Proceedings of the 1997 Winter Simulation Conference
ed. S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson

DESIGN OF EXPERIMENTS FOR FITTING SUBSYSTEM METAMODELS

Russell R. Barton
Department of Industrial and Manufacturing Engineering
The Pennsylvania State University
University Park, PA 16802 U.S.A.

ABSTRACT

For complex systems, traditional methods of experience-based design are ineffective: the design task must be supported by simulations. Conceptual design and system-level detailed design based on numerical simulation models is limited because of the difficulty in integrating disparate subsystem models to predict overall system behavior. A metamodel-based integration strategy allows simulation results from multiple submodels to be combined into a system-level simulation. The development of a metamodel-based integration strategy for system-level design depends on effective experiment design strategies for fitting and updating subsystem metamodels.

1 INTRODUCTION

Complex numerical and/or discrete-event simulation models of proposed or existing real systems are often used to estimate the effects on system performance due to changes to the system design. For complex systems, it is often the case that no single system-level model exists. Instead, different subsystems (or different aspects of performance) are represented by separate simulation models.

Conceptual design and system-level detailed design based on existing simulation models is difficult because of the need to integrate the inputs and outputs of the disparate subsystem models to predict overall system behavior. This raises important challenges for researchers in this area: to integrate disparate disciplinary models and to define a design selection algorithm for the multiple objective/multiple decision maker setting, and to do this in a computationally efficient way.

Present integration technology is based on application-specific large scale software involving iterative runs of the disciplinary subroutines linked by special executive programs and databases. The approach is time consuming, costly, computationally expensive, and application specific. As a consequence, integrated system simulators have been developed only for high-value applications, such as aircraft structural design (Neill et al. 1990).

An alternate strategy is to build *metamodels* for each subsystem simulator using a common form, and integrate the metamodels rather that the original simulation codes. Metamodels are mathematical approximations to the discipline-specific product and process models used in engineering design. This use of the term metamodel, which follows that of Kleijnen (1975), is different from Tomiyama et al. (1989), who use the term to refer to a model of the design process.

A metamodel-based integration technology permits a greater portion of the code development to be application independent, and the speed of execution for the metamodel-based integrated system permits a greater variety of design/optimization algorithms to be applied. There are three key research issues that must be addressed to make metamodel-based system-level design practical:

i) integration architecture for subsystem models,

ii) design of experiments for fitting subsystem metamodels, and

iii) measures of metamodel fidelity.

This paper presents a discussion of the second issue: designing experiments for fitting subsystem metamodels. The next section provides a description of the problem and shows an example of conventional system integration. Next, a brief description of a metamodel-based integration strategy is presented. A proposal for a general subsystem metamodel experiment design strategy is then followed by a simple example to illustrate the advantage of this semi-sequential design strategy. The last section provides a summary of research issues.

2 PROBLEM STATEMENT

The system-level design depends on numerical measures of system performance, y_k, $k = 1, ..., p$. These, in turn, are mathematical functions that depend on each other and on a set of design parameters, x_j, $j = 1, ..., d$. That is,

$$y_k = f_k(\mathbf{x}, \mathbf{y}),\ k = 1, ..., p.$$

Note that each y may depend on any x_j or y_k but need not depend on all other y's nor on all of the design parameter elements in $\mathbf{x}$. The system-level design task is to determine values for the components of $\mathbf{x}$ that result in a desirable performance vector $\mathbf{y}$. Typically, the functions are not computed independently, but rather in subsets corresponding to specific simulation/analysis programs which can be viewed as vector-valued functions, say g_m, $m = 1, ..., r$.

For example, in modeling a product and its manufacturing system, y_1 might be the tensile strength of a critical part, y_2 the material cost per unit, y_3 the average manufacturing flow time, y_4 the average value of work in process, y_5 the capital equipment cost, and y_6 the overall cost of production per unit. Typically, the calculation of these functions requires two or more separate software programs. In our example, the first two quantities might be calculated from product design parameters using CAD/CAE software (g_1). The third and fourth might be calculated using a discrete-event simulation model of the manufacturing operation (g_2), and the fifth and sixth using simple accounting models (g_3 and g_4).

