DESIGN OF EXPERIMENTS IN THE MÖBIUS MODELING FRAMEWORK

BY

PATRICK GERALD WEBSTER

B.A., Rockford College, 1992
M.S., University of Illinois at Urbana-Champaign, 1994

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2002

Urbana, Illinois

ABSTRACT

Performance and dependability modeling in system design is a complicated process. Model solutions often require large amounts of computer time. As the models become more complex, the solution time required for the performance and dependability measures increases. Models that are more complex also contain more parameters that can vary and affect the reward measures. The techniques of Design of Experiments (DOE) can increase the efficiency of the modeling process and alleviate some of the problems introduced by sophisticated system models.

This thesis reviews the theory behind several popular DOE strategies, including fractional factorial and response surface experimental designs. These theories are then realized as a DOE graphical user interface, fully integrated into the Möbius modeling tool. The interface allows the user to build an empirical model of the reward measure using linear regression techniques. The empirical model can be used to predict reward behavior within the specified design region. Standard errors and confidence intervals are given for all predicted values. Contour plots of predicted reward measures are also available. Screening experimental designs are available to determine which model parameters have a significant impact on the reward measures. Both graphical and tabular methods are available for ranking the sensitivity of the reward measures to each model parameter. All model solutions from simulators and numerical solvers can be imported directly into the DOE interface using the integrated results database, alleviating the need to import results manually.

Following a detailed description of the interface features, we present an example highlighting the features of the DOE interface in Möbius. The example investigates the reliability of a fault-tolerant multicomputer system. Plackett-Burman designs are used to screen the system parameters to determine which have the largest influence on reliability. A two-level factorial design is then used to determine the interaction among the significant system parameters. Central composite designs are then employed to determine the optimal coverage levels for a given number of computers in the system. Finally, the empirical model is used to predict system reliability at a specified confidence level and within a specified prediction interval.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 9975019, by the Defense Advanced Research Projects Agency (DARPA) ITO under Contract No. DABT63-96-C-0069, and by the Interstate Electronics Corporation. Any opinions, findings, and conclusions or recommendations expressed in this material are mine and do not necessarily reflect the views of the National Science Foundation, DARPA/ITO, or Interstate Electronics. I am grateful for the support of those organizations.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Page

1. INTRODUCTION

1.1 Performance and Dependability Modeling

As the field of system design matures, accurate prediction of system performance and dependability before producing the final product is becoming increasingly important. Modern systems in a variety of disciplines from computer to mechanical to biological engineering are becoming more complex and more costly to build. The ability to use a model (description) of the system to estimate performance and dependability can save design time and reduce production cost. To gather this system data, the engineer must decide which modeling formalism (language) to use, what performance or dependability metrics to define, and what solution methods to employ on the model.

Different modeling formalisms, such as queuing networks or SANs (stochastic activity networks), may be chosen to represent the system. The formalism choice depends heavily on the system characteristics. The level of detail at which the system should be represented depends on the performance and dependability measures under study and on the designer's intuition. System modeling is an art, requiring the designer to focus on system components relevant to the measures and to accurately represent the system behavior. The choice of performance or dependability measure simply depends on what system characteristic the modeler is interested in, such as availability or mean time to failure.

After the model is created and the measures are defined, a solution method must be selected. There are several solution types available, such as numerical methods and simulation. Like the modeling language choice, the solver choice also depends on the structure of the underlying model. For example, a queuing network with rare events may lend itself to an analytical (mathematical) solver rather than a simulator. Alternatively, a SAN with a very large state space may require too much time to solve analytically, but may converge to a solution quickly using simulation.

1.2 Experimentation in Möbius

Möbius is an extensible modeling tool for stochastic, discrete-event systems [1], [2], [3]. Möbius is unique among modeling tools in its ability to support integration of new modeling formalisms, solvers, and measures [4]. One of the goals of Möbius is to create a complete modeling environment in which the design engineer can experiment with new ideas while minimizing the necessity to interact with other software tools for information exchange. A fully integrated database for solution results helps to realize this goal [5]. Another aid to experimentation is an integrated sensitivity analysis and measure optimization module. This is the Design of Experiments study editor, which is the focus of this thesis. Before explaining the details of Design of Experiments in Möbius, some background information must be defined.

Model specification takes place at several levels in Möbius. Figure 1.1 shows the steps in model specification and solution in Möbius. The lowest level is the *Atomic* model level. Each atomic model consists of a single model. Atomic models can be composed or

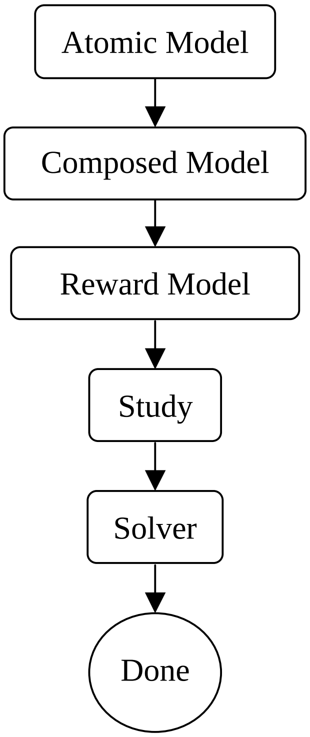


Figure 1.1: Model specification and solution in Möbius

connected with other models. Above the atomic level is the *Composed* model level. Composed models are built structurally from two or more atomic or composed models [6]. Next is the *Reward* model level. Reward models consist of performance and dependability measures (*Reward Variables*) defined on the underlying composed or atomic model. For example, consider a SAN model of a computer system on which the reliability of the entire system and the availability of the backside bus are to be measured. The system reliability and the bus availability are each reward variables, which together form a reward model.

During the specification of atomic, composed, and reward models, *global variables* can be used to parameterize model characteristics. A global variable is a variable that can be defined in one or more models across any number of model levels. Values are not assigned to the global variables during the model specification phase. Global variables are assigned values at the next level: the *Study* level.

The assignment of a specific value to each global variable forms an *experiment*. Each reward model can be solved for each experiment. A *study* is a collection of experiments. Studies vary model parameters in an organized manner in order to reveal how specific system attributes affect the reward variables. Continuing the above computer system example, suppose a module of redundant memory chips is installed. The effect on system reliability as the number of redundant memory chips varies is to be studied. The global variable is the number of memory chips. Several experiments assign unique values to this global variable. A state space is generated on the model for each experiment. A transient analytic solution is run for several time points for each experiment. The probability that the system is still up after a given time is one measure of the system's reliability.

1.3 Motivation

Performance and dependability modeling software tools have become increasingly powerful in recent years. Engineers have the ability to model increasingly complex systems using only a moderate amount of computing resources. However, despite the technological advances in system modeling, there remain several obstacles hindering the prediction of system behavior. Two facts contribute to these obstacles: the fact that system models have grown in both scale and intricacy of detail, and the fact that modeling software tools do not provide the appropriate feedback to the engineer during the design process.

As system models become more elaborate, the number of variables that can be parameterized increases rapidly. Difficulties arise from having such a large number of model parameters. The main problem with modeling large systems is in deciding how to make the best use of the computing resources available. The first step in optimizing use of available resources is reducing the number of model parameters to vary. Varying model parameters that do not contribute to the reward variables being measured wastes experimentation time. Selection of the parameters to vary requires detailed knowledge of the underlying model, and often relies on the designer’s intuition and experience with similar systems. Another way to optimize use of computing resources is to reduce the number of values assigned to the model parameters. Increasing the number of values for a particular model parameter requires more experiments if all values are to be tested. If there are several parameters being varied, then the number of experiments needed to test each combination can grow to an unmanageable number. The engineer should focus on a range of parameter values over which the reward variables are expected to change significantly. Again, the art of modeling requires detailed knowledge of the model, which is best gained from previous experience. Unfortunately, software tools cannot automatically grant an engineer intuition, but they can provide valuable information that, over time, can be used to develop a knowledge base useful for solving future problems.

The second problem contributing to the engineer’s difficulty with efficiently predicting system behavior is the fact that today’s modeling tools do not provide the engineer with the necessary feedback during the design process. The *design space* consists of all possible system configurations. Each configuration consists of unique values assigned to each system parameter. The experimentation process is often iterative, consisting of several sets of experimental runs, each producing results requiring analysis. The traversal through the design space ends when the desired system configuration is obtained. Figure 1.2 gives an abstract view of the navigation through the design space from an initial run toward the desired system configuration. In general, it is desirable to minimize the amount of experimentation time needed to determine the desired configuration. After running a set of experiments, the

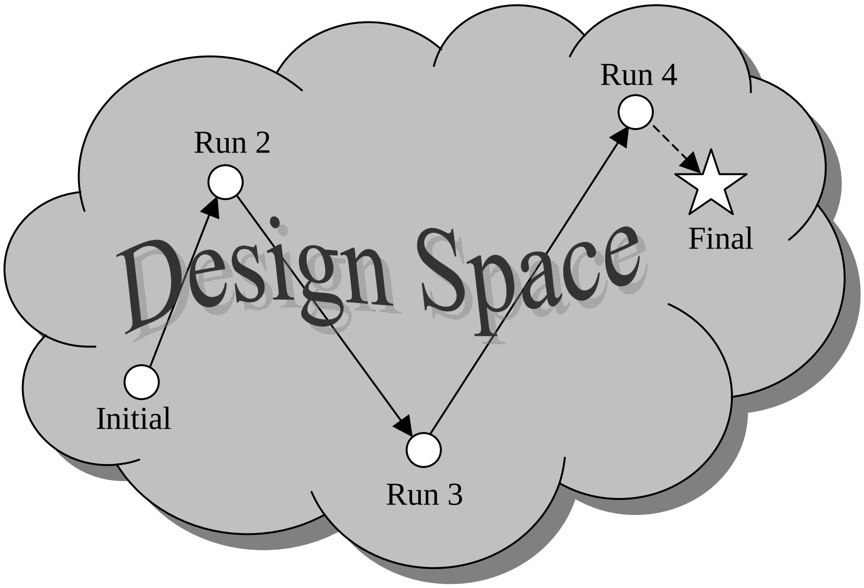


Figure 1.2: Design space navigation

engineer may find that the results do not meet the desired specification. Further experimentation is necessary to find which model parameter values are acceptable. It would be useful if the modeling software could analyze the results and provide information suggesting which parameter values to choose for future runs. This iterative feedback would help the engineer to efficiently arrive at a desired model configuration. Without such feedback, the engineer may incorrectly guess which values cause the model to converge to the desired specification, resulting in a waste of experimentation time. The feedback obtained during this efficient navigation of the design space can also be used to reveal less expensive model configurations that meet the specifications, rather than exceed them.

Often the goal is to find the model parameter values that optimize a particular reward variable. For example, in a multiprocessor computer system, one may wish to find the number of processors and the memory failure rate that maximize the reliability of the system. It would be desirable for the software to provide both graphical and numerical descriptions of how close a particular system configuration is to the desired optimum. Another desirable feature would be the ability to perturb various model parameters within a small range of values and see how the reward variable is affected without actually having to run additional experiments. In Figure 1.2, the short dotted line from run 4 signifies the small perturbation of design parameters that is necessary in order to reach the final configuration.

As mentioned previously, one problem facing the design engineer is that of deciding what values to assign to the model parameters. Often, the sensitivity of a reward variable to a model parameter is unknown. The engineer must choose a range of values over which to vary the model parameters. It would be useful if the experimentation software could provide the engineer with feedback revealing how sensitive reward variables are to the model parameters being varied. It may be the case that a particular model variable has a negligible effect on a reward variable. In that case, varying the negligible parameter produces no useful information and wastes experimentation time. Knowledge that a model characteristic has little effect on a reward variable allows the engineer to fix that characteristic's value and to concentrate on other model parameters that have significant impact.

1.4 Research Objectives

The goal of this research is to integrate into Möbius an experimentation interface that will help the design engineer to efficiently model the behavior of complex systems with different configurations. This facilitation of the exploration of different model configurations allows the engineer to spend less time gathering results from experimental runs and more time building a better model, which will eventually be realized as a better system. The aforementioned integrated experimentation interface is the Design of Experiments (DOE) study editor. The term "study editor" refers to the user interface in Möbius used to define studies. Figure 1.1 showed that the study specification takes place after the reward model has been defined but before the model is solved. That is the traditional method of model specification and solution in Möbius. The DOE study editor introduces a feedback loop in the design process, as seen in Figure 1.3. This additional study editor allows the user to analyze the results of experiments and to modify the study. The modified study creates additional experimental runs that can be analyzed when completed. The feedback provided by the DOE study editor, which is the same as the feedback discussed in the previous section, will help the engineer to predict system performance and dependability more efficiently and accurately.

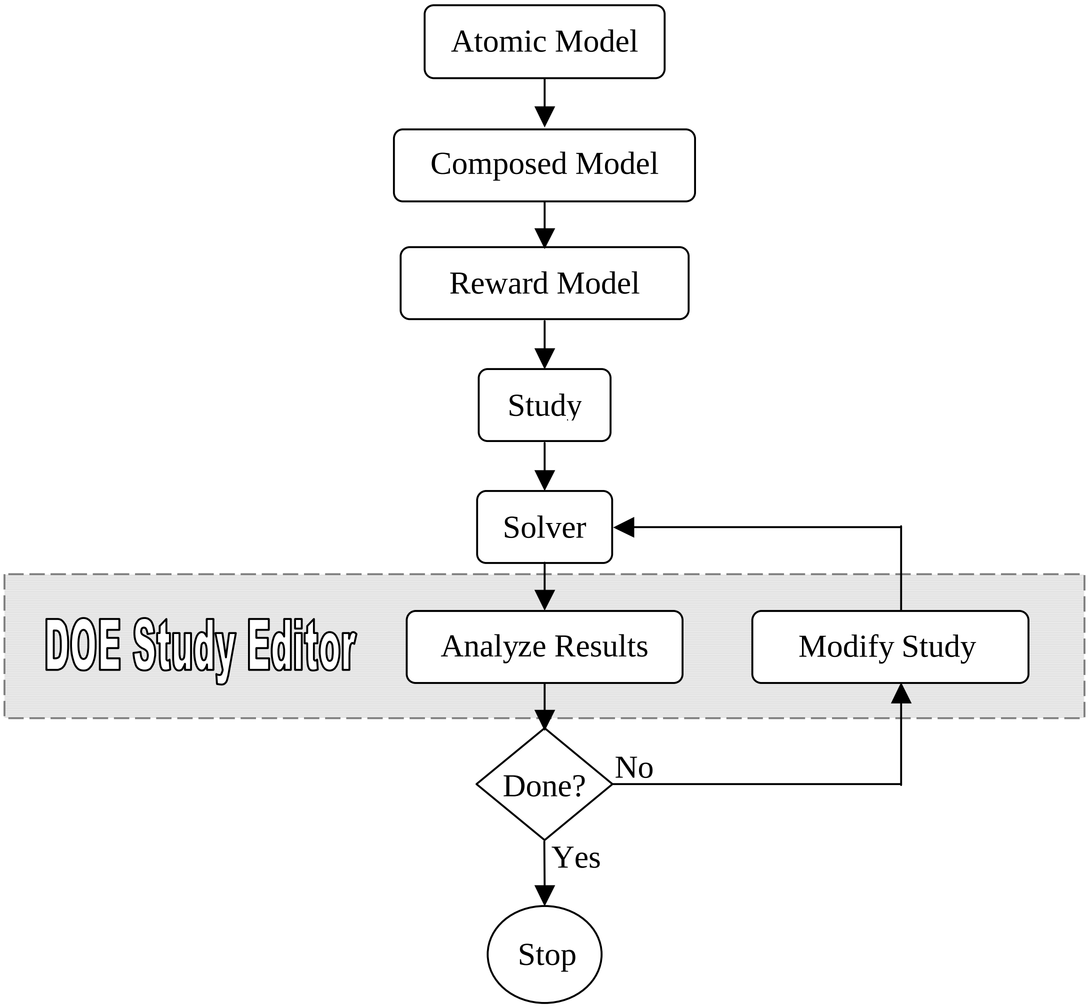


Figure 1.3: Model solution with the DOE study editor

To help the engineer's experimentation process, the DOE study editor will provide:

- a method of screening model parameters to determine if they have significant impact on the reward variables being studied.
- both graphical and numerical means to reveal how significant each model parameter is to each reward variable.
- an interface to perturb the model parameters and receive immediate graphical or numerical feedback with the predicted reward variable values.
- a graphical method to reveal local extrema of reward variables as a function of the model parameters.

- an intelligent method of reducing the number of experiments necessary to measure the desired system behavior.
- the ability to automatically retrieve experimental results from the Möbius results database, alleviating the need to manually copy in data for analysis.

The remaining chapters provide a detailed description of how the DOE study editor achieves its goals in the Möbius modeling framework. The theory of Design of Experiments necessary to build Möbius DOE studies is presented in Chapters 2 and 3. Chapter 4 describes the functionality of the DOE interface. Chapter 5 presents a detailed example illustrating the key features in the DOE study editor. The final chapter summarizes the research results and discusses possible future work.

2. REVIEW OF DESIGN OF EXPERIMENTS

2.1 Introduction to DOE

The statistical design of experiments is the process of planning experiments so the results collected can be analyzed by statistical methods, producing valid and objective conclusions. Statistically designed experiments offer the engineer a valid basis for developing an empirical model of the system being investigated. This model can be manipulated mathematically, graphically, or by other means that will be explained later in this thesis. DOE is divided into two main categories: the design of the experiments and the statistical analysis of the results. The rest of this chapter will review existing theories in the design of experiments. The next chapter will review popular techniques used in the statistical analysis of experimental results.

2.2 Basic Terminology

Before discussing how experiments are designed, it is necessary to introduce some DOE vocabulary. For a system model, there are model parameters that can be varied. These model parameters are called *factors*. The values assigned to factors are called *levels* [7]. For example, in a computer system, the factor *cache size* can be assigned levels of *32 Mbytes* or *64 Mbytes*. Here, *cache size* is assigned numerical values and hence is called a *numerical* (or *quantitative*) factor. The factor *cache replacement policy* can be assigned levels of *random* or *least-recently-used*. Here, the levels of *cache replacement policy* cannot be arranged in order of magnitude on a numerical scale. *Cache replacement policy* is called a *categorical* (or *qualitative*) factor.

The goal of any experiment is to vary the input factors so that the reasons for change in the output response can be identified. In DOE, this output response is simply called the *response*. In system modeling, the response is the performance or dependability measure being studied. For example, in a satellite communication system, the availability and mean-time-to-failure (MTTF) of a satellite node are two responses that can be investigated.

2.3 Experimentation Strategies

Several strategies can be used in the planning and conducting of experiments. A common strategy is the *best-guess* method, in which the experimenter uses his experience and knowledge of the system to guess which factor levels will best reveal the reasons for any change observed in the response. This method can work very well if the experimenter has excellent knowledge of the system under study. Often, the system is complex and its behavior is difficult to predict. The best-guess method may require many experiments before an acceptable result is achieved. It is also difficult to determine if the acceptable result is optimal [7].

Another popular experimentation strategy is the *one-factor-at-a-time* approach, in which one factor is varied while all other factors remain fixed at a nominal level. The nominal levels can be any values that the experimenter wishes to use as a baseline for comparisons. After each factor has been varied and all experiments are complete, the results show the effect that each varied factor has on the response. For example, the performance of a microprocessor is to be studied as the cache size and program size are varied. Figure 2.1 shows the performance as each factor is varied from its nominal value of `small`. The graphs suggest that performance increases slightly as cache size increases, but decreases sharply as program size increases. Alternatively, consider the same performance study with nominal levels set to `large`. The results in Figure 2.2 suggest that performance increases sharply as cache

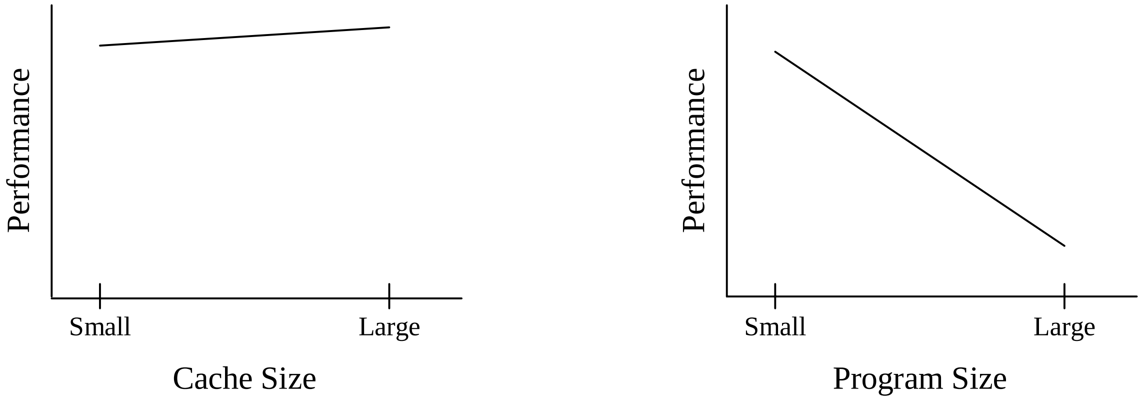


Figure 2.1: One-factor-at-a-time results at `small` nominal levels

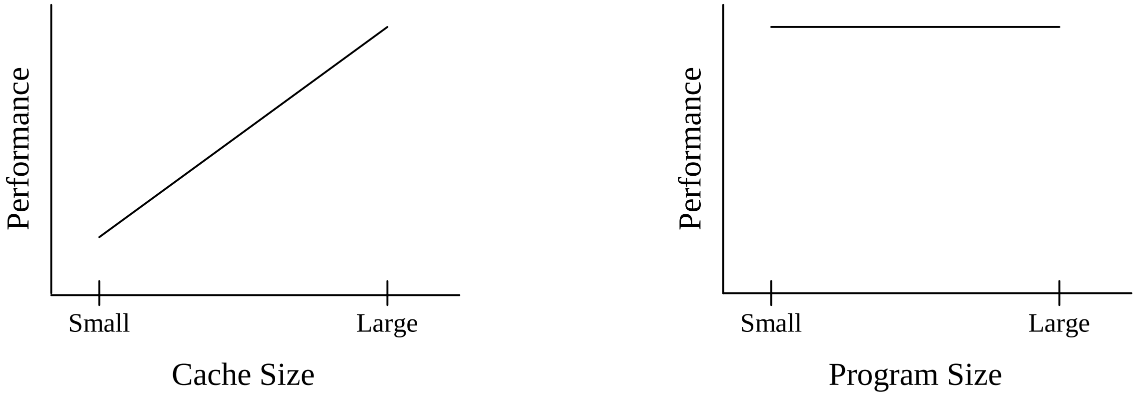


Figure 2.2: One-factor-at-a-time results at `large` nominal levels

size increases, but is unaffected by program size. The interpretation of the results varies as the nominal levels vary. This is a serious problem and can lead to bogus conclusions. The source of the problem is that neither result reveals any information about the *interaction* between the two factors [7].

An interaction occurs when the effect of one factor on the response varies as another factor changes levels. Figure 2.3 shows the interaction between program size and cache size.

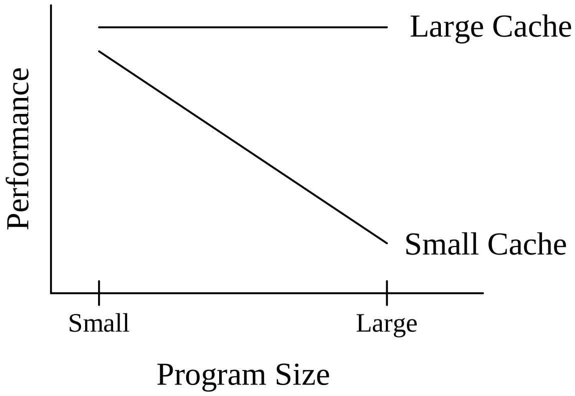


Figure 2.3: Interaction between program size and cache size

For a large cache, program size has no effect on performance, but for a small cache, program size is inversely proportional to performance. The obvious interpretation is that the large program does not fit in the small cache, producing a higher cache miss rate and hence decreased performance.

The effect that a single factor has on a response is called a *main effect*. In the example above, Figures 2.1 and 2.2 showed a graphical representation of the main effects produced by cache size and program size. The combined effect on a response from two or more factors is an *interaction* [7]. The previous example showed the effect that the *two-factor interaction* (2FI) of cache size and program size had on microprocessor performance. Interactions can comprise any number of factors. The abbreviation for any N-factor interaction is "NFI." Although interactions can be shown graphically as in the above example, numerical estimations of factor effects are the most useful in practice.

The estimation of factor effects is only the first step in interpreting the results of an experimental design. After effects are estimated, an empirical model is constructed and statistical testing is performed to determine the validity of the model. In the remainder of this chapter, we discuss the construction of the types of experimental designs available in Möbius and the calculation of the effects for each type. The empirical model-building and statistical analysis of the model will be discussed in the next chapter.

2.4 Factorial Designs

One experimentation strategy that considers interactions is the *factorial* approach. Rather than vary factors one at a time, factorial experimental designs vary factors together. One of the most important features of the factorial design is its efficient use of experimental data. Factorial designs allow the experimenter to perform the minimal number of experiments necessary to estimate the main effects and any interactions of interest. Factorial designs are by far the most commonly used experimental designs in practice. The remainder of this section will present the details of the types of factorial designs available in Möbius.

