

IDEA: INTERPOLATING DATA FOR ENERGY ANALYSIS

Eleni Primikiri⁺, Michael Kokkolaras^{*} and Panos Y. Papalambros^{*}

⁺Taubman College of Architecture and Urban Planning

^{*}Department of Mechanical Engineering

University of Michigan

Ann Arbor, MI, 48109, USA

ABSTRACT

In recent years major advances have been made in the development of computational tools for architects to facilitate building performance evaluation. Most tools require expert knowledge and remain accessible primarily to building engineers or specialized architects. This paper presents a methodology of interpolating data for energy analysis to enable easier implementation of building simulation in the design process. Architects and other building project participants can visualize and analyze complex thermal performance of buildings for different design scenarios, requiring little specialized knowledge and without a burdensome computational cost.

INTRODUCTION

Recent advances in the building industry have focused on energy conscious design. The resulting simulation tools have had limited impact on building design in practice because they lack the ability to represent simulation results adequately. (Mahdavi and Gurtekin, 2001). Also, most detailed simulations are not readily accessible to the typical practicing architect, as the need for specialized knowledge has increased significantly. Most effort in building performance visualization has focused in one or two dimensional representation of data derived from the analysis rather than three-dimensional representation that would be more useful to designers. (Primikiri and Harris, 2002)

This paper does not address visualization data directly; rather, it looks at how to make building performance data available to the architect with the least possible specialized knowledge. The key for overcoming the lack of design oriented building performance simulation tools is not entirely due to lack of visualization means but also due to lack of an efficient methodology for generating performance data. Such methodologies would support design decision-making and create opportunities for optimization as part of the design process.

The proposed methodology of interpolating data for energy analysis enables implementation of building

thermal performance analysis, visualization and optimization in the design process. The approach involves the generation of simplified models using data extracted from running expensive simulations. These models are commonly known as surrogate models, metamodels or responsive surface models (Papalambros and Wilde, 2000, Sasena, 1998)

The resulting prediction model replaces the computationally expensive simulations and delivers prompt responses to design changes that enable the evaluation of alternative design scenario. An additional strength of this approach is an ability lacking in current tools namely to handle easily the complex interactions of multiple design variables. This, in turn, allows the creation of design optimization problems that can be solved efficiently to determine the best alternative design scenario. (Papalambros and Wilde, 2000)

The rest of the paper is organized as follows. The next section describes the proposed methodology in details. Each step is analyzed, namely the theoretical background, the implementation steps and the metamodel type selection. An example to test this methodology is described in the Case Study section. A simple room is selected for experimental purposes and the methodology is applied. The results of the case study are mentioned in the last section, where the capabilities and limitation of this methodology are discussed.

METHODOLOGY

Theoretical Background

Surrogate models help designers understand the relationships among design parameters in order to specify the system with the desired behavior (Pacheco et al, 2001).

The common objective in many simulation studies tends to be towards attaining a level of understanding for the nature of relationships between the input variables and the output values of the real system that is under study. Although the simulation model itself is a simplification of the real-world system, it can be sufficiently complex for understanding

multidimensional input-output interrelations: “Sometimes a simpler (analytic) model may be used as an auxiliary to the simulation model in order to better understand the more complex model and to provide a framework for testing hypothesis about it; This auxiliary model is sometimes referred to as a Metamodel” (Friedman, 1996).

Metamodels can be often useful in conjunction with Computational Fluid Dynamics (CFD) simulations for fluid and thermal flow analysis that is computationally expensive. This is indeed the case for thermal analysis of buildings and the article focuses on the development of a surrogate model from CFD simulations.

Assuming for simplicity of explanation, a **real system** with only one output variable the input-output relationship can be described as

$$\mu = \phi(x_1, x_2, \dots, x_q) \quad (1)$$

where μ represents the real system response, and x_q 's are the factors (variables) that determine the value of that response. The function ϕ describes the effect of the factors on the response (Friedman, 1996).

In the **simulation model** the function ϕ is unknown to the modeler, and is approximated as closely as possible by f ,

$$y = f(x_1, x_2, \dots, x_q, \varepsilon) \quad (2)$$

where y is the simulation response, q is the number of input variables x_q , and ε represents the simulation error (Friedman, 1996).