These models share some inputs: a design variable specifying the kind of manufacturing equipment (x_4) is an input to the discrete event simulation subsystem model and the simple accounting subsystem model for y_5. Also, some subsystem model outputs are required as inputs to other models. For example, the calculation of y_6 will require y_2 and y_5 as inputs. Figure 1 shows a network representation of the input and output structure for this example, based on an illustrative but arbitrary allocation of six design parameters, x_1 - x_6. It is coincidental that $p = d$ in this example.

2.1 Existing Integration Technology

Typical of the multidisciplinary approach in use today, ASTROS (Neill et al. 1990) provides multidisciplinary integration technology via an executive program which calls separate optimization, modeling, and database routines. This general structure is illustrated in Figure 2.

Each subsystem analysis code corresponds to a g_m. Sobieszczanski-Sobieski and Haftka (1996) developed a similar structure.

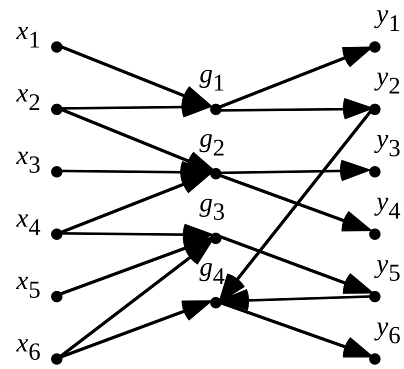


Figure 1: The Relationship Between Design Parameters, Subsystem Models, and Performance Measures for a Simple Example

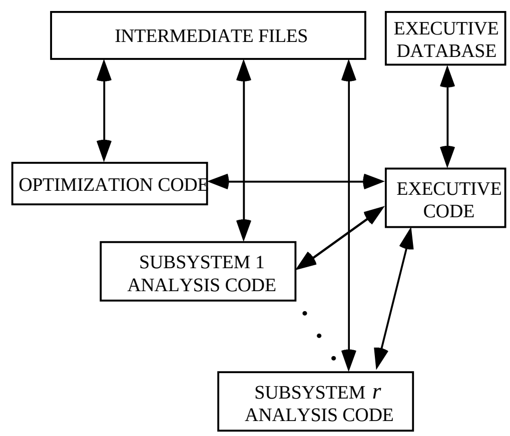


Figure 2: Typical Structure for Existing Multi-disciplinary Integration Technology

A key feature of this integration strategy is the definition of database structures for communication between subsystem analysis codes and the system-level executive program. Westfechtel proposed an object-oriented data structure following Reddy et al. (1993) to include data and analysis tools for integrating computer-aided design, computer-aided process planning and NC code generation.

Problem difficulty depends not just on the nature of the subsystem response functions, but on the interconnectedness of the subsystem models. The easiest topology results when each g is in a separate component of the graph. The most difficult is when the tripartite graph is complete: every subsystem depends on every design parameter and every (other) subsystem output.

These difficulties affect experiment design strategies for fitting subsystem metamodels.

Existing system-level design strategies focus on the integration of existing discipline-specific detailed design codes. Further, the emphasis has been on optimization, yet many system-level design tasks are multidisciplinary and multiobjective, and cannot be expressed in an optimization framework.

2.2 The Nature of Multidisciplinary Multiobjective Design

System-level design involves tradeoffs among multiple objectives that require different engineering and business disciplines to calculate and to assess. The design task requires multicriteria decision making. Zionts cites ten myths of multiple criteria decision making, including the myth of a single decision maker (2), the myth of an optimal solution (4), the myth of limiting consideration to non-dominated (Pareto-optimal) solutions (5), and the myth of the existence of a utility or value function (6).

