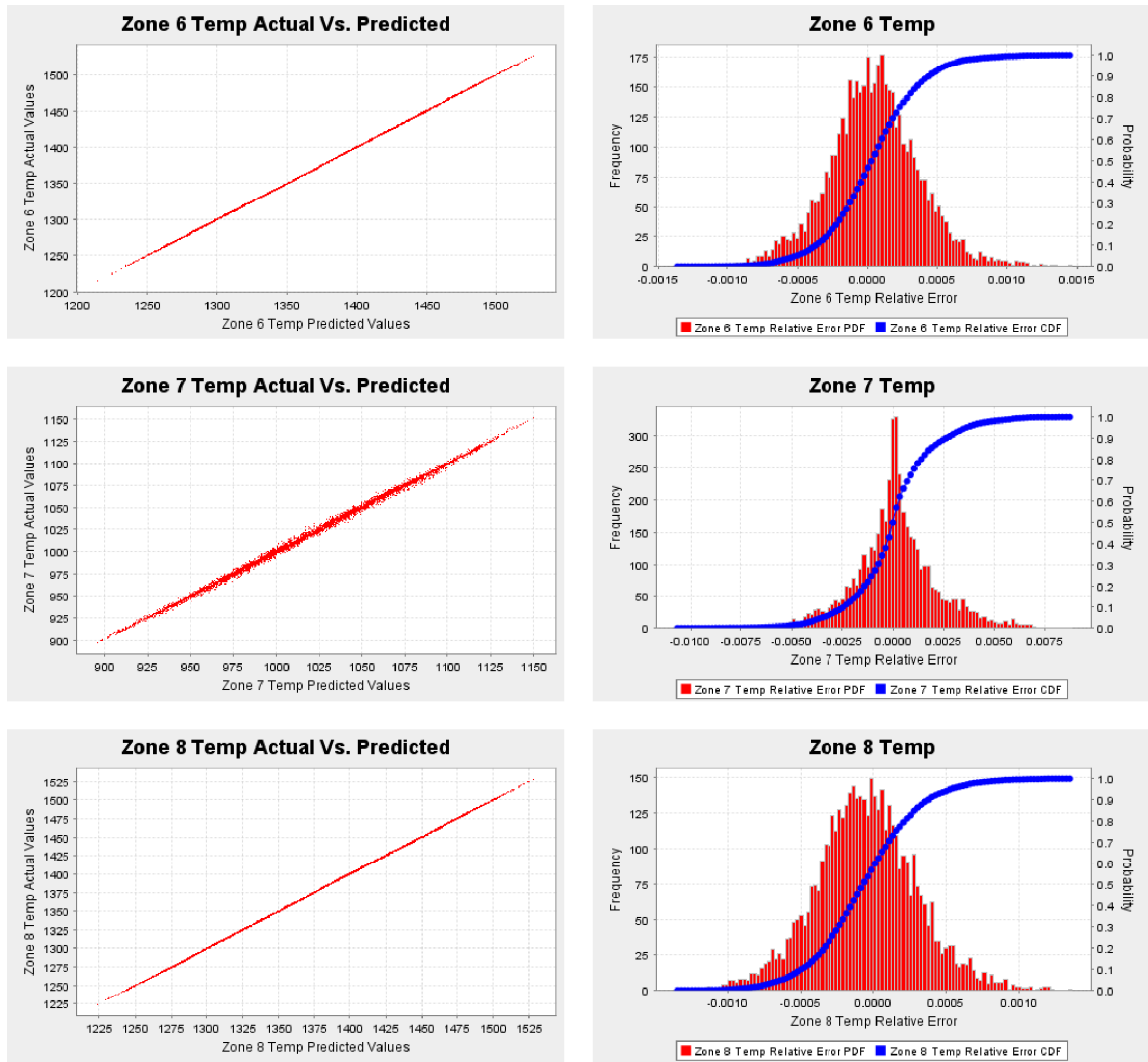


Figure 69: Mincut-4 Partitioning Error Distributions - Temperatures Zone 6, 7, 8

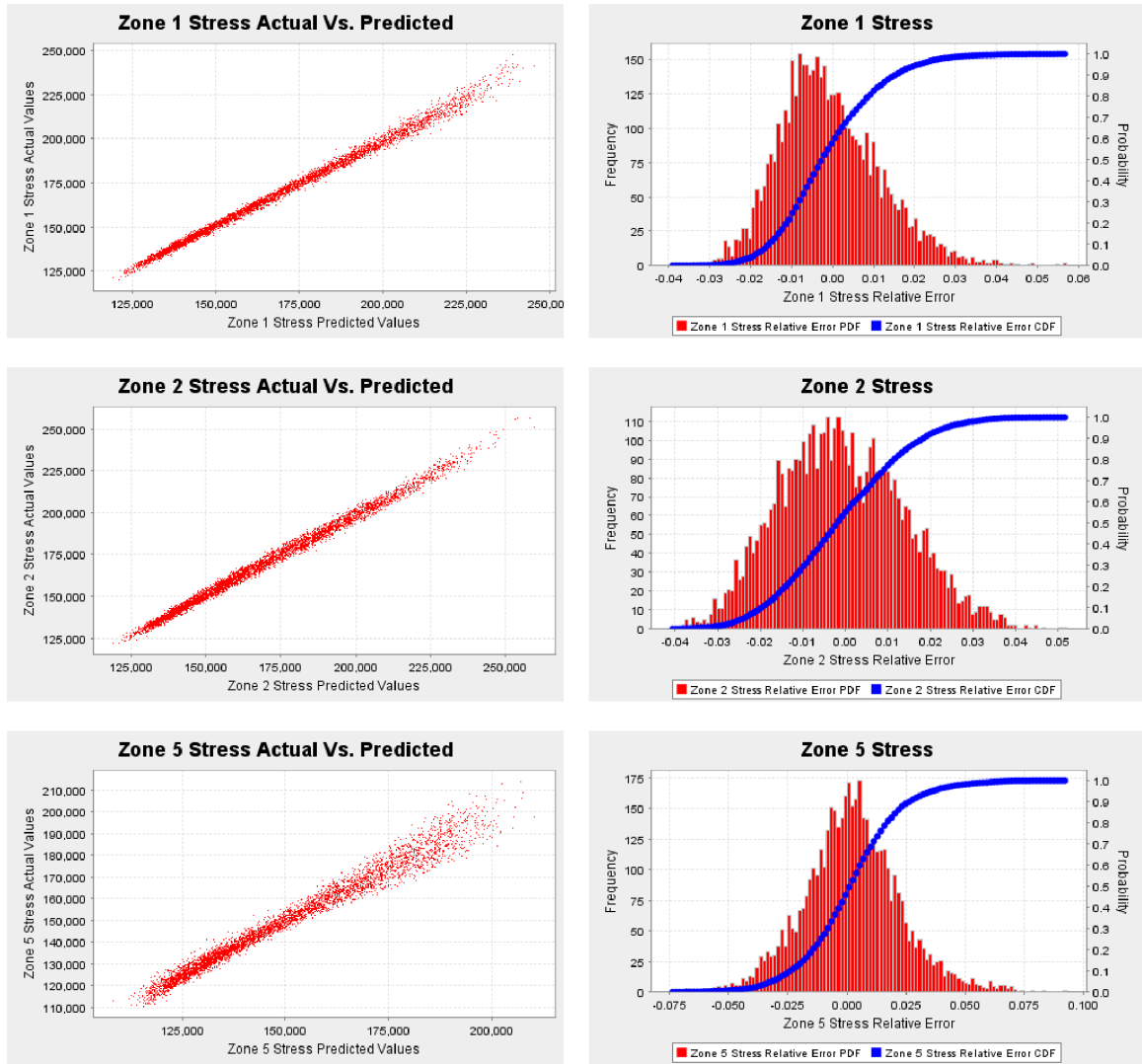


Figure 70: Mincut-5 Partitioning Error Distributions - Stresses Zone 1, 2, 5

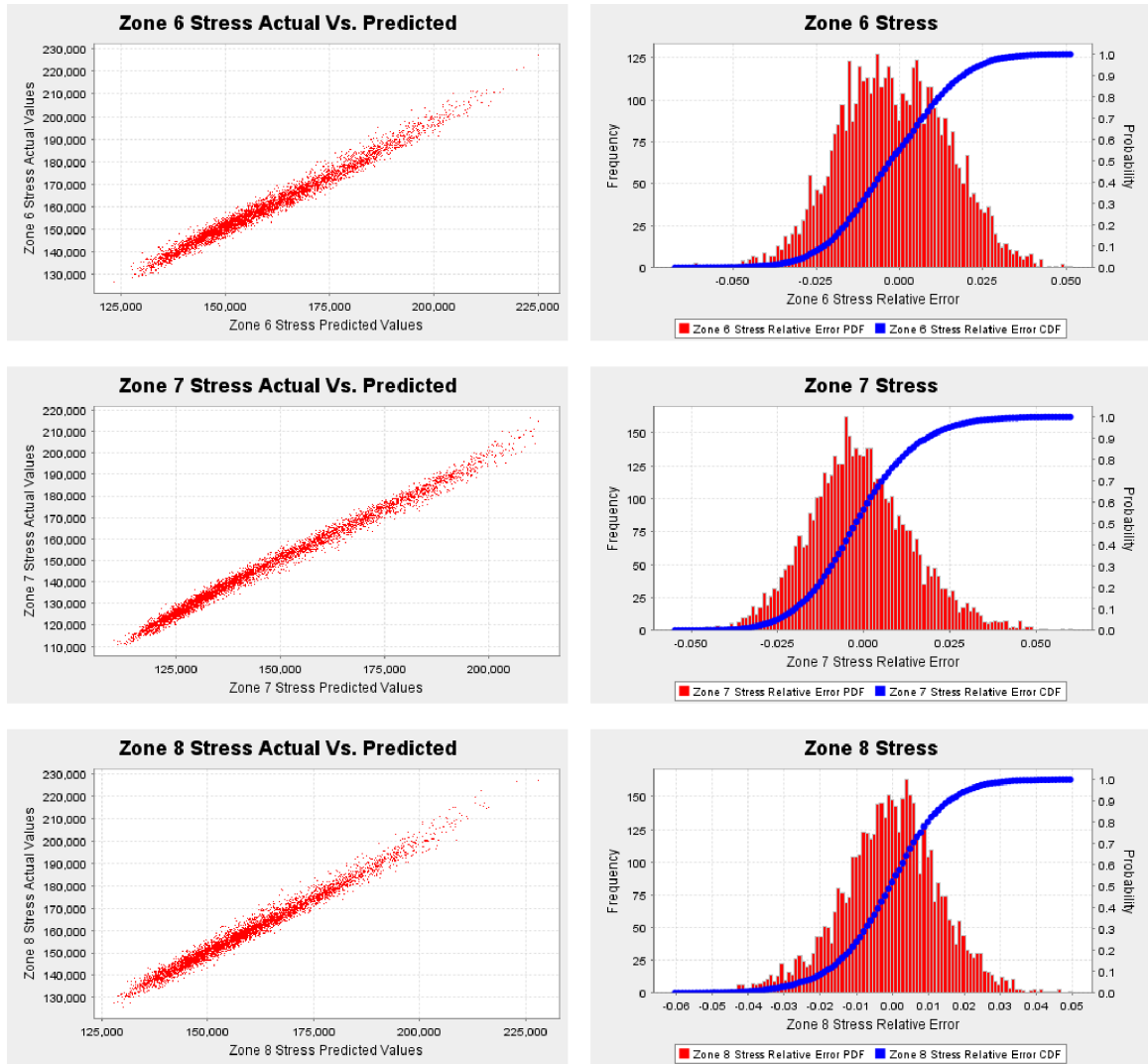


Figure 71: Mincut-5 Partitioning Error Distributions - Stresses Zone 6, 7, 8

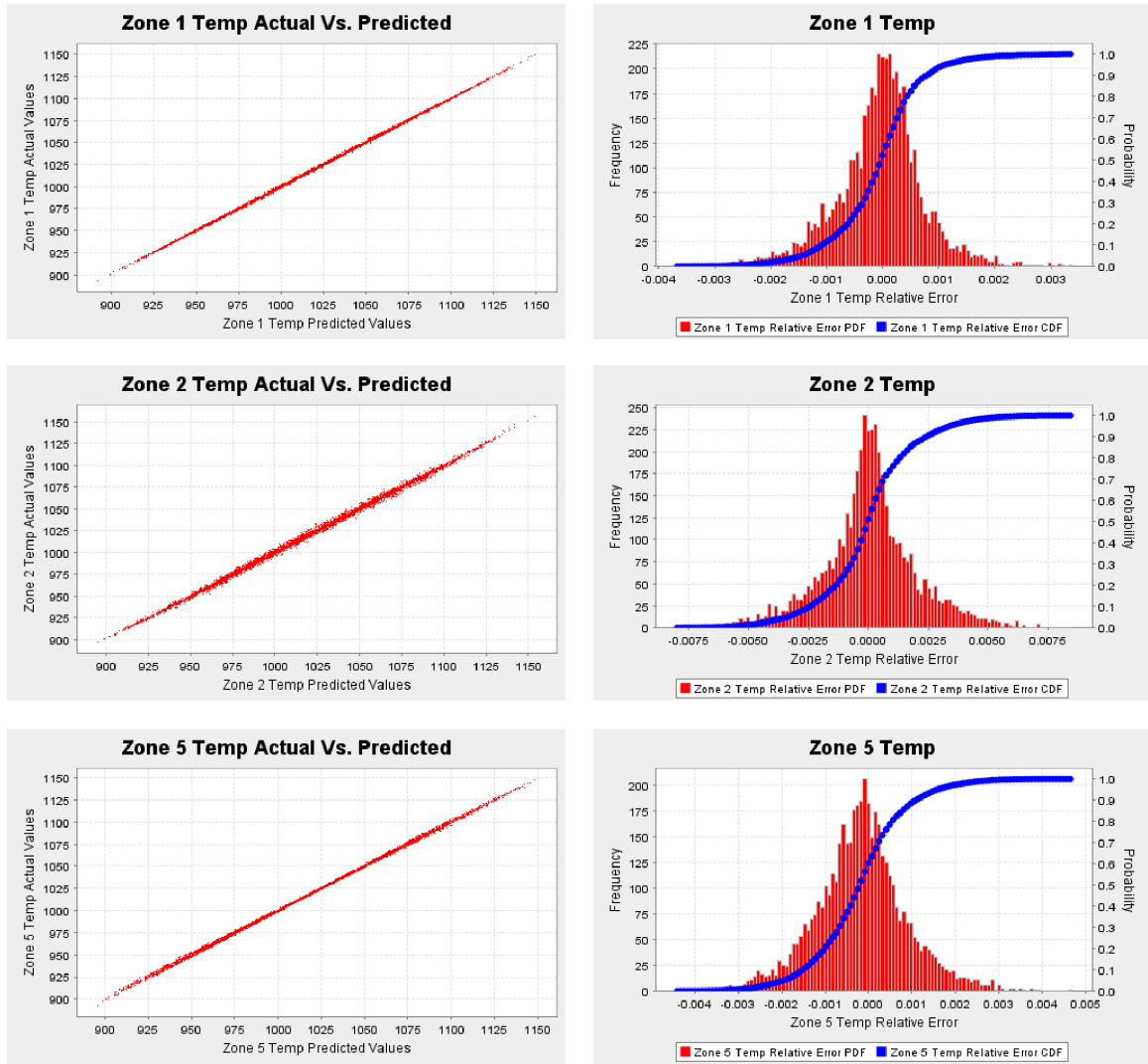


Figure 72: Mincut-5 Partitioning Error Distributions - Temperatures Zone 1, 2, 5

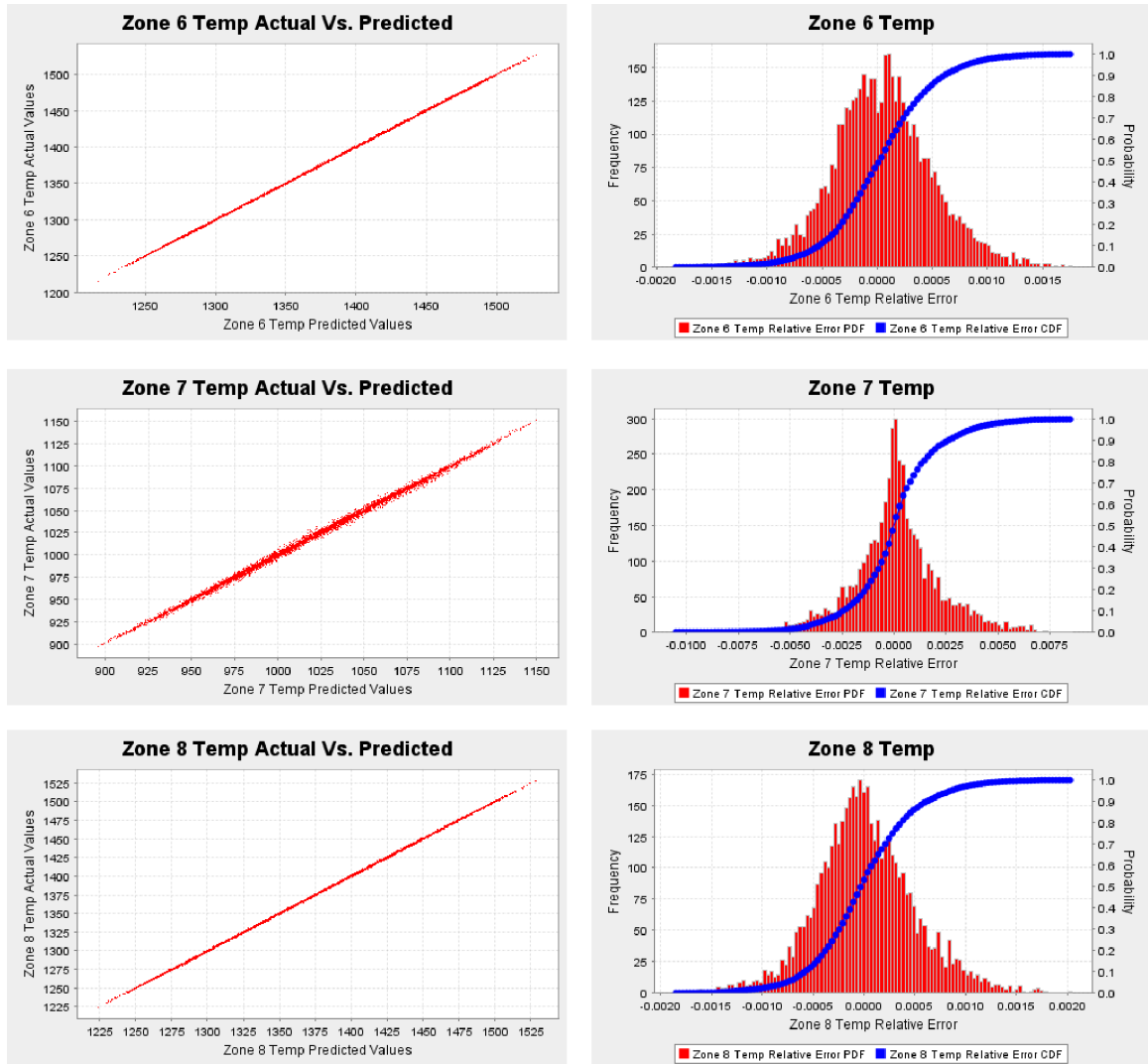


Figure 73: Mincut-5 Partitioning Error Distributions - Temperatures Zone 6, 7, 8

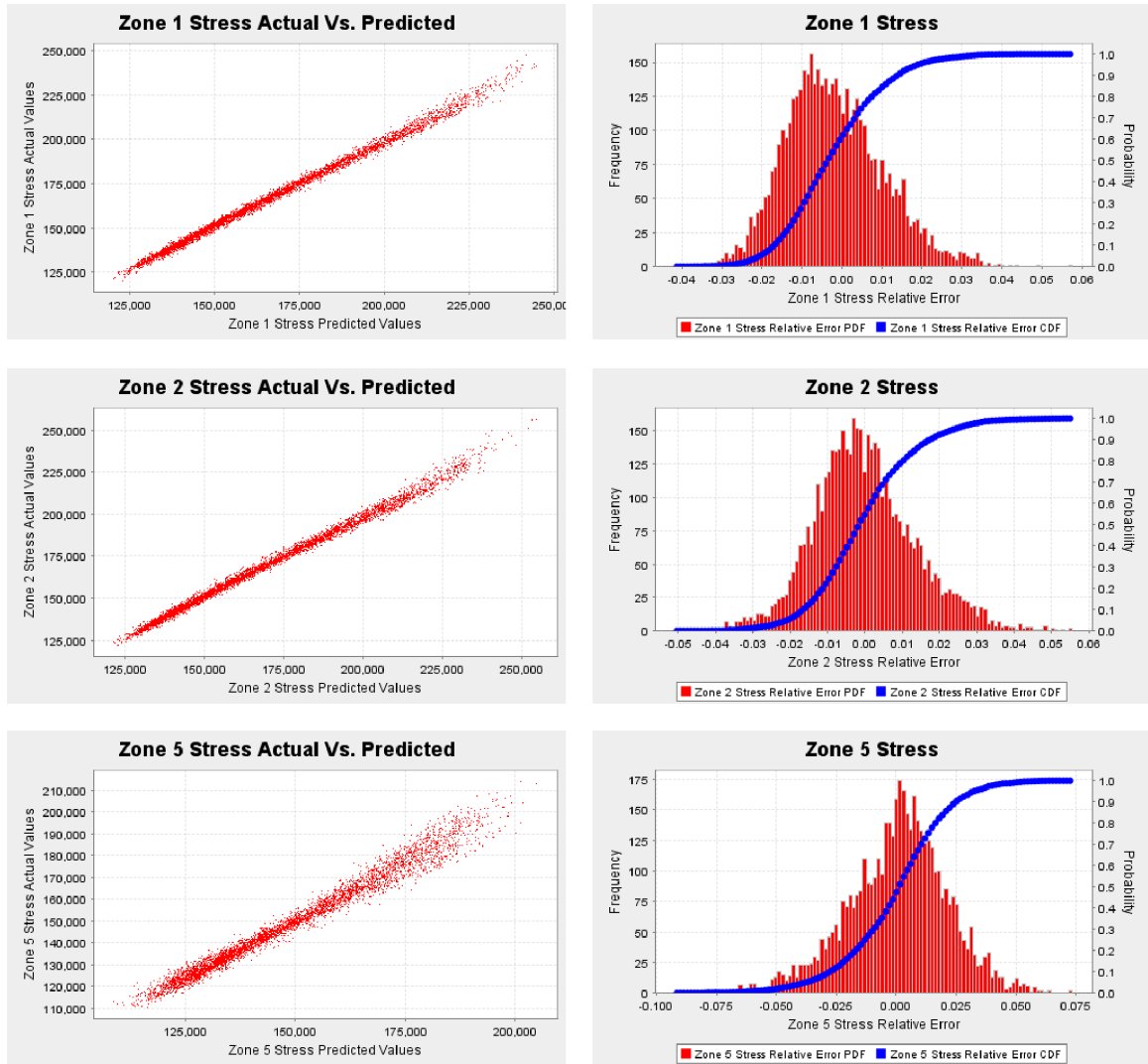