2.4.1 Full factorial

Full factorial designs are the most complete type of factorial design, because the full range of combinations of all levels of all factors is tested [8]. To illustrate the full factorial design, consider a study to design a workstation that yields the highest performance. The workstation has three parameters that will be varied: amount of RAM, number of processors,

Table 2.1: Full factorial levels for workstation

Factors	Levels
A: RAM (GB)	1,2,4,8,16
B: Num_processors	1,2,3,4
C: Hard Disk (GB)	300, 500, 700, 900

and hard drive size. Table 2.1 shows the possible levels to assign each workstation parameter. Each factor is assigned a corresponding capital letter for abbreviated future reference. Table 2.1 shows five possible values for amount of RAM, and four values for both number of processors and hard disk size. In order to measure performance at all combinations of all three factors at every level, the number of experiments needed, *n*, is equal to the product of the number of levels of each factor:

$$\begin{aligned} n &= (\text{RAM levels}) \times (\text{Processor levels}) \times (\text{Hard disk levels}) \\ &= 5 \times 4 \times 4 = 80 \end{aligned}$$

The advantage of the full factorial design is that it allows for computation of all main effects and all possible interactions. The disadvantage is that the cost to perform all experiments is usually too high. One solution to the problem is to reduce the number of variable factors in the study by assigning some factors to a fixed level. Another way to reduce the number of experiments is to reduce the number of levels for the factors. The most common full factorial design assigns only two levels to each factor. This type of experimental design is known as the 2^k *design*. The number of experiments in a 2^k design is equal to 2^k, where k is the number of factors [9]. Table 2.2 shows the levels in the workstation example when it is converted to a 2^3 design. This reduced design requires only eight experiments.

Table 2.2: 2^3 factorial levels for workstation

Factors	Levels	
	-	+
A: RAM (GB)	1	16
B: Num_processors	1	4
C: Hard Disk (GB)	300	900

Often, the response either increases or decreases continuously as a factor is increased from its minimum to maximum value. As a result, reducing a factor to two levels allows one to measure the effect that the factor has on the response as it is varied between the two extreme values. If the effect is significant, then more experiments can be performed at intermediate values in order to get a better understanding of the behavior of the response within the intermediate range of the factor. If the effect is insignificant, then the factor can be held constant, simplifying further experimentation. The process of determining which factors have a significant effect on the response is called *screening*. Fractional factorial designs (discussed in the next section) and 2^k designs are often used in the screening process. The low number of experiments necessary to perform a screening experimental design makes it ideal for an initial study designed to justify a more detailed examination of a factor's effect on the response. A few screening experiments performed early in a study can save valuable experimentation time later on [8].

After the workstation example has been reduced to a 2^3 design, it is easy to construct the *design matrix* in Table 2.3. A design matrix lists the factor levels for each experiment.

Table 2.3: Design matrix in geometric notation for 2^3 workstation

Experiment	**Factor**		
	A	B	C
1	-	-	-
2	+	-	-
3	-	+	-
4	+	+	-
5	-	-	+
6	+	-	+
7	-	+	+
8	+	+	+

Because each factor has only two levels, the notation + and – is used to represent the high and low levels, respectively. It is known as *geometric notation*. Other notations use a +1 and –1 or a 1 and 0 to represent the high and low levels of a factor. The geometric notation is derived from the corresponding *cube plot*, as seen in Figure 2.4 [8]. In the cube plot, each vertex is labeled with the corresponding experiment number in the design. Cube plots and their two-dimensional equivalent, the *square plot*, are useful for analyzing empirical models of the response when the vertices are labeled with response values. They provide an intuitive feel for how the response changes as each factor is varied from its low to high level.

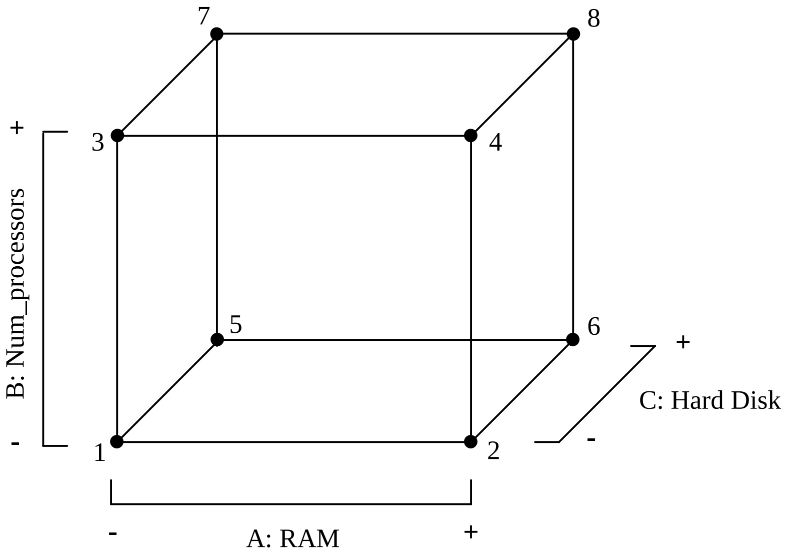


Figure 2.4: Cube plot with vertices labeled with experiment numbers

The number of *i*-factor interactions is easily calculated in a 2^k design. In the workstation example, the three 2FIs are AB, AC, and BC, and the one 3FI is ABC. In general, for any 2^k design, the number of *i*-factor interactions m_i is

$$m_i = \binom{k}{i} = \frac{k!}{i!(k-i)!} \qquad\qquad i \le k$$

Note that $i = 1$ for main effects. An estimate for the effect that each *i*-factor interaction has on the response can be calculated in a 2^k design.

To aid in the calculation of effects, a *sign table* is created [9]. Table 2.4 consists of the sign table for the workstation example. Before illustrating how the effects are calculated, we will explain the construction of the table entries. The first four columns of Table 2.4 are simply the design matrix from Table 2.3. Each factor is labeled as a main effect. The next four columns list all interactions among all factors. The interaction column entries are computed by multiplying the corresponding main effect entries in each row. A product with an even number of minus signs is positive (+). A product with an odd number of minus signs is negative (-). For example, the ABC interaction for experiment 2 is a plus sign, which is the product of the plus sign from main effect A and the two minus signs from main effects B and C. The identity column of all plus signs is labeled with **I** and is used to calculate the average response over all experiments. The last column consists of example response data represent-

Table 2.4: Sign table for 2^3 workstation design

Experiment	Effects							I	Response
	Main Effects			Interactions					Performance
	A	B	C	AB	AC	BC	ABC		
1	-	-	-	+	+	+	-	+	3
2	+	-	-	-	-	+	+	+	5
3	-	+	-	-	+	-	+	+	4
4	+	+	-	+	-	-	-	+	8
5	-	-	+	+	-	-	+	+	4
6	+	-	+	-	+	-	-	+	6
7	-	+	+	-	-	+	-	+	4
8	+	+	+	+	+	+	+	+	8

ing workstation performance scaled from 1 to 10. All data in the workstation example is fictitious. The example is presented for illustration purposes only.

The average response $\bar{y}$ is calculated by multiplying the identity column **I** with the response column and dividing the result by the number of experiments (2^k). In the workstation example, the average performance is

$$\begin{aligned} \bar{y} &= \frac{\mathbf{I}_{column} \cdot \mathbf{Response}_{column}}{2^k} \\ &= \frac{+3+5+4+8+4+6+4+8}{2^3} = \frac{42}{8} = 5.25 \end{aligned}$$

To estimate the effects, we first need to calculate the *contrast* of each effect [8]. In general, a contrast is a linear combination of parameters of the form

$$\Gamma = \sum_{i=1}^{n} c_i u_i \qquad \ni \qquad \sum_{i=1}^{n} c_i = 0$$

For an effect contrast, the constants c_i are +1 or –1, according to the corresponding entry in the sign table. The parameters u_i for any effect are the response values. For example, the contrast for main effect A is the product of the A column with the response column in the sign table:

$$contrast_A = (-3+5-4+8-4+6-4+8) = 12$$

The symbol l_i represents the estimate of effect *i*. To calculate an effect estimate, simply divide the effect's contrast by $2^{k\text{ -1}}$:

$$l_i = \frac{contrast_i}{2^{k-1}}$$

For example, the estimate of the main effect A on workstation performance is

$$l_A = \frac{contrast_A}{2^{3-1}} = \frac{12}{4} = 3$$

The remaining effect estimates are calculated similarly. Table 2.5 lists all effect estimates for the 2^3 workstation design. The first column in Table 2.5 lists the three main effects and the four interactions. Based on the magnitudes of the effect estimates, it appears that RAM (A) and the number of processors (B) have the largest effects on performance. It also appears that the interaction of these two factors (AB) may also have a significant impact on processor performance. In order to test whether this interaction is statistically significant, we must first construct an empirical model. The discussion of the analysis will be delayed until the next chapter.

Table 2.5: Factor effect estimates for workstation example

Effect	**Effect Estimate**
A	3.0
B	1.5
C	0.5
AB	1.0
AC	0.0
BC	-0.5
ABC	0.0

2.4.2 Fractional factorial

For systems with a large number of factors, even 2^k factorial designs can outgrow the resources of the experimenter. *Fractional factorial* designs alleviate this problem by requiring that only a fraction of the full 2^k experiments be run. The drawback is that less information is available from fractional factorial results than from the full 2^k factorial results [7].

Specifically, it is not possible to estimate all interactions among all factors in a fractional factorial design. However, the inability to estimate all interactions is rarely a problem, because most interactions are not significant. The *sparsity of effects principle* states that most systems are driven primarily by some of the main effects and low-order interactions, and most high-order interactions are negligible [8]. Often, the experimenter has enough familiarity with the system being modeled that some interactions are known to be insignificant. This a priori knowledge eliminates the necessity to perform those experiments that measure the insignificant effects.

For any number of factors, there are several factorial designs from which to choose. The experimenter chooses the design *fraction* with the minimum number of experiments that allow for the calculation of all factor effects and interactions of interest. For example, a *half-fraction* performs half the number of experiments that a full 2^k design does; a *quarter-fraction* performs one-fourth the number of experiments, and so forth. It is very common to count all interactions of order three or higher as insignificant. Therefore, only main effects and 2FIs need to be estimated. Figure 2.5 shows the minimum design fraction necessary to estimate all main effects and 2FIs for up to 15 factors, assuming all interactions of order three or higher to be insignificant. For less than eight factors, the experimenter must still

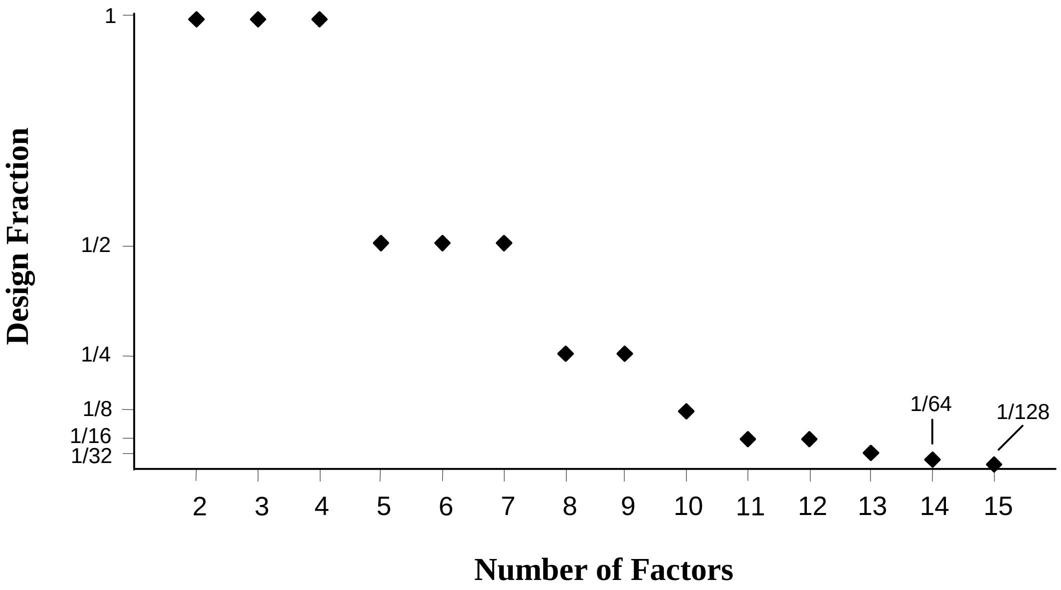


Figure 2.5: Minimal design fraction needed to estimate 2FIs, ignoring 3FIs and higher

perform at least half of the full 2^k experiments to get the desired effect estimates. However, as the number of factors increases, the savings increase dramatically. In the extreme case of 15 factors, rather than perform the full 2^k design of over 32,000 experiments, the experimenter need only perform $2^{15} \times 1/128 = 256$ experiments.

Another advantage of fractional factorial designs is the fact that they can be performed iteratively. It is possible to combine the runs from previously run fractional factorial designs into a larger design that estimates the desired main effects and interactions. Experimentation is a learning process. Experimental designs usually start as highly fractionated factorial designs used for screening. After we learn which factors are significant and which ones should be dropped from later experiments, the design may evolve into a less fractionated factorial design in which interactions can be explored. If desired, the region of exploration can be changed for some factors and examined in detail using more exhaustive methods, such as response surface techniques, which will be discussed later. It is generally a mistake to employ a single, large, exhaustive experimental strategy at the beginning of a study. At the start of a study, an experimenter does not usually know which factors are important, the ranges over which the important factors should be varied, and the number of levels to use. At each incremental phase of a well-planned study, the experimenter will learn more about the unknowns and can make decisions on how to proceed to the next step. The result is an economical and efficient learning process to determine the behavior of the system being modeled [8].

2.4.3 Half-fraction factorial

The simplest fractional factorial design consists of exactly half of the runs of a full 2^k design, and is called a *half-fraction* or a *$2^{k\text{-}1}$* design because it consists of $½ \times 2^k = 2^{k-1}$ experiments. The first decision to make when constructing a half-fraction is that of choosing which $2^{k\text{-}1}$ experiments from the full 2^k design should be selected. The half-fraction of the 2^3 workstation example from the previous section will be constructed in order to illustrate the $2^{k\text{-}1}$ design. The two choices for the $2^{3\text{-}1}$ design are shown in Figure 2.6. The vertices are chosen so that no selected vertex is adjacent to another selected vertex. The experiment numbers used for the vertices from each design are the same as those used for the full 2^3 de-

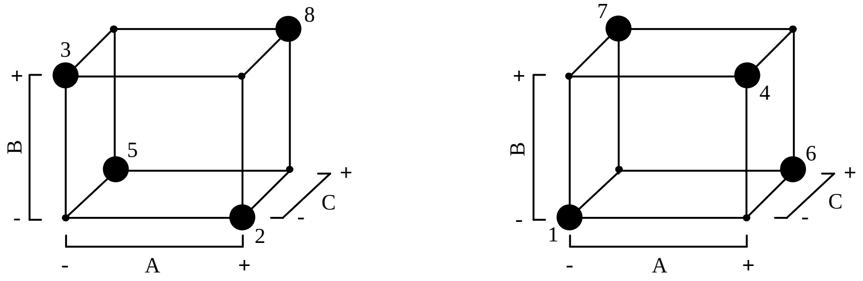


Figure 2.6: The two one-half fractions of the 2^3 design

sign from Table 2.4. The choice of vertices ensures that the half-fraction is *orthogonal*. Orthogonality is a necessary property that must be satisfied by all two-level factorial designs in Möbius. The details of orthogonality will be discussed later. In practice, it does not matter which half-fraction is chosen. Both fractions belong to the same *family*. When two or more fractional factorials combine to form a complete 2^k design, the fractional factorial designs are said to belong to the same family. The fraction on the left side of Figure 2.6 will be chosen for illustration [8].

Table 2.6 shows the sign table for the 2^{3-1} workstation example. The rows are taken directly from Table 2.4 and placed in *standard order*. Notice that the ABC interaction column is identical to the identity column. The *defining relation* for this design is

$$I = ABC$$

The defining relation is used to calculate the *alias structure* or *confounding* of the design. The drawback to fractional factorial designs is that independent estimates can only be calculated for some effects. Two or more effects are said to be *aliased* or *confounded* if it is not possible to determine the effect to which the calculated effect estimate corresponds. For ex-

Table 2.6: Sign table for 2^{3-1} workstation design

Experiment	**Main Effects**			**Interactions**				**I**	**Response**
	A	B	C	AB	AC	BC	ABC		Performance
5	-	-	+	+	-	-	+	+	4
2	+	-	-	-	-	+	+	+	5
3	-	+	-	-	+	-	+	+	4
8	+	+	+	+	+	+	+	+	8

ample, a main effect may be aliased to a 2FI. If a priori knowledge of the system being modeled can lead the modeler to conclude that the 2FI is negligible, then the effect estimate can be assigned to the main effect. If there exists no external knowledge to distinguish one effect from the other, and if the effect is large enough to warrant investigation, then further experiments will be necessary in order to *de-alias* or *unconfound* the two effects [8], [9].

In the workstation example, the alias structure for the 2^{3-1} design can be computed by multiplying each effect by the defining relation. Some rules must be followed when multiplying effects. First, the multiplicative identity is I. That is, for any effect X,

$$\mathrm{I} \times \mathrm{X} = \mathrm{X}$$

Second, any effect raised to an even power equals the identity, and any effect raised to an odd power is equal to the original effect. So, for any effect X,

$$\mathrm{X}^{2n} = \mathrm{I} \qquad , \qquad \mathrm{X}^{2n+1} = \mathrm{X} \qquad\qquad \forall n \in \mathrm{Z}^{+}$$

where *n* is any nonnegative integer. To calculate the aliases for main effect A, multiply the defining relation by A:

$$\mathrm{A} \times \mathrm{I} = \mathrm{A} \times \mathrm{ABC}$$
$$\mathrm{A} = \mathrm{A}^{2} \times \mathrm{BC} = \mathrm{I} \times \mathrm{BC} = \mathrm{BC}$$

Similar multiplications of each effect and the defining relation yield two more unique equations:

$$\mathrm{B} = \mathrm{AC}$$
$$\mathrm{C} = \mathrm{AB}$$

Each main effect is aliased with a 2FI. Careful inspection of Table 2.6 reveals that the columns are identical in each pair of aliased main effect and 2FI [9].

Calculation of effect estimate l_i is done in the same manner as in the previous section according to the formula

$$l_i = \frac{contrast_i}{2^{(k-1)-1}} \,.$$

The three aliased effect estimates are presented in Table 2.7. Actually, because each main effect is aliased, the estimates of effects A, B, and C are really the estimates of A+BC, B+AC, and C+AB. The common notation for effect estimation aliasing is

$$l_A \rightarrow A + BC$$
$$l_B \rightarrow B + AC$$
$$l_C \rightarrow C + AB$$

For example, the effect estimate l_C is actually equal to the *sum* of the estimate of effect C and the estimate of effect AB. It appears from Table 2.7 that the effects of B and C are equal. We know from previously calculating the effect estimates of the full 2^3 design that l_C is only 0.5, but l_{AB} is 1.0. The two effects add to 1.5 to produce the entry for l_C in Table 2.7. This example shows how important it is to pay attention to the alias structure in a fractional factorial design. If we ignore aliasing and accept Table 2.7 as the final result, we would conclude that C (hard disk size) has the same effect on workstation performance as B (the number of processors). In fact, it is not C, but the *interaction* of A and B that contributes the most to the change in observed performance. Hard disk size is negligible compared to the other factors. However, there is no way to determine how the aliased effect of l_C in the 2^{3-1} design is distributed between l_C and l_{AB} without performing the remaining four experiments to complete the 2^3 design. Performing additional experiments to de-alias a possibly significant effect is common and appeals well to the iterative nature of fractional factorial experimental design.

Table 2.7: Aliased effect estimates for 2^{3-1} workstation design

Effect	Aliased Effects	Effect Estimate
A	A + BC	2.5
B	B + AC	1.5
C	C + AB	1.5

2.4.4 $2^{k\text{-}p}$ fractional factorial

The previous section presented a special case of the general 2^{k-p} fractional factorial design, where k is the number of factors and p defines the *fraction* as

$$\text{fraction} = \frac{1}{p}$$

The half-fraction presented earlier had $k = 3$ and $p = 1$ for a total of $½ \times 2^3 = 2^{3-1} = 4$ experiments. Designs with lower fractions require fewer experiments [9]. In this section, the construction of the general $2^{k\text{-}p}$ fractional factorial design will be presented.

The first step in constructing a fractional factorial design is choosing the fraction. Two main issues influence the choice of fraction. The first is the number of experiments that can be afforded. For example, if a limited amount of simulation time is budgeted for a particular phase of a study, then a fraction must be chosen so as not to exceed the allocated experiment budget. The second issue affecting the choice of fraction is the amount of aliasing that is allowed. For example, if only main effects, not interactions, need to be measured, then a lower fraction may suffice. If, however, interactions cannot be ruled out as negligible, then a less fractionated (higher fraction) design is necessary.

One aid in the construction of a fractional factorial design is the *design resolution*, which summarizes the aliasing structure of a fractional factorial design. The official definition states that a design is of resolution *R* if no *x*-factor effect is aliased with any effects containing fewer than $R - x$ factors [8]. An effect containing *x* factors is said to be of *order x*. The best way to understand design resolution is to use the mnemonics in Table 2.8. One should think of design resolution as the sum of two numbers. For resolution *R*, to determine the order *x* of the smallest interaction that is aliased with a main effect (one-factor interaction), solve

$$R = 1 + x$$

The 1 represents the order of a main effect and *x* represents the *x*-factor interaction aliased to the main effect. Table 2.8 lists the definitions for the three most commonly used design

Table 2.8: Mnemonics for understanding design resolution

Resolution	**Mnemonic**	**Definition**
III	1 + 2	Main effects aliased with 2FIs
IV	1 + 3	Main effects aliased with 3FIs
	2 + 2	2FIs aliased with 2FIs
V	1 + 4	Main effects aliased with 4FIs
	2 + 3	2FIs aliased with 3FIs

resolutions. Design resolution is denoted as a Roman numeral. For resolution V $(=1+4)$ designs, main effects (or one-factor interactions) are aliased with 4FIs, following the mnemonic $1+4=5$. The more complete interpretation is that one-factor interactions are aliased with *at least* 4FIs and may be aliased to higher-order interactions. The resolution just states the worst-case (lowest-order) effect aliased to a given effect. For example, a fractional factorial design to estimate only main effects (ignoring interactions) would require only a resolution III design, which is the minimum resolution required to ensure that no main effect is aliased to another main effect. Most screening designs have resolution III. Alternatively, a design that estimates all main effects and 2FIs, ignoring 3FIs and higher, requires a resolution of at least V. A resolution IV design would not suffice, because it would alias some 2FIs with other 2FIs. In Section 2.4.2, Figure 2.5 can be more correctly described as a graph of the minimum design fraction required for a resolution V design for up to 15 factors. However, the definition of design resolution had not yet been presented.

In general, the design with the highest possible resolution for the required level of fractionation is desired. Higher-resolution designs alias main effects to higher-order interactions, which are more likely to be negligible than lower-order interactions. That allows for independent estimates of main effects when the aliased interactions are assumed to be insignificant. Design resolution is typically embedded into the $2^{k\text{-}p}$ notation as a subscript. The design presented in the previous section was a 2_{III}^{3-1} design, because at least one main effect was aliased to a 2FI, giving it resolution III [7].

The construction of any $2^{k\text{-}p}$ design is divided into two phases: the *base design* construction and the *design generator* selection. The base design of any $2^{k\text{-}p}$ design is simply a full two-level factorial in $k-p$ factors [8]. Table 2.9 shows an example 2_{III}^{6-3} design, with a box around the base design consisting of factors A, B, and C. Any factor can be included in the base design; A, B, and C were chosen for convenience. After building the base design, the remaining *p* factor columns are assigned values according to the *design generators*. A design generator is simply a product of some of the base design effects. Table 2.9 shows that column D is assigned values equal to the product of columns A and B. Similarly, $\mathrm{E}=\mathrm{AC}$ and $\mathrm{F}=\mathrm{BC}$. Design generator selection is very important, because the generators determine the aliasing structure and resolution of the design. Fortunately, Möbius automati-

Table 2.9: 2_{III}^{6-3} design highlighting base design

Experiment	**I**	**Base Design**			**Design Generators**		
		A	B	C	AB D	AC E	BC F
1	+	-	-	-	+	+	+
2	+	+	-	-	-	-	+
3	+	-	+	-	-	+	-
4	+	+	+	-	+	-	-
5	+	-	-	+	+	-	-
6	+	+	-	+	-	+	-
7	+	-	+	+	-	-	+
8	+	+	+	+	+	+	+

cally selects the design generators that produce the highest resolution and best aliasing structure for any given fraction.

Once the design generators have been selected, the defining relation can be calculated. The complete alias structure of the design can be calculated directly from the defining relation. Consider the 2_{III}^{6-3} design from Table 2.9. The three design generators are

$$\mathrm{D} = \mathrm{AB}$$
$$\mathrm{E} = \mathrm{AC}$$
$$\mathrm{F} = \mathrm{BC}$$

Multiplying each equation by its left-hand side yields the *defining contrast subgroup* [8]:

$$\mathrm{I} = \mathrm{ABD}$$
$$\mathrm{I} = \mathrm{ACE}$$
$$\mathrm{I} = \mathrm{BCF}$$

The defining relation consists of the products of all combinations of *words* in the defining contrast subgroup. In our example, there are $\binom{3}{2} = 3$ products from multiplying pairs of contrast subgroup words, and $\binom{3}{3} = 1$ product from multiplying all three words:

$$(ABD) \times (ACE) = BCDE$$
$$(ABD) \times (BCF) = ACDF$$
$$(ACE) \times (BCF) = ABEF$$
$$(ABD) \times (ACE) \times (BCF) = DEF$$

When combined with the defining contrast words, the entire defining relation is

$$\mathrm{I} \to \mathrm{I} + \mathrm{ABD} + \mathrm{ACE} + \mathrm{BCF} + \mathrm{DEF} + \mathrm{ABEF} + \mathrm{ACDF} + \mathrm{BCDE}$$

In general, for *p* design generators, the number of words in the defining relation is

$$\sum_{i=1}^{p} \binom{p}{i} = 2^{p} - 1$$

The design resolution is equal to the shortest word in the defining relation [7]. The shortest word in the above defining relation has length 3, and hence defines a resolution III design.