Generally, numerical combinations of multiple objectives are referred to as utility functions, although many names for such functions appear in the multidisciplinary optimization literature. Messac (1996) created a system-level objective function based on the sum of interdependent 'preference functions' constructed for each design objective. He proposed preference functions as an alternative to von Neumann-Morgenstern utility functions (von Neumann and Morgenstern 1953, Luce and Raiffa 1957) because of the difficulty in determining the appropriate utility function for an engineering design problem (Thurston et al. 1994). Yoshimura and Kondo (1995) developed an objective combining design performance and manufacturing cost using utility theory. The calibration of the utility function used an estimation of '50% satisfactory' designs for both performance and manufacturing cost, rather than the usual lottery equivalence calculation.

While Thurston et al. (1994) recommend the use of utility functions, Hazelrigg (1996a, 1996b, p.300) points out that such utility functions do not generally exist for groups of decision makers (Arrow 1951, Fishburn 1987). The difficulty described by Hazelrigg relates to myths (2), (5), and (6) of Zionts and exposes the conflict between group decision making and optimization. The concept of transitivity in design says that if design A is preferred to design B, and design B is preferred to design C, then design A is preferred to design C. This assumption is at the core of many optimization methods based on local improvement. For single-objective decisions, under the assumption of a single well-defined value of the objective for each particular set of design parameters this concept is reasonable, and thus the success of mathematical programming methods for single-objective design optimization.

Unfortunately, in engineering design there is often more than one objective, and more than one decision maker. The concept of transitivity for group (or even individual) ranking of choices has many difficulties when the choice is based on multiple characteristics or objectives, and so the search for a global optimum design based on pairwise comparisons (or local improvement) may not be appropriate (DeLong 1991). Instead, a comparison among Pareto optimal designs or design regions (based on one or more multiobjective functions) should be provided to decision makers, who may choose a design using democratic or other procedures. It is not necessary that the Pareto-optimal designs will form a single connected set in design parameter space. In fact Pareto-optimal regions of design space may be disconnected regions that are full-dimensional or lower dimensional such as segments of curves, or even points. Thus the phrase *multiobjective design optimization* may be an oxymoron; a more appropriate goal might be *multiobjective design selection*.

An effective experiment design strategy for metamodel-based system-level design must recognize this nature of the multidisciplinary multiobjective design problem. In some cases optimization is appropriate. In others, tradeoff studies involve the identification and exploration of local Pareto optimal regions, local sensitivity analysis and robust design.

3 METAMODEL-BASED SYSTEM DESIGN

Sobieszczanski-Sobieski and Haftka (1996) discuss four advantages of metamodel approximations for use with design space search codes: i) the need for a large number of response evaluations as part of the design process, ii) as a tool to integrate software from different disciplines and perhaps different machines (for example, Tai et al. 1995 and Giunta et al. 1995), iii) to overcome jagged response surfaces that arise from numerical roundoff or deliberately incorporated random factors, and iv) to permit visualizations of the entire design space (Mistree et al. 1994). They cite three metamodel approaches for global approximations: simplified physical models (with scale factors calibrated by the full model), polynomial response surface approximations, and neural networks.

Quadratic response surfaces are the most commonly used metamodels, although recent developments suggest other metamodel types could provide better global approximation (Barton 1992, 1994). Osyczka and Zajac (1990) use response surface metamodels fit via face-centered composite designs for complex optimization tradeoffs that multicriteria optimization requires. They suggest interactive graphical methods for the selection of a Pareto-optimal design, and note the accuracy limitations of the metamodeling approach.

A metamodel-based integration strategy offers an effective way to address many computational difficulties in multidisciplinary optimization. First, a metamodeling strategy permits a general integration strategy that can be implemented for all sources and combinations of disciplinary models. The metamodeling strategy simplifies optimization and/or examination of the system performance over the system design space, and, depending on the metamodel form, subsystem metamodels may be combined analytically to form the system metamodel, perhaps automatically as well. Second, new approximation methods reduce concerns about global model fidelity that are an inherent shortcoming of polynomial approximations. Third, the metamodeling approach permits rapid evaluation of system performance for alternative designs, relieving the limitation of examining one or a few candidates and relying on the questionable concept of a single, algorithm-determined 'optimal' design.

A general metamodel-based multidisciplinary design integration strategy that links approximation models for all disciplinary submodels to form an overall system model is illustrated in Figure 3.