Figure 74: Mincut-6 Partitioning Error Distributions - Stresses Zone 1, 2, 5

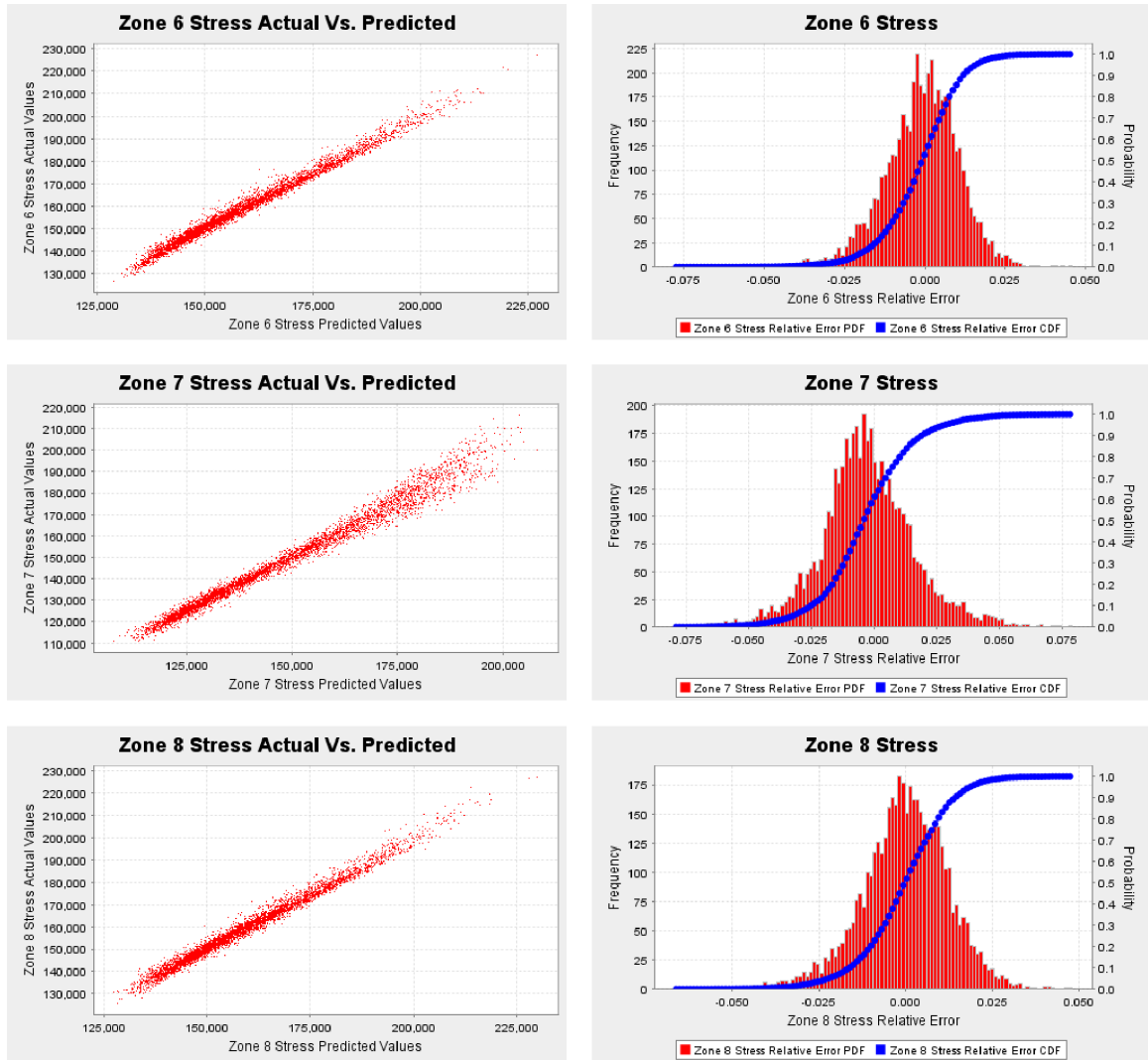


Figure 75: Mincut-6 Partitioning Error Distributions - Stresses Zone 6, 7, 8

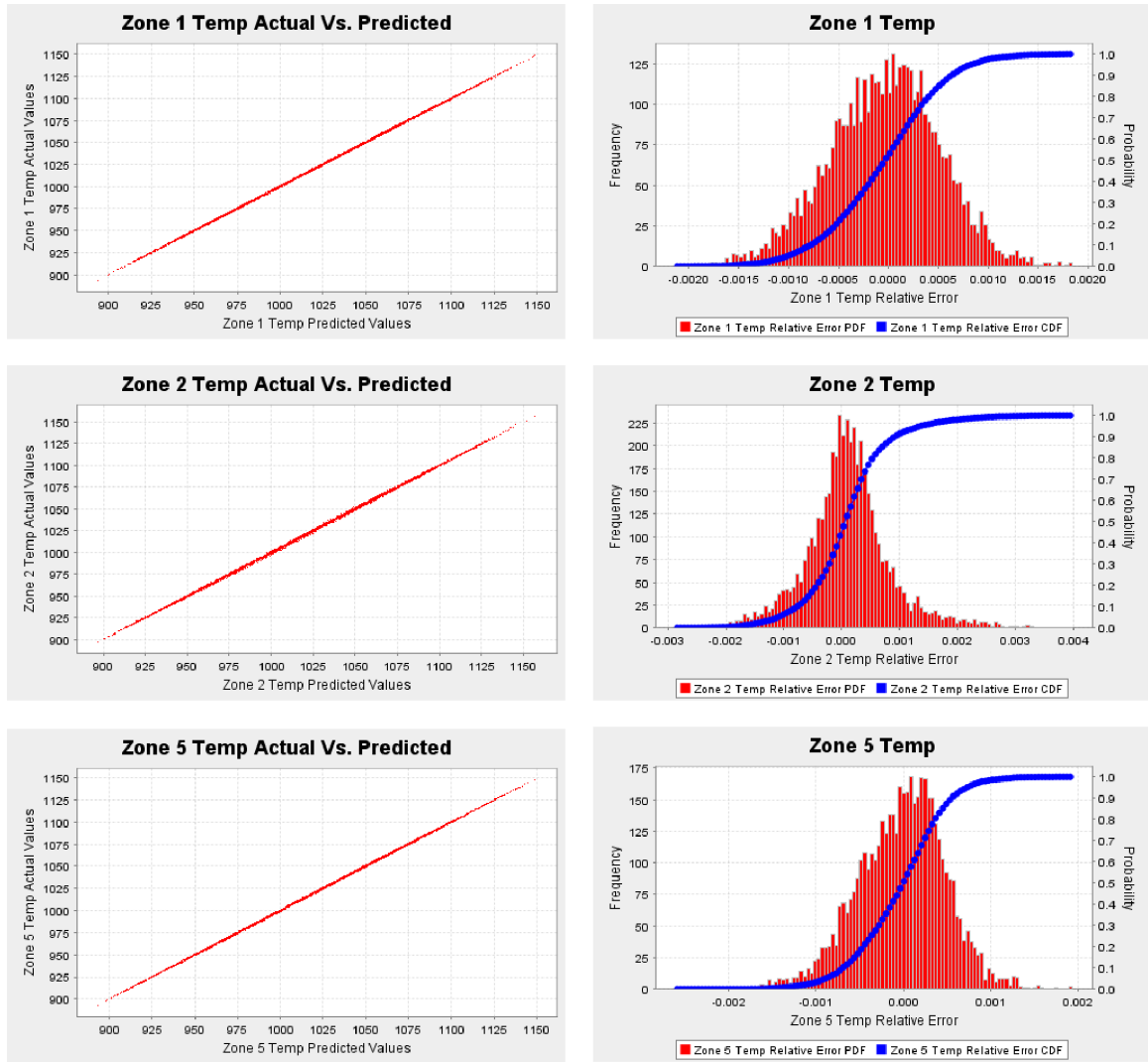


Figure 76: Mincut-6 Partitioning Error Distributions - Temperatures Zone 1, 2, 5

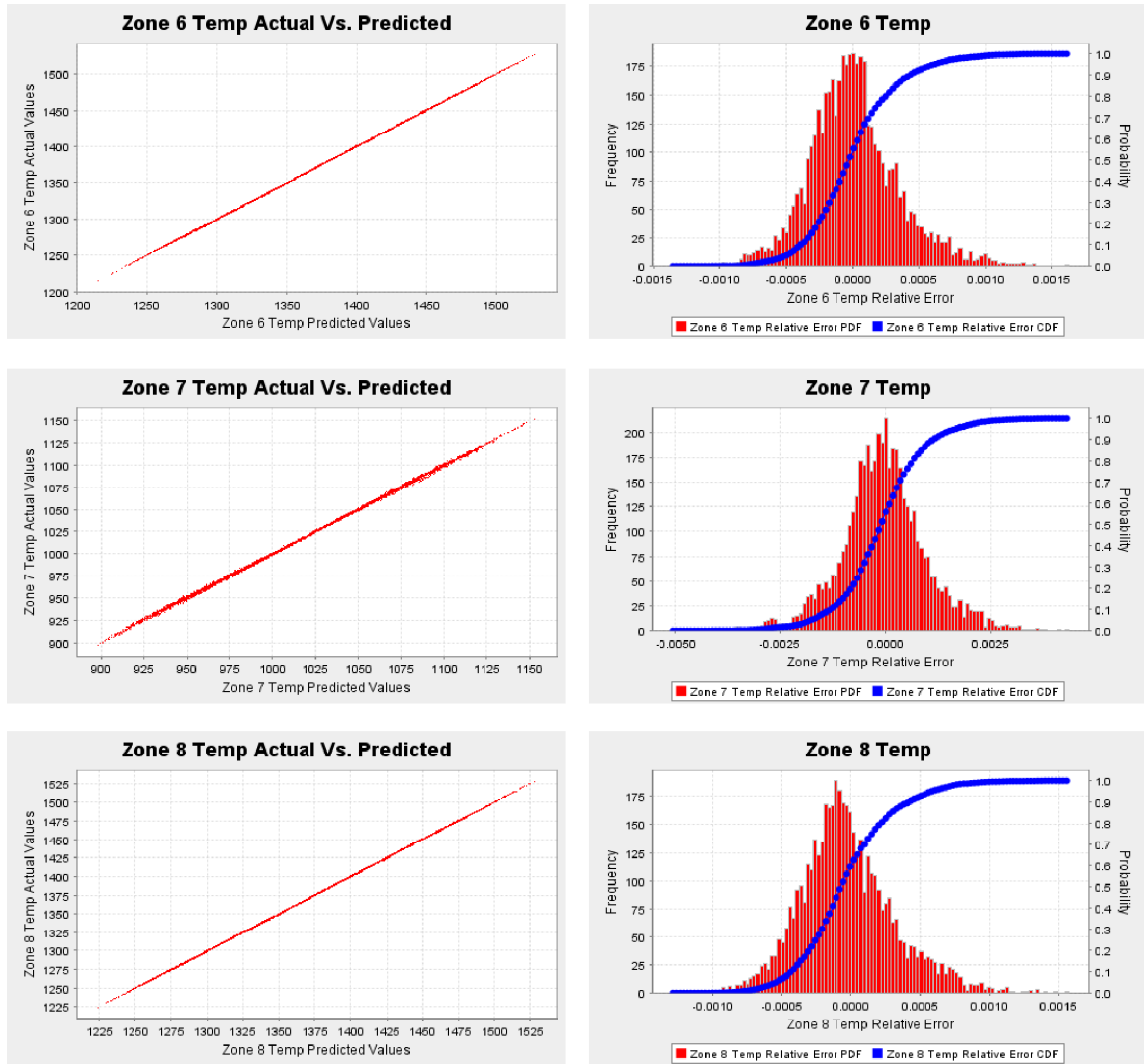


Figure 77: Mincut-6 Partitioning Error Distributions - Temperatures Zone 6, 7, 8

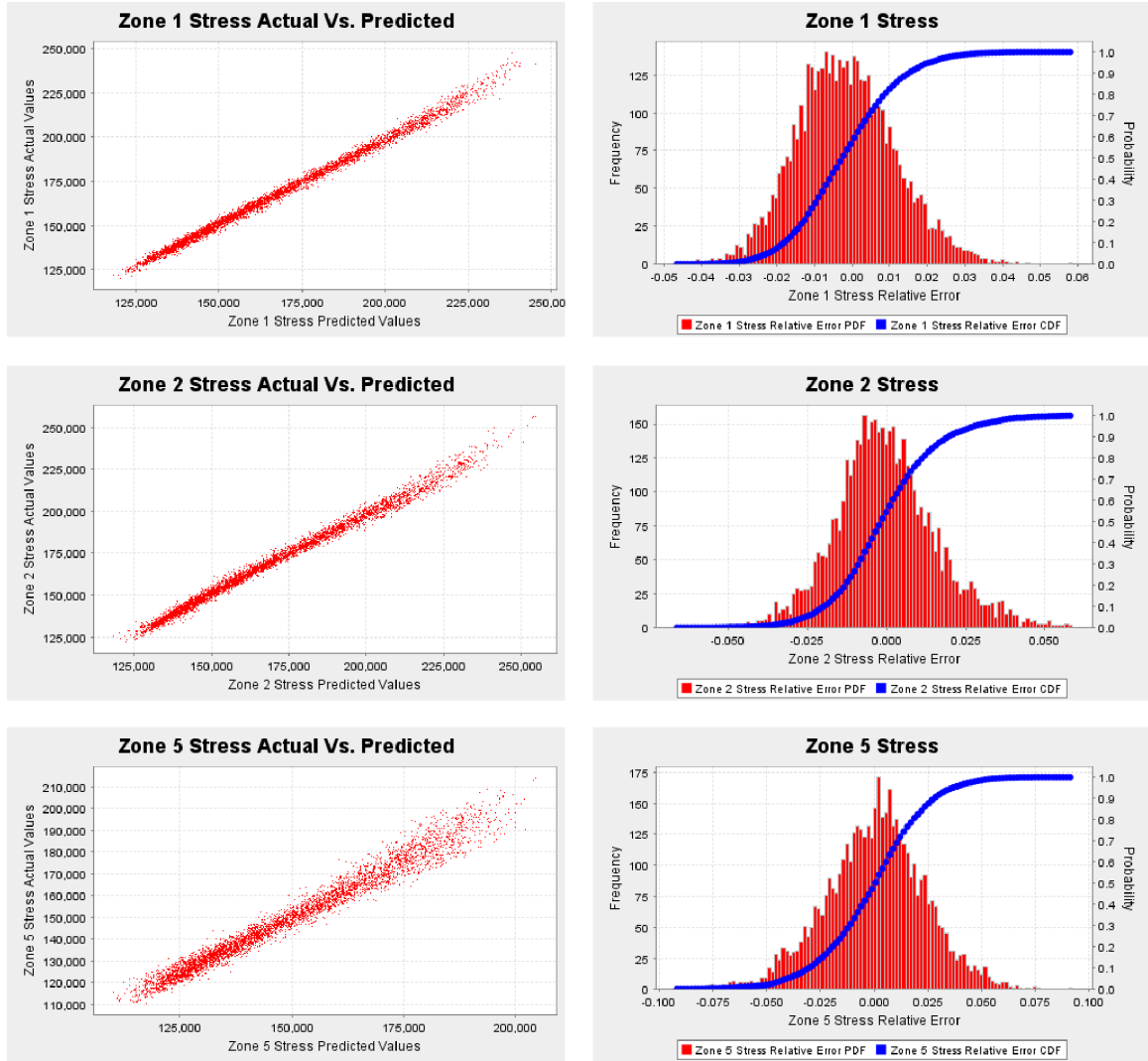


Figure 78: Mincut-7 Partitioning Error Distributions - Stresses Zone 1, 2, 5

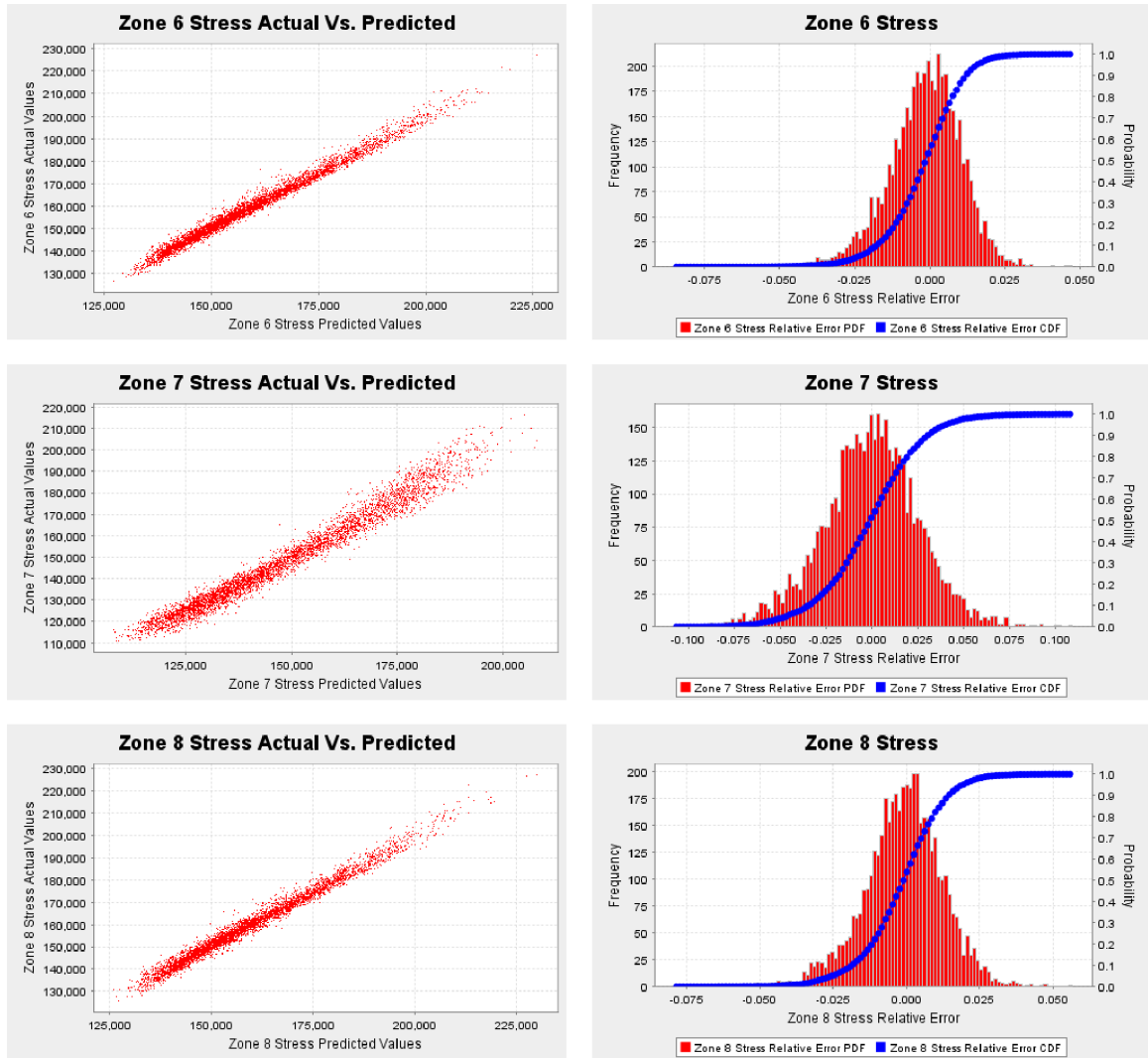