In the **metamodel** simulation the unknown function f that represents the real system ϕ is approximated by another function \hat{f} but in this case only with respect to the input variables (x_k) that affect the output values.

$$y = \hat{f}(x_1, x_2, \dots, x_k, \varepsilon) \quad (3)$$

where y is the simulation output, k is the number of input variables x_k , and ε represents the metamodel error. (Friedman, 1996)

The effectiveness of this methodology allows as aforementioned for detailed understanding of the model under study. It allows for generalizations, it facilitates responds in what if questions and provides the ability for multiple hypothesis testing.

Implementation Steps

Three principal steps are typically taken in creating a surrogate model. The first and most important step is deciding the type of metamodel to be used.

For this model, sample data points are required. The identification and the selection of these points from the input design space is a task completed in the second step. Although there exist several methods that are commonly used for this type of selections, the Design of Experiments (DOE) methods are helpful in determining the set of input variables that best represent the design space. The terms “design space” in this instance do not refer to the geometric space; rather the space defined by the input variables.

Finally, after the metamodel is selected and created its effectiveness is examined on the final step with a simulation validation technique.

Metamodel selection

There are several types of metamodels typically used in engineering. Some of them will be described in order to give an overview of what is available and to explicate the final selection.

The most common and simplest metamodel is the use of polynomials. For a given data set, one can determine the regression coefficients by minimizing the mean of the sum of squared errors of the predicted output values for the data set. This technique is called least-square fit. This technique is simple to use, the underlying theory is easy to understand and is one of the most computationally inexpensive metamodels. However, when the data sets are large, or more terms are added to the polynomial, the prediction becomes less accurate and thus less efficient. (Sasena, 1998)

Another type of metamodel that is currently widely used is called Kriging. It is a type of metamodel based on Spatial Correlation Functions (SCFs). The main idea behind SCF metamodels is that the errors in the predicted values of a set of simulation runs are not independent, but comprise a systematic function of the input variables. While the error is treated systematically so that the uncertainty can be taken into account in the Kriging technique, the underlying theory can be complex for use and computationally costly (Sasena, 1998).

A different type of metamodel is the Artificial Neural Network. ANNs are non-linear regression models used for function estimation, complex curve fitting and pattern recognition. They are nonparametric models and their weights and biases have no particular meaning in relation to the problems to which they are applied (Orr, 1996). They are based on the “black box” technique where the primary focus is on the understanding of input-

output system relations rather than the underlying mathematical equations that describe it. Due to this, ANN is selected as the metamodel for the proposed methodology

To use this metamodel, a set of input variables and its corresponding output values is required. The network is then “mapped” and “trained” so that it allows for new output values or responses to be predicted for a given input set. For a functional relation of the type,

$$y = f(x) \quad (4)$$

the independent variable is x ($x = [x_1 \ x_2 \ \dots \ x_q]$) and the dependent is y .

Thus, the training set consists of paired values of the independent (input) variables and the dependent (output) variables.

$$T = \{(x_i, \hat{y}_i)\}_{i=1}^p \quad (5)$$

where x_i are the input variables; \hat{y}_i are the corresponding output with the noise; and p is the number of pairs used for training. The noise in this case represents the assumed error of the predicted values (Orr, 1996). In fact, the training set only specifies \hat{y}_i which is equal to y_i plus a small amount of unknown noise. This error will be assessed in the third step of the metamodel development. This is facilitated by ANN’s property that allows one portion of the data to be used for training the neural network while the rest can be used for testing the accuracy of the model.

Design of Experiments

To determine which sample points are appropriate for training the ANN, DOE techniques are going to be used. There are three important criteria that should be taken under consideration (Lumani et al, 1995):

- The samples must portray nonlinearity: The number of input variables determines nonlinearity and can be captured by the samples.
- Samples must be spread over the input design space ensuring symmetry of the resulting distribution. To enable the model to conduct a sound prediction it has to be trained using data sampled over the entire input design space.
- A minimum number of training samples should be used. Most of the times data collection can be expensive and time consuming.

Random sampling is a widely used technique for generating training samples in ANN. This sampling technique has many drawbacks, among which the most important is its inability for guaranteed symmetry in the entire input space (Lumani et al,

1995). The samples are distributed over the design space, but not necessarily evenly spread.

Alternatively, factorial designed experiments are commonly used for balanced sampling. These techniques are popular in the statistical literature and are primarily used in linear models. The use of factorial design for nonlinear models requires multiple levels for each input variable and can be costly if the model requires many input variables and levels (Lumani et al, 1995). For example, a full factorial design for d input variables and L input levels for each input variable requires L^d number of samples.