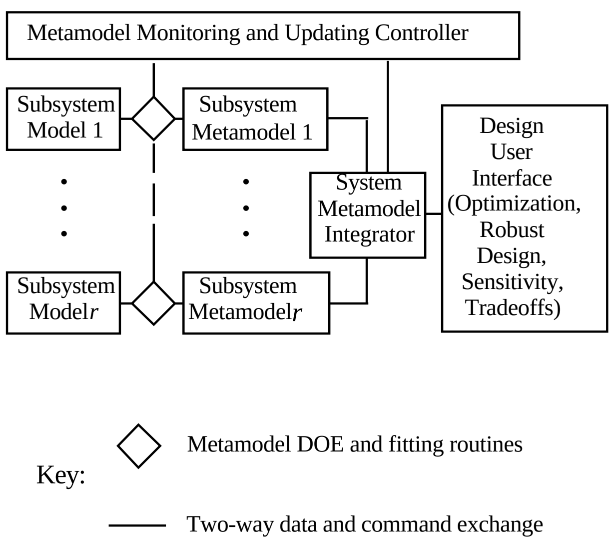


Figure 3: Proposed Structure for Metamodel-based Integration Strategy

Full subsystem disciplinary model analyses and perhaps optimizations are run for specific sets of design parameter values to fit the metamodels. The number of runs and the design parameter settings for each run are determined by the DOE/Fitting modules. The DOE strategy may differ for different metamodel types. The fitting modules calibrate the appropriate metamodel to the disciplinary run data, and determine the adequacy of the metamodel fit for each of the output variables. The DOE/Fitting module can request additional disciplinary subsystem runs if necessary. The integration module combines the discipline-specific metamodels based on an integration model, to produce a system metamodel which has as inputs all design parameters and as output all performance measures. The system metamodel is exercised by a user through a user interface that includes graphical exploration of the design space, identification and exploration of the Pareto region, and utility-based optimization. The fidelity of the metamodel representations is maintained by monitoring the design regions of current interest to the user, assessing the adequacy of the fit (validity) of the subsystem metamodels and the system metamodel, and prescribing additional calibration runs through the individual DOE modules. The additional runs may be conducted automatically, or requested of the user, depending on the difficulty in interfacing the discipline-specific subsystem codes to the Updating Controller.

4 EXPERIMENT DESIGN STRATEGY

The validity of model-based design depends on the validity of the model, which in turn depends on the validity of the subsystem models. This is true as well for metamodel-based design, but subsystem (meta)model validity also depends on the experiment designs used to fit the metamodels, the metamodel types, and the intended use (Sargent 1991).

The traditional emphasis for simulation metamodels has been simultaneously defined designs such as fractional factorial or central composite designs, used to fit polynomial metamodels. Occasionally the design process is two-stage: a screening fractional factorial design to identify the design parameters having significant effect on performance, followed by a higher order model and a more complete design on the remaining set of parameters (see Donohue et al. 1993 and Chen et al. 1996).

Since simulations typically compute results for one design point at a time, sequential designs may provide an advantage. There has been little work in this area, although such a strategy is used by Tu and Barton (1997) to develop efficient designs for metamodels used for Monte-Carlo estimates of yield. A sequential approach might be used in place of simultaneous designs for steps 1 and 4a in the strategy described below.

4.1 Semi-sequential Design Strategies

A form of sequential design has been applied to metamodel-based design. Designs are sometimes updated as optimization iterations progress, focusing on the region of the current optimization iterates. Each update may consist of a simultaneous set of runs, rather than an individual run, and so the method might be called a *semi-sequential* strategy.

For example, Toporov (1992) proposed a metamodeling method based on quadratic polynomial approximation of the response and constraint functions (fit by weighted least-squares), followed by a nonlinear programming optimization applied to the approximating objective and constraints. A sequential experiment design for fitting the approximations was modified after each optimization cycle, excluding points no longer in the 'region of interest,' and adding $n + 1$ new evaluations for a problem with n design parameters (Toporov et al. 1993). A move-limit strategy and a new weighting procedure for the least squares fitting were proposed by Toporov et al. (1996) to try to improve optimization performance in conjunction with an adaptive discretization error strategy for the finite element calculations. Toporov's *region of interest* will play a role when the experiment designs are updated based on recent design iterations. A too-rapid reduction in the *region of interest* size resulted in premature termination of the optimization process. The work of Toporov and co-authors did not address metamodel-based integration: the authors assumed the existence of a single integrated numerical code for calculating multidisciplinary objective and constraint function values.