Figure 79: Mincut-7 Partitioning Error Distributions - Stresses Zone 6, 7, 8

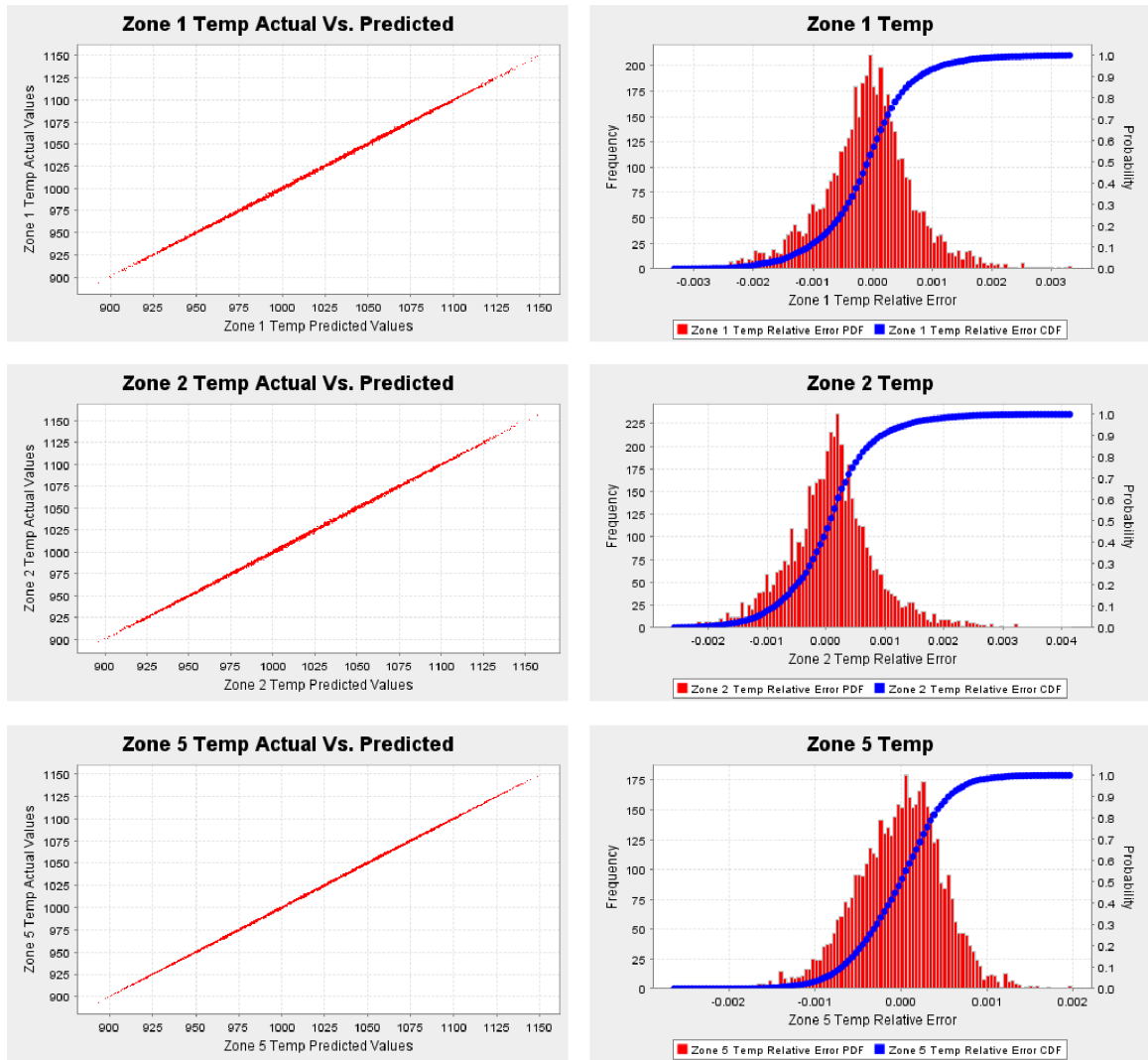


Figure 80: Mincut-7 Partitioning Error Distributions - Temperatures Zone 1, 2, 5

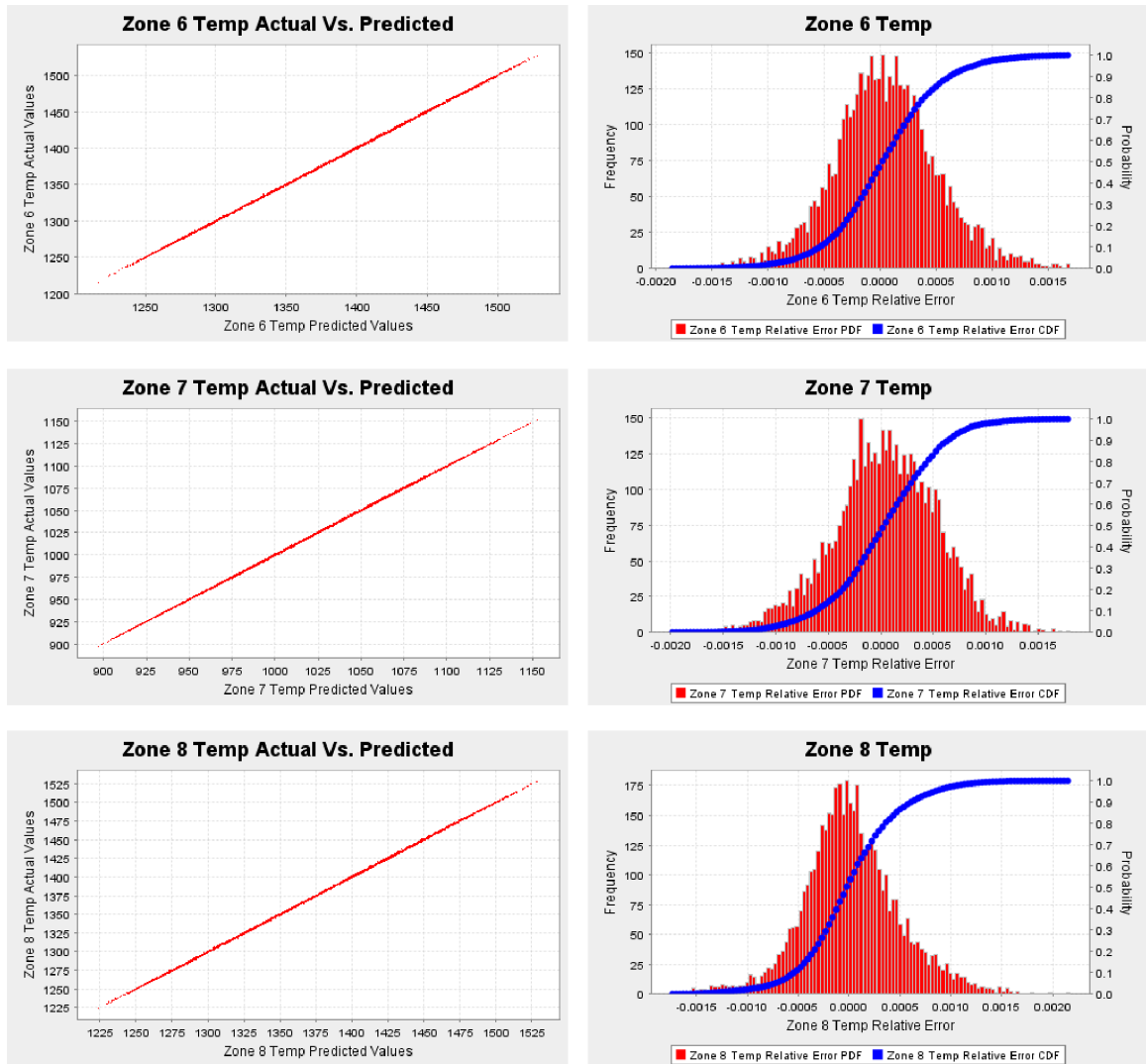


Figure 81: Mincut-7 Partitioning Error Distributions - Temperatures Zone 6, 7, 8

Appendix B

TVS TEST CASE RESULTS

This appendix contains actual versus predicted charts from the validation process for the the TVS model test case. Section B.1 contains the actual versus predicted charts for each response and metamodel type. In addition, the histograms of the relative error distributions for the validation data are also shown.