The aliases for the 2_{III}^{6-3} design are calculated by multiplying the defining relation by each effect. For example, the aliases of A are

$$\mathrm{A} \to \mathrm{A} + \mathrm{BD} + \mathrm{CE} + \mathrm{BEF} + \mathrm{CDF} + \mathrm{ABCF} + \mathrm{ADEF} + \mathrm{ABCDE}$$

For any $2^{k\text{-}p}$ design, the maximum number of effects that can be estimated is $2^{k\text{-}p}$ (including the average effect I) [9]. Table 2.10 lists the entire alias structure of the 2_{III}^{6-3} design. The assumption that three-factor and higher-order interactions are negligible simplifies the alias structure. In Table 2.10, low-order (more relevant) effects are emphasized in boldface.

Table 2.10: Alias structure for 2_{III}^{6-3} design

Effect	**Aliased Effects**
A	**A + BD + CE** + BEF + CDF + ABCF + ADEF + ABCDE
B	**B + AD + CF** + AEF + CDE + ABCE + BDEF + ABCDF
C	**C + AE + BF** + ADF + BDE + ABCD + CDEF + ABCEF
D	**D + AB + EF** + ACF + BCE + ACDE + BCDF + ABDEF
E	**E + AC + DF** + ABF + BCD + ABDE + BCEF + ACDEF
F	**F + BC + DE** + ABE + ACD + ABDF + ACEF + BCDEF
AF	**AF + BE + CD** + ABC + ADE + BDF + CEF + ABCDEF

Table 2.11: 2^3 workstation sign table

Experiment	**X**								**y**
	I	A	B	C	AB	AC	BC	ABC	
1	1	-1	-1	-1	1	1	1	-1	3
2	1	1	-1	-1	-1	-1	1	1	5
3	1	-1	1	-1	-1	1	-1	1	4
4	1	1	1	-1	1	-1	-1	-1	8
5	1	-1	-1	1	1	-1	-1	1	4
6	1	1	-1	1	-1	1	-1	-1	6
7	1	-1	1	1	-1	-1	1	-1	4
8	1	1	1	1	1	1	1	1	8

2.4.5 Effects estimation using matrices

The methods of effect estimation presented in the previous sections involved multiplying columns of the design's sign table. Column multiplication is best performed using matrix math. In this section, the method of effect estimation used in Möbius will be presented.

Consider once again the 2^3 workstation example from Section 2.4.1. The sign table is repeated in Table 2.11 with the columns reordered and relabeled. The usual geometric notation of **-** and **+** for factor levels is replaced with the numbers –1 and 1, respectively. The values –1 and 1 represent the *coded* values of the low and high factor levels. Coded values are equal to the *natural* values scaled from –1 to +1. For example, the natural low and high levels for factor C (hard disk size) in the workstation example are 300 and 900 GB, respectively. For any factor X, the average level $\bar{x}$ is defined as

$$\bar{x} = \frac{x_{low} + x_{high}}{2}$$

The *center* x_{cen} of factor X is defined as

$$x_{cen} = \frac{x_{high} - x_{low}}{2}$$

The relationship between the natural and coded values of factor X is

$$x_{coded} = \frac{x_{natural} - \bar{x}}{x_{cen}}$$

For example, setting hard disk size to the natural high level of 900 GB yields a coded value of

$$c_{coded} = \frac{900 - \bar{c}}{c_{cen}} = \frac{900 - \frac{300+900}{2}}{\frac{900-300}{2}} = \frac{900-600}{300} = 1$$

All internal calculations in Möbius use coded factor values in order to increase numerical stability [10].

Two quantities, the matrix **X** and the vector of observed values **y** extracted from Table 2.11, are

$$\mathbf{X} = \begin{bmatrix} 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 & 1 & -1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 \\ 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} 3 \\ 5 \\ 4 \\ 8 \\ 4 \\ 6 \\ 4 \\ 8 \end{bmatrix}$$

Note that the columns of **X** are orthogonal. That is, the inner product of any number of columns is zero. Therefore, $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ is diagonal:

$$\mathbf{X}^{\mathrm{T}}\mathbf{X} = \begin{bmatrix} 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 8 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8 \end{bmatrix}$$

Using the method of least squares, it can be shown that the solution to the system

$$\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{y}$$

is the *effect estimate vector* $\hat{\boldsymbol{\beta}}$. The first element of $\hat{\boldsymbol{\beta}}$ is the average response value, and the remaining elements of $\hat{\boldsymbol{\beta}}$ are the *half-effects*. A half-effect is simply equal to one-half of the corresponding effect estimate [7]. The method of least squares measures the effect across one unit change in a factor, whereas the contrast method measures the effect across two factor units (-1 to +1). The ordering of the estimated half-effects in $\hat{\boldsymbol{\beta}}$ is the same as in the columns of **X**. Solving for $\hat{\boldsymbol{\beta}}$ yields

$$\begin{aligned} \mathbf{X}\hat{\boldsymbol{\beta}} &= \mathbf{y} \\ (\mathbf{X}^{\mathrm{T}}\mathbf{X})\hat{\boldsymbol{\beta}} &= \mathbf{X}^{\mathrm{T}}\mathbf{y} \\ \hat{\boldsymbol{\beta}} &= (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y} \end{aligned} \tag{2.1}$$

In general, matrix inversion is nontrivial [8]. However, as shown earlier, $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ is diagonal. Therefore, the inverse of $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ is trivial, as is the computation of $\hat{\boldsymbol{\beta}}$. Table 2.12 lists the values of $\hat{\boldsymbol{\beta}}$ and the corresponding effect estimates from Table 2.5 in Section 2.4.1. Note that each effect estimate, with the exception of the average, is double the corresponding value in $\hat{\boldsymbol{\beta}}$. Effect estimates will equal double the corresponding least squares estimate for any *balanced* design [8]. In balanced designs, the low and high levels of any factor occur an equal number of times. All default experimental designs in Möbius are balanced, but can be made unbalanced if the user modifies the factor levels.

The vector $\hat{\boldsymbol{\beta}}$ is important not only in effect estimation, but also in the construction of the empirical model of the response. The next chapter will present the details of least squares estimation and multiple linear regression model construction.

Table 2.12: Comparison of least squares $\hat{\boldsymbol{\beta}}$ vector to effect estimates

Effect	$\hat{\boldsymbol{\beta}}$	**Effect Estimate**
I	5.25	5.25
A	1.5	3.0
B	0.75	1.5
C	0.25	0.5
AB	0.5	1.0
AC	0.0	0.0
BC	-0.25	-0.5
ABC	0.0	0.0

2.4.6 Plackett-Burman designs

One special type of two-level fractional factorial design developed by Plackett and Burman [11] is useful for screening a large number of factors. The number of experiments in Plackett-Burman (PB) designs is required to be a multiple of four. The Möbius DOE study editor allows up to 83 factors to be studied in one PB design. The number of factors is one, two, or three less than the number of experiments.

During an initial study of a large system with many factors, the experimenter may not know which factors have a significant influence on the response. PB designs test the largest number of factor main effects with the least number of experiments. The effects of interactions cannot be estimated. The aliasing structure is very complex and will not be presented. Interested readers should consult [11] for a detailed presentation of PB aliasing. All PB designs have resolution III, and every main effect is partially aliased with all 2FIs not involving itself. For example, in the 11-factor PB design, the BC interaction is aliased with the main effects A, D, E,…, K. PB designs are also useful for robustness testing [8]. In a robustness test, all main effects are assumed to be insignificant. Any main effects that violate this assumption are easily identifiable, and further experiments can be performed to reveal the exact nature of the effect.

Factorial designs in which the number of factors is one less than the number of experiments are called *saturated* [7]. PB designs in which the number of experiments is a power of two are equivalent to the corresponding 2^{k-p} design with the same number of factors. To show equivalence, the **X** matrix must be both row and column permuted until the 2^{k-p} fractional factorial **X** matrix is achieved. All **X** matrices in PB designs are *Hadamard matrices* [12]. An $n \times n$ Hadamard matrix $\mathbf{H}_n$ has the unique property

$$\mathbf{H}_n^{\mathrm{T}} \mathbf{H}_n = n\mathbf{I}_n$$

where $\mathbf{I}_n$ is the $n \times n$ identity matrix. This property simplifies effect calculations. The effect estimate vector $\hat{\boldsymbol{\beta}}$ defined in Equation (2.1) simplifies to

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y} = (n\mathbf{I}_n)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y} = \frac{1}{n}\mathbf{X}^{\mathrm{T}}\mathbf{y}$$

The experimenter should use caution when employing PB designs. If any interaction is significant, the fact that it is aliased to more than one main effect can lead to bogus conclusions about a main effect's significance. PB designs are encouraged only when there are too many factors for a 2^{k-p} design. Any main effects that do not appear be negligible should be investigated further using a higher-resolution design in order to determine if an interaction is the cause of the observed change in the response.

2.5 Response Surface Designs

One of the goals of experimental design in system modeling is to collect enough data to develop an empirical model that can be manipulated mathematically to predict system behavior within some region of the design space. In the next chapter, the methods of least squares and multiple linear regression will be used to develop this empirical model. For now, it will be assumed that the empirical model of the response *y* is a polynomial of the form

$$y = f(\mathbf{x}) + \varepsilon$$

where **x** is a vector of factor values, *f* is some polynomial function of the factors, and ε is the error between the observed response and the value predicted by *f*(**x**). The expected value of the response is

$$E(y) = f(\mathbf{x}) = \eta\,.$$

The surface represented by $\eta = f(\mathbf{x})$ is the *response surface* [8].

For example, consider the following response surface defined by two factors A and B:

$$\eta = 3 + 13x_{\mathrm{A}} + 69x_{\mathrm{B}} - 5x_{\mathrm{A}}^2 - 7x_{\mathrm{B}}^2 + 3x_{\mathrm{A}}x_{\mathrm{B}}$$

A contour plot of the surface is shown in Figure 2.7. Response contour values are computed by substituting x_{A} and x_{B} with the corresponding A and B coordinates on the graph and evaluating η. The plot shows that the response is maximized within the design region shown. If the goal was to maximize the response, the experimenter can read from the plot the input factor values that achieve the goal. The accuracy of the optimal factor levels depends on how accurately the empirical model represents the real system behavior. In order to effectively estimate the model parameters that accurately represent the system behavior, specific experimental designs called *response surface designs* must be constructed.

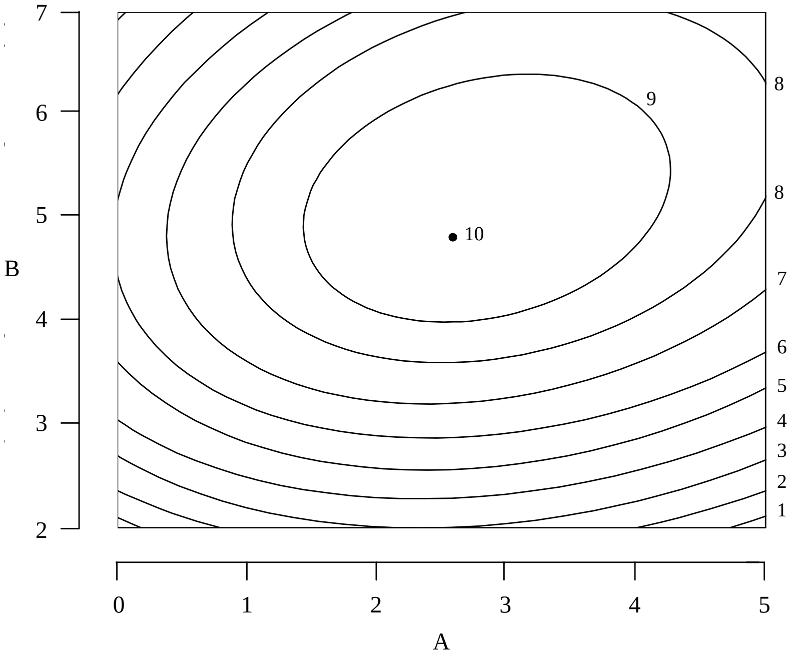


Figure 2.7: Contours of example response surface

Response surface designs are typically not performed during the early stages of system modeling. The sequential nature of experimental design dictates that negligible factors must be eliminated and significant interactions must be identified before in-depth analysis is performed on any specific factor. Response surface designs are employed when a higher-order (quadratic or cubic) polynomial model is necessary to accurately predict a response or when the experimenter simply wants more detail of the response behavior in a specific region. That increased detail facilitates the optimization and system tuning desired by the experimenter [13]. The next two sections present the construction of the two most popular response surface designs: the *central composite* and the *Box-Behnken* designs.

2.5.1 Central composite

After a two-level factorial design has been completed, the resulting empirical model may not predict the observed response values well. A model that shows such inaccuracy is said to exhibit *lack-of-fit*, meaning that the polynomial model does not fit well to the observed data. The most common cause of lack-of-fit in a factorial design is the inability of the model to predict response curvature in the design region. In order to effectively model a response with quadratic behavior, it is necessary to use a polynomial with degree higher than that provided by factorial designs. The most popular experimental design used for modeling quadratic response behavior is the *central composite design* or *CCD* [10].

A CCD with k factors consists of a base factorial design with 2^{k-p} experiments, $2k$ *axial* or *star* experiments, and one *center point* experiment. Figure 2.8 shows a CCD with two factors. The vertices of the square in Figure 2.8 represent the four 2^2 factorial experiments. Each of the four axial points is at distance α from the center point at the origin. The axial and center points provide additional information to the factorial data, so that a quadratic or higher-degree polynomial model can be fit to the response [13].

The fact that each CCD contains a two-level factorial design makes the CCD well-suited for sequential experimentation. Often, the factorial experiments are already complete before the CCD is created. Early factorial experiments may produce a model that does not fit the observed response. Only $2k + 1$ additional experiments need to be run in order to complete the CCD design, enabling the creation of a higher-order model to better fit the data. Table 2.13 shows the design matrix and effects columns for the design in Figure 2.8. The first four rows correspond to the base 2^2 factorial design.

All two-level factorial designs are orthogonal, meaning that $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ is a diagonal matrix. This diagonal matrix is easily invertible, and effect estimation is quite simple. Unfortunately, response surface designs are not orthogonal. From Table 2.13,

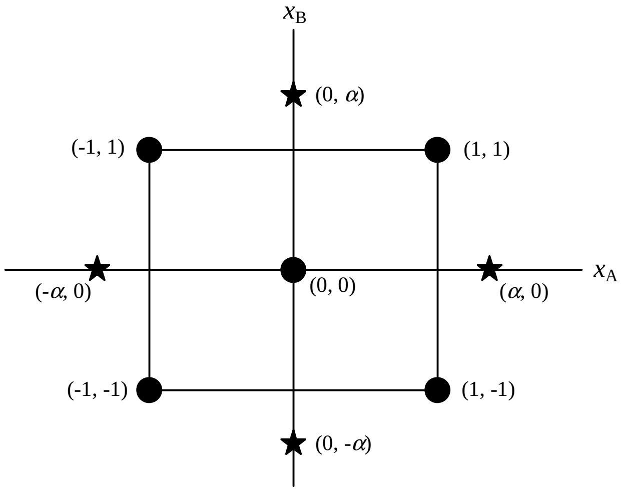


Figure 2.8: Central composite design with two factors

Table 2.13: Central composite design matrix and effects columns for two factors

Experiment	**X**					
	I	A	B	A^2	B^2	AB
1	1	-1	-1	1	1	1
2	1	1	-1	1	1	-1
3	1	-1	1	1	1	-1
4	1	1	1	1	1	1
5	1	$-\alpha$	0	α^2	0	0
6	1	α	0	α^2	0	0
7	1	0	$-\alpha$	0	α^2	0
8	1	0	α	0	α^2	0
9	1	0	0	0	0	0

$$\mathbf{X} = \begin{bmatrix} 1 & -1 & -1 & 1 & 1 & -1 \\ 1 & 1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -\alpha & 0 & \alpha^2 & 0 & 0 \\ 1 & \alpha & 0 & \alpha^2 & 0 & 0 \\ 1 & 0 & -\alpha & 0 & \alpha^2 & 0 \\ 1 & 0 & \alpha & 0 & \alpha^2 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Setting $\alpha = \sqrt{2}$, for example, yields

$$\mathbf{X}^{\mathrm{T}}\mathbf{X} = \begin{bmatrix} 9 & 0 & 0 & 8 & 8 & 0 \\ 0 & 8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 8 & 0 & 0 & 0 \\ 8 & 0 & 0 & 12 & 4 & 0 \\ 8 & 0 & 0 & 4 & 12 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 \end{bmatrix}$$

Clearly, $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ is not a diagonal matrix. Effect estimation for response surface designs follows the same formula as in factorial designs:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}$$

However, the computation is more complex without orthogonality.

Often, the goal is to find the optimal response value in the region of interest. Because the location of the optimum is unknown, it would be desirable to have the empirical model's prediction error be uniform throughout the design region. Unfortunately, a uniform prediction error would require many design points within the design region. Another option would be to require that the prediction error be equal at all points at a given distance from the design center. Box et al. [7] showed that setting $\alpha = \sqrt[4]{2^{k-p}}$ makes the variance of the predicted response $\hat{y}$ constant on spheres centered at the design center. A design that provides equal precision of estimation in all directions from the center is called a *rotatable* design [13]. Figure 2.9 shows the contours of the standard deviation of $\hat{y}$ for the quadratic polynomial model with two factors. Because the contours are concentric circles, the standard deviation of $\hat{y}$ remains unchanged as the design is rotated about the center.

One limitation to a rotatable CCD is that the factor values at the axial points must be valid in the system being modeled. Figure 2.8 illustrates that the axial points can exceed the low and high limits of the base factorial design. If those values are not realistic, then a perfectly rotatable CCD may not be possible. Luckily, another type of CCD, called the *face-centered cube*, alleviates the problem of unrealistic factor levels while providing a relatively consistent standard deviation of predicted response [8]. Figure 2.10 shows a three-factor face-centered cube CCD. Note that each axial point is at the center of a face of the cube. There are only three levels for each factor in the design: the low and high factorial levels and the center point. A contour plot of the standard deviation of the predicted response $\hat{y}$ is presented in Figure 2.11. The plot shows that even though the design is not rotatable, the standard deviation varies little over a relatively large region around the center [13].

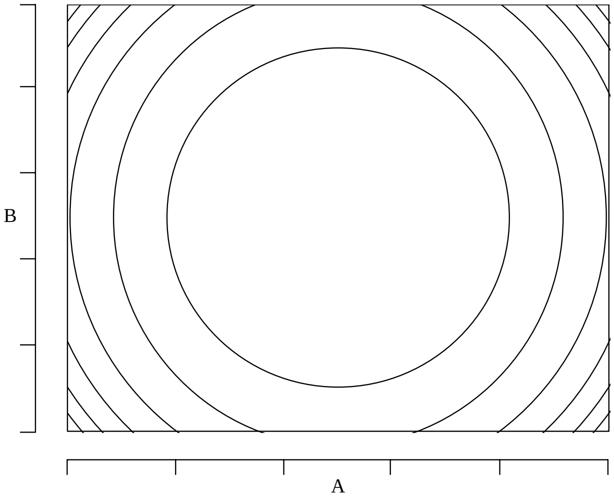


Figure 2.9: Contours of constant standard deviation of $\hat{y}$ for rotatable CCD

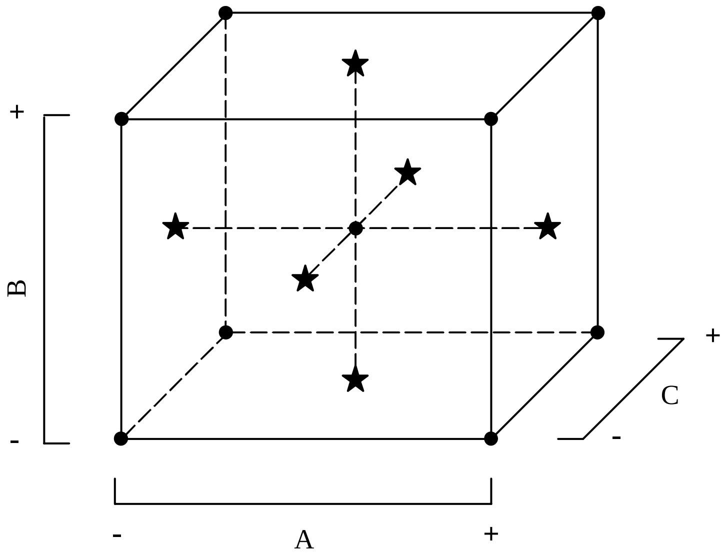


Figure 2.10: Face-centered cube with three factors

2.5.2 Box-Behnken

Another response surface design that uses three levels for each factor is the *Box-Behnken* or *BBD* design, named after its creators Box and Behnken [14]. Figure 2.12 shows the BBD with three factors. With the exception of the center point, each design point lies at the midpoint of an edge of the cube. In addition, each point lies on a sphere with radius $\sqrt{2}$ and center at the origin. BBDs are well-suited for spherical regions, for which prediction at

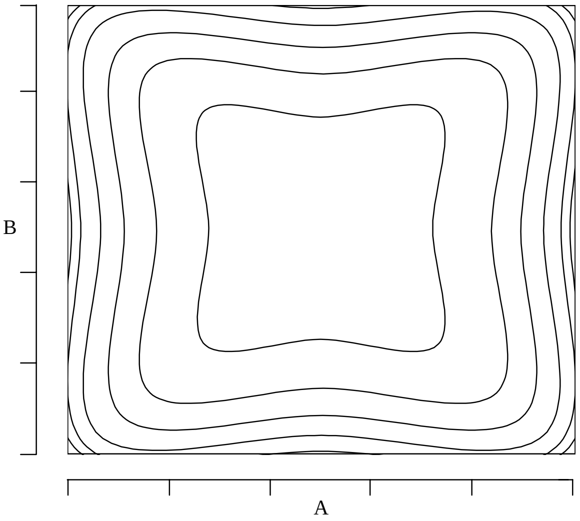


Figure 2.11: Contours of constant standard deviation for the face-centered cube

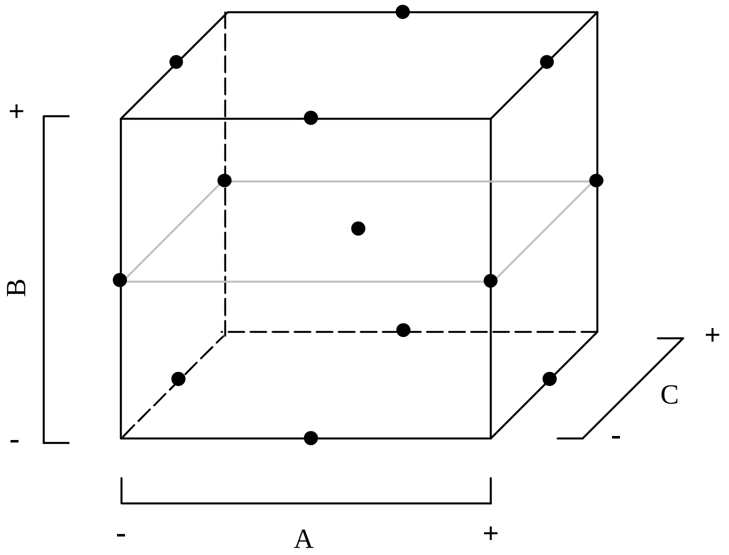


Figure 2.12: Box-Behnken design with three factors

the corners of the design region is not needed. When quadratic polynomial models are being estimated, BBDs are rotatable or nearly rotatable [8].

Table 2.14 shows the design matrix for the BBD with three factors. Each BBD is composed of blocks of two-level factorial runs centered at the origin. The three-factor BBD in Table 2.14 has three blocks of 2^2 factorial runs. Each block is outlined with a dotted line. One advantage of the BBD is that, like the face-centered cube CCD, it has no factor levels that exceed the low and high limits of the embedded factorial design, ensuring that all factors are assigned realistic values. BBDs are in general more economical than CCDs when quadratic models are being fitted [13]. One disadvantage of the BBD is that the special block structure (called an *incomplete block*) is not available for 8, 13, 14, 15, or more than 16 factors. The general iterative nature of DOE is to select only the vital few factors for in-depth analysis. Therefore, it is unlikely that a single response surface design would be created for more than 16 factors.

Table 2.14: Design matrix for Box-Behnken design with three factors

Experiment	**A**	**B**	**C**
1	-1	-1	0
2	1	-1	0
3	-1	1	0
4	1	1	0
5	-1	0	-1
6	1	0	-1
7	-1	0	1
8	1	0	1
9	0	-1	-1
10	0	1	-1
11	0	-1	1
12	0	1	1
13	0	0	0

3. STATISTICAL ANALYSIS OF EXPERIMENTAL RESULTS

3.1 Analysis Flow

After completing an experimental design and gathering all experimental data, the experimenter needs to make the most effective use of the results. The goal is to predict system behavior within the region of interest in the design space. In order to perform this prediction, it is necessary to construct an empirical model of the response. To build the empirical model, the experimenter selects the main effects and interactions that are thought to have a significant effect on the response. Möbius then calculates the parameters for a *regression model* that uses the selected effects to predict the response values. Figure 3.1 shows the flow of events from model construction to response prediction. The initial model is constructed from any potentially significant effects based on the effect estimates (as calculated in the previous chapter). The regression model is then analyzed and determined to be either good or bad. A *good* model satisfies the following three requirements:

- Model must fit experimental data accurately.
- Each effect must be statistically significant.
- Assumptions on input data must be valid.