There is no point in requesting an additional subsystem analysis whenever a subsystem metamodel evaluation is needed. In this case the subsystem models could be called directly, and the metamodel structure would provide no benefit. The metamodel monitoring and updating function shown in Figure 3 suggests a different approach, monitoring the adequacy of the metamodel fit as design iterations progress and conducting additional experimental runs to improve metamodel fits only when required. The controller must decide when additional runs are needed and where in design parameter space they should be conducted.

Determining *when* additional runs are needed is really a validation issue, and so validation research may be useful in answering this question. Yesilyurt and Patera (1993) provide one example of such a strategy. They use metamodels for deterministic subproblems that can be expressed as a functional applied to a field satisfying initial-boundary-value conditions. The strategy was applied to modeling two-dimensional laminar flow and convective heat transfer as a function of two design parameters, eddy promoter displacement and radius. The paper describes two metamodel types: piecewise-constant metamodels based on a Voronoi subdivision about sampled design points, and a bivariate interpolation method for scattered data.

The primary contribution of this work is the validation methodology that the authors propose. Monte Carlo samples of the design parameter vector are drawn, according to a Bayesian importance distribution function, r. For each vector, the difference between the metamodel prediction, y' and the model prediction, y is computed. The maximum of these differences is used in a probability statement of the form: "the r-measure of the set of design parameters where the metamodel error exceeds $E_{\max}$ is less than e_1 with probability greater than $1 - e_2$." The validation methodology sets the number of samples required for a given (e_1, e_2). Unfortunately, for high-dimensional design vectors, a small r-measure may still produce an unacceptably large region with poor fit. Yesilyurt and Patera's metamodel-based optimization procedure redefines r at the end of an optimization cycle using Monte Carlo validation bound to identify a reduced search region where the simulation model response is likely to be better than a pre-defined value. Unfortunately, in practice they observed that the search regions were not reduced substantially using this approach.

Determining *where* additional runs are needed can be based on self-assessment of the metamodel. For example, prediction intervals can be computed for any parameter vector for polynomial regression models using the general linear model assumptions, and new runs can be specified at points in the current *region of interest* with large prediction intervals. Similar prediction error estimates can be evaluated for spatial correlation models, again based on probability model assumptions (Sacks et al. 1989). In addition, cross-validation can be used for self-assessment of many metamodel types, and bootstrapping may be useful for some experiment design types.

The general structure of a semi-sequential strategy for updating subsystem metamodels is shown below. This structure helps to clarify areas where further research is needed.

Step 1. Establish Initial Designs and Fit Each Subsystem Metamodel: *Initial designs will generally be determined by the metamodel type and the number of subsystem design parameters. Possible designs include Plackett-Burman, factorial, fractional factorial, central composite, small composite (Draper 1985), Latin hypercube and other orthogonal arrays (Owen 1992, Tang 1993).*

Step 2. Determine Current *Region of Interest* in Design Parameter Space: *The current region of interest may be determined in a number of ways. It may be* *specified by the user, or defined as the convex hull of a specified number of most recent iterates, or defined as the rectangular region bounded by the coordinatewise maxima and minima for each design parameter over the specified number of most recent iterates, for example.*

Step 3. Monitor Current *Region of Interest*: *If the current region of interest changes, as indicated for example by i) a specified number of iterates outside the current region, ii) a specified reduction in the space spanned by iterates within the current region, or iii) a signal from the user, then go to Step 4. Otherwise, return to Step 2.*