B.1 Charts of Individual Metamodeling Scheme Error Distributions

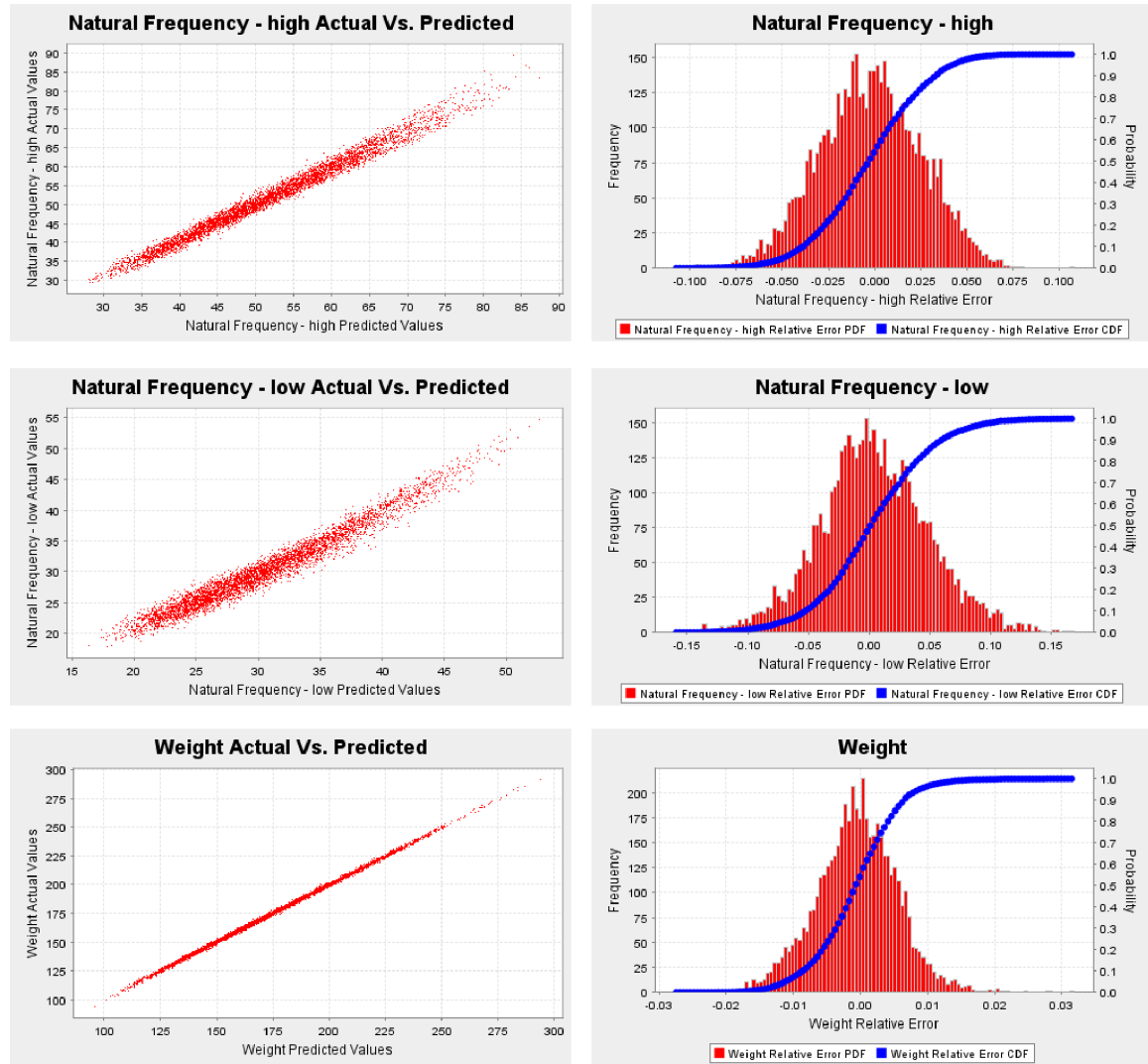


Figure 82: All-at-once Min. Run Res. V CCD Error Distributions

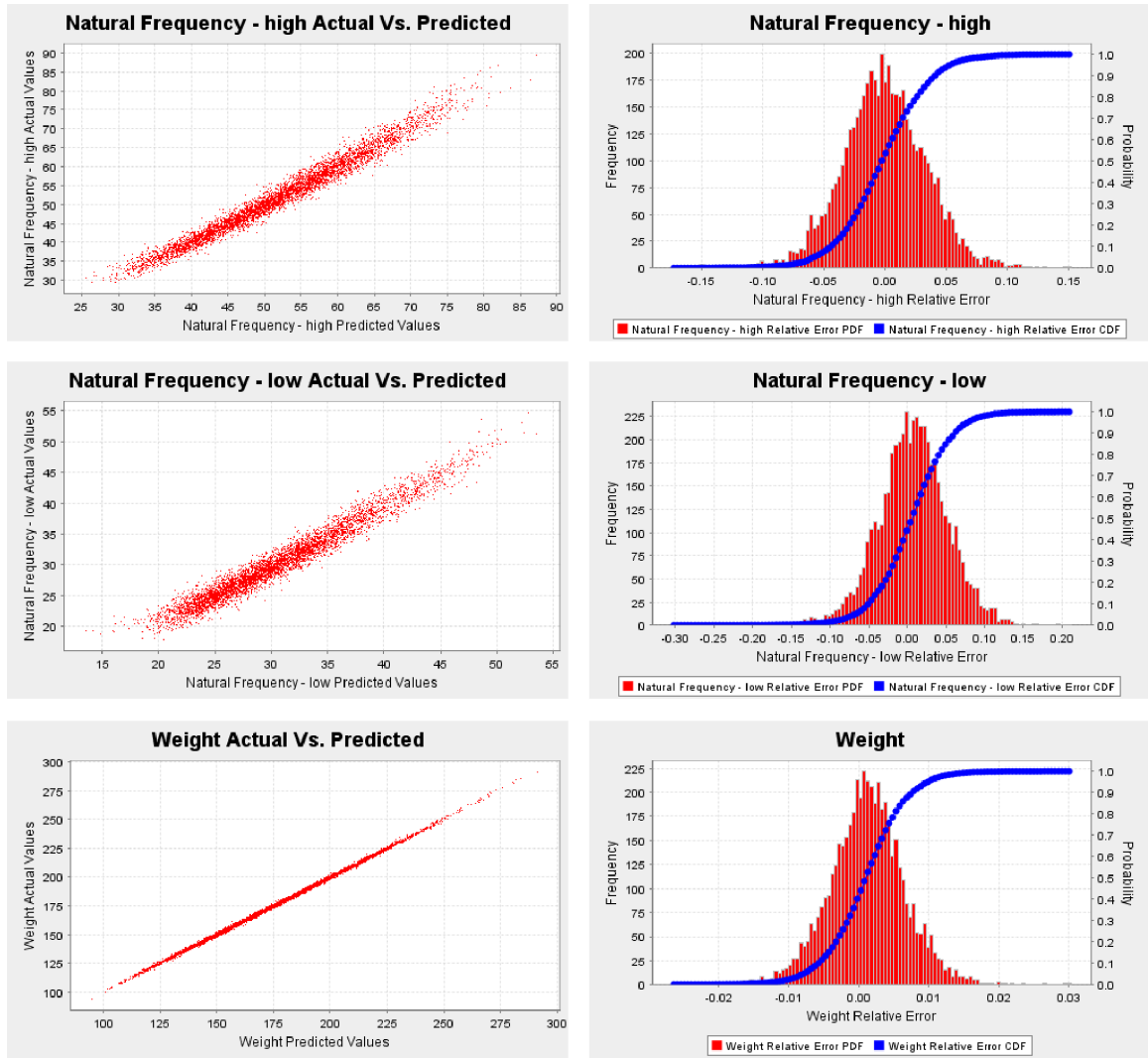


Figure 83: All-at-once Optimal Latin Hypercube Error Distributions

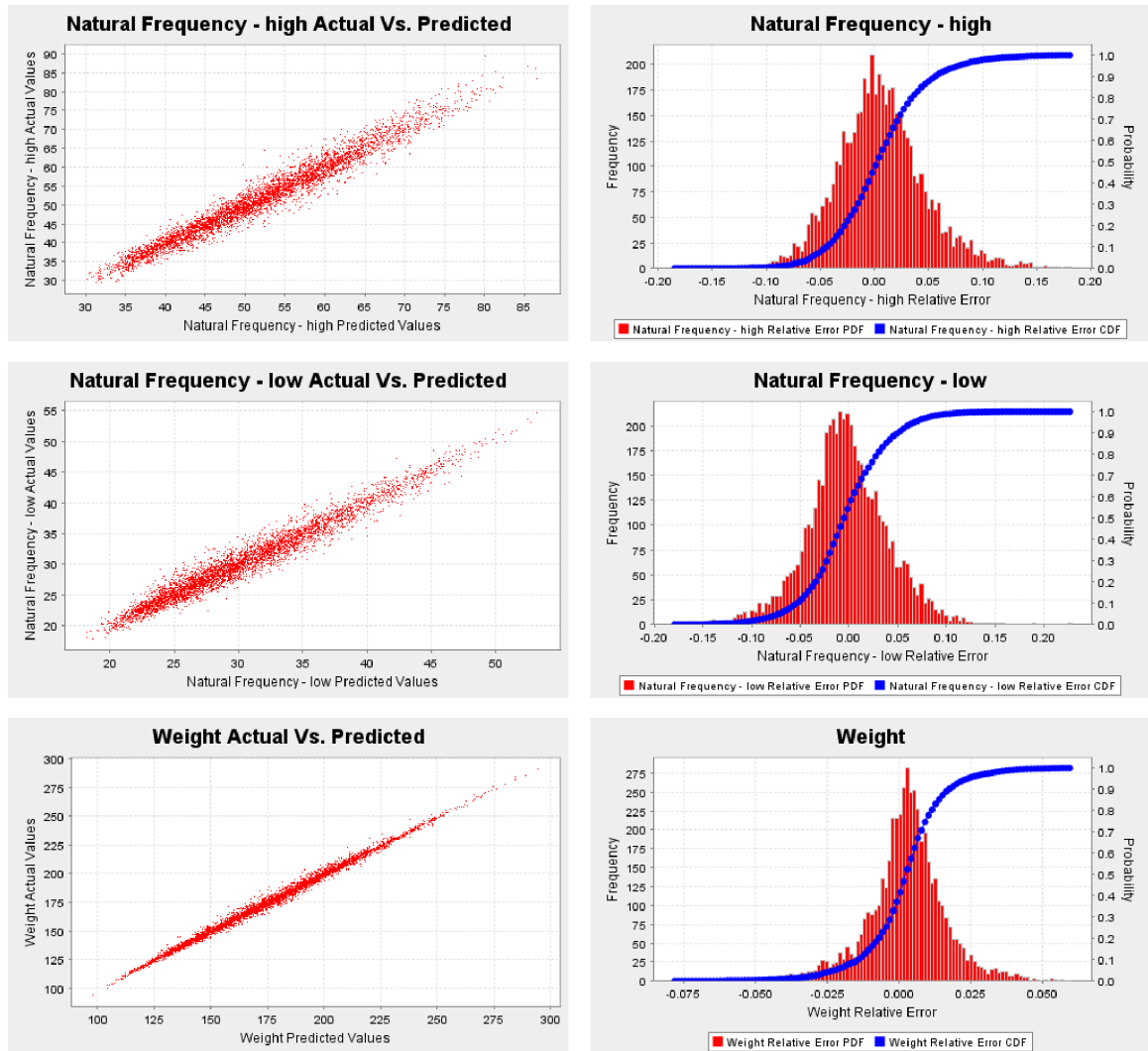


Figure 84: Variable-type Partitioning Error Distributions

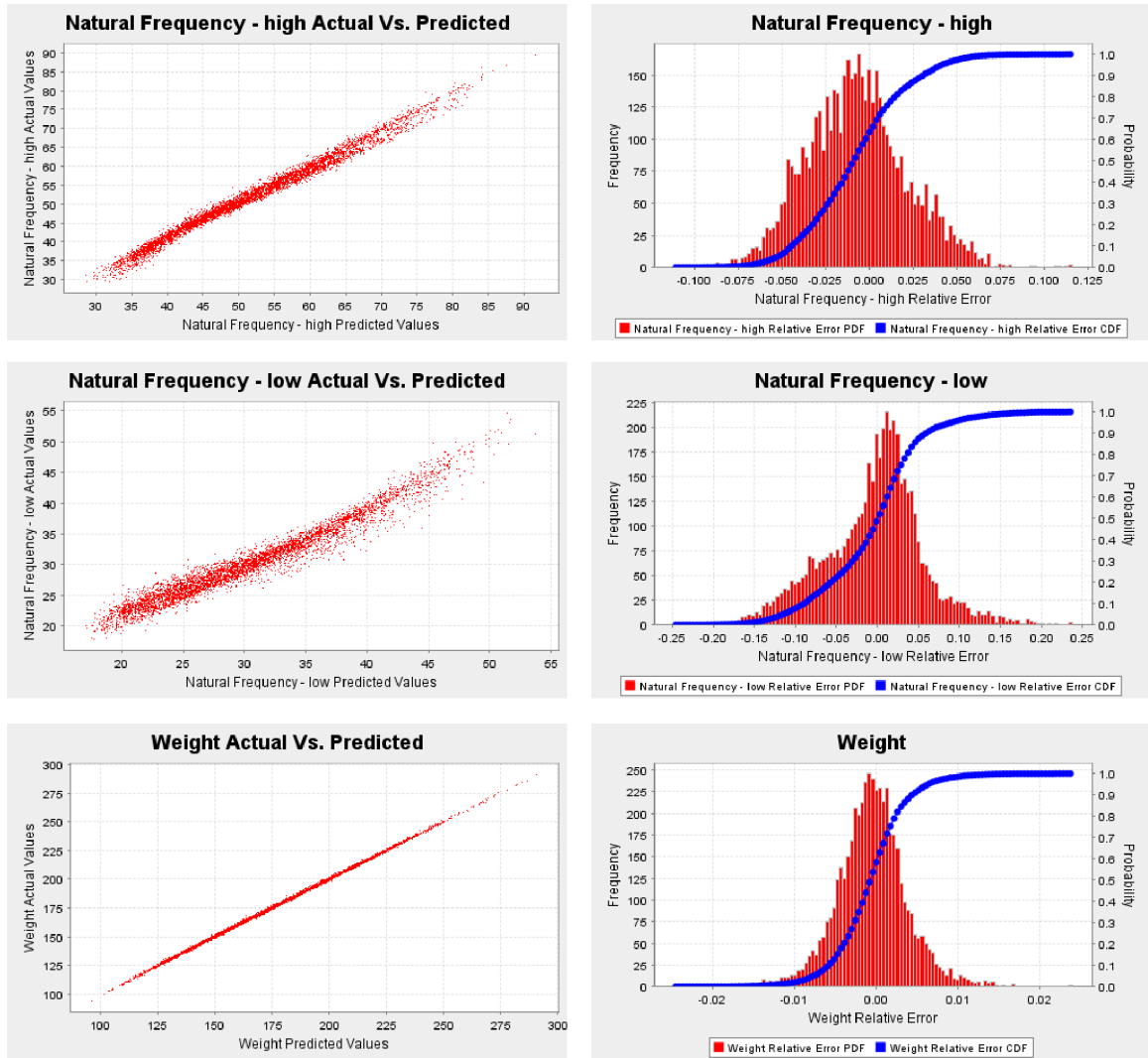


Figure 85: Subsystem Partitioning Error Distributions

Appendix C

THE HOLLOW I-BEAM ANSYS MACRO

```
! Changes for 6-3-2005, Jack Zentner:
! fixed temp and heat transfer coeff mappings
! fixed missing comma in the zone 7V thermal bc' table
! implemented the nodal zone mapping correctly (removed zones 3 and 4 from zones
! 5 and 7, respectively)
! re-implemented the selection of the max stress node per zone
! seperated out zones 3, 4, 5V and 7v into their own zones.
! upgraded model RSE equations

!10.17.02 simp44:
! Compared to simp40 single nodes are considered to be a cluster
!providing main independent variables
! dealing with transitional areas: variables like
! TINT, HINT,TEXT,HEXT will have independent values for zones 1,2,5,7,6,8
! while tranzition between those values will be given by linear
! function
!check where how to get the running time

/PREP7

*SET,TEXT1,1500.0
*SET,TEXT2,1500.0
*SET,TEXT5,1500.0
```

*SET,TEXT6,1500.0

*SET,TEXT7,1500.0

*SET,TEXT8,1500.0

*SET,HEXT1,800

*SET,HEXT2,800

*SET,HEXT5,800

*SET,HEXT6,800

*SET,HEXT7,800

*SET,HEXT8,800

*SET,TINT1,900

*SET,TINT2,900

*SET,TINT5,900

*SET,TINT6,900

*SET,TINT7,900

*SET,TINT8,900

*SET,HINT1,800

*SET,HINT2,800

*SET,HINT5,800

*SET,HINT6,800

*SET,HINT7,800

*SET,HINT8,800

*SET,TH1,0.25

*SET,TH2,0.25

*SET,TH5,0.25

*SET,TH6,0.25

*SET,TH7,0.25

*SET,TH8,0.25

*SET,TH3,1.1

*SET,TH4,1.1

*SET,LENGTH,8

*SET,HEIGHT,8

*SET,WB,8

*SET,WT,8

*SET,DB,1.75

*SET,DT,1.75

*SET,DW,2

*SET,RAD,0.5

*SET,CRYSANG,45

*SET,PRESIN,100

*SET,PRESDEL,200

*SET,THDIV,2

*SET,WDVUPT,10

*SET,WDVUPS,4

*SET,WDVUPB,4

*SET,WDVLOT,10 ! note: changed on 10.13.02 to maintain the symmetry of the mesh!