A common technique used in ANN is the Latin Hypercube Sampling (Lh). Lh allows samples to be selected for a design space with N dimensions exceedingly sparsely at M points (Lumani et al, 1995).

More specifically, in Lh the first point is selected randomly, from the M^N nodes of the design. This point is considered a representative of its strata leaving a $(M - 1)^N$ new design space. The same procedure is followed until the design space is empty and there are no more levels.

For example, in Figure 1, assuming a case of two input variables x_1 and x_2 the variable design space is described by a grid of 9^2 points. In a full factorial technique, all of the 9^2 points would have to be used. In Lh, only 9 points are enough to describe all the design space.

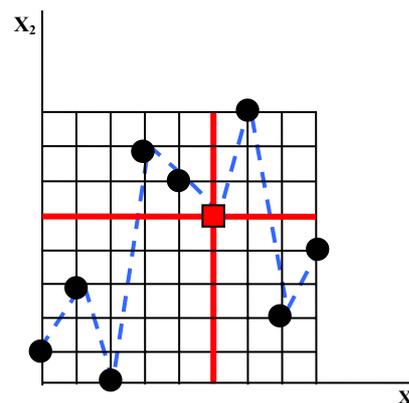


Figure1. Latin Hypercube sample points selection in a x-y dimension space

The first sample point is selected randomly (square dot on Figure 1) and is considered the representative of the row and column of its level. Thus by eliminating these axes, the grid size is 8^2 . This procedure is followed until there are no more points

left. On Figure 1, the points selected represent one of the many scenarios that can be generated using Lh.

There are more than one Latin hypercubes that can be generated for the same experiment (see Figure 2). The criterion posed on the selection of the Lh that best represents the design space is to maximize the minimum (Euclidean) distance of the sample points. The maximum minimum distance would mean that the sample points are more scattered in the design space. In Figure 2, the first Lh has the maximum minimum Euclidean distance, and thus is the best. This criterion is mainly used so that extreme cases like the second graph on the same figure can be avoided.

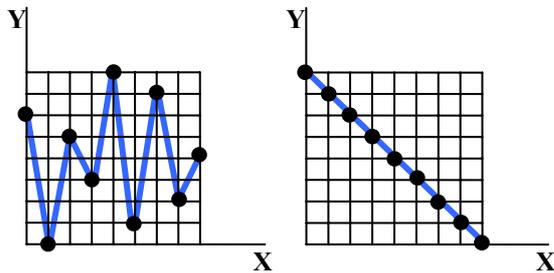


Figure 2 Two different Latin hypercubes. The Euclidean distance is shown by the line

This technique succeeds in fulfilling all aforementioned criteria and thus it is going to be used in this methodology.

Validation

After the sample points are selected and the ANN is created, an important step is to examine the metamodel's validity. The metamodel would need to be tested for two types of validity, *internal* and *external*. (Friedman, 1996) *Internal* validity indicates the degree of error of the metamodel while approximating the simulation model. *External* validity reflects the degree to which the metamodel accurately approximates the real-world system. In this paper only internal validity will be assessed.

After the ANN is prepared the model is ready to be used for prediction on new sample points. However, frequently models fail to perform as well on new data. To estimate this error, there are two practical and commonly used techniques, cross validation and the *jackknife* approach.

In the cross-validation technique, if the data is not scarce, the set of available input-output measurements can be divided into two parts: one part for training and one part for monitoring the errors (Orr, 1996). Thus, all different models trained in the training set can be compared on the test set. The

portion of the data to be kept for validation is varying.

One variation is to partition the original set of p patterns into a training set of size $p-1$ and a test of size 1 and average the absolute error or the relative error on the left-out pattern over the p possible ways of obtaining such a partition. This variation is intended to avoid the possible bias introduced by replying in any one particular division into test and train components. This technique is called leave-one-out (LOO) cross-validation (Orr, 1996) or else jackknife approach (Friedman, 1996).

The advantage of this method lies in the fact that all the data can be used for training and none has to be held back in a separate test set. This is the main reason for employing this technique for this model formulation. With the use of absolute-error and relative error calculations the model can be validated.

When x is the input, the trained model predicts the output $f(x)$. Since the true output $y(x)$ is given, the absolute-error can be calculated as follows:

$$AE = |y(x) - f(x)| \quad (6)$$

AE give a suggestion on how good the prediction is. The calculation of the relative is a better indication for the accuracy of the prediction:

$$RE = \frac{y(x) - f(x)}{y(x)} \quad (7)$$

CASE STUDY

While the proposed methodology seems feasible on a theoretical level, the most challenging task appears on its implementation. Thus, a case study is conducted to demonstrate the functionality of the methodology and to determine the limitations and advantages of such approach.