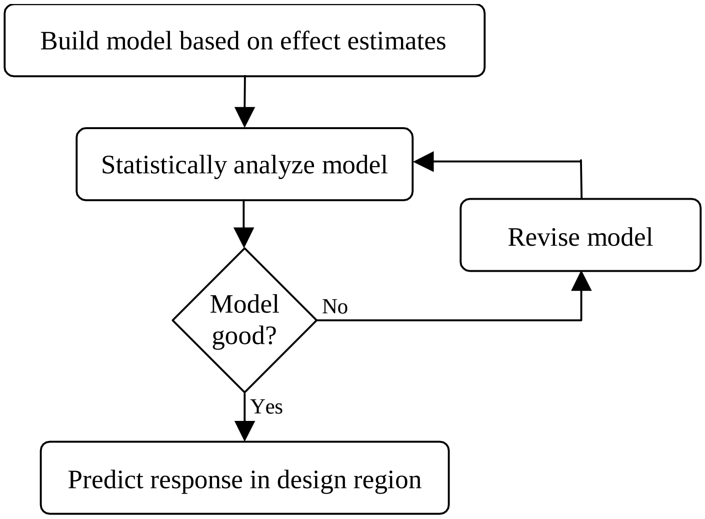

Build model based on effect estimates
Statistically analyze model
Revise model
Model good?
No
Yes
Predict response in design region

Figure 3.1: Flow of model construction and analysis

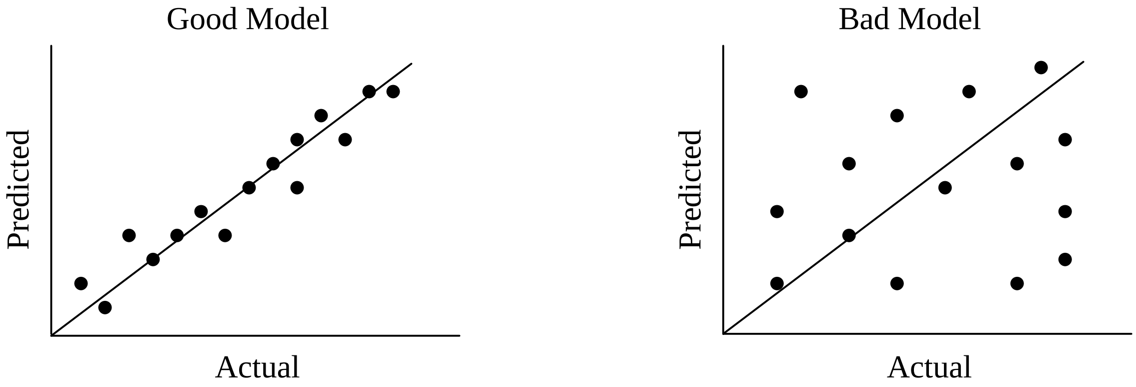


Figure 3.2: Comparison of good and bad model fit

The requirement of accurate fit is illustrated in Figure 3.2 with both good and bad models for comparison. On the plot of predicted response values vs. actual experimental values, a good model will fit points close to the 45^{o} line between the two axes. The second requirement for a good regression model is that all effects in the model contribute significantly to the predicted value. After a model is analyzed, any insignificant effects are removed, and any new potentially significant effects are added to the model. The final requirement for a good model is that all assumptions made about the data must be correct. For example, if the errors do not appear to have constant variance, then a *transformation* of the actual observed response values may be necessary in order to stabilize the variance. Transformations and assumptions made on the input data will be discussed in more detail in Sections 3.8 and 3.9.

At each analysis iteration, the revised model is evaluated, and the process continues until a good model is created. Only after rigorously validating the model is it possible to do response prediction. This chapter will present the statistical analysis necessary for validating model assumptions and testing for significance of model effects. First, the details of linear regression pertaining to the construction of empirical models used in experimental design will be presented.

3.2 Linear Regression Models

Complex systems can contain responses that require elaborate mathematical models for exact prediction. Low-order polynomial models are commonly used to approximate responses [8]. *Multiple linear regression models* predict values of a response or *dependent* *variable y* that depends on *k predictor* or *independent variables* x_1, x_2, … x_k. The independent variables represent factor or global variable values and their interactions in the system being modeled. In general, the multiple linear regression model may be written as

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k + \varepsilon \tag{3.1}$$

The error term ε measures the difference between the predicted and observed response values. The regression is *multiple* when *k* is greater than one, that is, when there is more than one independent variable in the model. The term *linear* is used because *y* is a function of the linear coefficients β_i. Consider the second-order response surface model with two factors:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_2^2 + \beta_5 x_1 x_2 + \varepsilon$$

This linear regression model is in the same form as Equation (3.1) with the substitutions $x_3 = x_1^2$, $x_4 = x_2^2$, and $x_5 = x_1 x_2$. The linearity of a regression model is determined by the linearity of the β_i coefficients, regardless of how nonlinear the response surface may be [8].

Table 3.1 shows the experimental data for calculating a regression model with *k* factors and *n* > *k* experiments [8]. Each row in the table contains the data for one experiment. The x_{ij} variables represent the effects and form the **X** matrix, which is similar to the **X** matrix used in effect estimation. The y_i variables represent the observed response values. The model for the data in Table 3.1 can be represented as

$$\begin{aligned} y &= \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_k x_{ik} + \varepsilon_i \\ &= \beta_0 + \sum_{j=1}^{k} \beta_j x_{ij} + \varepsilon_i \qquad i = 1, 2, \ldots, n \end{aligned} \tag{3.2}$$

Table 3.1: Multiple linear regression data

X					**y**
I	$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	**…**	$\boldsymbol{x_k}$	
1	x_{11}	x_{12}	…	x_{1k}	y_1
1	x_{21}	x_{22}	…	x_{2k}	y_2
1	$\vdots$	$\vdots$		$\vdots$	$\vdots$
1	x_{n1}	x_{n2}	…	x_{nk}	y_n

Rewriting Equation (3.2) in matrix form yields

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

The matrix **X** and the response vector **y** are taken from Table 3.1, and $\boldsymbol{\beta}$ and $\boldsymbol{\varepsilon}$ are written as

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{bmatrix}, \qquad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

It is assumed that the ε_i are independent and identically distributed normal random variables with equal variances. The mean of ε_i is assumed to be zero:

$$\sum_{i=1}^{n} \varepsilon_i = 0$$

The goal of linear regression is to compute the vector $\boldsymbol{\beta}$ that makes the regression model most accurately predict the true response behavior. The method of least squares finds the estimates of $\boldsymbol{\beta}$ such that the sum of the squares of the errors $\boldsymbol{\varepsilon}$ is minimized. The least squares function *L* is

$$\begin{aligned} L = \sum_{i=1}^{n} \varepsilon_i^2 &= \boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{\varepsilon} = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^{\mathrm{T}}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \\ &= \mathbf{y}^{\mathrm{T}}\mathbf{y} - 2\boldsymbol{\beta}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{y} + \boldsymbol{\beta}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\boldsymbol{\beta} \end{aligned}$$

using the fact that $(\mathbf{y}^{\mathrm{T}}\mathbf{X}\boldsymbol{\beta})^{\mathrm{T}} = \boldsymbol{\beta}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{y}$ is a scalar [9]. To minimize *L*, the partial derivative of *L* with respect to $\boldsymbol{\beta}$ is evaluated at $\hat{\boldsymbol{\beta}}$ and set equal to zero. Solving for $\hat{\boldsymbol{\beta}}$ yields

$$\begin{aligned} \left.\frac{\partial L}{\partial \boldsymbol{\beta}}\right|_{\hat{\boldsymbol{\beta}}} &= -2\mathbf{X}^{\mathrm{T}}\mathbf{y} + 2\mathbf{X}^{\mathrm{T}}\mathbf{X}\hat{\boldsymbol{\beta}} = 0 \\ \mathbf{X}^{\mathrm{T}}\mathbf{X}\hat{\boldsymbol{\beta}} &= \mathbf{X}^{\mathrm{T}}\mathbf{y} \\ \hat{\boldsymbol{\beta}} &= \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y} \end{aligned} \tag{3.3}$$

$\hat{\boldsymbol{\beta}}$ is the estimate of $\boldsymbol{\beta}$ that minimizes *L*. Note that Equation (3.3) is identical to the effect estimate vector in Equation (2.1) from the previous chapter. Effect estimation uses the same method of least squares estimation that linear regression uses. The fitted regression model is

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}} \tag{3.4}$$

For k effects and $n > k$ experiments, an empirical model can be used to estimate the response y using Equation (3.4).

The *residual* e_i for experiment i is defined as

$$e_i = y_i - \hat{y}_i$$

The vector of residuals is

$$\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}}$$

The residual is the difference between the observed experimental value and the predicted value from the regression model [9]. A quantity commonly used to measure the error of a regression model is the *residual sum of squares* or *error sum of squares* (*SSE*):

$$\begin{aligned} SSE &= \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i-1}^{n} e_i^2 = \mathbf{e}^{\mathrm{T}}\mathbf{e} \\ &= (\mathbf{y} - \hat{\mathbf{y}})^{\mathrm{T}} (\mathbf{y} - \hat{\mathbf{y}}) \\ &= (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})^{\mathrm{T}} (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) \\ &= \mathbf{y}^{\mathrm{T}}\mathbf{y} - \mathbf{y}^{\mathrm{T}}\mathbf{X}\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{y} + \hat{\boldsymbol{\beta}}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\hat{\boldsymbol{\beta}} \\ &= \mathbf{y}^{\mathrm{T}}\mathbf{y} - 2\hat{\boldsymbol{\beta}}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{y} + \hat{\boldsymbol{\beta}}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{y} \\ &= \mathbf{y}^{\mathrm{T}}\mathbf{y} - \hat{\boldsymbol{\beta}}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{y} \end{aligned} \tag{3.5}$$

The method of least squares calculates the regression model parameters that minimize *SSE*.

To illustrate the construction of a regression model, a model of workstation performance will be calculated using the data from the 2^3 factorial design presented in Section 2.4.5. Table 3.2 repeats the previously calculated effect estimates for the 2^3 design. Using the effect estimates, the experimenter must choose any potentially significant effects to include in

Table 3.2: Effect estimates for 2^3 workstation example

Effect	$\hat{\boldsymbol{\beta}}$	**Effect Estimate**
I	5.25	5.25
A	1.5	3.0
B	0.75	1.5
C	0.25	0.5
AB	0.5	1.0
AC	0.0	0.0
BC	-0.25	-0.5
ABC	0.0	0.0

the regression model. Two interactions, AC and ABC, have a zero effect estimate; therefore, they will not be included in the model. It will be assumed that a priori knowledge of the system reveals that the number of processors and hard disk size do not interact; hence, interaction BC will not be selected for the model. A priori knowledge of the system should never be underestimated. The system being modeled should rarely be treated as a black box. There is usually some knowledge of how factors interact in a system, and when properly used, that knowledge can significantly increase the efficiency of the analysis. The remaining three main effects and interaction AB will be included in the model, which will be of the form

$$\hat{y} = \beta_0 + \beta_1 x_A + \beta_2 x_B + \beta_3 x_C + \beta_4 x_A x_B$$

In general, to solve for $\hat{\boldsymbol{\beta}}$, the system $\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}$ is solved using the following equation:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{3.6}$$

The construction of the **X** matrix consists of selecting the *k*+1 columns from the **X** matrix used in the effect estimation that correspond to the effects included in the model [13]. For the model above, which contains four effects, **X** will consist of a first column of ones, and four columns corresponding to effects A, B, C, and AB taken from the **X** matrix in Table 2.11 in Section 2.4.5:

$$\mathbf{X} = \begin{array}{c} \begin{array}{ccccc} & x_A & x_B & x_C & x_A x_B \end{array} \\ \left[\begin{array}{rrrrr} 1 & -1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 \\ 1 & 1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 1 & 1 \end{array}\right] \end{array}$$

The above matrix is orthogonal because it is a submatrix of an orthogonal matrix. A nice property of solving $\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}$ with the above **X** matrix is that the solution values will be identical to those obtained during the effects estimation. Therefore, there is no need to solve any additional linear system of equations after the original effect estimation is calculated. This is true for all orthogonal experimental designs, including all two-level factorial designs. The model parameters are computed by selecting the corresponding values from the effect estimate vector in Table 3.2. The regression model parameters are

$$\hat{\boldsymbol{\beta}} = \begin{bmatrix} 5.25 \\ 1.5 \\ 0.75 \\ 0.25 \\ 0.5 \end{bmatrix}$$

and the solved model is

$$\hat{y} = 5.25 + 1.5x_{\mathrm{A}} + 0.75x_{\mathrm{B}} + 0.25x_{\mathrm{C}} + 0.5x_{\mathrm{A}}x_{\mathrm{B}} \tag{3.7}$$

Table 3.3 lists the observed and predicted values of the regression model for each experiment [9]. The error and squared error for each observation are also presented. The error sum of squares (*SSE*) is shown to be equal to 0.5. Any other choice of regression parameters $\hat{\boldsymbol{\beta}}$ would result in a higher *SSE*.

Before accepting the above model as an accurate predictor of the response, it is necessary to perform additional statistical tests on the model. The additional tests measure the significance of each term in the model and diagnose the assumptions made on the errors. The following sections present the methods for performing these statistical tests.

Table 3.3: Error calculation for 2^3 workstation design

Experiment	y_i	$\hat{y}_i$	e_i	e_i^2
1	3	3.25	-0.25	0.0625
2	5	5.25	-0.25	0.0625
3	4	3.75	0.25	0.0625
4	8	7.75	0.25	0.0625
5	4	3.75	0.25	0.0625
6	6	5.75	0.25	0.0625
7	4	4.25	-0.25	0.0625
8	8	8.25	-0.25	0.0625
Σ	42	42	0	0.5

3.3 Allocation of Variation

It is desirable for the regression model to predict the response with minimum variability. Consider an alternative to the regression model that uses only the mean response $\bar{y}$ as the predicted value. The mean-only model would have a larger error than the regression model error. Specifically, the *SSE* without regression would be

$$\sum_{i=1}^{n}(y_i-\bar{y})^2$$

The above quantity is a measure of the variation of *y* and is referred to as the *total sum of squares* or *SST* [9]. The *SST* can be calculated as follows:

$$SST=\sum_{i=1}^{n}(y_i-\bar{y})^2=\sum_{i=1}^{n}y_i^2-n\bar{y}^2=\mathbf{y}^{\mathrm{T}}\mathbf{y}-n\bar{y}^2 \tag{3.8}$$

The sum of the squares explained by the regression (*SSR*) is equal to the difference between *SST* and *SSE*:

$$SSR=SST-SSE$$

Rewriting the above equation yields

$$SST=SSR+SSE$$

Therefore, the total variation *SST* is composed of the variation explained by the regression (*SSR*) and the variation not explained by the regression (*SSE*). *SSR* can be written as

$$\begin{aligned}SSR&=SST-SSE\\&=\mathbf{y}^{\mathrm{T}}\mathbf{y}-n\bar{y}^2-\mathbf{y}^{\mathrm{T}}\mathbf{y}+\hat{\boldsymbol{\beta}}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{y}\\&=\hat{\boldsymbol{\beta}}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{y}-n\bar{y}^2\end{aligned}$$

Each sum of squares presented above has a corresponding *degree of freedom*, which is equal to the number of independent terms in the sum [9]. Equation (3.8) shows that *SST* sums *n* terms $(y_i-\bar{y})^2$. Although there are *n* summands, only *n*–1 are independent. One observed value y_j must be chosen in order to satisfy

$$\sum_{i=1}^{n}(y_i-\bar{y})=0$$

All remaining *n*–1 y_i terms are free to vary, and therefore the degrees of freedom (DF) of *SST* is

$$\mathrm{DF}(SST)=n-1$$

The calculation of *SSE* in Equation (3.5) requires that *k*+1 $\hat{\beta}_i$ terms first be calculated; each of those terms uses one degree of freedom from the *n* total summands. Therefore,

$$\mathrm{DF}(SSE) = n - (k+1) = n - k - 1$$

Degrees of freedom are additive. Therefore, the total degrees of freedom available is partitioned just as the sum of squares is partitioned [9]:

$$\begin{aligned} SST = SSR + SSE \Rightarrow \mathrm{DF}(SST) &= \mathrm{DF}(SSR) + \mathrm{DF}(SSE) \\ \Rightarrow \mathrm{DF}(SSR) &= \mathrm{DF}(SST) - \mathrm{DF}(SSE) \\ &= n - 1 - (n - k - 1) \\ &= k \end{aligned}$$

Dividing a sum of squares by its degrees of freedom yields the *mean square* for the corresponding quantity. The mean square is an estimate of variance. Estimates for the regression model variance *(MSR)* and the error variance *(MSE)* are

$$MSR = \frac{SSR}{k} \qquad\qquad MSE = \frac{SSE}{n - k - 1}$$

The mean squares are used in the calculation of the *analysis of variance* (ANOVA) [9].

3.4 Analysis of Variance

The usefulness of a multiple linear regression model can be measured by testing certain hypotheses about the regression model parameters. One procedure that tests for the significance of regression is the *analysis of variance* or *ANOVA*. ANOVA requires that the errors $\boldsymbol{\varepsilon}$ in the model $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ be independent and normally distributed with mean zero and variance σ^2, commonly abbreviated $\boldsymbol{\varepsilon} \sim \mathrm{NID}(0, \sigma^2)$. It is also assumed that all factors are measured without error and that, therefore, the observed responses *y* are independent and normally distributed with mean $\mathbf{X}\boldsymbol{\beta}$ and variance σ^2 [8].

The general multiple linear regression model is of the form

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k + \varepsilon$$

The test for significance of regression determines if there is a linear relationship between the response *y* and some subset of the independent variables x_j. The hypotheses are

$$\begin{aligned} &H_0 : \beta_1 = \beta_2 = \cdots = \beta_k = 0 \\ &H_1 : \beta_j \neq 0 \quad \text{for at least one } j \end{aligned}$$

The null hypothesis H_0 is rejected if at least one x_j contributes significantly to the model [8]. Testing the above hypotheses is equivalent to performing an *F-test* on the ratio of mean squares MSR/MSE. The details of the F-test are presented in [15] and [16]. Essentially, the F-test relies on the fact that *MSR* and *MSE* have a chi-square distribution. Therefore, the ratio MSR/MSE, called the *F-value,* has an $\mathrm{F}_{[k,\ n-k-1]}$ distribution with *k* numerator degrees of freedom and $n-k-1$ denominator degrees of freedom. The model's significance increases as the F-value increases. The *P-value* is equal to the proportion of the area under the curve of the F-distribution that lies beyond the observed F-value. The P-value is the probability that a model could obtain its F-value strictly by chance. If the P-value for the model is less than the significance threshold α, then the null hypothesis H_0 is rejected, and the regression is assumed to be significant [15]. The significance test information is usually summarized in an ANOVA table such as Table 3.4.

The analysis of variance will be calculated for the workstation model in Equation 3.7. The sums of squares and degrees of freedom are calculated using the formulae from the previous sections and are presented in Table 3.5. The mean squares and F-value are

$$MSR = \frac{25}{4} = 6.25 \qquad , \qquad MSE = \frac{0.5}{3} = 0.1667 \qquad , \qquad \mathrm{F{-}value} = \frac{6.25}{0.1667} = 37.5$$

The P-value of 0.0068 is calculated using the built-in F-distribution algorithms in Möbius and is shown in Table 3.5. The small P-value reveals that there is only a 0.68% possibility that a model F-value of 37.5 could occur strictly by chance. The default significance threshold in Möbius is 0.05. The P-value is far below the threshold, and therefore the model is labeled as significant in the last column of the ANOVA table.

Table 3.4: ANOVA table

Source	Sum of Squares	DF	Mean Square	F-Value
Model	*SSR*	k	*MSR*	*MSR/MSE*
Residual	*SSE*	$n-k-1$	*MSE*	
Total	*SST*	$n-k$		

Table 3.5: ANOVA table for workstation example

Source	Sum of Squares	DF	Mean Square	F-Value	P-Value	Significant?
Model	25	4	6.25	37.5	0.0068	Significant
Residual	0.5	3	0.1667			
Total	25.5	7				

3.5 Testing Individual Regression Terms

The previous section presented the ANOVA procedure for determining the significance of an entire regression model. It is often useful to know the significance of the individual effects in the model. The general regression model of the form

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k + \varepsilon$$

has k independent variables $x_1, x_2, \ldots, x_k$, each of which corresponds to a main effect or to an interaction of several factors. The method for calculating each effect's significance involves partitioning the regression sum of squares into additive components corresponding to each effect. The method for calculating each effect's partitioned sum of squares depends on the type of experimental design on which the regression model is based. For a detailed presentation of the different types of sums of squares, the reader may consult [15] or [17].

For all orthogonal designs, including all factorial designs in Möbius, the sum of squares for each effect is calculated using the *partial* or *Type III* sum of squares method [17]. A partial sum of squares (*SSP*) is the difference between the regression sum of squares (*SSR*) of the *full* regression model (including $\beta_0, \beta_1, \ldots, \beta_k$) and the *SSR* of the model with one term $\beta_j x_j$ removed. If the regression sum of squares of the model with the j^{th} term ($\beta_j x_j$) removed is denoted $SSR_{(j)}$, then the partial sum of squares of the j^{th} effect *SSP(j)* is

$$SSP(j) = SSR - SSR_{(j)}$$

The *SSP* can be calculated for each effect corresponding to $x_1, x_2, \ldots, x_k$ in the full model. *SSR* will always be greater than or equal to $SSR_{(j)}$ because the removal of a nonzero term from the full model always results in a regression model with a decreased *SSR* and increased *SSE* [17]. For orthogonal designs in Möbius, the sum of all *k SSP(i)* is equal to the *SSR* of the full model:

$$\sum_{i=1}^{k} SSP(i) = SSR$$

For nonorthogonal designs, including all response surface designs in Möbius, the sum of squares for each model effect is calculated using the *sequential* or *Type I* sum of squares method. The method of partial sum of squares is not used in nonorthogonal designs because the *SSPs* for each effect may not sum up to *SSR* for the full model. The method for calculating the sequential sum of squares (*SSS*) is similar to that for the partial sum of squares, except that effects are added to the model sequentially, starting with the mean-only model $y = \beta_0$ and finishing with the full regression model. For example, the *SSS* for the first effect added to the model, denoted by *SSS*(1), is equal to the difference between the regression sum of squares of the model $y = \beta_0 + \beta_1 x_1$ and the regression sum of squares of the mean-only model, denoted *SSR*(0):

$$SSS(1) = SSR(1) - SSR(0)$$

In general, the *SSS* for the i^{th} effect is

$$SSS(i) = SSR(i) - SSR(i-1) \qquad i = 1, 2, \ldots, k$$

The sum of the *SSS* for each effect in the regression model is equal to *SSR* for the full model:

$$\sum_{i=1}^{k} SSS(i) = SSR$$

Regardless of which method is used to calculate an effect's sum of squares, the notation used in the remainder of this thesis to refer to the i^{th} effect's sum of squares will be *SS(i)*. The *k* degrees of freedom of *SSR* are distributed equally among the *k SS(i)*. Therefore, each of the *k* effects in the full model possesses one degree of freedom [18]. The mean square of any individual effect corresponding to variable x_i is

$$MS(x_i) = \frac{SS(i)}{\mathrm{DF}(x_i)} = \frac{SS(i)}{1} = SS(i)$$

That is, the mean square of any effect is equal to the sum of squares for that effect.

After the mean squares for all effects are calculated, F-values can be computed for each effect using $\mathrm{F}_{[1,\, n-k-1]}$. P-values can also be computed and the significance of each effect tested, completing the analysis of variance on the model. The sums of squares for each effect in the workstation model are presented in the ANOVA in Table 3.6. The ANOVA

Table 3.6: ANOVA for workstation model showing each effect's significance

Source	Sum of Squares	DF	Mean Square	F-Value	P-Value	Significant?
Model	25	4	6.25	37.5	0.0068	Significant
A	18	1	18	108	0.0019	Significant
B	4.5	1	4.5	27	0.0138	Significant
C	0.5	1	0.5	3	0.1817	Not Significant
AB	2	1	2	12	0.0405	Significant
Residual	0.5	3	0.1667			
Total	25.5	7				

table lists separate rows displaying the significance of each effect in the model. Note that the sum of each individual effect's sum of squares equals the regression model sum of squares $(18+4.5+0.5+2=25)$. Table 3.6 shows that factor C, hard disk size, does not contribute significantly to the response. Therefore, C will be removed from the model. The revised model without C will be

$$\hat{y} = 5.25 + 1.5x_{\mathrm{A}} + 0.75x_{\mathrm{B}} + 0.5x_{\mathrm{A}}x_{\mathrm{B}}$$

The ANOVA for the revised model is presented in Table 3.7. The P-value in the revised model is lower (more significant) than in the previous model, as expected.

In general, all model effects that are not significant should be removed from the model. The exception occurs when the model retains any insignificant effect that is required

Table 3.7: ANOVA for the revised workstation model

Source	Sum of Squares	DF	Mean Square	F-Value	P-Value	Significant?
Model	24.5	3	8.1667	32.6667	0.0028	Significant
A	18	1	18	72	0.0011	Significant
B	4.5	1	4.5	18	0.0132	Significant
AB	2	1	2	8	0.0474	Significant
Residual	1	4	0.25			
Total	25.5	7				

in order to support *hierarchy* for another significant model effect. For example, if interaction AB is included in a model, then both *parent* main effects A and B must be included in the model before an ANOVA can be performed. Well-formulated models support model hierarchy. The interested reader should consult [19] or [20] for a detailed presentation of model hierarchy.