*SET,WDVLOS,4

*SET,WDVLOB,4

*SET,WDVWEB,6

*SET,WDVFIL,3

*SET,LENDIV,20

*SET,NNUMB,8

*SET,FILEST,'stress'

*SET,FILEEXT,'dat'

*CFOPEN,FILEST,FILEEXT

MPREAD,mp2,lis

*SET,CFAC,3600*144

*SET,HEXT1,HEXT1/CFAC

*SET,HEXT2,HEXT2/CFAC

*SET,HEXT5,HEXT5/CFAC

*SET,HEXT6,HEXT6/CFAC

*SET,HEXT7,HEXT7/CFAC

*SET,HEXT8,HEXT8/CFAC

*SET,HINT1,HINT1/CFAC

*SET,HINT2,HINT2/CFAC

*SET,HINT5,HINT5/CFAC

*SET,HINT6,HINT6/CFAC

*SET,HINT7,HINT7/CFAC

*SET,HINT8,HINT8/CFAC

! CROSS SECTION IS defined by points numeration is from left upper

! corner in rows FIRST OUTER CONTOUR, ALSO ONLY THE GEOMETRY IS DEFINED AT THIS

! POINT, MORE kp MAY BE REQUIRED FOR NICE MESH

K,1,-WT/2,0,HEIGHT/2
 K,4,WT/2,0,HEIGHT/2
 K,5,-WT/2,0,HEIGHT/2-DT
 K,6,-RAD-DW/2,0,HEIGHT/2-DT
 K,7,RAD+DW/2,0,HEIGHT/2-DT
 K,8,WT/2,0,HEIGHT/2-DT
 K,101,-RAD-DW/2,0,HEIGHT/2-DT-RAD! AUXILIRARY POINT (CENTER FOR THE
 FILLET)
 K,9,-DW/2,0,HEIGHT/2-DT-RAD!
 K,102,RAD+DW/2,0,HEIGHT/2-DT-RAD! AUXILIRARY POINT (CENTER FOR THE
 FILLET)
 K,10,DW/2,0,HEIGHT/2-DT-RAD
 K,11,-DW/2,0,-HEIGHT/2+DB+RAD!lower part
 K,103,-RAD-DW/2,0,-HEIGHT/2+DB+RAD!lower part
 K,12,DW/2,0,-HEIGHT/2+DB+RAD!lower part
 K,104,RAD+DW/2,0,-HEIGHT/2+DB+RAD!lower part
 K,13,-WB/2,0,-HEIGHT/2+DB
 K,14,-RAD-DW/2,0,-HEIGHT/2+DB!lower part
 K,15,RAD+DW/2,0,-HEIGHT/2+DB!lower part
 K,16,WB/2,0,-HEIGHT/2+DB
 K,17,-WB/2,0,-HEIGHT/2
 K,20,WB/2,0,-HEIGHT/2

 LSTR,1,4 !to simplify meshing
 LSTR,1,5
 LSTR,4,8
 LSTR,5,6
 LSTR,7,8
 LARC,6,9,101,RAD

LARC,10,7,102,RAD

LSTR,9,11

LSTR,10,12

LARC,11,14,103,RAD

LARC,12,15,104,RAD

LSTR,13,14

LSTR,15,16

LSTR,13,17

LSTR,16,20

LSTR,17,20

! internal contour

K,21,-WT/2+TH5,0,HEIGHT/2-TH6

K,22,WT/2-TH5,0,HEIGHT/2-TH6

K,23,-WT/2+TH5,0,HEIGHT/2-DT+TH5

K,24,-RAD-DW/2,0,HEIGHT/2-DT+TH5

K,25,RAD+DW/2,0,HEIGHT/2-DT+TH5

K,26,WT/2-TH5,0,HEIGHT/2-DT+TH5

K,27,-DW/2+TH2,0,HEIGHT/2-DT-RAD!

K,28,DW/2-TH1,0,HEIGHT/2-DT-RAD

K,29,-DW/2+TH2,0,-HEIGHT/2+DB+RAD!lower part

K,30,DW/2-TH1,0,-HEIGHT/2+DB+RAD!lower part

K,31,-WB/2+TH7,0,-HEIGHT/2+DB-TH7

K,32,-RAD-DW/2,0,-HEIGHT/2+DB-TH7

K,33,RAD+DW/2,0,-HEIGHT/2+DB-TH7

K,34,WB/2-TH7,0,-HEIGHT/2+DB-TH7

K,35,-WB/2+TH7,0,-HEIGHT/2+TH8

K,36,WB/2-TH7,0,-HEIGHT/2+TH8

LSTR,21,22

LSTR,21,23

LSTR,22,26

LSTR,23,24

LSTR,25,26

LARC,24,27,101,TH3!note that TH3 here actually defines the radius for the inside

LARC,25,28,102,TH3

LSTR,27,29

LSTR,28,30

LARC,29,32,103,TH4!note that TH4 here actually defines the radius for the inside

LARC,30,33,104,TH4!

LSTR,31,32

LSTR,33,34

LSTR,31,35

LSTR,34,36

LSTR,35,36

A,1,4,22,21

A,1,21,23,5

A,22,4,8,26

A,5,23,24,6

A,25,26,8,7

A,24,27,9,6

A,25,7,10,28

A,9,27,29,11

A,28,10,12,30

A,11,29,32,14

A,30,12,15,33

A,13,14,32,31

A,15,16,34,33

A,13,31,35,17

A,34,16,20,36

A,35,36,20,17

!meshing parameters

!THICKNESS: 33,34,35,36,37,38,39,40,41,42,43,44..48

LSEL,S,,33,48

CM,LTHICK,LINE

LSEL,S,,1,17,16! lines 1,17

CM,LWUPT,LINE! top of the upper flange

LSEL,S,,2,3

LSEL,A,,18,19

CM,LWUPS,LINE !sides of the upper flange

LSEL,S,,4,5

LSEL,A,,20,21

CM,LWUPB,LINE !"bottoms" of the upper flange

LSEL,S,,6,7

LSEL,A,,22,23

LSEL,A,,26,27

LSEL,A,,10,11

CM,LWFIL,LINE ! fillets

LSEL,S,,8,9

LSEL,A,,24,25

CM,LWWEB,LINE !"bottoms" of the flange

LSEL,S,,12,13

LSEL,A,,28,29

CM,LWLOB,LINE !"bottoms" of the flange

LSEL,S,,14,15

LSEL,A,,30,31

CM,LWLOS,LINE !sides of the lower flange

LSEL,S,,16,32,16! lines 16,32

CM,LWLOT,LINE! top of the LOWER flange

ALLSEL,ALL

LESIZE,LTHICK,,THDIV

LESIZE,LWUPT,,WDVUPT

LESIZE,LWUPS,,WDVUPS

LESIZE,LWUPB,,WDVUPB

LESIZE,LWLOT,,WDVLOT

LESIZE,LWLOS,,WDVLOS

LESIZE,LWLOB,,WDVLOB

LESIZE,LWWEB,,WDVWEB

LESIZE,LWFIL,,WDVFIL

ASEL,S,,1,16

CM,AROOT,AREA

ALLSEL,BELOW,AREA

ET,1,SHELL57

```

MSHAPE,0,2D
MSHKEY,1 !
AMESH,ALL
ET,2,SOLID70
TYPE, 1
EXTOPT,ESIZE,LENDIV/2,0,
EXTOPT,ACLEAR,1
VEXT,ALL,,,LENGTH/2
*GET,NUMNOD,NODE,,COUNT

! *****
! The following section of code defines the nodal componets associated with a
! given zone
!selecting nodal components based on the location
NSEL,S,LOC,Z,HEIGHT/2-TH6,HEIGHT/2
CM,NZONE6,NODE

NSEL,S,LOC,Z,-HEIGHT/2,-HEIGHT/2+TH8
CM,NZONE8,NODE

NSEL,S,LOC,Z,HEIGHT/2-DT-RAD,HEIGHT/2-DT+TH5
! the small delta (0.01) is to make sure that the nodes on the border of
! zones 3 and 5 are included in zone 5
NSEL,U,LOC,X,-DW/2-RAD+.01,DW/2+RAD-.01
CM,NZONE5,NODE

NSEL,S,LOC,Z,-HEIGHT/2+DB-TH8,-HEIGHT/2+DB
! the small delta (0.01) is to make sure that the nodes on the border of
! zones 4 and 7 are included in zone 7

```

NSEL,U,LOC,X,-DW/2-RAD+.01,DW/2+RAD-.01

CM,NZONE7,NODE

NSEL,S,LOC,Z,-HEIGHT/2+DB+RAD,HEIGHT/2-DT-RAD

NSEL,R,LOC,X,-DW/2-RAD,0

CM,NZONE2,NODE

NSEL,S,LOC,Z,-HEIGHT/2+DB+RAD,HEIGHT/2-DT-RAD

NSEL,R,LOC,X,0,DW/2+RAD

CM,NZONE1,NODE

NSEL,S,LOC,Z,HEIGHT/2-DT-RAD,HEIGHT/2-DT+TH5

CMSEL,U,NZONE5

CMSEL,U,NZONE2

CMSEL,U,NZONE1

CM,NZONE3,NODE

NSEL,S,LOC,Z,-HEIGHT/2+DB-TH7,-HEIGHT/2+DB+RAD

CMSEL,U,NZONE7

CMSEL,U,NZONE2

CMSEL,U,NZONE1

CM,NZONE4,NODE

NSEL,S,LOC,Z,HEIGHT/2,HEIGHT/2-DT

CMSEL,U,NZONE6

CMSEL,U,NZONE5

CMSEL,U,NZONE3

CM,NZONE5V,NODE

NSEL,S,LOC,Z,-HEIGHT/2,-HEIGHT/2+DB

CMSEL,U,NZONE8

CMSEL,U,NZONE7

CMSEL,U,NZONE4

CM,NZONE7V,NODE

! *****

!Lumps all of the back face areas created from the VEXT command into
!one compoent called ATIP

ASEL,S,,17!tip:

ASEL,A,,22,78,4

CM,ATIP,AREA

ASEL,S,,49

!error - CM,AZONE1,AREA !external surface for zone 1 (X- POSITIVE)

CM,AZONE2,AREA !external surface for zone 2 (X- NEGATIVE)

ASEL,S,,47

!error - CM,AZONI1,AREA !INTERNAL surface for zone 1

CM,AZONI2,AREA !INTERNAL surface for zone 2 (X- NEGATIVE)

ASEL,S,,51

!error - CM,AZONE2,AREA !external surface for zone 2 (X- NEGATIVE)

CM,AZONE1,AREA !external surface for zone 1 (X- POSITIVE)

ASEL,S,,53

!error - CM,AZONI2,AREA !external surface for zone 2 (X- NEGATIVE)

CM,AZONI1,AREA !external surface for zone 2 (X- POSITIVE)

ASEL,S,,41

CM,AZONE3L,AREA !external surface for zone 3 (Z- POS) -left(adjacent to zone 2)

ASEL,S,,43

CM,AZONE3R,AREA !external surface for zone 3 (Z- POS) -right(adjacent to zone 1)

ASEL,S,,39

CM,AZONI3L,AREA !INTERNAL surface for zone 3 (Z- POS) -left(adjacent to zone 2)

ASEL,S,,45

CM,AZONI3R,AREA !INTERNAL surface for zone 3 (Z- POS) -right(adjacent to zone 1)

ASEL,S,,57

CM,AZONE4L,AREA !external surface for zone 4 (Z- NEG) -left(adjacent to zone 2)

ASEL,S,,59

CM,AZONE4R,AREA !external surface for zone 4 (Z- NEG) -right(adjacent to zone 1)

ASEL,S,,55

CM,AZONI4L,AREA !INTERNAL surface for zone 4 (Z- NEG) -left(adjacent to zone 2)

ASEL,S,,61

CM,AZONI4R,AREA !INTERNAL surface for zone 4 (Z- NEG) -right(adjacent to zone 1)

ASEL,S,,25,27,2!azone5 33,36,25,27

CM,AZONE5V,AREA ! external 5 vertical (tranzitional part)

ASEL,S,,33,36,3

CM,AZONE5H,AREA !EXTERNAL actually it looks like we need further

!split zones 5 and 7 into two 5v and 7v(vertical, where transition
!takes place and 5h and 7h (horizontal where BC are constant)

ASEL,S,,23,29,6!AZONI5 23,31,35,29
CM,AZONI5V,AREA !INTERNAL

ASEL,S,,31,35,4
CM,AZONI5H,AREA !INTERNAL

ASEL,S,,18
CM,AZONE6,AREA !external surface for zone 6 !Z POSITIVE

ASEL,S,,20
CM,AZONI6,AREA !INTERNAL surface for zone 6

ASEL,S,,63,67,4! zone7 67, 75,63,73
CM,AZONE7H,AREA !EXTERNAL HORIZ

ASEL,S,,73,75,2
CM,AZONE7V,AREA !EXTERNAL VERT

ASEL,S,,64,69,5! ZONI7 71, 64,69,77
CM,AZONI7H,AREA !INTERNAL HORIZ

ASEL,S,,71,77,6
CM,AZONI7V,AREA !INTERNAL

ASEL,S,,80
CM,AZONE8,AREA !external surface for zone 8 Y NEGATIVE

ASEL,S,,79

CM,AZONI8,AREA !INTERNAL surface for zone 8

! now let us create total components

CMSEL,S,AZONE1 !EXTERNAL

CMSEL,A,AZONE2

CMSEL,A,AZONE3L

CMSEL,A,AZONE3R

CMSEL,A,AZONE4L

CMSEL,A,AZONE4R

CMSEL,A,AZONE5H

CMSEL,A,AZONE5V

CMSEL,A,AZONE6

CMSEL,A,AZONE7H

CMSEL,A,AZONE7V

CMSEL,A,AZONE8

CM,AZONE,AREA

CMSEL,S,AZONI1 ! INTERNAL

CMSEL,A,AZONI2

CMSEL,A,AZONI3L

CMSEL,A,AZONI3R

CMSEL,A,AZONI4L

CMSEL,A,AZONI4R

CMSEL,A,AZONI5H

CMSEL,A,AZONI5V

CMSEL,A,AZONI6

CMSEL,A,AZONI7H

CMSEL,A,AZONI7V

CMSEL,A,AZONI8

CM,AZONI,AREA

ALLSEL,ALL

!this is a solution part

/SOLU

ANTYPE,STATIC

!first let us define regular (constant) bc's:

SF,AZONE1,CONV,HEXT1,TEXT1

SF,AZONI1,CONV,HINT1,TINT1

SF,AZONE2,CONV,HEXT2,TEXT2

SF,AZONI2,CONV,HINT2,TINT2

SF,AZONE5H,CONV,HEXT5,TEXT5

SF,AZONI5H,CONV,HINT5,TINT5

SF,AZONE6,CONV,HEXT6,TEXT6

SF,AZONI6,CONV,HINT6,TINT6

SF,AZONE7H,CONV,HEXT7,TEXT7

SF,AZONI7H,CONV,HINT7,TINT7

SF,AZONE8,CONV,HEXT8,TEXT8

SF,AZONI8,CONV,HINT8,TINT8

! we are defining tables for transitioning area

! let us make sure that z is always increasing

! here we have transtion in the fillets

*DIM,HFTBE3L,TABLE,2,,Z! H for tranzitional area 3,left external

*DIM,TMTBE3L,TABLE,2,,Z! T for tranzitional area 3,left external

HFTBE3L(1,0)=HEIGHT/2-DT-RAD,HEIGHT/2-DT! kp 9 and 6 respectively

HFTBE3L(1,1)=HEXT2,HEXT5 ! transition from zone 2 to 5

TMTBE3L(1,0)=HEIGHT/2-DT-RAD,HEIGHT/2-DT! kp 9 and 6 respectively

TMTBE3L(1,1)=TEXT2,TEXT5 ! transition from zone 2 to 5

SF,AZONE3L,CONV,%HFTBE3L%,%TMTBE3L%

*DIM,HFTBE3R,TABLE,2,,Z! H for tranzitional area 3R external

*DIM,TMTBE3R,TABLE,2,,Z! T for tranzitional area 3R external

HFTBE3R(1,0)=HEIGHT/2-DT-RAD,HEIGHT/2-DT! kp 10 and 7 respectively

HFTBE3R(1,1)=HEXT1,HEXT5 ! transition from zone 1 to 5

TMTBE3R(1,0)=HEIGHT/2-DT-RAD,HEIGHT/2-DT! kp 10 and 7 respectively

TMTBE3R(1,1)=TEXT1,TEXT5 ! transition from zone 1 to 5

SF,AZONE3R,CONV,%HFTBE3R%,%TMTBE3R%

*DIM,HFTBI3L,TABLE,2,,Z! H for tranzitional area 3l internal

*DIM,TMTBI3L,TABLE,2,,Z! T for tranzitional area 3l internal

HFTBI3L(1,0)=HEIGHT/2-DT-RAD,HEIGHT/2-DT+TH5! kp 27 and 24 respectively

HFTBI3L(1,1)=HINT2,HINT5 ! transition from zone 2 to 5

TMTBI3L(1,0)=HEIGHT/2-DT-RAD,HEIGHT/2-DT+TH5! kp 27 and 24 respectively

TMTBI3L(1,1)=TINT2,TINT5 ! transition from zone 2 to 5

SF,AZONI3L,CONV,%HFTBI3L%,%TMTBI3L%

*DIM,HFTBI3R,TABLE,2,,Z! H for tranzitional area 3R internal

*DIM,TMTBI3R,TABLE,2,,Z! T for tranzitional area 3R internal

HFTBI3R(1,0)=HEIGHT/2-DT-RAD,HEIGHT/2-DT+TH5! kp 28 and 25 respectively

HFTBI3R(1,1)=HINT1,HINT5 ! transition from zone 1 to 5

TMTBI3R(1,0)=HEIGHT/2-DT-RAD,HEIGHT/2-DT+TH5! kp 28 and 25 respectively

TMTBI3R(1,1)=TINT1,TINT5 ! transition from zone 1 to 5

SF,AZONI3R,CONV,%HFTBI3R%,%TMTBI3R%

*DIM,HFTBE4L,TABLE,2,,Z! H for tranzitional area 4l external

*DIM,TMTBE4L,TABLE,2,,Z! T for tranzitional area 4l external
HFTBE4L(1,0)=-HEIGHT/2+DB,-HEIGHT/2+DB+RAD! kp 14 and 11 respectively
HFTBE4L(1,1)=HEXT7,HEXT2 ! transition from zone 2 to 7
TMTBE4L(1,0)=-HEIGHT/2+DB,-HEIGHT/2+DB+RAD! kp 14 and 11 respectively
TMTBE4L(1,1)=TEXT7,TEXT2 ! transition from zone 2 to 7
SF,AZONE4L,CONV,%HFTBE4L%,%TMTBE4L%