This simulation selection is based on the fact that CFD simulations require plenty of time to generate. Thus approximating, interpolating or extrapolating the flow fields using intermediate existing flow fields would be an effective way of obtaining an understanding of the flow behavior with changes in parameters such as geometry and time. (Shahnawaz, 1999).

The case study considers a room of 10x15 square feet, for velocity magnitude prediction with inlet and outlet ventilation ducts. Properties of the diffusers like the diameter, their position and the inlet air flow rate and temperature, are controlled.

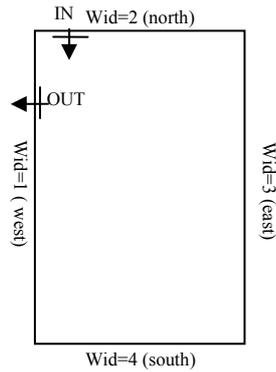


Figure 3 The room model with its variables

There is a mix of discrete and continuous variables used: The wall number of inlet duct (wid) and the wall number of the outlet duct (wod) represents the orientation (Figure 3). The normalized position of the inlet duct (pid), the normalized position of the outlet duct (pod), the inlet duct diameter (did) and the outlet duct diameter (dod) are geometric variables. Finally the inlet duct velocity (iv) and inlet duct temperature (it) are also varying. Velocity magnitude will be used for the output value. The bounds of these variables are listed on Table 1.

Table 1.
The boundaries of the 8 input variables
 $X = [wid, wod, pid, pod, did, dod, it, iv]$

	wid	wod	pid	pod	did	dod	It	iv
Ub	1	1	0.1	0.1	0.6	0.6	68	1.5
Lb	4	4	0.9	0.9	0.9	0.9	78	2.5

While preparing the ANN, the most difficult task is to create the training sets. As described before in the methodology section, the next task is the selection of sample points. The number of sample points (p in equation 5) used for the ANN training is decided empirically. One hundred Latin hypercubes were generated and the one with the maximum Euclidean distance has been chosen as the best.

The next task after selecting the sample points which is a set of input vectors is to generate the geometric models for the CFD simulation for each of these vectors. GAMBIT is chosen as the modeling tool (FLUENT, 1999). An automated routine has been created to facilitate this process. In GAMBIT a simple mesh technique is used that creates mesh fields of 925 nodes. In Figure 4 one case is illustrated with the inlet and outlet ducts shown by the arrows.

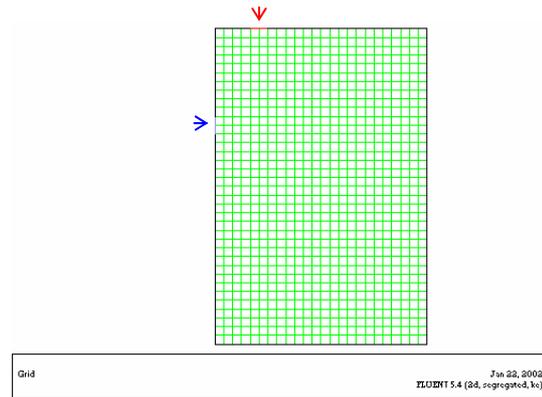


Figure 4. Mesh example

After the geometric models are created they are simulated in FLUENT, for generating the output; the velocity magnitudes (FLUENT, 1999). It is assumed that the outside temperature of the walls was 60° F, and the wall material is the same for all four walls using an 8-inch common brick wall the properties of which are shown on Table 2. The calculations are based on a steady state flow. Another automated routine has been created to store all velocity magnitude data.

Table 2
The values the 8-inch common brick wall properties

Specific Heat	Cp	0.2 Btu/lb*F
Density	P	120 lb/ft ³
Thermal Conductivity	K	0.42 Btu/h*ft*F

With this step all data are available for the ANN's training. The final task is the validation. With the leave-one-out technique the absolute value and the relative error are shown on Figure 5 and 6 respectively. The absolute error values are between 0 and 0.07 except of 4 cases that are between 0.15 and 0.25 and one case that is close to 0.42. Similarly for the relative error it is between a range of 0 to 12% except of 2 cases that are close to 30%, one case close to 50% and one close to 80%. Overall these results were expected and are satisfactory.