3.6 Regression Model Diagnostics

There are several diagnostic procedures for testing the adequacy of a regression model. This section will present several procedures to test whether the regression model provides an adequate approximation of the system being modeled. The reader may consult [18] for a more complete presentation.

The first useful quantity for diagnosing a regression model is the mean response μ. The mean response is calculated during effect estimation. It is equal to the first element in the effect estimate vector $\hat{\boldsymbol{\beta}}$:

$$\mu = \hat{\beta}_0$$

The mean response is the constant term in the regression model and is sometimes referred to as the *intercept*. An estimate of the standard deviation, σ, of the error in the design is equal to the square root of the residual mean square:

$$\sigma = \sqrt{MSE}$$

σ should be kept as small as possible [8]. The *coefficient of variation, CV*, measures the unexplained or residual variability in the data as a percentage of the mean [8]:

$$CV = 100\left(\frac{\sigma}{\mu}\right)$$

The *coefficient of multiple determination* R^2 is defined as

$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$$

R^2 measures the amount of response variation that is explained by the regression model [10]. A perfect regression model would explain all variation and *SSE* would be zero, resulting in R^2 being equal to one. R^2 always increases as nonzero terms are added to the model, even if the terms do not contribute significantly to the response. Therefore, R^2 is not the best measurement of a “good” regression model. Another diagnostic, called the *adjusted* R^2, measures the amount of variation explained by the regression model, adjusted for the number of effects in the model:

$$R_{adj}^2 = 1 - \frac{\dfrac{SSE}{n-k-1}}{\dfrac{SST}{n-1}} = 1 - \frac{n-1}{n-k-1}\left(1 - R^2\right)$$

R_{adj}^2 increases as significant effects are added to the model, but can decrease as unnecessary ones are added. If R^2 and R_{adj}^2 differ dramatically, there is a good chance that insignificant effects have been added to the model [8].

A measure of how the regression model fits each observed response in the design is obtained from the *prediction residual error sum of squares*, or *PRESS*. The *PRESS* is calculated by computing the prediction error $e_{(i)}$ for each experiment. For the i^{th} experiment, the regression model is fit to the observed response values from the other n–1 experiments, excluding the i^{th} experiment’s response value y_i. That results in n regression models, each with a different $\hat{\boldsymbol{\beta}}$ vector [8]. Each model is then used to predict the value of its missing response y_i. The predicted value for the i^{th} experiment is $\hat{y}_{(i)}$, and the prediction error is

$$e_{(i)} = y_i - \hat{y}_{(i)} \tag{3.9}$$

The *PRESS* is computed as

$$PRESS = \sum_{i=1}^{n} e_{(i)}^2$$

An easier method for computing *PRESS* will be presented in the next section. *PRESS* is most useful when used to compute an approximate R^2 for prediction:

$$R_{prediction}^2 = 1 - \frac{PRESS}{SST}$$

The prediction R^2 measures the amount of variability that the regression model is expected to explain when predicting new response values. If the prediction R^2 is not within 0.2 of the adjusted R^2, there may be a problem with the data or the model [8]. Methods for dealing with such problems will be discussed in Section 3.9.

One final statistic useful for determining model adequacy measures the signal-to-noise ratio. The *adequate precision* compares the range of predicted values $\hat{y}$ to the average prediction error:

$$Adequate\ Precision = \frac{\hat{y}_{\max} - \hat{y}_{\min}}{\frac{(k+1)MSE}{n}}$$

where *k* is the number of effects in the regression model. A value greater than four indicates that the model's signal-to-noise ratio is adequate for exploring the design space [8].

Table 3.8 lists some diagnostics for the workstation model from the previous section. Fortunately, R^2 is close to the adjusted R^2. The adjusted R^2 reveals that the model explains 93.14% of the variation in the observed data from the original least squares fit. The prediction R^2 indicates that the model is expected to explain about 84.31% of the variation when predicting new observations. The adequate precision is much larger than four, indicating that the model has a strong signal within the design space. The overall predictive capability of the model is satisfactory, according to our criteria.

Table 3.8: Model adequacy statistics for workstation model

R^2	Adjusted R^2	Prediction R^2	Adequate Precision
0.9608	0.9314	0.8431	12.7279

3.7 Influential Data and Outliers

Occasionally, observed response values from certain experiments distinguish themselves from the majority of observed design points. The distinguished data points may exert a disproportionately large influence on the regression model's prediction or may stray far from the values predicted by the majority of observed values. Well-designed experiments facilitate methods for detecting these potentially problematic data points [8]. This section describes three such methods.

A useful quantity for diagnosing regression models is the hat matrix **H**. Using Equation (2.1) from Section 2.4.5, it is possible to derive **H** as

$$\begin{aligned}\hat{\mathbf{y}} &= \mathbf{X}\hat{\boldsymbol{\beta}} \\ &= \mathbf{X}(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y} \\ &= \mathbf{H}\mathbf{y}\end{aligned}$$

The hat matrix $\mathbf{H} = \mathbf{X}(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}$ maps the vector of observed values **y** into the vector of predicted values $\hat{\mathbf{y}}$. **H** is purely a function of the design's **X** matrix, and the diagonal elements h_{ii} measure the location of the i^{th} design point in *x*-space. All diagonal elements are greater than or equal to zero. More details on the interpretation of **H** can be found in [18] or [21]. **H** is useful for identifying influential experimental observations. The elements h_{ij} of **H** can be interpreted as the amount of leverage that observed response y_j exerts on predicted response $\hat{y}_i$. The *leverage* value for experiment *i* is equal to the diagonal element h_{ii} of **H** [10]. The relationship

$$\sum_{i-1}^{n} h_{ii} = rank(\mathbf{H}) = rank(\mathbf{X}) = k+1$$

states that the sum of all leverage points is equal to the number of parameters in the regression model (*k* effect estimates and 1 intercept). The *k*+1 units of leverage are distributed across the *n* leverage values, and the average leverage value is $(k+1)/n$ [18]. In Möbius, the number of experiments *n* is always greater than the number of effects estimated *k*; therefore, the maximum value of any leverage point is one. A rule of thumb is to declare any leverage value higher than twice the average value as exerting a large influence on the predicted response [8].

The most common tool for identifying high leverage points is a scatter plot of leverage values for each experiment. Figure 3.3 shows a scatter plot for the leverage values in the workstation model. The leverage plot shows that all leverage values are equal. This is not a coincidence. All orthogonal designs, including all factorial designs in Möbius, will have leverage values equal to $(k+1)/n$ for each experiment. Leverage values depend on the distance between the design points and the design center. All design points in factorial designs lie on vertices of a hypercube centered at the design center, and hence are equidistant from the design center. Therefore, all leverage values are designed to be equal, ensuring that no experimental value exerts a disproportionate influence on the predicted response [18]. Leverage plots are most useful in response surface designs.

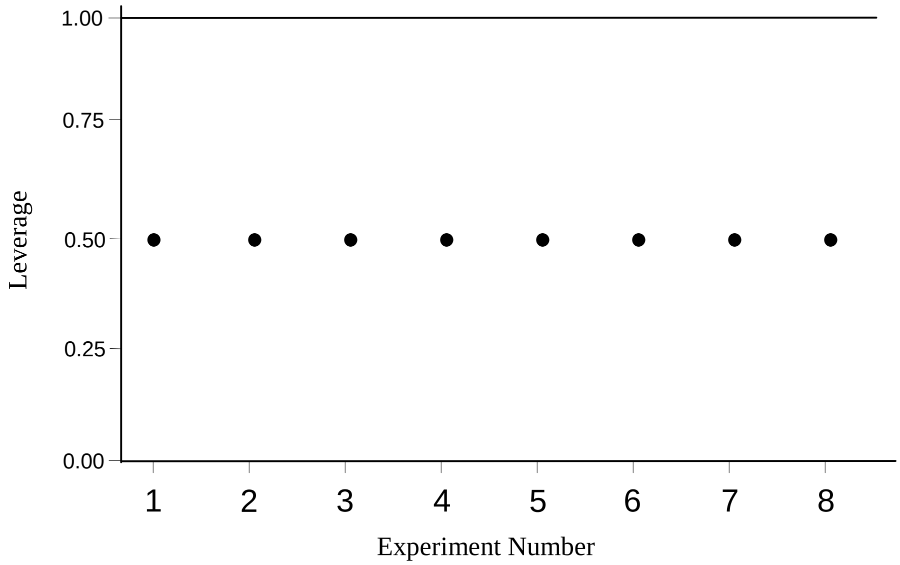


Figure 3.3: Leverage values for workstation model

The hat matrix **H** is also used to calculate the PRESS for the regression model. The previous section showed that in order to define the *n* prediction errors $e_{(i)}$ in Equation (3.9), *n* regression models must be fit. It turns out that no additional regressions need to be fit, due to the relationship

$$e_{(i)} = \frac{e_i}{1 - h_{ii}}$$

where e_i is the i^{th} residual $y_i - \hat{y}_i$ [8]. Therefore, the *PRESS* is easily calculated as

$$PRESS = \sum_{i=1}^{n} e_{(i)}^2 = \sum_{i=1}^{n} \left(\frac{e_i}{1 - h_{ii}} \right)^2$$

Another diagnostic for influential experimental data points is the *Cook's distance* D_i [16]. The Cook's distance is the squared distance between the least squares estimate based on all *n* experiments, $\mathbf{X}\hat{\boldsymbol{\beta}}$, and the estimate obtained by removing the i^{th} experiment, say $\mathbf{X}\hat{\boldsymbol{\beta}}_{(i)}$:

$$D_i = \frac{(\hat{\boldsymbol{\beta}}_{(i)} - \hat{\boldsymbol{\beta}})^{\mathrm{T}} \mathbf{X}^{\mathrm{T}} \mathbf{X} (\hat{\boldsymbol{\beta}}_{(i)} - \hat{\boldsymbol{\beta}})}{(k+1)\ MSE} \qquad i = 1, 2, \ldots, n$$

It is not necessary to compute *n* different least squares estimates to obtain Cook's distance. The hat matrix **H** simplifies the calculation:

$$D_i = \frac{e_i^2 \, h_{ii}}{(k+1)\, MSE\, (1-h_{ii})^2},$$

where e_i is the i^{th} residual. Any experimental point that has a Cook's distance greater than one is considered to exert a disproportionately large influence on the regression coefficients, and hence on the predicted response. The scatter plot of Cook's distances for all experiments in the workstation example is presented in Figure 3.4. None of the D_i exceeds one, so no overly influential points are detected. An equivalent calculation for Cook's distance is

$$D_i = \frac{r_i^2}{(k+1)} \cdot \frac{h_{ii}}{(1-h_{ii})},$$

where r_i is the *studentized residual,* which will be presented in the next section [13].

An observed response that deviates much further from the predicted response than most observed values do is called an *outlier*. All outliers must be investigated to determine their impact on the regression model. If the outlier is a "bad" value, then it should be eliminated, or the experiment should be repeated. One possible cause for a "bad" value could be a simulation that does not converge due to premature termination. Outliers need not be bad values. Outliers may simply reveal an inadequacy in the chosen model. For example, if a

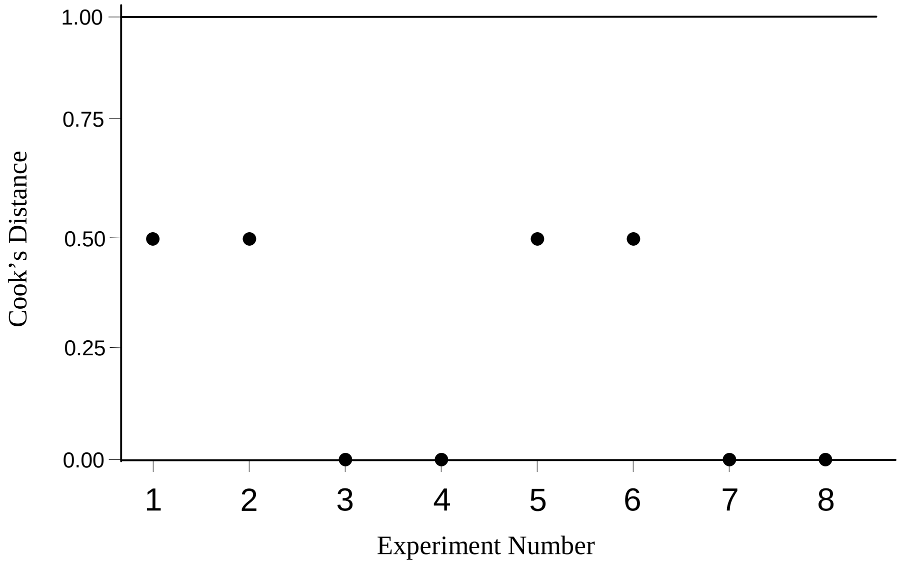


Figure 3.4: Cook's distance for workstation example

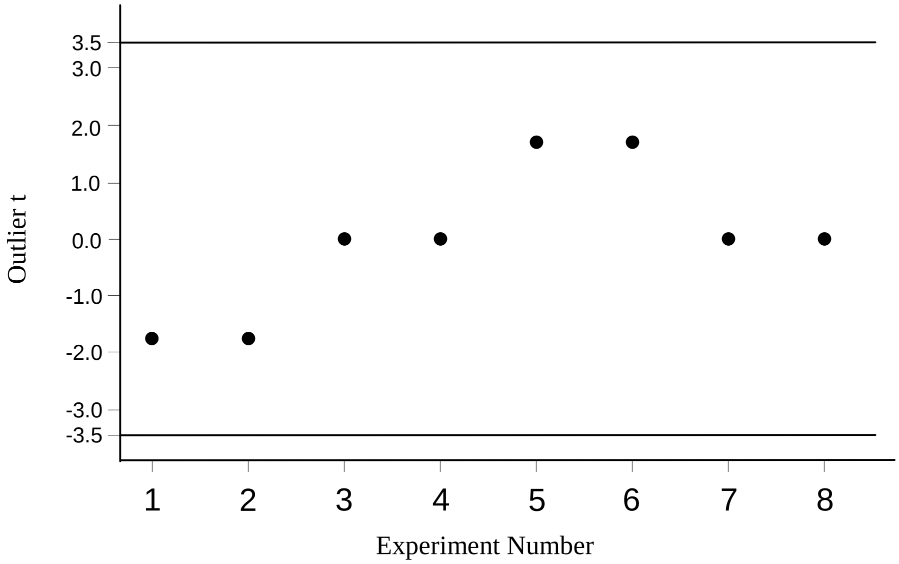


Figure 3.5: Outlier t values for the workstation example

response is a cubic function of the model factors, a second-order response surface may fail to map onto some of the outlying observed values [7].

The diagnostic used to detect outliers is the *Outlier t* statistic, also known as the *externally studentized residual* or *R-student* [21]. The outlier t measures the number of standard deviations between the i^{th} observed response value and the predicted response value using a model with the i^{th} experiment deleted. To calculate the outlier t for experiment *i*, we first compute an estimate of the standard deviation of the model with the i^{th} experiment removed, $\hat{\sigma}_{(i)}$:

$$\hat{\sigma}_{(i)} = \sqrt{\frac{(n-k-1)MSE - \frac{e_i^2}{1-h_{ii}}}{n-k-2}}$$

The outlier t for each experiment is calculated as [8]

$$t_i = \frac{e_i}{\hat{\sigma}_{(i)}\sqrt{1-h_{ii}}} \qquad i = 1, 2, \ldots, n$$

A scatter plot of t_i versus experiment number is used to reveal any outliers. Any t_i with magnitude greater than 3.5 is considered an outlier. The workstation example outlier t plot is shown in Figure 3.5. None of the data points exceeds 3.5 in magnitude.

3.8 Verifying the ANOVA Assumptions

The analysis of variance assumes that errors are independent and normally distributed with zero mean and constant variance. An experimenter must check the validity of those assumptions before relying on the ANOVA results. This section presents methods for examining the residuals of the regression model in order to investigate whether the ANOVA assumptions have been violated. In general, departures from the assumptions usually mean that the residuals contain structure that is not accounted for in the model. Identifying that structure and, if possible, adding a term representing it to the original regression model lead to a better model [15].

The assumption that the errors have zero mean is always valid in Möbius. The least squares fit guarantees that the sum of the residuals (errors) is zero; therefore, the residual mean will always be zero.

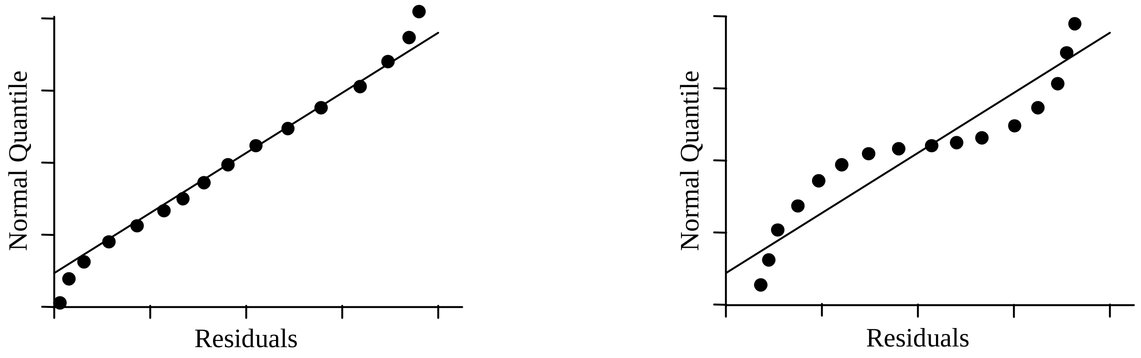
It is possible to check the normality assumption by plotting the residuals on a *normal quantile plot*. In general, quantile plots are used to verify whether a set of data conforms to a hypothesized distribution. The vertical axis of a normal quantile plot consists of the quantiles of a standard normal distribution (with mean zero and variance one) corresponding to the ordered residuals e_i, which are plotted on the horizontal axis. The quantiles are computed from the inverse cumulative distribution function of the normal distribution. Let the observed cumulative frequency f_i of the i^{th} residual e_i be

$$f_i = \frac{i-0.5}{n} \qquad\qquad i = 1, 2, \ldots, n$$

The i^{th} quantile q_i corresponding to the i^{th} residual e_i is computed using a common approximation of the inverse CDF of $N(0,1)$:

$$q_i = \left[f_i^{0.14} - (1-f_i)^{0.14}\right] \qquad\qquad i = 1, 2, \ldots, n$$

If the normal distribution adequately describes the residuals, the plotted points will fall approximately along a straight line [22]. The determination of whether the data represent a straight line is subjective. Figure 3.6(a) shows an example normal quantile plot of a set of normally distributed residuals. The quantiles fall close to the straight line throughout the central portion of the plot. The extrema deviate slightly from linearity. In general, slight deviations at the extrema of the plot do not constitute grounds for dismissing the normality



(a) Normal distribution

(b) Non-normal distribution

Figure 3.6: Example normal quantile plots of residuals

of the data. The F-test in the analysis of variance is robust enough to tolerate small deviations from normality in the residuals [15]. Figure 3.6(b) shows significant deviation from linearity throughout the entire range of data. Such a strong violation of the normality assumption indicates that the regression model is not appropriate.

In practice, statisticians prefer to work with *scaled residuals*, which often convey more information than the ordinary least squares residuals do [8]. The variance of the i^{th} residual is

$$V(e_i) = \sigma^2 (1 - h_{ii}) \qquad\qquad i = 1, 2, \ldots, n$$

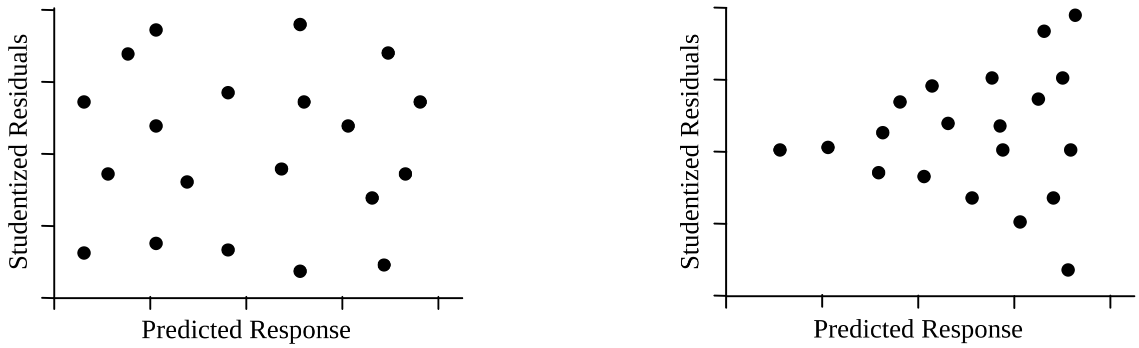
where h_{ii} is the leverage value for the i^{th} experiment $(0 \le h_{ii} \le 1)$. High leverage points can significantly reduce the variance of the corresponding residual. In general, violations of assumptions are more likely to occur at remote locations in the design space rather than near the design center. Leverage values tend to increase with distance from the design center. Therefore, remote residuals tend to have lower variances than residuals near the center [8]. One type of scaled residual that takes this inequality of variance into account is the *studentized residual*. Studentized residuals r_i are residuals converted to a scale approximately representing the standard deviation of an individual residual from the center of the residual distribution. This rescaling reflects the number of standard deviations between the actual and predicted response values:

$$r_i = \frac{e_i}{\sqrt{MSE(1 - h_{ii})}}$$

The studentized residuals have a constant variance of one. In cases with high leverage, only the studentized form of residuals will produce valid diagnostic graphs [10].

The independence assumption on the errors is difficult to prove. It is possible to verify that the errors are independent of specific quantities; however, that does not eliminate the possibility that the errors may depend on something else. One quantity that the errors are definitely independent of in Möbius is experiment number. Deterministic computer solution of the response variables is independent of the time order of the experiments. The results are exactly repeatable every time the experiment is run. In a laboratory environment, external factors such as experimenter skill can change over the course of a study, resulting in a correlation among the residuals. No such external factors exist in Möbius, and residuals remain independent of experiment number [9].

To test whether the errors are independent of the predictor variables (model factors), we analyze a scatter plot of the studentized residuals versus the predicted response. There should be no visible trend in the data if the assumption of independence of the residuals is to hold. The same plot is used to verify that the residuals have constant variance. Again, the plot should be structureless if the constant variance assumption is to hold. Figure 3.7(a) shows a structureless scatter plot of studentized residuals. No trend is evident from the plot, and the variance is assumed to be constant. Figure 3.7(b) shows an increasing spread as the predicted response increases. That implies that the error distribution depends on the predic-



(a) Constant Variance

(b) Nonconstant Variance

Figure 3.7: Scatter plots to detect constant variance of residuals

tor variables, and that the regression model does not fully take into account the effects from these predictor variables. One possible method to stabilize the variance is to apply a *transformation* to the response data. Response transformations are discussed in the next section.

3.9 Transformations

The term *transformation* refers to applying a function to the observed response values y to create the *transformed* values y'. For example, if the square root function is applied to the observed response, the transformed regression model is

$$y' = \sqrt{y} = \beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k + \varepsilon$$

The analysis of the regression model is performed as usual, but the conclusions apply to the transformed response [8].

The use of transformations should be investigated in three situations. The first is when the response data cover several orders of magnitude and the sample size is relatively small. If the ratio $y_{\max}/y_{\min}$ is greater than 10, then a transformation may reduce the range of variability of the response. The second case in which a transformation may be appropriate is when a priori knowledge of the system being modeled dictates that a function of the response, rather than the response itself, is the better variable to use in the regression model. For example, in a queuing model, the experimenter may measure interarrival times y for requests. It may be known that the number of requests per unit time $1/y$ has a linear relationship with the predictor variables x_i. In that case, the reciprocal transformation $1/y$ should be used [9]. The third reason to use a transformation is to stabilize nonconstant variance of the residuals. In the previous section, the scatter plot of studentized residuals versus predicted response was used to detect nonconstant variance of the residuals. There exist specific transformations that can stabilize the variance of residuals exhibiting certain behavior. The remainder of this section will explore the types of transformations available in Möbius.

The most common type of transformation is the variance-stabilizing *power transformation* [21]. Power transformations express the transformed response y' as a power of the original response y:

$$y' = y^{\lambda}$$

Suppose that the standard deviation σ of the response y is proportional to some power α of the response mean μ:

$$\sigma \propto \mu^{\alpha}$$

It can be shown that the standard deviation $\sigma_{y'}$ of the transformed response is related to the mean of the original response as

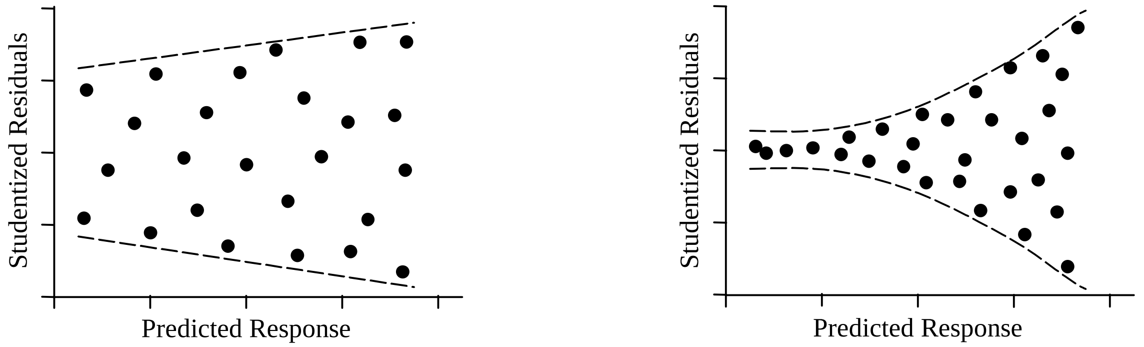
$$\sigma_{y'} \propto \mu^{\lambda+\alpha-1}$$

In order to stabilize the variance of the transformed response, the value of λ must be set to $1-\alpha$. Table 3.9 lists the relationships between the standard deviation and the mean of the response for some of the power transformations [8]. Given a particular relationship in the first column, application of the corresponding transformation will produce a transformed response with constant variance. For example, if the standard deviation of the response is equal to a constant *c*, then no transformation is necessary. If the variance of the response is equal to the mean, as in a Poisson distribution, then the square root transformation will make the variance constant. The power transformations have little effect if the range of response values is less than two or three [9].