Step 4. Assess Fit of Metamodels on Current *Region of Interest* Against Current Fit Criteria: *The fit criteria may change depending on the stage of the design process and on the particular submodel. Fit may be assessed by validation runs selected according to a validation run experiment design strategy, or by cross-validation, or by a metamodel-specific self-measure of prediction error, or by a combination of these methods.*

Step 5a. Update the Metamodel Using a Revised Experiment Design: *If the current fit criterion is not satisfied, determine a set of existing and new runs to fit the metamodel. Conduct the new set of runs of the subsystem model. Use the selected set of new and existing runs to fit a new subsystem metamodel. Repeat Step 4.*

Step 5b. Update Estimate of Metamodel Uncertainty over Current *Region of Interest*: *If the current fit criterion is satisfied, update the estimate of the subsystem metamodel uncertainty and return to Step 2.*

The example in the next section implements a very simple version of this approach.

5 AN EXAMPLE

In this example, the subsystem model has a single design parameter, a single numerical output, and is deterministic. It is the function $f(x) = 1/(1+x^2)$ over an initial *region of interest* corresponding to the interval [-5,5]. Runge (1901) shows the failure of polynomial basis functions to approximate this response function, and this situation is also described in Schumaker (1981). A simple quartic polynomial fitted by least squares will be used for the metamodel. The design goal is assumed to be maximization of the function over the initial region of interest.

The example is not meant to illustrate a *good* strategy, but only to give an instance of the general procedure described in Section 4. The example extends through three iterations of Step 5b. The resulting fit is compared at each iteration with a simultaneous design consisting of 101 evenly spaced points (0.1 increment) between -5 and 5.

Step 1: the initial design consists of 21 evenly spaced *x* values between -5 and 5. The resulting least-squares fit is shown in the top plot of Figure 4. This figure shows three curves: the true subsystem model response is coded as 'A,' the 101-run simultaneous design metamodel is plotted as 'B,' and the 21-run initial semi-sequential design as 'C.' The 21-run design produces virtually the same fit as the simultaneous design in this example, and so the 'B' and 'C' plot symbols overlap.

Step 2: Based on the form of the initial metamodel, the *region of interest* is arbitrarily reduced to the interval [–1, 1].

Step 3: Since the *region of interest* changed, Step 4 is executed.

Step 4: Fit may be assessed in a number of ways. In this example, whenever the *region of interest* changes, the fit is assumed to be unsatisfactory.

Step 5a: A design of 21 evenly spaced points in [-1, 1] is used to augment the initial 21-run design. The results are shown in the middle plot in Figure 4. Note that the error is reduced in comparison with the 'C' metamodel in the top plot, for the current *region of interest.* (The error is more extensive elsewhere.)

Step 4: The fit is assumed adequate after an augmenting design.

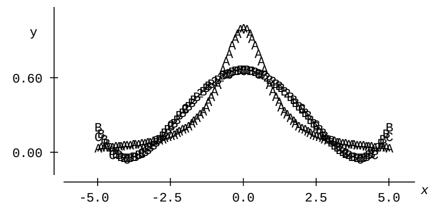
Step 5b: No error estimate is provided in this example.

Step 2 . . . Step 5b: The process repeats, with an arbitrarily selected *region of interest* of [-.5, +.5] at this iteration, and [-.25, +.25] at the third iteration. The results of the third iteration are shown in the lower plot. The fit shows further improvement over the *region of interest* [-.25, .25] in comparison with the upper plots.

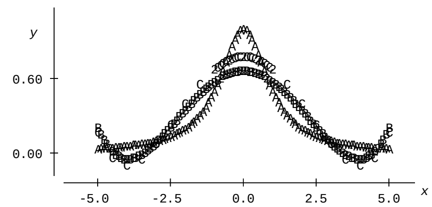
The advantage of a semi-sequential strategy is illustrated even by this simple example: the middle and lower sequential metamodels based on 42 and 84 runs, respectively, both show greater fidelity in the region near the maximum than the 101-run simultaneous design.