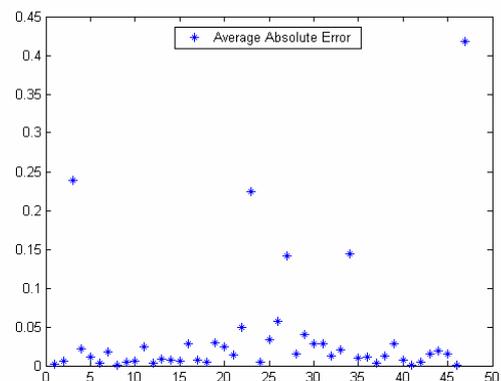


Figure 5 Absolute error of the 50 samples

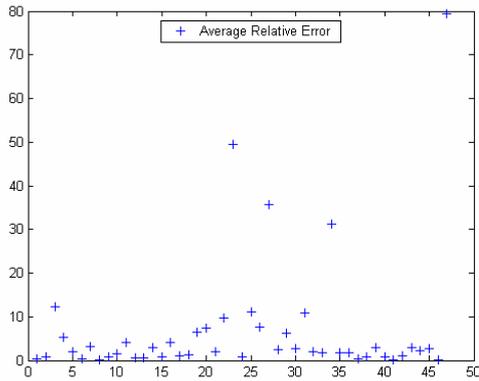


Figure 6 Absolute value of relative error of the 50

Figures 7 and 8 compare the output of a sample case simulated in FLUENT to the one predicted from ANN. The match is acceptable.

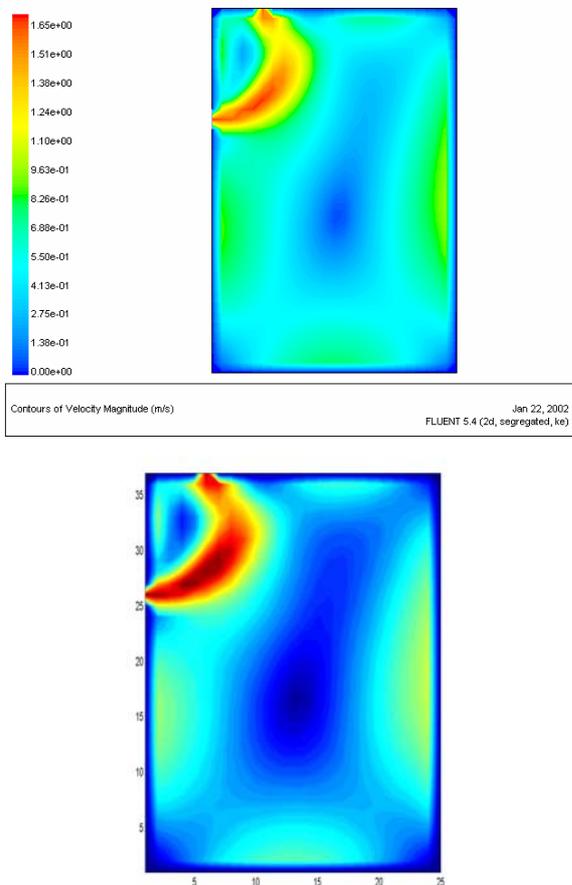


Figure 7 and 8 Two velocity magnitude fields are shown generated from Fluent and from predicting model respectively

Even in the case of a 0.2 average absolute error and of a 50% relative error, the comparison shown on Figures 9 and 10 indicates that a practicing architect can depend on the predicted output to make decisions for the building and obtain an overall idea of the flow inside the room.

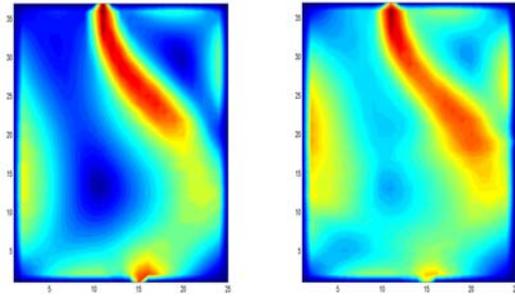


Figure 9 and 10 The velocity magnitude of a case with 0.2 absolute error and of 50% relative error.

The automated method for building the geometric model with GAMBIT as well as with FLUENT simulation was done using C++. For the development of the ANN, the Matlab neural network toolbox (Mathworks, 2002) was used. A routine was used for validating the model. Finally for getting the output of FLUENT and feeding it to the Matlab, a sorting function was used, again implemented in C++.