The transformations in Table 3.9 are listed in order of increasing *strength*. The strength of a transformation refers to the amount of scaling that the transformation applies to the plot of residuals versus predicted response. In practice, the experimenter should observe the plot of studentized residuals versus predicted response and determine the amount of scaling necessary to bring the plot to a constant variance [10]. For example, Figure 3.8 shows

Table 3.9: Variance stabilizing power transformations of the form $y' = y^{\lambda}$

σ to μ Relationship	α	$\lambda = 1-\alpha$	Power Transformation
$\sigma \propto c$	0	1	None
$\sigma \propto \mu^{0.5}$	0.5	0.5	Square Root
$\sigma \propto \mu$	1	0	Natural Log
$\sigma \propto \mu$	1	0	Base 10 Log
$\sigma \propto \mu^{1.5}$	1.5	-0.5	Reciprocal Square Root
$\sigma \propto \mu^{2}$	2	-1	Reciprocal



(a) Near constant

(b) Far from constant

Figure 3.8: Nonconstant residual variances requiring different strength transformations

two residual plots with different deviations from constant variance. Figure 3.8(a) shows residuals that increase slightly as the predicted response increases. The spread is nearly constant, and only a weak transformation, such as the square root, is necessary to stabilize the variance. Figure 3.8(b) shows a rapid increase in the spread of residuals as the predicted response increases. A strong transformation, such as the reciprocal, may be necessary to equalize the residuals.

Three other transformations, presented in Table 3.10, apply to bounded responses. The *arc sine square root* transformation is useful for binomial data or when the response represents a proportion or percentage. The *omega* transformation maps response values be-

Table 3.10: Useful transformations for bounded response data

Transformation	**Equation**	**Restrictions**
Arc Sine Sqrt	$y' = \text{ArcSin}\sqrt{y}$	$0 < y < 1$
Omega	$y' = 10\text{Log}_{10}\left(\frac{y}{1-y}\right)$	$0 < y < 1$
Logit	$y' = \text{Ln}\left(\frac{y - Lower}{Upper - y}\right)$	$Lower < y < Upper$

tween zero and one to the real line. Omega is most useful when the response values are close to zero or one, and has little effect if most values are near one-half. The transformed data is in units of decibels (**dB**). A more general transformation similar to omega is the *logit* transformation, which maps any range of values to the real line. The user supplies upper and lower bounds [9].

In addition to stabilizing the residual variance, transformations can make the distribution of the response closer to the normal distribution, and improve the fit of the model to the observed response data. It is important to remember that the analysis results apply to the transformed response, not to the original values. Therefore, model prediction is performed in the transform domain. That is generally not a problem, because inverse transformations are easily computed, if necessary. The transformation selection process is often trial-and-error. Möbius provides an interface that allows the experimenter to easily investigate different transformations until the desired one is achieved. A more in-depth discussion of transformations is presented in [10].

3.10 Confidence Intervals

Several statistical quantities are estimated during analysis of designed experiments. Each estimate has a certain amount of uncertainty. Therefore, it is necessary to state confidence intervals for all estimates. This section presents an overview for calculating confidence intervals for the regression coefficients and the predicted response. Statistical details can be found in [18].

The general form of the regression model is $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ with independent and normally distributed errors. The least squares vector $\hat{\boldsymbol{\beta}}$ estimates $\boldsymbol{\beta}$ to produce the regression model $\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}$. Let C_{ii} be the i^{th} diagonal element of $(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}$. The *standard deviation of errors* or *standard error* for each regression coefficient $\hat{\beta}_i$ is

$$se_i = \sqrt{(MSR) \cdot C_{ii}}$$

where *MSR* is the regression model variance estimate presented in Section 3.3 [8]. It can be shown that the statistics

$$\frac{\hat{\beta}_i - \beta_i}{se_i} \qquad i = 0,1,\ldots,k$$

have a *t distribution* with $n-k-1$ degrees of freedom. A $100(1-\alpha)$ percent confidence interval (CI) for each regression coefficient is

$$\hat{\beta}_i - t_{\frac{\alpha}{2},n-k-1}\left(se_i\right) \le \beta_i \le \hat{\beta}_i + t_{\frac{\alpha}{2},n-k-1}\left(se_i\right)$$

Confidence intervals can also be calculated for the mean response at a particular point $\mathbf{x}_p$:

$$\mathbf{x}_p = \begin{bmatrix} 1 \\ x_{p1} \\ x_{p2} \\ \vdots \\ x_{pk} \end{bmatrix}$$

The mean response at $\mathbf{x}_p$ is

$$\mu_{y|\mathbf{x}_p} = \beta_0 + \beta_1 x_{p1} + \beta_2 x_{p2} + \ldots + \beta_k x_{pk} = \mathbf{x}_p^{\mathrm{T}}\boldsymbol{\beta}$$

The unbiased estimated mean response at $\mathbf{x}_p$ is

$$\hat{y}(\mathbf{x}_p) = \mathbf{x}_p^{\mathrm{T}}\hat{\boldsymbol{\beta}}$$

The standard error of the estimated mean is

$$se_{\hat{y}(\mathbf{x}_p)} = \sqrt{(MSR)\cdot \mathbf{x}_p^{\mathrm{T}}(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{x}_p}$$

The statistic

$$\frac{\hat{y}(\mathbf{x}_p) - \mu_{y|\mathbf{x}_p}}{se_{\hat{y}(\mathbf{x}_p)}}$$

has a *t* distribution with $n-k-1$ degrees of freedom [8]. A $100(1-\alpha)$ percent CI on the mean response at $\mathbf{x}_p$ is

$$\hat{y}(\mathbf{x}_p) - t_{\frac{\alpha}{2},n-k-1}\left(se_{\hat{y}(\mathbf{x}_p)}\right) \le \mu_{y|\mathbf{x}_p} \le \hat{y}(\mathbf{x}_p) + t_{\frac{\alpha}{2},n-k-1}\left(se_{\hat{y}(\mathbf{x}_p)}\right)$$

The goal of developing a regression model is usually to predict response values at points within the design region. Each predicted point has a corresponding prediction interval (PI). The standard error for the predicted value at a single point is

$$se_{\hat{y}(\mathbf{x}_p)} = \sqrt{(MSR)\cdot(1 + \mathbf{x}_p^{\mathrm{T}}(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{x}_p)}$$

A $100(1-\alpha)$ percent PI for the predicted response at some point $\mathbf{x}_p$ is

$$\hat{y}(\mathbf{x}_p)-t_{\frac{\alpha}{2},n-k-1}\left(se_{\hat{y}(\mathbf{x}_p)}\right)\le\ y(\mathbf{x}_p)\ \le\ \hat{y}(\mathbf{x}_p)+t_{\frac{\alpha}{2},n-k-1}\left(se_{\hat{y}(\mathbf{x}_p)}\right)$$

The prediction interval will always be wider than the corresponding confidence interval. The PI is the interval that contains the individual observations $100(1-\alpha)$ percent of the time. The CI is the interval that contains the mean of an infinite number of future observations $100(1-\alpha)$ percent of the time [8].

Care should be taken when predicting future response values. The standard deviation of the prediction increases with distance from the design center. That results in a wider confidence interval and a less accurate prediction. A regression model that fits the observed data well within the region of the original observations may not fit well outside of that region [7].

4. DOE STUDY EDITOR INTERFACE

4.1 User Interface Flow

Chapter 1 presented the modeling flow in Möbius. The first steps in the modeling process are the definitions of the atomic, composed, and reward models. It is at these three modeling levels that all global variables are defined, but not assigned values. The study level is where global variables are assigned values. Figure 4.1 shows the modeling flow from the study level onward for a DOE study. During the study *definition*, all global vari-

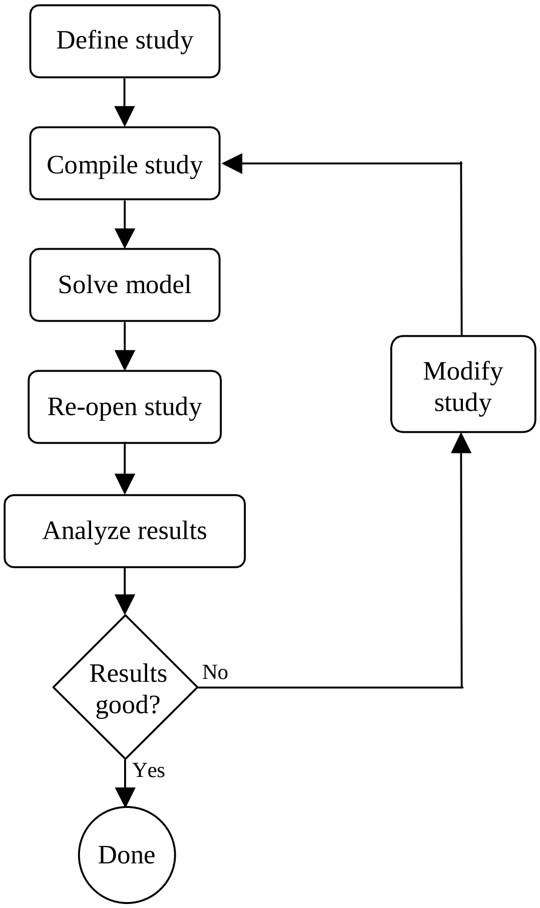


Figure 4.1: Flow of model solution with the DOE study editor

ables are assigned values for each experiment. After a study is defined, it is saved and compiled into object code, which is later linked in by the solver during the model solution step. After all experiments are complete, the DOE study is reopened, and the results from the experiments are analyzed. If necessary, the study can be modified, and new experiments can be performed. This iterative process continues until the desired results are achieved. The remainder of this chapter will present the details of the *graphical user interface* of the DOE study editor.

4.2 Defining a DOE Study

Figure 4.1 shows that the first step of the DOE experimentation process is to *define* a DOE study. There are several steps in the study definition process. Figure 4.2 presents the flow of events involved in defining a DOE study. The remainder of this section will present the details of each step of the DOE study definition.

The creation of a new DOE study is similar to the creation of any new model in Möbius. First, from the Möbius project window, the type of module must be selected. The user must select the study node by clicking on the tree node named “Study” and then press the

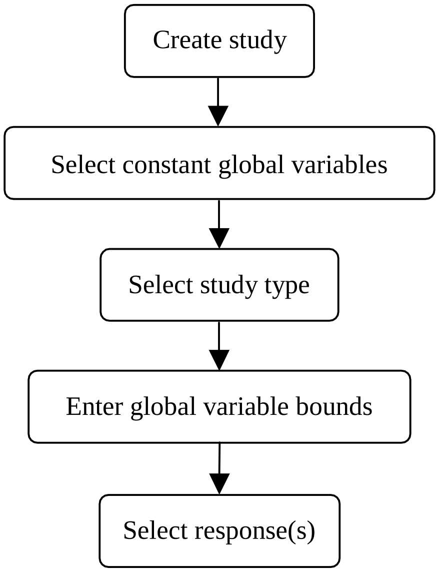
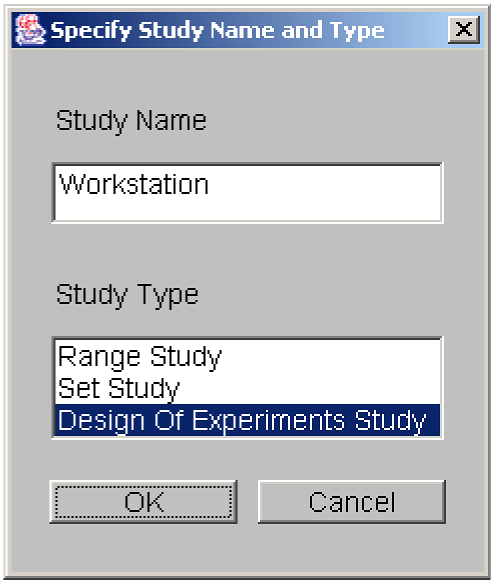

Create study
Select constant global variables
Select study type
Enter global variable bounds
Select response(s)

Figure 4.2: The flow of events in the DOE study definiton process



Workstation

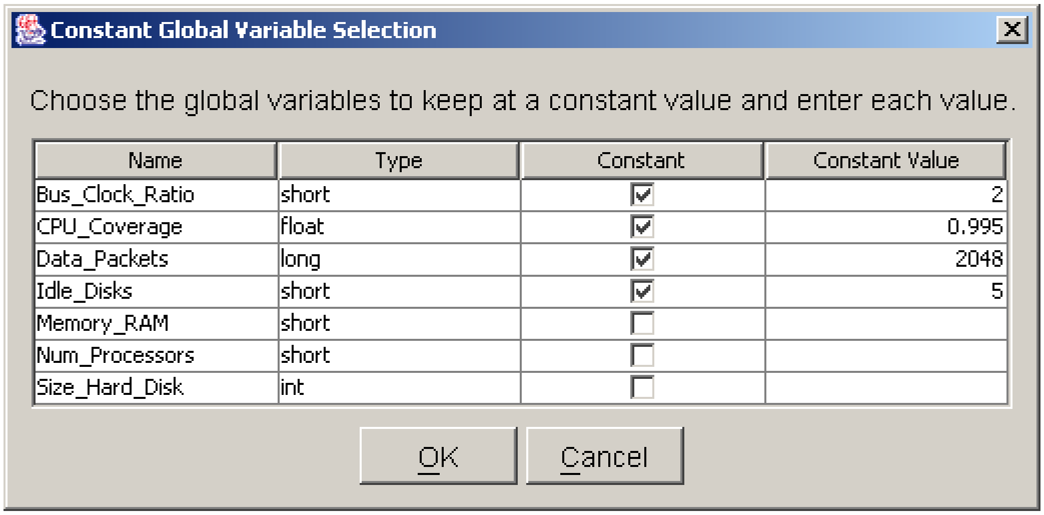
Range Study
Set Study
Design Of Experiments Study

Figure 4.3: DOE study creation dialog

the New button. A dialog asking for the study name and type is presented. Figure 4.3 shows a dialog from the creation of a study named "Workstation" of type "Design of Experiments Study." After pushing the OK button, the user selects the reward model upon which the study will be based. After selecting the appropriate reward model and pushing OK, the user is presented with the Constant Global Variable Selection dialog.

The purpose of the Constant Global Variable Selection dialog is to help the user select which global variables are to be held at constant values throughout the study. There are several reasons why some global variables would remain fixed to specific values during a study. One reason is that the variation of some model parameters may be relevant to another study, but not to the current one. Another reason to fix global variables to constant values is that a previous DOE study may have revealed that the variables do not significantly affect the reward model under study, so can be fixed to convenient values. Alternatively, the optimal values of some global variables may already have been determined from earlier studies. In that case, the global variables can be fixed to their optimal values. Any global variables that are held constant will not be included in the empirical model of the response created by the DOE analysis.

Figure 4.4 shows a Constant Global Variable Selection dialog with a list of all global variable names and types defined in the underlying model. To designate a global variable as constant, its checkbox in the Constant column must be selected by a single mouse click.



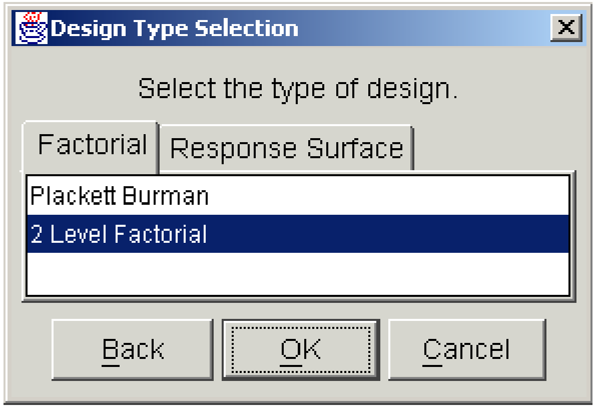
Choose the global variables to keep at a constant value and enter each value.

Name	Type	Constant	Constant Value
Bus_Clock_Ratio	short	☑	2
CPU_Coverage	float	☑	0.995
Data_Packets	long	☑	2048
Idle_Disks	short	☑	5
Memory_RAM	short	□	
Num_Processors	short	□	
Size_Hard_Disk	int	□	

Figure 4.4: Constant global variable selection dialog

Once that box has been selected, the value can be entered into the corresponding row under the Constant Value column. There are two ways to enter values into a table in the DOE study editor. One is to single-click the table cell and simply start typing the value. The other is to double-click the table cell. A blinking caret will appear, and all text will be left-justified while the user is typing. Regardless of which method is used, pushing the Enter key will commit the new values to the table. After being entered, all numbers will be right-justified in their table cells. When all of the constant global variables have been selected, the user can push the OK button to proceed to the next step. At any point during the study definition process, the user may push the Cancel button in order to cancel the study definition process. After pushing the Cancel button, the user is asked whether the study is to be saved. If it is, it can be reopened later, and the definition process can resume. If the user chooses not to save the study, then it will be permanently discarded.

The next step in the DOE study definition is the selection of the design type. Figure 4.5 shows the Design Type Selection dialog. The two categories of designs, factorial and response surface, are presented in separate tab panels. Selecting the Factorial tab shows a list from which either a Plackett-Burman or a two-level factorial design may be selected. Selecting the Response Surface tab displays a list with a choice of either a Central Composite or a Box-Behnken design. After selecting the desired design type from the appropriate list, the user may either proceed to the next step by pushing the OK button, or return to the



Design Type Selection

Select the type of design.

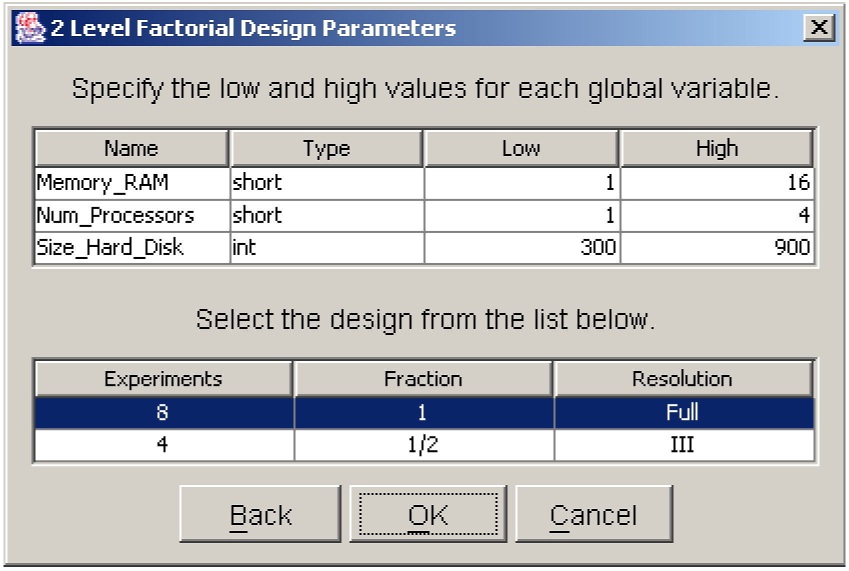
Plackett Burman

2 Level Factorial

Figure 4.5: Design type selection dialog

previous dialog by pushing the Back button. At any point in the study definition process, the user may move forward or backward to any step and make changes to the study.

After the user has selected the design type, a dialog prompts the user to enter parameters for the chosen design. All design types require that both lower and upper bounds be specified for each nonconstant global variable. In factorial designs, global variables are varied over exactly two values. The values are equal to the low and high values entered in the design parameters dialog. Figure 4.6 shows an example 2-Level Factorial Design Parameters dialog. The top table lists each nonconstant global variable name and type. A low and



2 Level Factorial Design Parameters

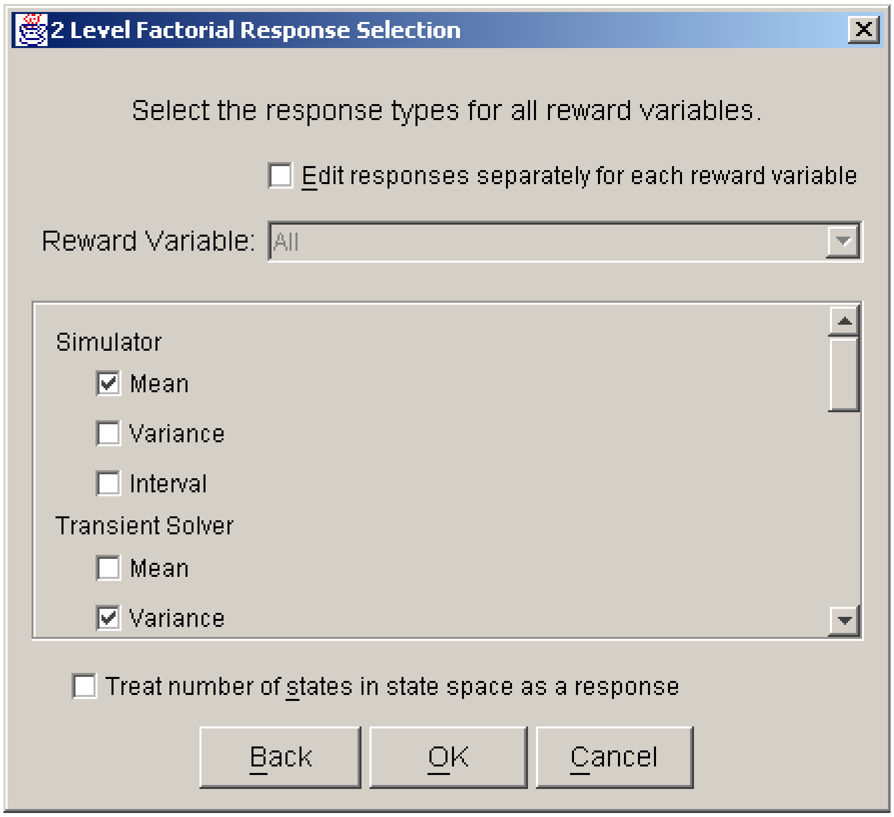
Specify the low and high values for each global variable.

Name	Type	Low	High
Memory_RAM	short	1	16
Num_Processors	short	1	4
Size_Hard_Disk	int	300	900

Select the design from the list below.

Experiments	Fraction	Resolution
8	1	Full
4	1/2	III

Figure 4.6: Design parameters dialog



2 Level Factorial Response Selection

Select the response types for all reward variables.

□ Edit responses separately for each reward variable

Reward Variable: All

Simulator

☑ Mean

□ Variance

□ Interval

Transient Solver

□ Mean

☑ Variance

□ Treat number of states in state space as a response

Figure 4.7: Response selection dialog

high bound must be entered for each variable. In two-level factorial and central composite designs, a second table displays a list of the different designs that are available. The user should select the design that best suits the available computing resources. A single mouse click will highlight the row of the selected design. In Figure 4.6, the full resolution design is selected. After all global variable bounds have been entered and the design fraction is selected, the study definition may proceed to the response selection step.

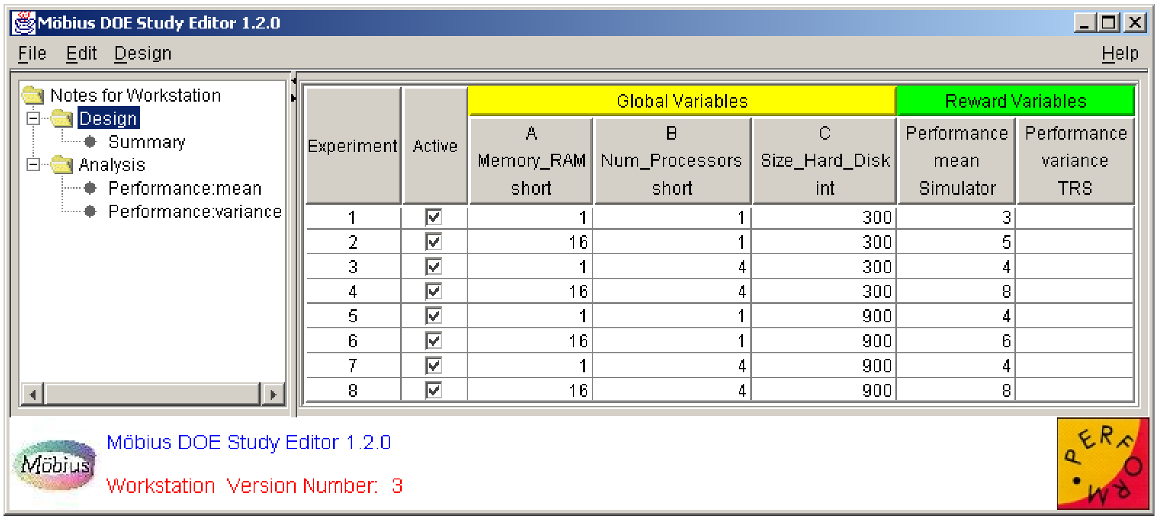
A reward model may consist of several reward variables. Each of the reward variables can be solved using a variety of solution methods available in Möbius. For example, the mean and variance of a reward variable measuring workstation performance may be calculated by a simulation or via several numerical solution methods. The solution from each solver can be represented by a unique response. Each response can be analyzed separately, allowing for the comparison of different solution methods on the same reward variable.