*DIM,HFTBE4R,TABLE,2,,Z! H for tranzitional area 4R external
*DIM,TMTBE4R,TABLE,2,,Z! T for tranzitional area 4R external
HFTBE4R(1,0)=-HEIGHT/2+DB,-HEIGHT/2+DB+RAD! kp 15 and 12 respectively
HFTBE4R(1,1)=HEXT7,HEXT1 ! transition from zone 1 to 7
TMTBE4R(1,0)=-HEIGHT/2+DB,-HEIGHT/2+DB+RAD! kp 15 and 12 respectively
TMTBE4R(1,1)=TEXT7,TEXT1 ! transition from zone 1 to 7
SF,AZONE4R,CONV,%HFTBE4R%,%TMTBE4R%

*DIM,HFTBI4L,TABLE,2,,Z! H for tranzitional area 4l internal
*DIM,TMTBI4L,TABLE,2,,Z! T for tranzitional area 4l internal
HFTBI4L(1,0)=-HEIGHT/2+DB-TH7,-HEIGHT/2+DB+RAD! kp 32 and 29 respectively
HFTBI4L(1,1)=HINT7,HINT2 ! transition from zone 2 to 7
TMTBI4L(1,0)=-HEIGHT/2+DB-TH7,-HEIGHT/2+DB+RAD! kp 32 and 29 respectively
TMTBI4L(1,1)=TINT7,TINT2 ! transition from zone 2 to 7
SF,AZONI4L,CONV,%HFTBI4L%,%TMTBI4L%

*DIM,HFTBI4R,TABLE,2,,Z! H for tranzitional area 4R internal
*DIM,TMTBI4R,TABLE,2,,Z! T for tranzitional area 4R internal
HFTBI4R(1,0)=-HEIGHT/2+DB-TH7,-HEIGHT/2+DB+RAD! kp 33 and 30 respectively
HFTBI4R(1,1)=HINT7,HINT1 ! transition from zone 1 to 7
TMTBI4R(1,0)=-HEIGHT/2+DB-TH7,-HEIGHT/2+DB+RAD! kp 33 and 30 respectively
TMTBI4R(1,1)=TINT7,TINT1 ! transition from zone 2 to 7

SF,AZONI4R,CONV,%HFTBI4R%,%TMTBI4R%

*DIM,HFTBE5,TABLE,2,,Z! H for tranzitional area 5v external

*DIM,TMTBE5,TABLE,2,,Z! T for tranzitional area 5v external

HFTBE5(1,0)=HEIGHT/2-DT,HEIGHT/2! kp 5 and 1 respectively

HFTBE5(1,1)=HEXT5,HEXT6 ! transition from zone 5 to 6

TMTBE5(1,0)=HEIGHT/2-DT,HEIGHT/2! kp 5 and 1 respectively

TMTBE5(1,1)=TEXT5,TEXT6 ! transition from zone 5 to 6

SF,AZONE5V,CONV,%HFTBE5%,%TMTBE5%

*DIM,HFTBI5,TABLE,2,,Z! H for tranzitional area 5v internal

*DIM,TMTBI5,TABLE,2,,Z! T for tranzitional area 5v internal

HFTBI5(1,0)=HEIGHT/2-DT+TH5,HEIGHT/2-TH6! kp 23 and 21 respectively

HFTBI5(1,1)=HINT5,HINT6 ! transition from zone 5 to 6

TMTBI5(1,0)=HEIGHT/2-DT+TH5,HEIGHT/2-TH6! kp 23 and 21 respectively

TMTBI5(1,1)=TINT5,TINT6 ! transition from zone 5 to 6

SF,AZONI5V,CONV,%HFTBI5%,%TMTBI5%

*DIM,HFTBE7,TABLE,2,,Z! H for tranzitional area 7v external

*DIM,TMTBE7,TABLE,2,,Z! T for tranzitional area 7v external

HFTBE7(1,0)=-HEIGHT/2,-HEIGHT/2+DB! kp 17 and 13 respectively

HFTBE7(1,1)=HEXT8,HEXT7 ! transition from zone 7 to 8

TMTBE7(1,0)=-HEIGHT/2,-HEIGHT/2+DB! kp 17 and 13 respectively

TMTBE7(1,1)=TEXT8,TEXT7 ! transition from zone 7 to 8

SF,AZONE7V,CONV,%HFTBE7%,%TMTBE7%

*DIM,HFTBI7,TABLE,2,,Z! H for tranzitional area 7v internal

*DIM,TMTBI7,TABLE,2,,Z! T for tranzitional area 7v internal

!error - HFTBI7(1,0)=-HEIGHT/2+TH8-HEIGHT/2+DB-TH7! kp 35 and 31 respectively

```

HFTBI7(1,0)=-HEIGHT/2+TH8,-HEIGHT/2+DB-TH7! kp 35 and 31 respectively
HFTBI7(1,1)=HINT8,HINT7 ! transition from zone 7 to 8
!error - TMTBI7(1,0)=-HEIGHT/2+TH8,-HEIGHT/2+DB-TH7! kp 35 and 31 respectively
TMTBI7(1,0)=-HEIGHT/2+TH8,-HEIGHT/2+DB-TH7! kp 35 and 31 respectively
TMTBI7(1,1)=TINT8,TINT7 ! transition from zone 7 to 8
SF,AZONI7V,CONV,%HFTBI7%,%TMTBI7%

```

```

AUTOTS,ON

```

```

NSUBST,50

```

```

KBC,0

```

```

EQSLV,PCG

```

```

SOLVE

```

```

finish

```

```

/POST1

```

```

*DIM,NDTEMP,ARRAY,NUMNOD

```

```

*VGET,NDTEMP(1),NODE,,TEMP

```

```

/PREP7

```

```

ETCHG,TTS! changing the elements to structural

```

```

LSCLEAR,ALL! clearing the loads

```

```

/SOLU

```

```

ANTYPE,STATIC

```

```

!LDREAD,TEMP,1,,,TEST,RTH ! for some reason it still uses the name "file"

```

```

!LDREAD,TEMP,1,,,FILE,RTH ! for all the extensions so we go along with that

```

```

LDREAD,TEMP,1,,,file,rth

```

```

SFA,AZONE,,PRES,PRESIN+PRESDEL

```

```

SFA,AZONI,,PRES,PRESIN

```

```

DA,AROOT,UY,0 ! only in Y-direction whole section is restricted

```

```

DK,1,ALL,0
DK,20,ALL,0
CMSEL,S,ATIP
NSLA,S,1
DSYM,SYMM,Y
ALLSEL,ALL
AUTOTS,ON
NSUBST,50
KBC,0
EQLV,PCG
SOLVE

```

```

/post1

```

!10.17.2 let us define the goals in postprocessing information

```

*DIM,SRES,ARRAY,10
*DIM,TRES,ARRAY,10
*DIM,INOD,ARRAY,10
*DIM,NCOOR,ARRAY,10,3

```

! This is the code to find the node in a given zone with the max stress

```

! CMSEL,S,NZONE1
! NSORT,S,EQV,0,0 !first
! *GET,SRES(1),SORT,,MAX !maximum value
! *GET,INOD(1),SORT,,IMAX !corresponding node
! *GET,NCOOR(1,1),NODE,INOD(1),LOC,X
! *GET,NCOOR(1,2),NODE,INOD(1),LOC,Y
! *GET,NCOOR(1,3),NODE,INOD(1),LOC,Z
! *SET,TRES(1),NDTEMP(INOD(1))! its temperature

```

```

! CMSEL,S,NZONE2
! NSORT,S,EQV,0,0 !first
! *GET,SRES(2),SORT,,MAX !maximum value
! *GET,INOD(2),SORT,,IMAX !corresponding node
! *GET,NCOOR(2,1),NODE,INOD(2),LOC,X
! *GET,NCOOR(2,2),NODE,INOD(2),LOC,Y
! *GET,NCOOR(2,3),NODE,INOD(2),LOC,Z
! *SET,TRES(2),NDTEMP(INOD(2))! its temperature

```

```

! CMSEL,S,NZONE3
! NSORT,S,EQV,0,0 !first
! *GET,SRES(3),SORT,,MAX !maximum value
! *GET,INOD(3),SORT,,IMAX !corresponding node
! *GET,NCOOR(3,1),NODE,INOD(3),LOC,X
! *GET,NCOOR(3,2),NODE,INOD(3),LOC,Y
! *GET,NCOOR(3,3),NODE,INOD(3),LOC,Z
! *SET,TRES(3),NDTEMP(INOD(3))! its temperature

```

```

! CMSEL,S,NZONE4
! NSORT,S,EQV,0,0 !first
! *GET,SRES(4),SORT,,MAX !maximum value
! *GET,INOD(4),SORT,,IMAX !corresponding node
! *GET,NCOOR(4,1),NODE,INOD(4),LOC,X
! *GET,NCOOR(4,2),NODE,INOD(4),LOC,Y
! *GET,NCOOR(4,3),NODE,INOD(4),LOC,Z
! *SET,TRES(4),NDTEMP(INOD(4))! its temperature

```

```

! CMSEL,S,NZONE5
! NSORT,S,EQV,0,0 !first

```



```

! *GET,SRES(5),SORT,,MAX !maximum value
! *GET,INOD(5),SORT,,IMAX !corresponding node
! *GET,NCOOR(5,1),NODE,INOD(5),LOC,X
! *GET,NCOOR(5,2),NODE,INOD(5),LOC,Y
! *GET,NCOOR(5,3),NODE,INOD(5),LOC,Z
! *SET,TRES(5),NDTEMP(INOD(5))! its temperature

```

```

! CMSEL,S,NZONE6
! NSORT,S,EQV,0,0 !first
! *GET,SRES(6),SORT,,MAX !maximum value
! *GET,INOD(6),SORT,,IMAX !corresponding node
! *GET,NCOOR(6,1),NODE,INOD(6),LOC,X
! *GET,NCOOR(6,2),NODE,INOD(6),LOC,Y
! *GET,NCOOR(6,3),NODE,INOD(6),LOC,Z
! *SET,TRES(6),NDTEMP(INOD(6))! its temperature

```

```

! CMSEL,S,NZONE7
! NSORT,S,EQV,0,0 !first
! *GET,SRES(7),SORT,,MAX !maximum value
! *GET,INOD(7),SORT,,IMAX !corresponding node
! *GET,NCOOR(7,1),NODE,INOD(7),LOC,X
! *GET,NCOOR(7,2),NODE,INOD(7),LOC,Y
! *GET,NCOOR(7,3),NODE,INOD(7),LOC,Z
! *SET,TRES(7),NDTEMP(INOD(7))! its temperature

```

```

! CMSEL,S,NZONE8
! NSORT,S,EQV,0,0 !first
! *GET,SRES(8),SORT,,MAX !maximum value
! *GET,INOD(8),SORT,,IMAX !corresponding node

```

```

! *GET,NCOOR(8,1),NODE,INOD(8),LOC,X
! *GET,NCOOR(8,2),NODE,INOD(8),LOC,Y
! *GET,NCOOR(8,3),NODE,INOD(8),LOC,Z
! *SET,TRES(8),NDTEMP(INOD(8))! its temperature

! CMSEL,S,NZONE5V
! NSORT,S,EQV,0,0 !first
! *GET,SRES(9),SORT,,MAX !maximum value
! *GET,INOD(9),SORT,,IMAX !corresponding node
! *GET,NCOOR(9,1),NODE,INOD(9),LOC,X
! *GET,NCOOR(9,2),NODE,INOD(9),LOC,Y
! *GET,NCOOR(9,3),NODE,INOD(9),LOC,Z
! *SET,TRES(9),NDTEMP(INOD(9))! its temperature

! CMSEL,S,NZONE7V
! NSORT,S,EQV,0,0 !first
! *GET,SRES(10),SORT,,MAX !maximum value
! *GET,INOD(10),SORT,,IMAX !corresponding node
! *GET,NCOOR(10,1),NODE,INOD(10),LOC,X
! *GET,NCOOR(10,2),NODE,INOD(10),LOC,Y
! *GET,NCOOR(10,3),NODE,INOD(10),LOC,Z
! *SET,TRES(10),NDTEMP(INOD(10))! its temperature

! This is the current code that specifies a given node per zone to extract
! the "max" stress and temp from. The node point chose corresponds to the
! max stress node based on the baseline input variable settings
*SET,INOD(1),94
*SET,INOD(2),100
*SET,INOD(3),92

```

```

*SET,INOD(4),141
*SET,INOD(5),70
*SET,INOD(6),3
*SET,INOD(7),136
*SET,INOD(8),219
*SET,INOD(9),40
*SET,INOD(10),193
*DO,I,1,10
*GET,SRES(I),NODE,INOD(I),S,EQV !maximum value
*GET,NCOOR(I,1),NODE,INOD(I),LOC,X
*GET,NCOOR(I,2),NODE,INOD(I),LOC,Y
*GET,NCOOR(I,3),NODE,INOD(I),LOC,Z
*SET,TRES(I),NDTEMP(INOD(I))! its temperature

*ENDDO

ALLSEL,ALL
*VWRITE,INOD(1),TRES(1),SRES(1),NCOOR(1,1),NCOOR(1,2),NCOOR(1,3)
(F10.0,2X,E14.6,2X,E14.6,2X,E14.4,5X,E14.4,2X,E14.4)
finish

```

REFERENCES

- [1] Abraham, B., Chipman, H., and Vijyan, K., “Some Risks in the Construction and Analysis of Supersaturated Designs,” *Technometrics*, Vol. 41, No. 2, 1999, pp. 135–141.
- [2] ACSYNT Institute, “ACSYNT Overview and Installation Manual,” 1992.
- [3] Alvarado, F. L., “Visualizing Sparse Matrix Computations,” *Presented at the IEEE International Symposium on Circuits and Systems, 1990*, IEEE, New Orleans, LA, May 1990.
- [4] Angelis, L., Bora-Senta, E., and Moyssiadis, C., “Optimal Exact Experimental Designs with Correlated Errors Through a Simulated Annealing Algorithm,” *Computational Statistics & Data Analysis*, Vol. 37, 2001, pp. 275–296.
- [5] ANSYS, Inc., *ANSYS 7.1 Documentation*, ANSYS, Inc., 2003.
- [6] Baker, A. P., *The Role of Mission Requirements, Vehicle Attributes, Technologies and Uncertainty in Rotocraft System Design*, Ph.D. thesis, Georgia Institute of Technology, 2002.
- [7] Baker, A. P. and Mavris, D., “Assessing the Simultaneous Impact of Requirements, Vehicle Characteristics and Technologies During Aircraft Design,” *Proceeding of the 39th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA, Reno, NV, January 2001, AIAA-2001-0533.
- [8] Balevic, D., Burger, R., and Forry, D., “Heavy-Duty Gas Turbine Operating and Maintenance Considerations,” Tech. rep., GE Energy, 2004, <http://www.ge-energy.com/.../ger3620k.pdf>, [Online; accessed 8-March-2006].
- [9] Bates, R. A., Buck, R. J., Riccomagno, E., and Wynn, H. P., “Experimental Design and Observation for Large Scale Systems,” *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 58, 1996, pp. 77–94.
- [10] Bates, S. J. and Sienz, J., “Formulation of the Optimal Latin Hypercube Design of Experiments Using a Permutation Genetic Algorithm,” *Proceedings of the 45th Annual Structures, Structural Dynamics and Materials Conference*, AIAA, Palm Springs, CA, April 2004, AIAA-2004-2011.
- [11] Box, G. E. P. and Draper, N. R., *Empirical Model-Building and Response Surfaces*, John Wiley & Sons, 1987.

- [12] Box, G. E. P. and Meyer, R. D., "An Analysis for Unreplicated Fractional Factorials," *Technometrics*, Vol. 28, No. 1, 1986, pp. 11–18.
- [13] Brandt, D. E. and Wesorick, R. R., "GE Gas Turbine Design Philosophy," Tech. rep., GE Energy, 1994, <http://www.ge-energy.com/.../ger3434d.pdf>, [Online; accessed 8-March-2006].
- [14] Branin, F. H. J., "Computer Methods of Network Analysis," *Proceedings of the IEEE*, Vol. 55, 1967, pp. 1787–1801.
- [15] Brooks, F. J., "GE Gas Turbine Performance Characteristics," Tech. rep., GE Energy, 2000, <http://www.ge-energy.com/.../ger3567h.pdf>, [Online; accessed 9-March-2006].
- [16] Bucher, C. G. and Bourgund, U., "A Fast and Efficient Response Surface Approach for Structural Reliability Problems," *Structural Safety*, Vol. 7, 1990, pp. 57–66.
- [17] Burdick, R. K., *Confidence Intervals on Variance Components*, Marcel Dekker, Inc., 1992.
- [18] Burnett, D. S., *Finite Element Analysis - From Concepts to Applications*, Addison-Wesley Publishing Company, 1987.
- [19] Campolongo, F. and Braddock, R., "The Use of Graph Theory in the Sensitivity Analysis of the Model Output - A Second Order Screening Method," *Reliability Engineering and System Safety*, Vol. 64, 1999, pp. 1–12.
- [20] Cela, R., Martinez, E., and Carro, A. M., "Supersaturated Experimental Designs. New Approaches to Building and Using it Part I. Building Optimal Supersaturated Designs by Means of Evolutionary Algorithms," *Chemometrics and Intelligent Laboratory Systems*, Vol. 52, 2000, pp. 1167–182.
- [21] Chase, D. L. and Kehoe, P. T., "GE Combined-Cycle Product Line and Performance," Tech. rep., GE Energy, 2000, <http://www.ge-energy.com/.../ger3574g.pdf>, [Online; accessed 28-March-2006].
- [22] Chen, X. and Lind, N. C., "Fast Probability Integration by Three-Parameter Normal Tail Approximation," *Structural Safety*, Vol. 1, 1983, pp. 269–276.
- [23] Chua, L. O. and Chen, L. K., "Diakoptic and Generalized Hybrid Analysis," *IEEE Transactions on Circuits and Systems*, Vol. CAS-23, 1976, pp. 694–705.
- [24] Cornell, C. A., "A Probability-Based Structural Code," *Journal of the American Concrete Institute*, Vol. 66, No. 12, 1969, pp. 974–985.
- [25] Cropp, R. A. and Braddock, R. D., "The New Morris Method: An Efficient Second-order Screening Method," *Reliability Engineering and System Safety*, Vol. 78, 2002, pp. 77–83.