CONCLUSIONS, LIMITATIONS AND FUTURE WORK

The methodology was applied in a very simplified example, of a two dimensional fixed size room space. Summarizing the steps that needed to be followed from a developer's perspective one should:

1. Select the input variables and their bounds
2. Select the desired output type (e.g temperature, velocity magnitude, etc.)
3. Select the metamodel to be used
4. Select a DOE to generate the sample points to be used in the metamodel
5. Simulate the sample points to get the output for the training sets
6. Conduct a validation

The metamodel is then ready to be used for predicting selected values. To deal with more complex scenarios of varying room sizes, or shapes, or even materials, a "library" of neural networks for predicting flow should be used. In the case study analyzed on the previous section, velocity magnitude was the output. Alternatively one could predict temperature distribution or any other FLUENT output. With this option a more global analysis is enabled. Getting this methodology a step further a different simulation could be used for lighting or acoustics' analysis.

From a user's perspective, once this methodology is automated, all that is needed is the selection of input values that are within the predefined bounds.

There are several limitations that were observed while testing the methodology. Some of them were clearly technical and can be overcome, but some of them are directly related to the methodology itself. So far a very simple two dimensional case has been studied that demonstrated acceptable results. However, it is uncertain how the system will perform when considering more complex scenarios for example with different room sizes. A valid argument can be that one cannot define the size of the “library” of such cases. This is an issue that must be addressed separately.

The methodology introduced in this paper comprises a useful tool for the prediction of energy data in buildings. This will enable the implementation of optimization techniques to support design decision-making. It provides a faster interaction interface for selecting building geometry and material properties than previously proposed techniques. The goal is better representation of data and less need for specialization: giving the designer rapid feedback while refining designs, and enabling informed choices to achieve optimum building performance.

ACKNOWLEDGMENTS

The authors would like to thank Horace H. Rackham School of Graduate Studies, the Department of Mechanical Engineering and Taubman College of Architecture and Urban Planning, of the University of Michigan for their support in this work.

The authors would also like to express their gratitude to prof. Ali Malkawi for his valuable contribution throughout the project completion.

REFERENCES

- FLUENT Inc., NH, USA, 1999.
- L. W. Friedman, 1996. *The simulation Metamodel*, Kluwer Academic Publishers, Nowell, MA
- Mathworks, *Matlab Release 12*, The MathWorks Inc., 2002, USA
- M. Lumani, A. Sudjianto and P. L. Johnston, 1995. “*Generating Efficient Training Samples for Neural Networks Using Latin Hypercube Sampling*”, Proceedings of ANNIE'95 Artificial Neural Networks in Engineering, September, 1995.
- E. Linden, J. Hellstrom, M. Cehlin, M. Sandberg, “*Virtual Reality Presentation of Temperature Measurements in a Diffuser for Displacement Ventilation*” 4th International Conference on Indoor Air Quality, Ventilation & Energy Conservation in Buildings, Changsha, Hunan, China, 2001
- A. Mahdavi, B. Gurtekin, “*Computational Support for the Generation and Exploration of the Design-Performance Space*”, Proceedings of the 7th International Building Simulation Conference (IBPSA), Rio de Janeiro, Brazil, August 2001.
- M. J. L. Orr, *Introduction to Radial Basis Function Networks*, Centre for Cognitive Science, University of Edinburgh, April 1996.
- J. E. Pacheco, C. H. Amon, S. Finger, “*Developing Bayesian Surrogates for Use in Preliminary Design*” Proceedings, DETC 2001, ASME 2001 Design Engineering Technical Conference and Computers and Information in Engineering Conference, Pittsburgh, PA, September 2001.
- P. Y. Papalambros, D.J. Wilde, *Principles of Optimal Design*, Cambridge University Press, New York, 1991.
- E. Primikiri and J. Harris, 2002, *Visualizing CFD data in the CAVE Automated Virtual Environment*, Proceedings of the Australian and New Zealand Architecture Science Association, Geelong, Australia, November, 2002.
- M. J. Sasena, 1998, *Optimization of Computer Simulations via Smoothing Splines and Kriging Metamodels*, Master’s Thesis, University of Michigan, .
- T. W. Simpson, 1998, *A concept Exploration Method for Product Family Design*, PhD Thesis, Georgia Institute of Technology, September, 1998.