Figure 4.7 shows a Response Selection dialog. A checkbox near the top, labeled "Edit responses separately for each reward variable," allows a different set of solution methods to be used for each reward variable. Usually, this checkbox is not selected, and all reward variables are solved using the same solution methods. A scrolling panel in the middle of the dialog lists all of the solver types available in Möbius. The user selects the desired solution methods by clicking the checkbox to the left of each solver name. At the bottom of the dialog window is a checkbox for selecting as a response the number of states in the state space. The purpose of the Response Selection dialog is to allow quick selection of a potentially large number of responses during an initial study definition. If desired, new responses can be added and existing responses deleted after the study definition process is complete.

Pushing the OK button completes the study definition process and opens the DOE study editor window, which is described in the next section. The study can be redefined at any time, as described later in this chapter.

4.3 DOE Study Editor Window

After a study is defined, the DOE study editor window appears. It is the main window and is always present, except during the study definition process. Figure 4.8 shows an example DOE study editor window. The data is from the fictitious workstation model from Chapters 2 and 3. The window has four areas. The first is the menu bar at the top. The menu functionality will be discussed later in the chapter. At the bottom is the logo panel,



		Global Variables			Reward Variables	
Experiment	Active	A Memory_RAM short	B Num_Processors short	C Size_Hard_Disk int	Performance mean Simulator	Performance variance TRS
1	☑	1	1	300	3	
2	☑	16	1	300	5	
3	☑	1	4	300	4	
4	☑	16	4	300	8	
5	☑	1	1	900	4	
6	☑	16	1	900	6	
7	☑	1	4	900	4	
8	☑	16	4	900	8	

Möbius DOE Study Editor 1.2.0

Workstation Version Number: 3

Figure 4.8: DOE study editor window

which is similar to the logo panel found in other Möbius modules. The lower central portion of the logo panel contains a status area that occasionally displays messages in red text about the status of an activity.

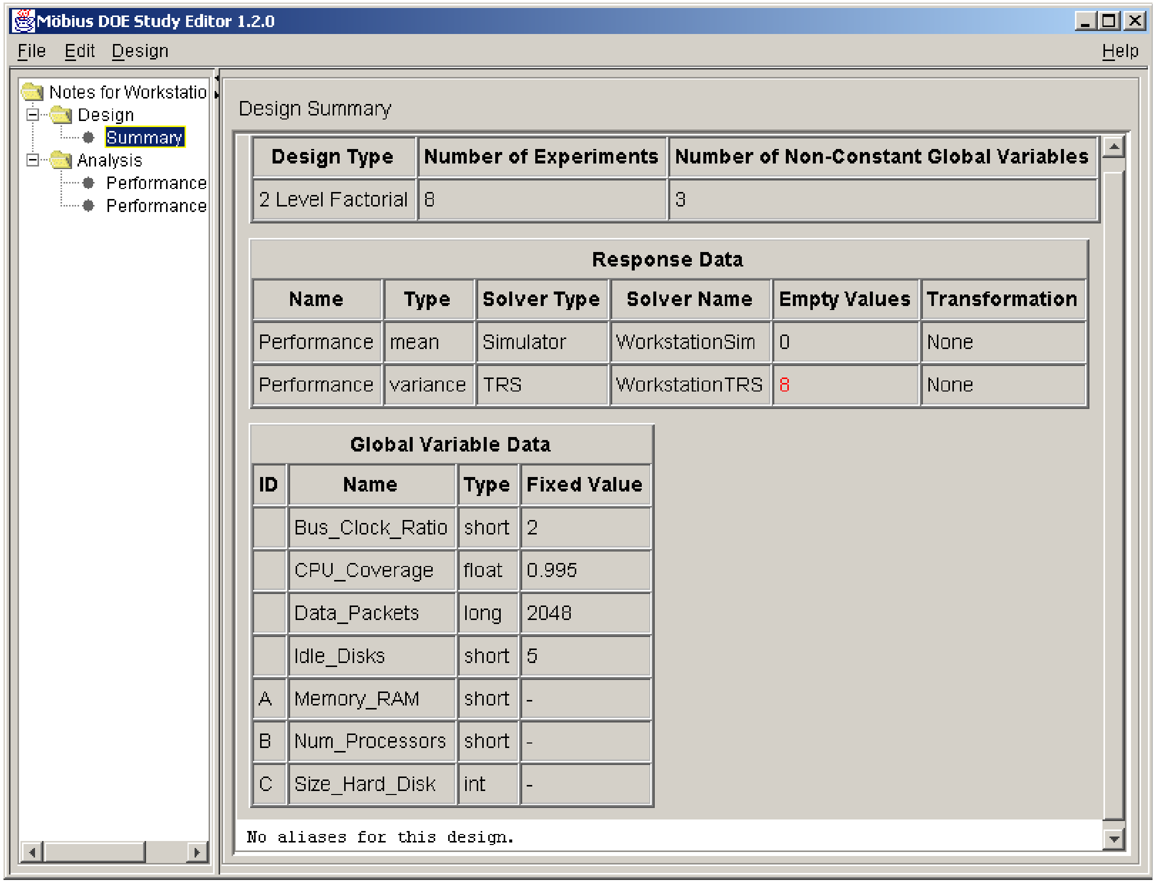
On the left side of the window is the *design tree* panel. The selected node of the design tree determines what is displayed in the *main panel*, which is the area directly to the right of the design tree. The design tree panel and main panel are separated by a movable *divider*. The mouse can be used to drag the divider left and right in order to give more screen area to a particular panel. At the top of the divider, as shown in Figure 4.8, there are two small triangles. Clicking one of the triangles moves the divider all the way to the edge of the window, causing either the design tree panel or the main panel to be hidden, depending on which triangle was clicked. If a panel is hidden, it can be revealed by a single click anywhere on the divider.

The graphical user interface for the DOE study editor was written using Java Swing classes. The appearance of the GUI varies depending on the operating system being used. The screen shots in this thesis were taken on the Microsoft Windows operating system. The DOE study editor will appear slightly different on other operating systems, such as UNIX or Linux. However, the functionality described in this thesis will be consistent across operating systems.

4.4 Design Tree

Most of the time spent using the DOE study editor involves analyzing the contents of the main panel. There are five different *views* available for display in the main panel. Each of the five views corresponds to one of the five different types of nodes in the design tree. Clicking on an unselected design tree node will change the contents of the main panel. This section gives a brief description of each design tree node.

At the top of the design tree is the Notes node. Selecting the Notes node displays an editable text area in the main panel. Any comments entered in the text area will be saved when the study is saved. The text area supports cut, copy, paste, undo, and redo functionality, either from the Edit menu or by keyboard shortcuts. Cut, copy, paste, undo, and redo are performed by typing Control-x, Control-c, Control-v, Control-z, and Control-Shift-z, respectively.



Design Summary

Design Type	Number of Experiments	Number of Non-Constant Global Variables
2 Level Factorial	8	3

Response Data

Name	Type	Solver Type	Solver Name	Empty Values	Transformation
Performance	mean	Simulator	WorkstationSim	0	None
Performance	variance	TRS	WorkstationTRS	8	None

Global Variable Data

ID	Name	Type	Fixed Value
	Bus_Clock_Ratio	short	2
	CPU_Coverage	float	0.995
	Data_Packets	long	2048
	Idle_Disks	short	5
A	Memory_RAM	short	-
B	Num_Processors	short	-
C	Size_Hard_Disk	int	-

```
No aliases for this design.
```

Figure 4.9: Summary panel

Below the Notes node is the Design node, which, when selected, displays the *design panel* in the main panel. The design panel presents all input data for the study, including all global variable and response values for each experiment. The design panel will be discussed in Section 4.6.

Selecting the Summary node displays a design summary, an example of which is shown in Figure 4.9. Response and global variable data are summarized in HTML-formatted tables. If the design is a two-level factorial type with less than full resolution, the alias structure is presented at the bottom of the summary panel. The text from the summary panel can be exported to a file using the Design menu's Export Text menu item. Further details of all menu items will be discussed in the next section.

The Analysis node contains a brief summary of the analysis procedure. Each response in the design panel has a corresponding tree node under the Analysis node. Response analysis requires several steps. Most of the functionality of the DOE study editor is used during response analysis. Before any analysis can be performed on a particular response, the user must enter response values for all experiments into the design panel. Details on how to do so will be presented in Section 4.6.

4.5 Menus

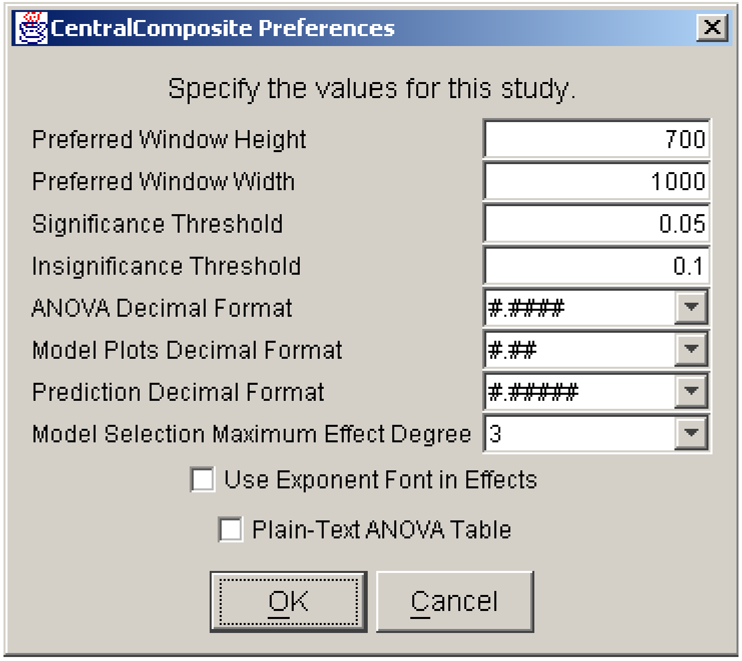
There are four menus in the DOE study editor. The first is the File menu, which performs the same functions as the File menus in other Möbius editors. The Save item saves any changes to the study to disk and compiles the global variable values for each experiment into object code. A dialog appears that shows the progress of the compilation and linking process. The object code will be linked in by the solver during model solution. The Document menu item saves an HTML description of the study in the study directory. The Close menu item closes the DOE study. If any changes were made, the user will be prompted to decide whether the changes should be saved.

The Edit menu provides basic cut, copy, and paste functionality in any editable text area or text field. There is also undo and redo functionality available only in the notes panel, as discussed earlier. Keyboard shortcuts are available for all menu items.

The Design menu contains items specific to the DOE study editor. The Export Regression Model menu item is only enabled when an appropriate analysis panel is displayed. The uses of exporting the regression model will be discussed in Section 4.10.

The Export Text menu item is used to save any displayed panel to a text or HTML-formatted file. If the notes or design panel is displayed, the contents of the panel will be saved to a text file. If the summary panel is displayed, the contents of the panel will be saved to an HTML file. Other uses of the Export Text menu item will be presented later.

The Redefine Design menu item is used to initiate the study definition process explained in Section 4.2. The user may wish to redefine a study to correct a mistake made during the initial study definition. He or she could also use redefinition to *augment* the design. Design augmentation involves adding more experiments to a study so that more information can be gained about the responses under study. For example, a half-fraction two-level factorial design may not present enough information for the user to build an accurate empirical model of the response. The design can be augmented with the other half-fraction in order to create the full resolution design, doubling the number of experiments in the study. However,


CentralComposite Preferences

Specify the values for this study.

Preferred Window Height	700
Preferred Window Width	1000
Significance Threshold	0.05
Insignificance Threshold	0.1
ANOVA Decimal Format	#.####
Model Plots Decimal Format	#.##
Prediction Decimal Format	#.#####
Model Selection Maximum Effect Degree	3

☐ Use Exponent Font in Effects

☐ Plain-Text ANOVA Table

Figure 4.10: Preferences dialog

half of the experiments will already have been completed. After the remaining half are run, an analysis of the full resolution 2^k design can be performed. One caveat of design redefinition is that all data in the response columns in the design panel are erased when a study is redefined. In order to restore the data, the user must re-query the results database to fill in the response columns. Instructions for gathering response data from the results database will be presented in the next section.

The Hide/Show Logo Panel menu item is useful for gaining valuable screen area. On small screens, it may be difficult to view some of the graphs in the analysis panels. Hiding the logo panel can increase the area available for viewing such graphs. That can reduce the amount of scrolling necessary in viewing the data. One disadvantage of hiding the logo panel is that the status bar will not be visible; furthermore, any important messages displayed there would also not be visible.

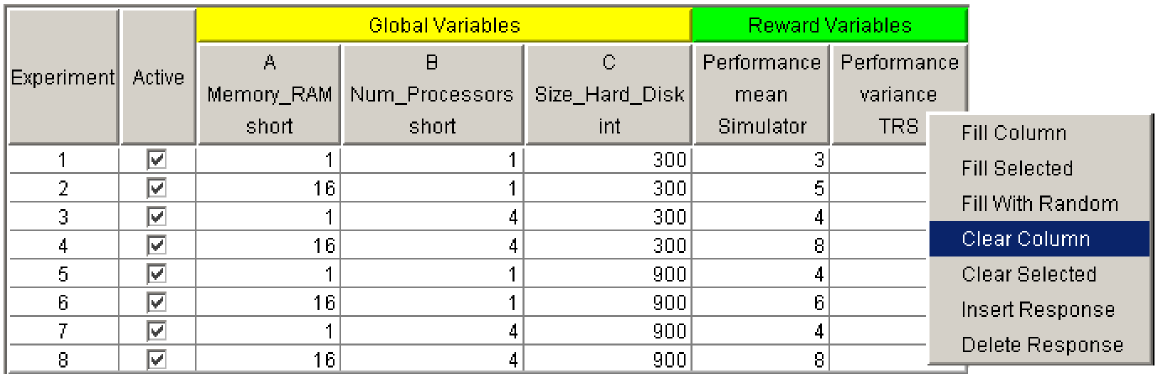
The last item in the Design menu displays a list of preferences for the current study. Figure 4.10 shows an example preferences dialog. The first two items allow the user to specify the preferred window dimensions. Once the preferred height and width are entered and the study is saved, the DOE study editor window will reopen with the specified window dimensions. This feature allows users to customize the window size to a particular large or small screen. The remaining preference items are specific to the analysis procedure and will be discussed later. It is important to note that saved preferences are relevant only to a specific study, not to all DOE studies.

The last menu is the Help menu on the right side of the menu bar. The About item displays information about copyrights and contact information for the PERFORM research group. The Reward Info item displays the name of the reward model on which the study is based.

4.6 Design Panel

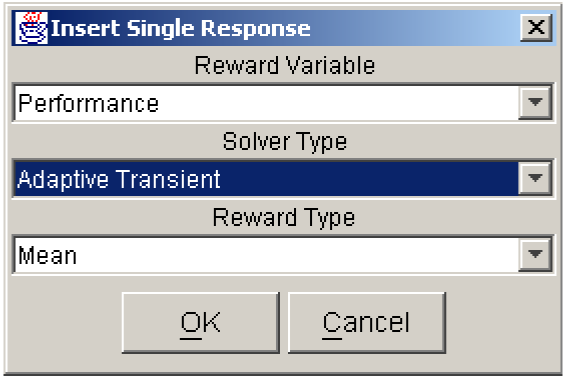
Whenever a DOE study is opened, the first view displayed in the main panel is the design panel. The design panel displays the entire experimental design in the *design table*. Figure 4.11 shows an example design table. The layout is very similar to that of the design matrix presented in Section 2.4.1. Each row represents an experiment. The global variable columns represent the factor values. The reward variable columns represent the responses. The Active column displays the experiments that will be performed by the solver, and replaces the Experiment Activator found in the Set and Range study editors. If a checkbox is selected, then the experiment for the corresponding row is active.

The user can activate several popup menus by right-clicking on certain column headers of the design table. The Active column has a popup menu with four choices for experiment activation. All experiments can be either activated or deactivated with a single menu



		Global Variables			Reward Variables	
Experiment	Active	A Memory_RAM short	B Num_Processors short	C Size_Hard_Disk int	Performance mean Simulator	Performance variance TRS
1	☑	1	1	300	3	
2	☑	16	1	300	5	
3	☑	1	4	300	4	
4	☑	16	4	300	8	
5	☑	1	1	900	4	
6	☑	16	1	900	6	
7	☑	1	4	900	4	
8	☑	16	4	900	8	

Figure 4.11: Design table



Reward Variable
Performance

Solver Type
Adaptive Transient

Reward Type
Mean

Figure 4.12: Response insertion dialog for design table

choice. Alternatively, experiments from only the selected rows can be activated or deactivated. Row selection is performed by left-clicking on a row with the mouse. The Shift and Control keys can be used with the mouse click to select a sequence or noncontiguous block of rows, respectively.

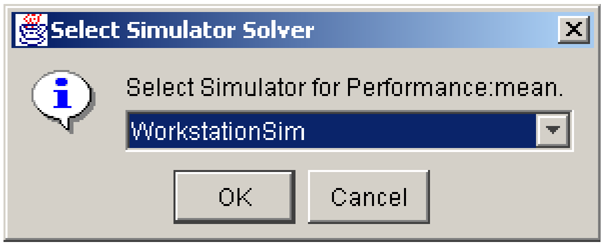
For response surface designs, a popup menu is available on the Experiment column header. The user may add experiments to the bottom of the table. Any user-added experiments may be removed from the table via the Remove Selected item on the Experiment column's popup menu. Only user-added experiments may be removed from the design table; the built-in design must remain intact. The ability to add experiments to factorial designs is not enabled. The factorial designs built in to Möbius are orthogonal. Adding arbitrary experiments would destroy the orthogonality and distort the analysis. Response surface designs do not require orthogonality for analysis. The user should avoid creating arbitrary experiments that are far from the design space of the original built-in design. Points far from the original design center can grossly distort the response surface, resulting in inaccurate prediction. For the same orthogonality reason, global variable values cannot be edited in the design table for factorial designs. They can, however, be edited in response surface designs. Again, the user is cautioned against entering factor levels that produce design points far from the original design.

The user can activate the response popup menu by right-clicking on any reward variable column header. The Insert Response item allows the user to add a single response column to the table. The dialog in Figure 4.12 appears, prompting the user to select the reward variable, the solver type, and the reward type. Pushing the OK button in the example shown in Figure 4.12 would add a new response measuring the mean performance using the adaptive transient solver. The Delete Response item removes a response column from the table. There must be at least one response, so the last response cannot be deleted.

There are three menu items in the response popup menu associated with filling the column with data. One item, which is very useful for experimenting with the DOE study interface, is the Fill With Random item. Selecting that menu item will fill the response column with uniformly distributed integers between 90 and 110. No analysis of a response can take place until the entire response column is filled with data. Therefore, the Fill With Random feature is a quick way for the user to start playing with the analysis tools, which will be described later. To analyze real data, the user must fill the response columns with solver data by querying the results database for the model solutions.

There are two popup menu items for filling a response column with data from the results database. The Fill Selected item fills in data only for the selected rows in the design table. The Fill Column item fills the entire column with the solved reward variable results. Selecting either of the two choices causes a dialog similar to the one in Figure 4.13 to appear. Because the column associated with the popup menu represents data from a Simulator, Figure 4.13 prompts the user to select from a list of simulators defined in the project. Once the solver is selected, the results database is queried, and the response values are retrieved and placed into the response column.

To erase values from a response column, the Clear Selected or Clear Column popup menu items can be selected. The Clear Selected item clears the response values only for the selected rows in the table. The Clear Column item erases the entire column. The user may



Select Simulator for Performance:mean.

WorkstationSim

Figure 4.13: Solver selection dialog for results database query

also edit the response values by typing the values directly into the table cells. Only after *every* response value in a column is entered may analysis of the response be performed.

4.7 Analysis Procedure

For every response column in the design table, there is a corresponding node under the Analysis node in the design tree. After all response values have been filled in for a particular response column, the corresponding tree node can be activated. Clicking on the response node reveals the analysis tabs in the main panel. There is a sequence of six tabs that, when clicked, reveal a panel for performing one step in the analysis procedure. The user interface for the tab panels will be presented in the remainder of the chapter. Figure 4.14 shows the analysis flow corresponding to the six tab panels. In summary, the goal is to build

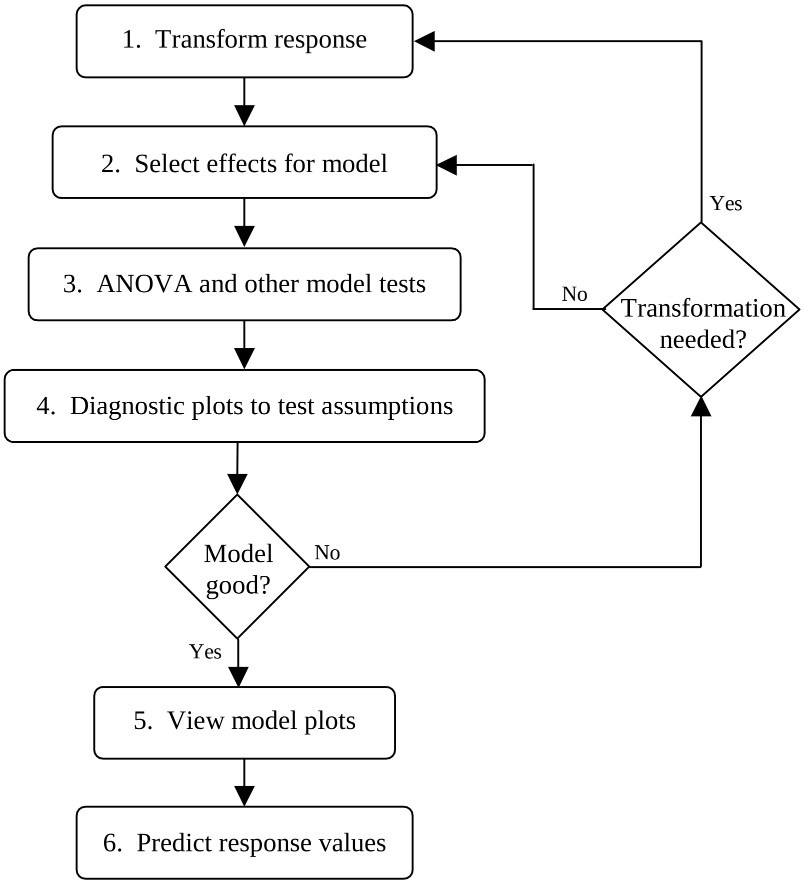
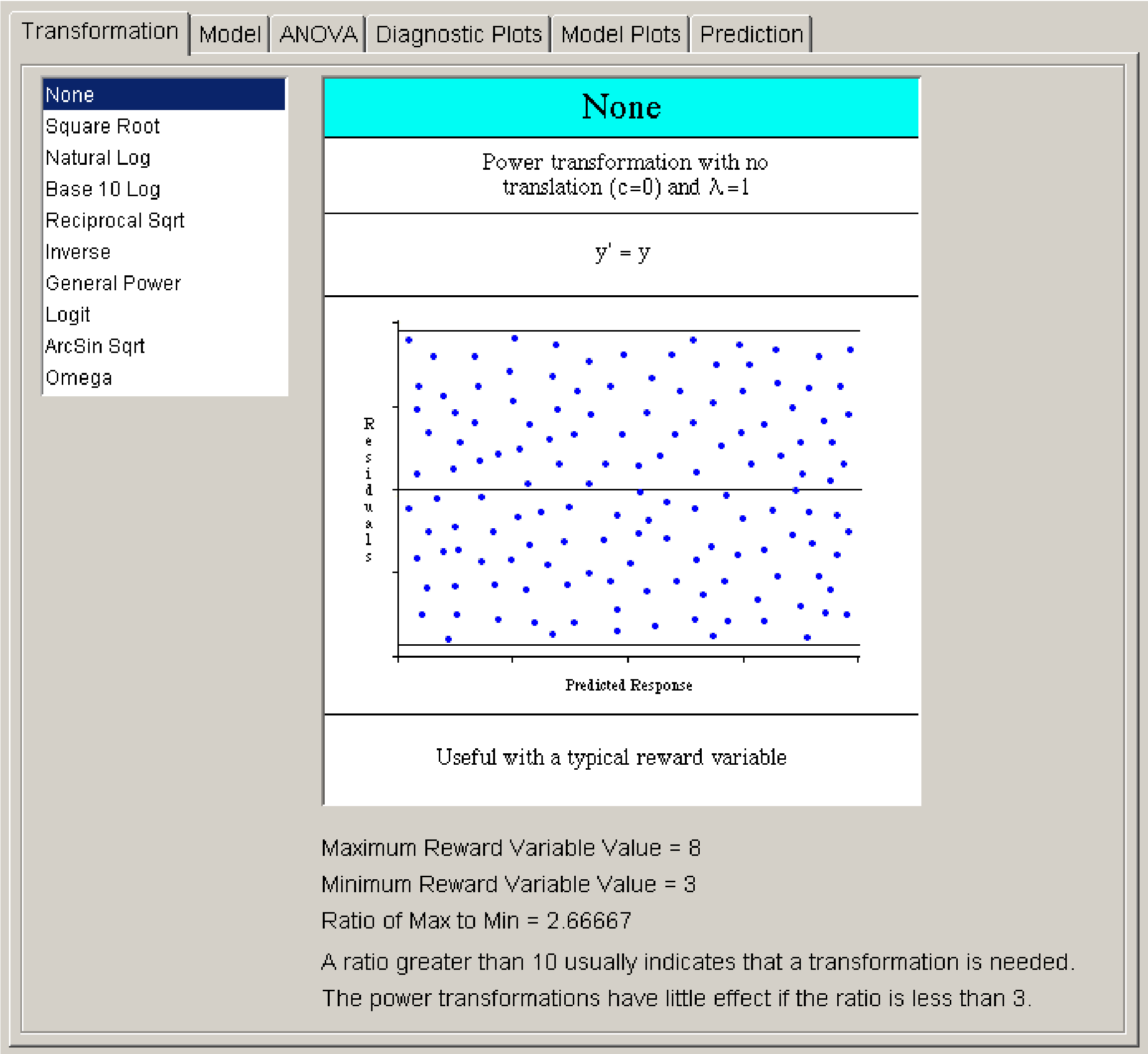

1. Transform response
2. Select effects for model
3. ANOVA and other model tests
4. Diagnostic plots to test assumptions
Model good?
No
Yes
Transformation needed?
Yes
No
5. View model plots
6. Predict response values

Figure 4.14: Analysis flow in the DOE study editor

and evaluate a regression model of the response and to use the regression model to predict response values at points in the design space other than the experimental points. Steps 1 and 2 transform the observed response values (if necessary) and build a regression model from the significant effects. Steps 3 and 4 analyze the model for significance and test the assumptions made on the data. If the model is good, then it can be used to produce graphs of the response and predict new values in the design space, as indicated in steps 5 and 6 of Figure 4.14. The six tabs are shown along the top of Figure 4.15. They progress from left to right as the analysis progresses.