6 SUMMARY

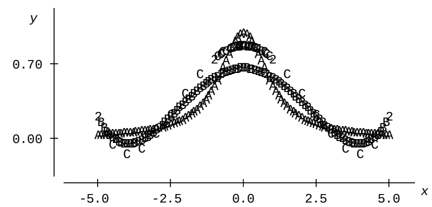
Integration strategies for system-level design are important in an era of concurrent engineering, when system-level design decisions involve detailed information about subsystem performance. Using a metamodel-based integration strategy permits an attractive compromise between model fidelity and the ease of subsystem integration.



A = True, B = Fitted quartic based on 101-run simultaneous design, C = Fit of first 21-run semi-sequential design



A = True, B = Fitted quartic based on 101-run simultaneous design, C = Fit of second 42-run semi-sequential design



A = True, B = Fitted quartic based on 101-run simultaneous design, C = Fit of fourth 84-run semi-sequential design

Figure 4: A Comparison of Semi-Sequential Designs with a 101-Run Simultaneous Design for Fitting $y = 1/(1+x^2)$

Critical to the success of such a strategy is an effective experiment design methodology. We have presented one general structure: a semi-sequential design approach. Key areas for research in developing this structure include i) how to define a *region of interest* and how to determine its current extent, ii) how to asses the adequacy of metamodel fit over the region, iii) how to select an experiment design composed of existing and new runs given a region of interest, a metamodel type, and an assessment of adequacy of fit, and iv) how to determine when the *region of interest* has changed enough to warrant reassessment.

ACKNOWLEDGMENTS

This research was supported in part by NSF grant DMI-9700040 and by a Total Quality Management Grant from the IBM Corporation. Thanks to Robin Barton and Martin Meckesheimer for their patience and proofreading.

REFERENCES

Arrow, K. J. 1951. *Social choice and individual values*. New York: John Wiley.

Barton, R. R. 1992. Metamodels for simulation input-output relations, *Proceedings of the 1992 Winter Simulation Conference*, ed. D. Goldsman, J. Swain, J. Wilson, 411-418. IEEE, Piscataway, New Jersey.

Barton, R. R. 1994. Metamodeling: a state of the art review, *Proceedings of the 1994 Winter Simulation Conference*, ed. J. D. Tew, M. S. Manivannan, D. A. Sadowski, and A. F. Seila, 237-244. IEEE, Piscataway, New Jersey.

Chen, W., J. K. Allen, D. P. Schrage and F. Mistree. 1996. Statistical experimentation for affordable concurrent design. *6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 921-930. AIAA, Reston, Virginia.

DeLong, H. 1991. *A refutation of Arrow's theorem.* Lanham, Maryland: University Press of America.

Donohue, J. M., E. C. Houck and R. H. Myers. 1993. A sequential experimental design procedure for the estimation of first- and second-order simulation metamodels. *ACM Transactions on Modeling and Computer Simulation* 3: 190-224.

Draper, N. R. 1985. Small composite designs. *Technometrics* 27: 173-180.

Fishburn, P. C. 1987. *Interprofile conditions and impossibility.* New York: Harwood Academic Publishers.

Hazelrigg, G. A. 1996a. The implications of Arrow's impossibility theorem on approaches to optimal engineering design. *Journal of Mechanical Design* 118: 161-164.

Hazelrigg, G. A. 1996b. *Systems engineering: an approach to information-based design.* Upper Saddle River, New Jersey: Prentice-Hall.

Kleijnen, J. P. C. 1975. A comment on Blanning's metamodel for sensitivity analysis: the regression metamodel in simulation. *Interfaces* 5: 21-23.

Luce, R. D. and H. Raiffa. 1957. *Games and decisions.* New York: John Wiley.

Messac, A. 1996. From the dubious art of constructing objective functions to the application of physical programming. *6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 1857-1867. AIAA, Reston, Virginia.

Mistree, F., B. Patel and S. Vadde. 1994. On modeling multiple objectives and multi-level decisions in concurrent design, in *Advances in Design Automation - 1994*, eds. B.J. Gilmore, D.A. Hoeltzel, D. Dutta, and H.A. Eschenauer. ASME DE-Vol. 69-2: 151-161.