- [26] Davis, L. B. and Black, S. H., "Dry Low NO_x Combustion Systems for GE Heavy-Duty Gas Turbines," Tech. rep., GE Energy, 2000, <http://www.ge-energy.com/.../ger3568g.pdf>, [Online; accessed 28-March-2006].
- [27] Davis, P. K. and Bigelow, J. H., "Motivated Metamodels," *Proceedings of the 2002 PerMIS Workshop*, NIST - Intelligent Systems Division, Location N/A, August 2002.
- [28] Davis, P. K. and Bigelow, J. H., "Motivated Metamodels: Synthesis of Cause and Effect Reasoning and Statistical Metamodeling," Tech. rep., Rand, 2003.
- [29] Dieter, G. E., *Engineering Design: A Materials and Processing Approach*, McGraw-Hill, 3rd ed., 1999.
- [30] Duff, I. S., "A Survey of Sparse Matrix Research," *Proceeding of the IEE*, Vol. 65, 1977, pp. 500–535.
- [31] Engineous Software, "Homepage," <http://www.engineous.com/>, [Online; accessed 1-August-2002].
- [32] Fiduccia, C. M. and Mattheyses, R. M., "A Linear-time Heuristic for Improving Network Partitions," *Presented at the 19th Conference on Design Automation*, IEEE, Location N/A, June 1982.
- [33] Fiessler, B., Neumann, H.-J., and Rackwitz, R., "Quadratic Limit States in Structural Reliability," *Journal of the Engineering Mechanics Division*, Vol. 105, 1979, pp. 661–676.
- [34] Flowers, T. L., Anderson, A. W., and Brunso, J., "Improvement of Nozzle Life in Gas Turbines," *American Society of Mechanical Engineers*, Vol. 79-GT-50, 1979, pp. 12.
- [35] Fox, E. P., "Methods of Integrating Probabilistic Design within an Organization's Design System using Box-Behnken Matrices," *Proceedings of the 34th Annual Structures, Structural Dynamics and Materials Conference*, AIAA, La Jolla, CA, April 1993, AIAA-93-1380-CP.
- [36] Fox, E. P., "The Pratt & Whitney Probabilistic Design System," *Proceedings of the 35th Annual Structures, Structural Dynamics and Materials Conference*, AIAA, Hilton Head, SC, April 1994, AIAA-94-1442.
- [37] Fox, E. P., "On the Accuracy of Various Probabilistic Methods," *Proceedings of the 41st Annual Structures, Structural Dynamics and Materials Conference*, AIAA, Atlanta, GA, April 2000, AIAA-2000-1631.
- [38] Gamma, E., Helm, R., Johnson, R., and Vlissides, J., *Design Patterns*, Addison-Wesley, 1st ed., 1995.

- [39] GE Energy, "H Systems - World's Most Advanced Combined Cycle Technology," <http://www.ge-energy.com/.../hsystembrochure.pdf>, 2004, [Online; accessed 6-March-2006].
- [40] GE Energy, "Gas Turbine and Combined Cycle Products," http://www.ge-energy.com/.../gasturbine_cc_products.pdf, 2005, [Online; accessed 14-April-2006].
- [41] Georgia Institute of Technology, "Guidelines for Ph.D. Dissertation Research," <http://www.grad.gatech.edu/thesis/index.html>, June 2006, [Online; accessed 3-June-2006].
- [42] Gilbert, D., *The JFree Chart Class Library - A Developers Guide*, Object Refinery Limited, 2004.
- [43] Gilmour, S. G. and Mead, R., "Stopping Rules for Sequences of Factorial Designs," *Applied Statistics*, Vol. 44, 1995, pp. 343–355.
- [44] Giunta, A. A., Wojtkiewicz Jr., S. F., and Eldred, M. S., "Overview of Modern Design of Experiments Methods for Computation Simulations," *Proceedings of the 41st AIAA Aerospace Sciences Meeting and Exhibit*, AIAA, Reno, NV, January 2003, AIAA-2003-0649.
- [45] Guardabassi, G. and Sangiovanni-Vincentilli, A., "A Two Level Algorithm for Tearing," *IEEE Transactions on Circuits and Systems*, Vol. CAS-23, 1976, pp. 783–791.
- [46] Hajj, I. N., "Sparsity Considerations in Network Solution by Tearing," *IEEE Transactions on Circuits and Systems*, Vol. CAS-27, 1980, pp. 357–366.
- [47] Haldar, A. and Mahadevan, S., *Probability, Reliability and Statistical Methods in Engineering Design*, John Wiley and Sons, Inc., 2000.
- [48] Haldar, A. and Mahadevan, S., *Reliability Assessment Using Stochastic Finite Element Analysis*, John Wiley and Sons, Inc., 2000.
- [49] Hale, M. A. and Craig, J. I., "IMAGE: A Design Integration Framework Applied to the High Speed Civil Transport," *Presented at the 1st Industry/Acedemy Symposium on Research for Future Supersonic and Hypersonic Vehicles*, N/A, Greensboro, NC, December 1994.
- [50] Hale, M. A. and Craig, J. I., "Techniques for Integrating Computer Programs into Design Architectures," *Presented at the 6th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA, Bellevue, WA, September 1996, AIAA-96-4166.
- [51] Hale, M. A. and Mavris, D. N., "Enabling Advanced Design Methods in an Internet-Capable Framework," *Presented at the 4th World Aviation Congress and Exposition*, SAE/AIAA, San Francisco, CA, October 1999, SAE/AIAA-1999-01-5578.

- [52] Hale, M. A. and Mavris, D. N., "Lean-Server Foundation for Collaborative Design," *Presented at 5th NASA National Symposium on Large-Scale Analysis, Design and Intelligent Synthesis Environments*, Publisher N/A, Williamsburg, VA, October 1999.
- [53] Halton, J. H., "On the Efficiency of Certain Quasi-Random Sequences of Points in Evaluating Multi-Dimensional Integrals," *Numerische Mathematik*, Vol. 2, 1960, pp. 84–90.
- [54] Hammersley, J. M., "Monte Carlo Methods for Solving Multivariate Problems," *Annals of the New York Academy of Sciences*, Vol. 86, 1960, pp. 844–874.
- [55] Harary, F., "Sparse Matrices and Graph Theory," *Large Sparse Sets of Linear Equations*, edited by J. K. Reid, Academic Press, 1971.
- [56] Harary, F., *Graph Theory*, Adison-Wesley, 1972.
- [57] Harold, E. R. and Means, W. S., *XML in a Nutshell*, O'Reilly & Associates, 1st ed., 2001.
- [58] Hasofer, A. M. and Lind, N. C., "Exact and Invariant Second-Moment Code Format," *Journal of the Engineering Mechanics Division*, Vol. 100, 1974, pp. 111–121.
- [59] Hayter, A. J., *Probability and Statistics for Engineers and Scientists*, PWS Publishing Company, 1st ed., 1995.
- [60] Heavlin, W. D. and Finnegan, G. P., "Columnwise Construction of Response Surface Designs," *Statistica Sinica*, Vol. 8, 1998, pp. 185–206.
- [61] Helton, J. C. and Davis, F. J., "Illustration of Sampling-Based Methods for Uncertainty and Sensitivity Analysis," *Risk Analysis*, Vol. 22, 2002, pp. 591–622.
- [62] Hill, P. and Peterson, C., *Mechanics and Thermodynamics of Propulsion*, Addison Wesley, 2nd ed., 1992.
- [63] Himmelblau, D. M., "Morphology of Decomposition," *Decomposition of Large-Scale Problems*, edited by D. M. Himmelblau, North-Holland Publishing Company, 1973.
- [64] Holcomb, D. R. and Carlyle, W. M., "Some Notes on the Construction and Evaluation of Supersaturated Designs," *Quality and Reliability Engineering International*, Vol. 18, 2002, pp. 299–304.
- [65] Hoschek, W., "The Colt Distribution: Open Source Libraries for High Performance Scientific and Technical Computing in Java," <http://hoschek.home.cern.ch/hoschek/colt/>, November 2002, [Online; accessed 2-November-2005].

- [66] Huff, D., *How to Lie with Statistics*, W. W. Norton & Company, 1954.
- [67] Huntington, D. E. and Lyrintzis, C. S., "Improvements to and Limitations of Latin Hypercube Sampling," *Probabilistic Engineering Mechanics*, Vol. 13, 1998, pp. 245–253.
- [68] Iman, R. L. and Conover, W. J., "A Distribution-free Approach to Inducing Rank Correlation Among Input Variables," *Communications in Statistics: Simulation and Computation*, Vol. 11, 1982, pp. 311–334.
- [69] Jin, R., Chen, W., and Sudjianto, A., "An Efficient Algorithm for Constructing Optimal Design of Computer Experiments," *Journal of Statistical Planning and Inference*, Vol. 134, 2004, pp. 268–287.
- [70] Junger, M., Rinaldi, G., and Thienel, S., "Practical Performance of Efficient Minimum Cut Algorithms," *Algorithmica*, Vol. 26, 2000, pp. 172–195.
- [71] Kernighan, B. W. and Lin, S., "An Efficient Heuristic Procedure for Partitioning Graphs," *The Bell System Technical Journal*, Vol. 49, 1970, pp. 291–307.
- [72] Knoernschild, K., *Java Design: Objects, UML, and Process*, Addison-Wesley, 1st ed., 2001.
- [73] Koch, P. N., *Hierarchical modeling and robust synthesis for the preliminary design of large scale complex systems*, Ph.D. thesis, Georgia Institute of Technology, 1997.
- [74] Koch, P. N., Mavris, D., and Mistree, F., "Partitioned, Multilevel Response Surfaces for Modeling Complex Systems," *AIAA Journal*, Vol. 38, 2000, pp. 875–881.
- [75] Kocis, L. and Whiten, W. J., "Computational Investigations of Low Discrepancy Sequences," *ACM Transactions on Mathematical Software*, Vol. 23, No. 2, June 1997, pp. 266–294.
- [76] Kron, G., *Tensor Analysis of Networks*, John Wiley & Sons, Inc., 1939.
- [77] Kron, G., "Equivalent Circuits to Represent the Electro-magnetic Field Equations," *Physical Review*, Vol. 64, 1943, pp. 126–128.
- [78] Kron, G., "Equivalent Circuits of the Elastic Field," *Journal of Applied Mechanics*, Vol. 11, 1944, pp. A149–A161.
- [79] Kron, G., "Electric Circuit Models of the Schrodinger Equation," *Physical Review*, Vol. 67, 1945, pp. 39–43.
- [80] Kron, G., "Equivalent Circuits of Compressible and Incompressible Fluid Flow Fields," *Journal of the Aeronautical Sciences*, Vol. 12, 1945, pp. 221–231.

- [81] Kron, G., "Numerical Solution of Ordinary and Partial Differential Equations by Means of Equivalent Circuits," *Journal of Applied Physics*, Vol. 16, 1945, pp. 172–186.
- [82] Kron, G., *Diakoptics - The Piecewise Solution of Large-Scale Systems*, Macdonald & Company, Ltd., 1963.
- [83] Lefebvre, A. H., *Gas Turbine Combustion*, Taylor & Francis, 2nd ed., 1999.
- [84] Lei, S., Allidina, A. Y., and Malinowski, K., "Clustering Technique for Rearranging ODE Systems," *Parallel Processing Techniques for Simulation*, edited by M. G. Singh, A. Y. Allidina, and B. K. Daniels, Plenum Press, 1985.
- [85] Lennart, L., *System Identification: Theory for the User*, Prentice Hall, 2nd ed., 1999.
- [86] Levy, F., "Distribution of Income," <http://www.econlib.org>, 1993, Library of Economics and Liberty, [Online; accessed 14-4-2006].
- [87] Li, W. W. and Wu, C. F. J., "Columnwise-Pairwise Algorithms With Applications to the Construction of Supersaturated Designs," *Technometrics*, Vol. 39, No. 2, 1997, pp. 171–179.
- [88] Lin, D. K. J., "A New Class of Supersaturated Designs," *Technometrics*, Vol. 35, No. 1, 1993, pp. 28–31.
- [89] Lin, D. K. J., "Generating Systematic Supersaturated Designs," *Technometrics*, Vol. 37, No. 2, 1995, pp. 213–225.
- [90] Lin, Y., *An Efficient Robust Concept Exploration Method and Sequential Exploratory Experimental Design*, Ph.D. thesis, Georgia Institute of Technology, 2004.
- [91] Liu, Z., *A Methodology for Probabilistic Remaining Creep Life Assessment of Gas Turbine Components*, Ph.D. thesis, Georgia Institute of Technology, 2002.
- [92] Liu, Z. and Mavris, D. N., "A Methodology for Probabilistic Creep-Life Assessment of Hot Gas Path Components," *Proceedings of the 45th Annual Structures, Structural Dynamics and Materials Conference*, AIAA, Palm Springs, CA, April 2004, AIAA-2004-1990.
- [93] Liu, Z., Volovoi, V., and Mavris, D., "Probabilistic Remaining Creep Life Assessment For Gas Turbine Components Under Varying Operating Conditions," *Proceedings of the 43th Annual Structures, Structural Dynamics and Materials Conference*, AIAA, Denver, CO, April 2002, AIAA-2002-1277.
- [94] Madsen, H. O., Krenk, S., and Link, N. C., *Methods of Structural Safety*, Prentice-Hall, Inc., 1986.

- [95] Martin, R. C., *Agile Software Development*, Prentice Hall, 2002.
- [96] Matta, R. K., Mercer, G. D., and Tuthill, R. S., "Power Systems for the 21st Century - "H" Gas Turbine Combined-Cycles," Tech. rep., GE Energy, 2000, <http://www.ge-energy.com/.../ger3935b.pdf>, [Online; accessed 8-March-2006].
- [97] Mavris, D. N. and Bandte, O., "A Probabilistic Approach to Multivariate Constrained Robust Design Simulation," *Presented at the 1997 World Aviation Congress*, AIAA and SAE, Anaheim, CA, October 1997.
- [98] Max, D. M. and Mitchell, T. J., "Exploratory Design for Computational Experiments," *Journal of Statistical Planning and Inference*, Vol. 43, 1995, pp. 381–402.
- [99] McCullers, L. A., "Flight Optimization System, User's Guide Version 5.95," Tech. rep., NASA Langley Research Center, 2001.
- [100] McKay, M. D., Beckman, R. J., and Conover, W. J., "A Comparison of Three Methods of Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, Vol. 21, 1979, pp. 239–245.
- [101] McLaughlin, B., *Java & XML*, O'Reilly & Associates, 2nd ed., 2001.
- [102] Melchers, R. E., "Simulation in Time-invariant and Time-variant Reliability Problems," *Reliability and Optimization of Structural Systems '91: Proceedings of the 4th IFIP WG 7.5 Conference, Munich, Germany, September 11-13, 1991*, edited by R. Rackwitz and P. Thoft-Christensen, Vol. 76 of *Lecture Notes in Engineering*, Springer-Verlag, 1992.
- [103] Metropolis, N. and Ulam, S., "The Monte Carlo Method," *Journal of the American Statistical Association*, Vol. 44, 1949, pp. 335–341.
- [104] Miller, A. J., *Subset Selection in Regression*, Chapman and Hall, 1990.
- [105] Miller, A. J., "A Fedorov Exchange Algorithm for D-Optimal Design," *Applied Statistics*, Vol. 43, No. 4, 1994, pp. 669–678.
- [106] Montgomery, D. C., *Design and Analysis of Experiments*, John Wiley and Sons, Inc., 5th ed., 2001.
- [107] Morris, A. J., "Supporting Distributed Multi-disciplinary Design and Optimization Teams," *Cranfield University Aerogram*, Vol. Vol 10, March 2002, pp. 21–26.
- [108] Morris, A. J., Syamsudin, H., Fielding, J. P., Guenov, M., Payne, K. H., Deasley, P. J., and Evans, S. and Thorne, J., "MACRO - A tool to support distributed MDO," *Presented at the 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA, Long Beach, CA, September 2000, AIAA-2000-4702.