Neill, D. J., E. H. Johnson and R. Canfield. 1990. ASTROS - a multidisciplinary automated structural design tool. *Journal of Aircraft* 27: 1021-1027.

Owen, A. 1992. Orthogonal arrays for computer experiments, integration, and visualization. *Statistica Sinica* 2: 439-452.

Reddy, R., K. Srinivas and V. Jagannathan. 1993. Computer support for concurrent engineering, *IEEE Computers* 26: 12-16.

Runge, C. 1901. Uber die darstellung willkürlicher functionen und die interpolation zwischen aquidistanten ordinaten. *Z. Angew. Math. Phys.* 46: 224-243.

Sacks, J., W. J. Welch, T. J. Mitchell and H. P. Wynn. 1989. Design and analysis of computer experiments. *Statistical Science* 4: 409-435.

Sargent, R. G. 1991. Research issues in metamodeling. *Proceedings of the 1991 Winter Simulation Conference*, ed. B. L. Nelson, W. D. Kelton, G. M. Clark, 888-893. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

Schumaker, L. L. 1981. *Spline functions: basic theory.* New York: Wiley.

Sobieszczanski-Sobieski, J. and R. T. Haftka. 1996. Multidisciplinary design optimization: a survey of recent developments. *34th Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, January 15-18, 1995. AIAA Paper 96-0711.

Tang, B. 1993. Orthogonal array-based latin hypercubes. *Journal of the American Statistical Association* 88: 1392-1397.

Thurston, D.L., J. V. Carnahan and T. Liu. 1994. Optimization of design utility. *ASME Journal of Mechanical Design* 116: 801-807.

Tomiyama, T., T. Kiriyama, H. Takeda, and D. Xue. 1989. Metamodel: a key to intelligent CAD systems. *Research in Engineering Design* 1: 19-34.

Toporov, V. V. 1992. Multipoint approximation method in optimization problems with expensive function values, in *Computational systems analysis 1992*, A. Sydow, ed. New York: Elsevier.

Toporov, V. V., A. A. Filatov and A. A. Polynkin. 1993. Multiparameter structural optimization using FEM and multipoint explicit approximations. *Structural Optimization* 6: 7-14.

Toporov, V. V., F. van Keulen, V. Markine and H. de Boer. 1996. Refinements in the multipoint approximation method to reduce the effects of noisy structural responses, *6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 941-951. AIAA, Reston, Virginia.

Tu, C.-H. and R. R. Barton. 1997. A metamodel strategy for Monte-Carlo yield estimation. *Proceedings of DETC'97.* 1997 ASME Design Engineering Technical Conferences, Sacramento, California. Paper DETC97/DTM-3870. To appear.

von Neumann, J. and O. Morgenstern. 1953. *Theory of games and economic behavior.* 3rd edition. Princeton: Princeton University Press.

Westfechtel, B. 1996. Integrated product and process management for engineering design applications. *Integrated Computer-Aided Engineering* 3: 20-35.

Yesilyurt, S. and A. T. Patera. 1993. *Surrogates for numerical simulations; optimization of eddy-promoter heat exchangers.* ICASE Report 93-50, NASA Contractor Report 191510. NASA Langley Research Center, Hampton, Virginia.

Yoshimura, M. and H. Kondo. 1995. Concurrent product design based on simultaneous processing of design and manufacturing information by utility analysis. *Proceedings of the 1995 Design Engineering Technical Conferences* ASME DE-Vol. 82, 25-32.

Zionts, S. 1993. Multiple criteria decision making: the challenge that lies ahead, in *Multiple criteria decision making,* G.H. Tzeng, H.F. Wang, U.P. Wen, P.L. Yu, eds., 17-26. Springer-Verlag, New York.

AUTHOR BIOGRAPHY

RUSSELL R. BARTON is an Associate Professor in the Department of Industrial and Manufacturing Engineering at The Pennsylvania State University. He received a B.S. in electrical engineering from Princeton University in 1973 and a Ph.D. in operations research from Cornell University in 1978. His current research interests include graphical methods for experiment design, optimization and metamodeling methods for simulation models, and statistical issues in modeling manufacturing yield.