- [109] Myers, R. H. and Montgomery, D. C., *Response Surface Methodology*, Wiley-Interscience, 2nd ed., 2002.
- [110] Neter, J., Kunter, M. H., Nachtsheim, C. J., and Wasserman, W., *Applied Linear Statistical Models*, Richard D. Irwin, Inc., 4th ed., 1996.
- [111] NIST/SEMATECH e-Handbook of Statistical Methods, “A Glossary of DOE Terminology,” <http://www.itl.nist.gov/div898/handbook>, 2006, [Online; accessed 17-May-2006].
- [112] Oehlert, G. W. and Whitcomb, P., “Small, Efficient, Equireplicated Resolution V Fractions of 2^k Designs and their Application to Central Composite Designs,” <http://www.statease.com/pubs/small15.pdf>, [Online; accessed 11-May-2005].
- [113] Ogbuobiri, E. C., Tinney, W. f., and W., W. J., “Sparsity-Directed Decompositions for Gaussian Elimination of matrices,” *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-89, 1970, pp. 141–150.
- [114] OpsResearch.com, “OR-Objects,” <http://opsresearch.com/OR-Objects/index.html>, January 2000, [Online; accessed 5-November-2005].
- [115] Pallos, K. J., “Gas Turbine Repair Technology,” Tech. rep., GE Energy, 2001, <http://www.ge-energy.com/.../ger3957b.pdf>, [Online; accessed 8-March-2006].
- [116] Park, J., “Optimal Latin-hypercube Designs for Computer Experiments,” *Journal of Statistical Planning and Inference*, Vol. 39, 1994, pp. 95–111.
- [117] Parter, S., “Use of Linear Graphs in Gauss Elimination,” *SIAM Review*, Vol. 3, 1961, pp. 119–130.
- [118] Perry, L. A., Montgomery, D. C., and Fowler, J. W., “Partition Experimental Designs for Sequential Process: Part I - First Order Models,” *Quality and Reliability Engineering International*, Vol. 17, 2001, pp. 429–438.
- [119] Perry, L. A., Montgomery, D. C., and Fowler, J. W., “Partition Experimental Designs for Sequential Process: Part II - Second Order Models,” *Quality and Reliability Engineering International*, Vol. 18, 2002, pp. 373–382.
- [120] Pheonix Integration, ““Homepage”,” <http://www.phoenix-int.com/>, August 2002, [Online; accessed 8-August-2002].
- [121] Phoenix Integration, “Improving The Engineering Process with Software Integration,” <http://www.phoenix-int.com/publications/>, July 2002, [Online; accessed 1-August-2005].
- [122] Probabilistic Methods Committee of the G-11 Reliability, Maintainability, Support and Logistics Division of SAE, “Perceptions and Limitations Inhibiting the Application of Probabilistic Methods,” Tech. rep., SAE, 1998.

- [123] R., J. V. and Hung, Y., "Orthogonal-Maximum Latin Hypercube Designs," Tech. rep., Georgia Institute of Technology, 2006.
- [124] Rackwitz, R., "Reliability Analysis - Past, Present and Future," *Presented at the 8th ASCE Specialty Conference on Probabilistic Mechanics and Structural Reliability*, ASCE, South Bend, IN, July 2000, PMC2000-RRR.
- [125] Rackwitz, R., "Reliability Analysis - A Review and Some Perspectives," *Structural Safety*, Vol. 23, 2001, pp. 365–395.
- [126] Rackwitz, R. and Fiessler, B., "Structural Reliability Under Combined Random Load Sequences," *Computers & Structures*, Vol. 9, 1978, pp. 489–494.
- [127] Rajashekhar, M. R. and Ellingwood, B. R., "A New Look at the Response Surface Approach for Reliability Analysis," *Structural Safety*, Vol. 12, 1993, pp. 205–220.
- [128] Reddy, J. N., *An Introduction to the Finite Element Method*, McGraw-Hill, 2006.
- [129] Riel, A. J., *Object-Oriented Design Heuristics*, Addison-Wesley Pub Co, 1st ed., 1999.
- [130] Robinson, D. and Atcitty, C., "Comparison of Quasi- and Pseudo-Monte Carlo Sampling for Reliability and Uncertainty Analysis," *Presented at the 1999 Structures, Structural Dynamics and Materials (SDM) Conference*, AIAA, St. Louis, MO, April 1999, AIAA-99-1589.
- [131] Robinson, D. G., "A Survey of Probabilistic Methods Used in Reliability, Risk and Uncertainty Analysis - Analytical Techniques I," Tech. rep., Sandia National Laboratories, 1998, SAND98-1189.
- [132] Romero, V., "Effect of Initial Seed and Number of Samples on Simple-Random and Latin-Hypercube Monte Carlo Probabilities (Confidence Interval Considerations)," *Presented at the 8th ASCE Specialty Conference on Probabilistic Mechanics and Structural Reliability*, ASCE, South Bend, IN, July 2000, PMC2000-176.
- [133] Romero, V. J., "Characterization, Costing and Selection of Uncertainty Propagation Methods for Use with Large Computational Physics Models," *Proceedings of the 42th Annual Structures, Structural Dynamics and Materials Conference*, AIAA, Seattle, WA, April 2001, AIAA-2001-1653.
- [134] Romero, V. J., "Toward a New Class of Hybridized Reliability Methods Designed for Efficiency and Robustness with Large Computer Models," *Proceedings of the 42th Annual Structures, Structural Dynamics and Materials Conference*, AIAA, Seattle, WA, April 2001, AIAA-2001-1679.

- [135] Romero, V. J. and Bankston, S. D., "Finite-Element Progressive-Lattice-Sampling Response Surface Methodology and Application to Benchmark Probability Quantification Problems," Tech. rep., Sandia National Laboratories, 1998, SAND98-0567.
- [136] Rose, D. J. and Bunch, J. R., "The Role of Partitioning in the Numerical Solutions of Sparse Systems," *Sparse Matrices and Their Applications*, edited by D. J. Rose and J. R. Bunch, Plenum Press, 1972.
- [137] Rose, D. J. and Bunch, J. R., "Symposium on Sparse Matrices and Their Applications," *Sparse Matrices and Their Applications*, edited by D. J. Rose and R. A. Willoughby, Plenum Press, 1972.
- [138] Rosen, K. H., *Discrete Mathematics and Its Applications*, Random House, 1988.
- [139] Rosenblatt, M., "Remarks on a Multivariate Transformation," *Annals of Mathematical Statistics*, Vol. 23, 1952, pp. 470–472.
- [140] Ross, S. M., *Simulation*, Academic Press, 3rd ed., 2002.
- [141] Sacks, J. and Schiller, S. B. and Welch, W. J., "Designs for Computer Experiments," *Technometrics*, Vol. 31, No. 1, 1989, pp. 41–47.
- [142] Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P., "Design and Analysis of Computer Experiments," *Statistical Science*, Vol. Vol 4, No. 4, 1989, pp. 409–435.
- [143] Saltelli, A., Chan, K., and Scott, E. M., *Sensitivity Analysis*, John Wiley & Sons, 2000.
- [144] Sangiovanni-Vincetelli, A., Chen, L. K., and Chua, L. O., "An Efficient Heuristic Cluster Algorithm for Tearing Large-Scale Networks," *IEEE Transactions on Circuits and Systems*, Vol. CAS-24, 1977, pp. 709–717.
- [145] SAS Institute Inc., "JMP - The Statistical Discovery Software," <http://www.jmp.com/>, June 2004.
- [146] Schilke, P. W., "Advanced Gas Turbine Materials and Coatings," Tech. rep., GE Energy, 2004, <http://www.ge-energy.com/.../ger3569g.pdf>, [Online; accessed 28-March-2006].
- [147] Schueller, G. I., Bucher, C. G., U., B., and Ouypornprasert, W., "On Efficient Computational Schemes to Calculate Structural Failure Probabilities," *Probabilistic Engineering Mechanics*, Vol. 4, 1989, pp. 10–18.
- [148] Schueller, G. I. and Stix, R., "A Critical Appraisal of Methods to Determine Failure Probabilities," *Structural Safety*, Vol. 4, 1987, pp. 293–309.

- [149] Scott, A. T. and Olds, J., "An Evaluation of Three Commercially Available Integrated Design Framework Packages for Use in the Space Systems Design Lab," Tech. rep., Georgia Institute of Technology, 2001.
- [150] Shin, H., "A Simple Yet Effective Technique for Partitioning," *IEEE Transactions on Very Large Scale Integrated (VLSI) Systems*, Vol. 1, 1993, pp. 380–386.
- [151] Shonkwiler, R., "Math 4255 Class Notes - Monte Carlo Methods," Tech. rep., Georgia Institute of Technology, 2004, <http://www.math.gatech.edu/~shenk/mcmain.pdf>, [Online; accessed 12-October-2005].
- [152] Siemens, "Combined Cycle Plant Ratings," <http://www.powergeneration.siemens.com>, 2006, [Online; accessed 23-May-2006].
- [153] Sobieszczanski-Sobieski, J., "Multidisciplinary Design Optimization: An Emerging New Engineering Discipline," *Presented at the World Congress on Optimal Design of Structural Systems*, Kluwer, Location N/A, August 1993.
- [154] Sobieszczanski-Sobieski, J., Agte, J. S., and Sandusky Jr., R. R., "Bilevel Integrated System Synthesis," *AIAA Journal*, Vol. Vol 38, 2000, pp. 164–172.
- [155] Sonntag, R. E. and Van Wylen, G. J., *Introduction to Classical Thermodynamics*, John Wiley & Sons, 3rd ed., 1991.
- [156] South West Research Institute, *Nessus Theoretical Manual*, 7th ed., October 2001.
- [157] Space Science and Engineering Center, "VisAD - Homepage," <http://www.ssec.wisc.edu/~billh/visad.html>, May 2002, [Online; accessed 1-May-2002].
- [158] Stat Ease, "Design Expert 7.0," <http://www.statease.com/>, 2005, [Online; accessed 5-November-2005].
- [159] Steward, D. V., "On an Approach to Techniques for the Analysis of the Structure of Large Systems of Equations," *SIAM Review*, Vol. 4, 1962, pp. 321–342.
- [160] Steward, D. V., "Partitioning and Tearing Sysytems of Equations," *Journal of the Society for Industrial and Applied Mathematics: Series B, Numerical Analysis*, Vol. 2, 1965, pp. 345–365.
- [161] Sun Microsystems Inc., "Java," <http://java.sun.com/>, May 2002, [Online; accessed 5-May-2002].
- [162] TechnoSoft Inc., "Homepage," <http://www.technosoft.com/>, August 2002, [Online; accessed 1-Ausust-2002].

- [163] Tewarson, R. P., "Sorting and Ordering Sparce Linear Systems," *Large Space Sets of Linear Equations*, edited by J. K. Reid, Academic Press, 1971.
- [164] Tewarson, R. P., *Sparse Matraces*, Academic Press, 1973.
- [165] Thacker, B. H., Riha, D. S., Millwater, H. R., and P., E. M., "Errors and Uncertainties in Probabilistic Engineering Analysis," *Presented at the 42nd Structures*, AIAA, Seattle, WA, April 2001.
- [166] The Apache Software Foundation, "The Jakarta Project," <http://jakarta.apache.org/>, May 2002, [Online; accessed 5-May-2002].
- [167] The Mathworks, "lhsdesign" - Matlab 7.0.1 Documentation," <http://www.mathworks.com/.../toolbox/stats/lhsdesign.html>, 2006, [Online; accessed 26-May-2006].
- [168] Turns, S. R., *An Introduction to Combustion - Concepts and Applications*, McGraw-Hill, Inc., 1996.
- [169] Valenti, M., "Reaching for 60 Percent," *Mechanical Engineering*, Vol. 124, 2002, pp. 35.
- [170] Volovio, V. V., Waters, M., and Mavris, D. N., "Comparative Assessment of Direct and Indirect Probabilistic Methods for Thermomechanical Analysis of Structural Components in Gas Turbines," *Proceedings of the ASME Turbo Expo 2003*, ASME, Atlanta, GA, June 2003, GT2003-38510.
- [171] Volovoi, V., Zentner, J., and Mavris, D., "Assessment of the Efficiency of Partitioned Response Surface Equations in Structural Applications," *Proceedings of the 44th Annual Structures, Structural Dynamics and Materials Conference*, AIAA, Denver, CO, April 2003, AIAA-2003-1751.
- [172] Von Mises, R., *Probability, Truth and Statistics*, Dover Publications, Inc., 1957.
- [173] Wallace, J. M., *A Framework for Conducting Mechanistic Based Reliability Assessments of Components Operating in Complex Systems*, Ph.D. thesis, Georgia Institute of Technology, 2003.
- [174] Wallace, J. M., "Improved Component Reliability Assessments Using Joint Probability Modeling," *Proceedings of the 45th Annual Structures, Structural Dynamics and Materials Conference*, AIAA, Palm Springs, CA, April 2004, AIAA-2004-1897.
- [175] Wallace, J. M. and Mavris, D., "Creep Life Uncertainty Assessment of a Gas Turbine Airfoil," *Proceedings of the 44th Annual Structures, Structural Dynamics and Materials Conference*, AIAA, Norfolk, VA, April 2003b, AIAA-2003-1484.

- [176] Weisstein, E. W., "Function." From MathWorld—A Wolfram Web Resource." <http://mathworld.wolfram.com/Function.html>, May 2006, [Online; accessed 25-May-2006].
- [177] Weisstein, E. W., "Gaussian Elimination." From MathWorld—A Wolfram Web Resource," <http://mathworld.wolfram.com/GaussianElimination.html>, 2006, [Online; accessed 21-May-2006].
- [178] Weisstein, E. W., "Linear System of Equations." From MathWorld—A Wolfram Web Resource." <http://mathworld.wolfram.com/LinearSystemofEquations.html>, 2006, [Online; accessed 21-May-2006].
- [179] Wickens, T. D., *The Geometry of Multivariate Statistics*, Lawrence Erlbaum Associates, 1995.
- [180] Wikipedia, "Peaking power plant — Wikipedia, The Free Encyclopedia," http://en.wikipedia.org/wiki/Peaking_power_plant, 2005, [Online; accessed 12-April-2006].
- [181] Wikipedia, "Base load power plant — Wikipedia, The Free Encyclopedia," http://en.wikipedia.org/wiki/Base_load_power_plant, 2006, [Online; accessed 12-April-2006].
- [182] Wikipedia, "Incidence matrix — Wikipedia, The Free Encyclopedia," http://en.wikipedia.org/wiki/Incidence_matrix, 2006, [Online; accessed 19-May-2006].
- [183] Wikipedia, "NP-complete — Wikipedia, The Free Encyclopedia," <http://en.wikipedia.org/wiki/NP-Complete>, 2006, [Online; accessed 21-May-2006].
- [184] Wikipedia, "Plasma (physics) — Wikipedia, The Free Encyclopedia," http://en.wikipedia.org/wiki/Plasma_Physics, 2006, [Online; accessed 14-April-2006].
- [185] Wikipedia, "Sparse Matrix — Wikipedia, The Free Encyclopedia," http://en.wikipedia.org/wiki/Sparse_matrix, 2006, [Online; accessed 19-May-2006].
- [186] Wolfram Research, "ArrayPlot — Mathematica 5.2 Documentation," <http://documents.wolfram.com/mathematica/.../ArrayPlot.html>, 2006, [Online; accessed 22-May-2006].
- [187] World Wide Web Consortium (W3C), "Simple Object Access Protocol (SOAP) 1.1," <http://www.w3.org/TR/soap/>, May 2000, [Online; accessed 5-May-2005].

- [188] World Wide Web Consortium (W3C), “Extensible Markup Language (XML) 1.0 (Second Edition),” <http://www.w3c.org/TR/REC-xml/>, May 2002, [Online; accessed 5-May-2005].
- [189] Wu, C. F. J. and Chen, Y., “A Graph-aided Method for Planning Two-level Experiments when Certain Interactions are Important,” *Technometrics*, Vol. 34, 1992, pp. 162–175.
- [190] Wu, C. F. J. and Hamada, M., *Experiments: Planning, Analysis, and Parameter Design Optimization*, Wiley-Interscience, 1st ed., 2000.
- [191] Wu, F. F., “Solution of Large-Scale Networks by Tearing,” *IEEE Transactions on Circuits and Systems*, Vol. CAS-23, 1976, pp. 706–713.
- [192] Wu, Y. T., “Computational Methods of Efficient Structural Reliability and Reliability Based Sensitivity Analysis,” *AIAA Journal*, Vol. 32, 1994, pp. 1717–1723.
- [193] Wu, Y. T., Millwater, H. R., and Cruse, T. A., “Advanced Probabilistic Structural Analysis Method for Implicit Performance Functions,” *AIAA Journal*, Vol. 28, 1990, pp. 1663–1669.
- [194] Wu, Y. T. and Wirsching, P. H., “New Algorithm for Structural Reliability Estimation,” *Journal of Engineering Mechanics*, Vol. 113, 1987, pp. 1319–1336.
- [195] Ye, K. Q., “Orthogonal Column Latin Hypercubes and Their Application in Computer Experiments,” *Journal of the American Statistical Association*, Vol. 93, No. 444, Dec. 1998, pp. 1430–1439.
- [196] Zecevic, A. I. and Siljak, D. D., “Balanced Decomposition of Sparse Systems for Multilevel Parallel Processing,” *IEEE Transactions on Circuits and Systems*, Vol. CAS-41, 1994, pp. 220–233.
- [197] Zentner, J., De Baets, P. W. G., and Marvris, D., “Formulation of an Integrating Framework for Conceptual Object-Oriented Systems Design,” *Presented at the SAE Aerospace Congress & Exhibition*, SAE, Montreal, Quebec, Canada, September 2003, SAE-2003-01-3053.
- [198] Zhao, Y. G. and Ono, T., “A General Procedure of First and Second-order Reliability Method (FORM-SORM),” *Structural Safety*, Vol. 21, 1999, pp. 95–112.