None

Power transformation with no translation (c=0) and λ=1

y' = y

Useful with a typical reward variable

Maximum Reward Variable Value = 8

Minimum Reward Variable Value = 3

Ratio of Max to Min = 2.66667

A ratio greater than 10 usually indicates that a transformation is needed.

The power transformations have little effect if the ratio is less than 3.

Figure 4.15: Six tab panels in analysis procedure with transformation tab selected

4.8 Transformations

The first step in the analysis procedure allows the user to transform the observed response values. After a transformation is applied, all analysis is done on data that are some function of the original solver values. Section 3.9 presented three situations in which a transformation may be required. The first occurs when the maximum-to-minimum ratio of the response values is large. The second occurs when a priori knowledge of the system results in the belief that a particular function of the response, rather than the response itself, is better suited to a regression model. Both of those situations require that the transformations take place before any analysis is performed. It is for that reason that the Transformation tab is placed first in the analysis flow.

The third situation in which a transformation is required occurs after an initial analysis has revealed that the assumptions on the response data have been violated. For example, a nonconstant variance might not be discovered until the Diagnostic Plots phase of the analysis. The user must then backtrack to the Transformation tab, build a new regression model, and re-analyze the data. It is very common to move backward and forward while experimenting with different model parameters. The interface allows the user to click on a previous tab at any time to try out new settings. Regression model analysis is often trial-and-error because the underlying model of the response behavior is unknown. The whole point of experimental design is to better understand the structure of the response.

Figure 4.15 shows the list of available transformations on the left side of the transformation panel. Selecting any transformation changes the graphic to the right of the transformation list. The first seven transformations belong to the *power* family. The graphic for each power transformation contains a sample of what the Residuals vs. Predicted Response plot should look like if the selected power transformation is applicable. All graphics contain information about when the selected transformation is most useful. Most transformations have restrictions on the input data, which are illustrated in each transformation's graphic. Many transformations have optional parameters that can be entered in the text fields that appear below the transformation list when appropriate. The most common parameter is the translation constant *c*. Adding the appropriate constant to each reward variable will translate the values to a range for which the selected transformation is valid. If the selected transformation is invalid for any response values, a dialog will warn about the offending values when the user tries to progress to the next analysis step. The ratio of maximum-to-minimum response values is presented below each graphic to aid in the decision on whether a transformation is necessary.

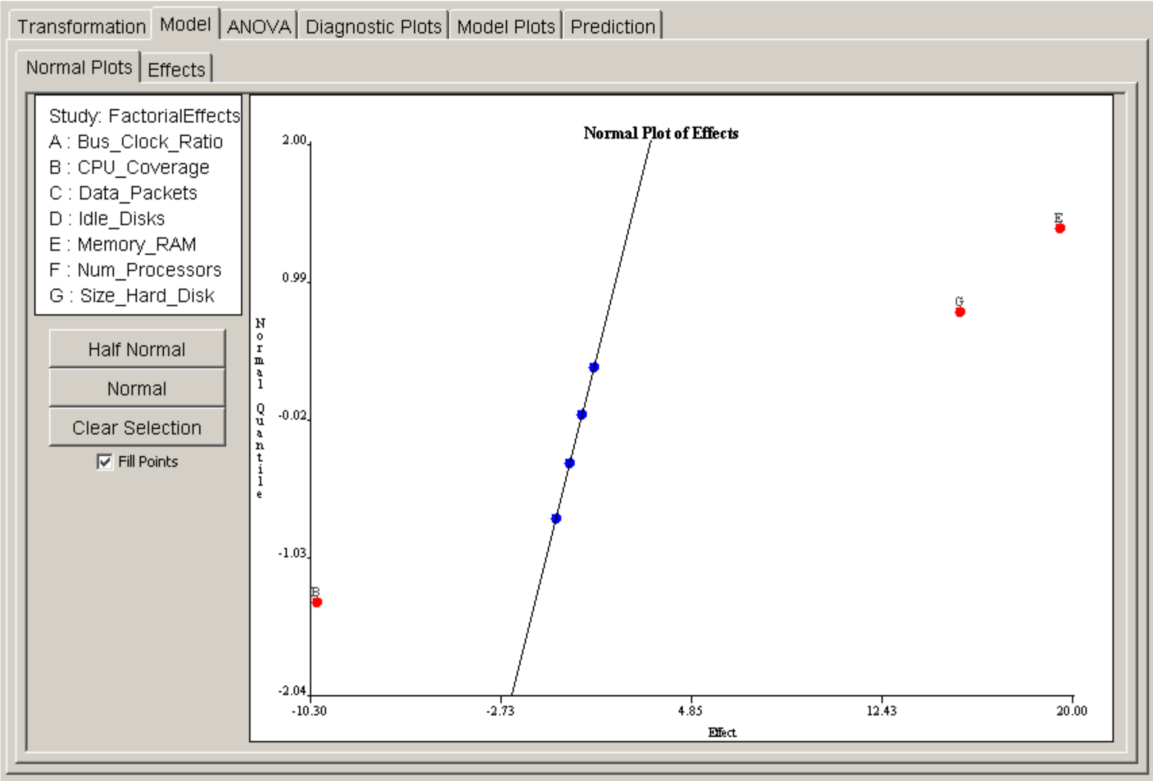
Most commonly, no transformation is applied to the response. The remaining analysis steps are identical regardless of whether the response has been transformed. If a transformation is used, it is important for the user to remember that regression model plots and prediction are presented in the transformed domain. Whenever possible, the response is labeled wrapped in the corresponding transformation function. For example, if the natural logarithm transformation is used, the response will be labeled “Ln[response_name].”

4.9 Model Selection

After the appropriate transformation is selected, the next step is to click on the Model tab and choose the effects to include in the regression model. The method for choosing effects varies depending on whether a factorial or a response surface design is being analyzed. Each case will be presented separately.

4.9.1 Factorial model selection

For factorial designs, one may select effects either graphically or from a table. Each selection interface has a separate tab panel within the model panel. The Normal Plots tab shows the normal quantile plot of effects. Figure 4.16 shows an example *normal plot* of effects. If the effects follow a normal distribution, then the normal quantile plot of ordered effects will follow a straight line. The outliers are the significant effects. On the normal plot of effects, outliers appear at the top and bottom of the plot. Figure 4.16 shows three outliers: two at the top and one at the bottom. To select an effect for inclusion in the regression model, the user simply clicks the point corresponding to the effect. The selected points will turn from blue to red, and the label of the chosen effect will be placed near the selected point. An ordinary least squares straight line will be drawn through all unselected blue points.



Study: FactorialEffects
A : Bus_Clock_Ratio
B : CPU_Coverage
C : Data_Packets
D : Idle_Disks
E : Memory_RAM
F : Num_Processors
G : Size_Hard_Disk

Figure 4.16: Normal plot of effects for factorial design

The most common graphical method for selecting effects is the *half-normal plot*. The half-normal plot is identical to the normal plot, but the absolute values of the effects are plotted. All outliers appear in the top right section of the plot. Figure 4.17 shows the same data as Figure 4.16, but plotted on a half-normal quantile plot. All three outliers (largest effects) are selected at the top right of the plot. In this example, main effects B, E, and G are included in the regression model.

The user may switch back and forth from normal to half-normal views of the effects plots by selecting the appropriate button under the factor list at the left side of the normal plots panel. The user can unselect all selected points by pressing the Clear Selection button. The Fill Points checkbox controls whether the plotted points are filled-in colored disks, or simply hollow circles. Deselecting the checkbox can make point labels easier to read on a very crowded plot.

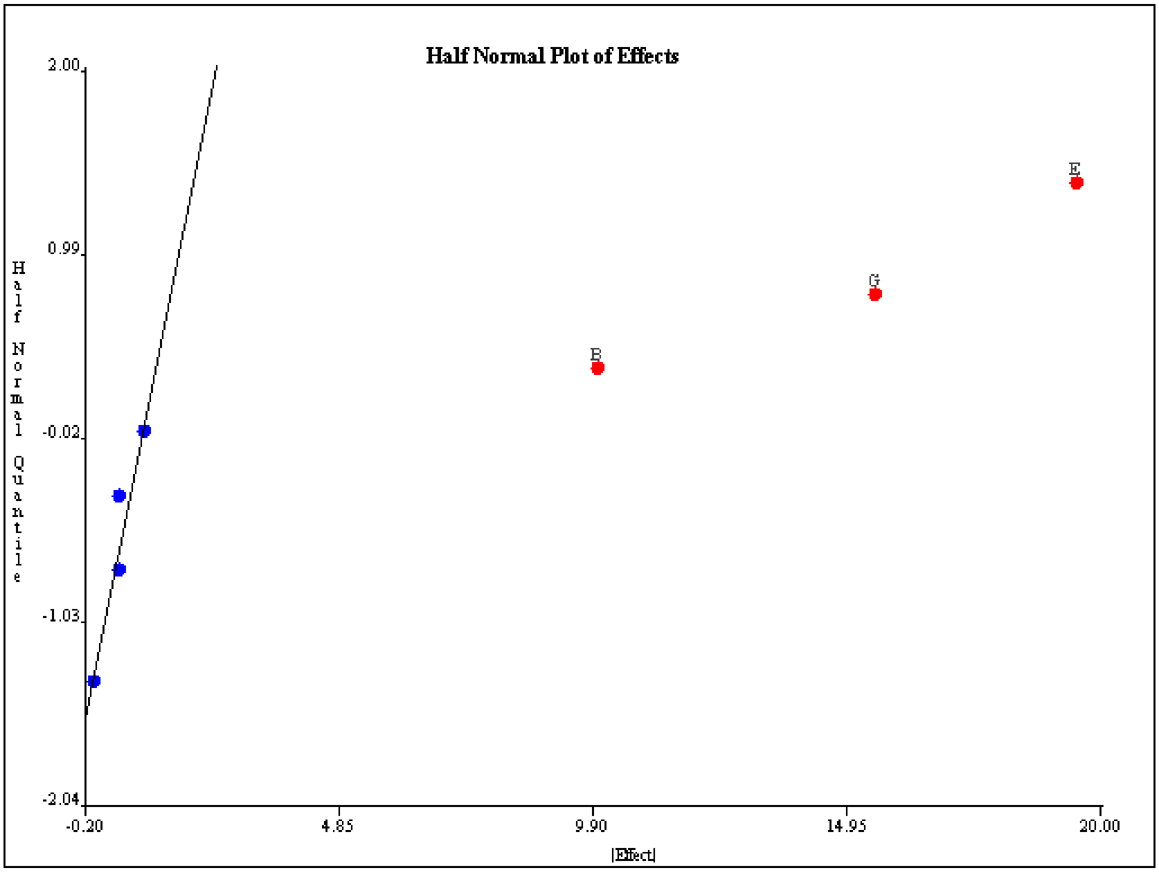
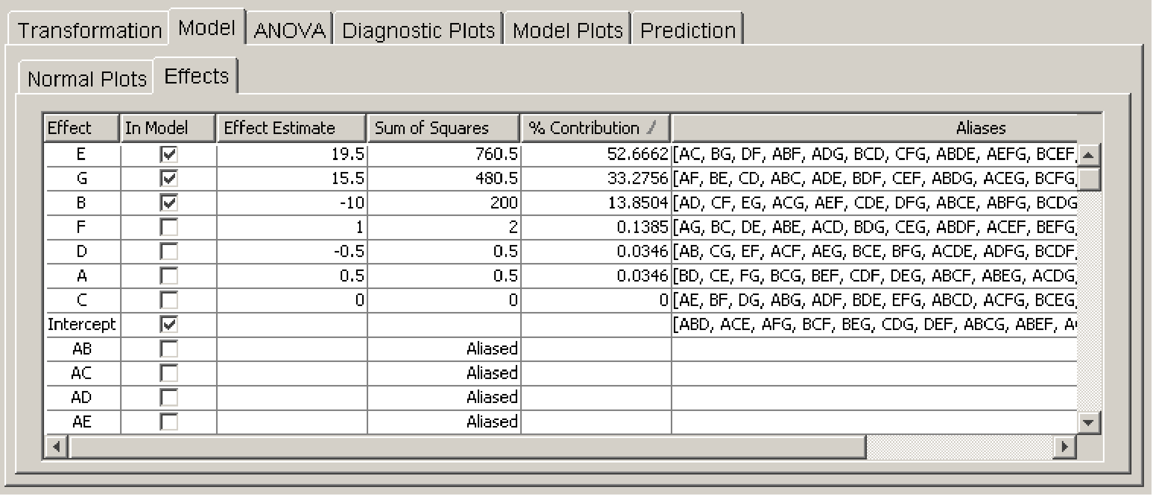


Figure 4.17: Half normal effects plot

The user can also select effects via the table in the Effects tab panel. Figure 4.18 shows the effects table corresponding to the data in the normal plots in Figures 4.16 and 4.17. Only the portion of the table listing the nonaliased effects is shown in order to conserve space. The first column lists all effects up to five factor interactions. The user can include in the regression model any effect that is not aliased by clicking on the corresponding checkbox in the In Model column. The next two columns list the effect estimate and sum of squares value for each effect. If an effect is aliased, the word “aliased” appears instead of a sum of squares value. The % Contribution column lists the percentage that the corresponding effect’s sum of squares contributes to the total sum of squares. The elements of the % Contribution column will always add to 100%, except in Plackett-Burman designs in which interactions contribute a nonzero sum of squares. Only main effects are listed for Plackett-Burman designs. The last column lists the aliases of the effects in the first column. Only



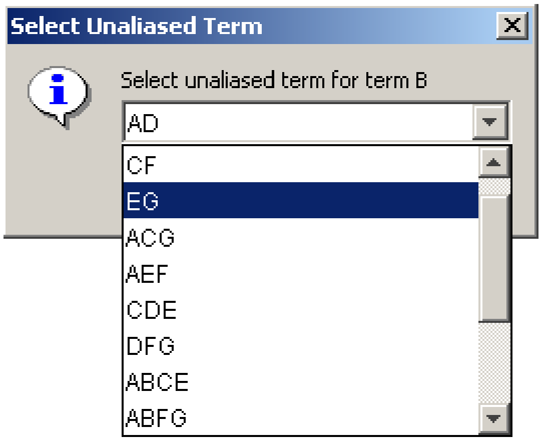
Effect	In Model	Effect Estimate	Sum of Squares	% Contribution	Aliases
E	☑	19.5	760.5	52.6662	[AC, BG, DF, ABF, ADG, BCD, CFG, ABDE, AEFG, BCEF,
G	☑	15.5	480.5	33.2756	[AF, BE, CD, ABC, ADE, BDF, CEF, ABDG, ACEG, BCFG,
B	☑	-10	200	13.8504	[AD, CF, EG, ACG, AEF, CDE, DFG, ABCE, ABFG, BCDG
F	☐	1	2	0.1385	[AG, BC, DE, ABE, ACD, BDG, CEG, ABDF, ACEF, BEFG,
D	☐	-0.5	0.5	0.0346	[AB, CG, EF, ACF, AEG, BCE, BFG, ACDE, ADFG, BCDF,
A	☐	0.5	0.5	0.0346	[BD, CE, FG, BCG, BEF, CDF, DEG, ABCF, ABEG, ACDG,
C	☐	0	0	0	[AE, BF, DG, ABG, ADF, BDE, EFG, ABCD, ACFG, BCEG,
Intercept	☑				[ABD, ACE, AFG, BCF, BEG, CDG, DEF, ABCG, ABEF, A
AB	☐		Aliased		
AC	☐		Aliased		
AD	☐		Aliased		
AE	☐		Aliased		

Figure 4.18: Factorial effects table

aliases for up to five factor interactions are listed. A complete list of the alias structure can be found in the Summary tree node.

Each column in the effects table is sortable. Clicking on any column header will sort the list in descending order. Clicking again will sort in ascending order. The standard procedure is to click on the % Contribution column header to bring the effects that explain the most observed variability of the response to the top of the column. Effects with larger percent contributions are more likely to be significant regression model terms. Figure 4.18 shows that the top three effects, E, G, and B, are selected for inclusion in the regression model. Those effects were also selected in the normal plots in Figures 4.16 and 4.17. Selecting an effect from the effects table is equivalent to selecting the same effect from the normal plot.

The effects table in Figure 4.18 shows that effects E and G explain 52.7% and 33.3% of the observed variability of the response. Suppose that experience with the system being modeled indicates that the interaction EG should exert a strong effect on the response. Suppose also that factor B is not expected to be influential on the response. However, according to the effects table, B explains 13.9% of the observed variability. Careful inspection of the aliases for effect B reveals that B is aliased to EG (and several other interactions). This observation would lead the experimenter to conclude that the 13.9% variability is due to interaction EG, not factor B. Inclusion of factor B in the regression model would lead to



Select Unaliased Term

Select unaliased term for term B

AD

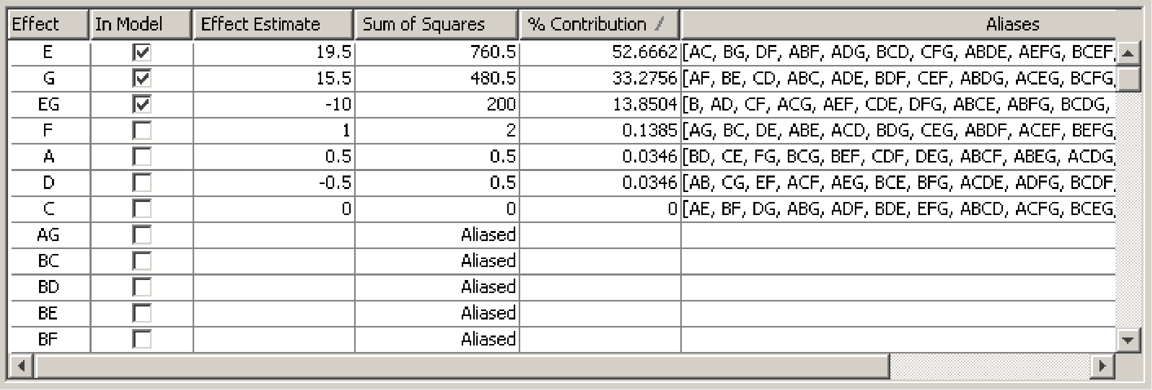
CF
EG
ACG
AEF
CDE
DFG
ABCE
ABFG

Figure 4.19: Edit aliases dialog

invalid conclusions. Fortunately, there is a way to alter the alias arrangement in the effects table so that the correct effect can be added to the regression model.

Right-clicking on any row of the Effect or Aliases column activates a popup menu allowing the aliases for the row's effect to be edited. Figure 4.19 shows the dialog that appears after the user has selected the popup menu item to edit the aliases for effect B. The popup shows a pull-down menu listing the aliases of B. After term EG has been selected, the effects table is updated as shown in Figure 4.20. Factors E, G, and their interaction EG will then be used to build the regression model.

Regression models in the DOE study editor are required to be *hierarchical*. In a hierarchical model, all parent effects for all interactions in the model must also be included in the model. For example, if interaction ABC is included in a regression model, the parent



Effect	In Model	Effect Estimate	Sum of Squares	% Contribution	Aliases
E	☑	19.5	760.5	52.6662	[AC, BG, DF, ABF, ADG, BCD, CFG, ABDE, AEFG, BCEF,
G	☑	15.5	480.5	33.2756	[AF, BE, CD, ABC, ADE, BDF, CEF, ABDG, ACEG, BCFG,
EG	☑	-10	200	13.8504	[B, AD, CF, ACG, AEF, CDE, DFG, ABCE, ABFG, BCDG,
F	□	1	2	0.1385	[AG, BC, DE, ABE, ACD, BDG, CEG, ABDF, ACEF, BEFG,
A	□	0.5	0.5	0.0346	[BD, CE, FG, BCG, BEF, CDF, DEG, ABCF, ABEG, ACDG,
D	□	-0.5	0.5	0.0346	[AB, CG, EF, ACF, AEG, BCE, BFG, ACDE, ADFG, BCDF,
C	□	0	0	0	[AE, BF, DG, ABG, ADF, BDE, EFG, ABCD, ACFG, BCEG,
AG	□		Aliased		
BC	□		Aliased		
BD	□		Aliased		
BE	□		Aliased		
BF	□		Aliased		

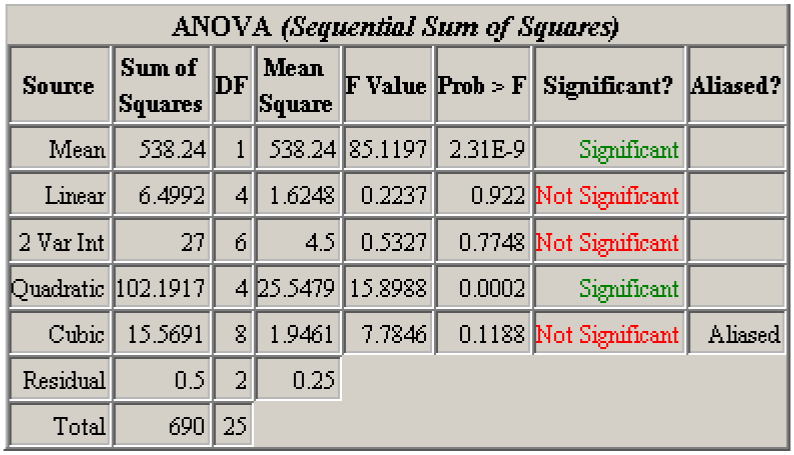
Figure 4.20: Updated effects table after aliases have been edited

effects A, B, C, AB, AC, and BC must also be included. Möbius automatically includes the terms necessary to make the selected model hierarchical. This automatic update is performed whenever the user switches to a different view in the main panel. Clicking from the Effects tab to the Normal Plots tab will automatically cause all parent effects to be added to the selected regression model.

4.9.2 Response surface model selection

The method of effect selection for response surface designs differs from the factorial method. There is no graphical procedure for selecting effects for response surface designs. Under the Model tab in response surface designs, there are two tabs to choose from: the Fit Summary and Model Selection tabs.

The fit summary panel summarizes the fit of several regression models to the observed response values. An ANOVA table is used to calculate the significance of each model. Figure 4.21 presents an example ANOVA table for five models. The mean regression model contains only the average response value. If β_0 is the average, then the model is simply $y = \beta_0$. The linear model contains the mean and all main effects. The two-variable interaction model contains all effects from the mean and linear models, plus all two-factor interactions. Each successive model adds more terms to the model. The quadratic model adds pure second-degree effects (A^2, B^2, C^2…). The cubic model adds all pure cubics (A^3, B^3, C^3…), all mixed terms (A^2B, A^2C, AB^2…), and all three-factor interactions (ABC). The



ANOVA *(Sequential Sum of Squares)*

Source	Sum of Squares	DF	Mean Square	F Value	Prob > F	Significant?	Aliased?
Mean	538.24	1	538.24	85.1197	2.31E-9	Significant	
Linear	6.4992	4	1.6248	0.2237	0.922	Not Significant	
2 Var Int	27	6	4.5	0.5327	0.7748	Not Significant	
Quadratic	102.1917	4	25.5479	15.8988	0.0002	Significant	
Cubic	15.5691	8	1.9461	7.7846	0.1188	Not Significant	Aliased
Residual	0.5	2	0.25				
Total	690	25					

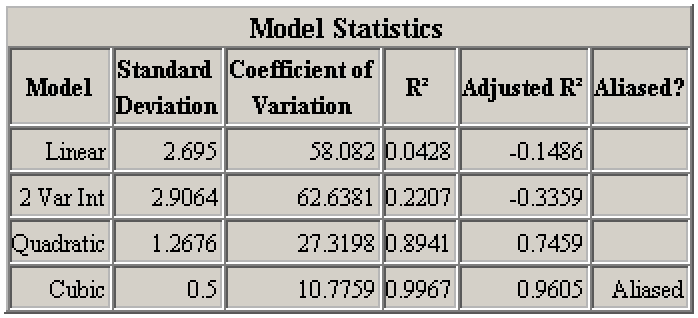
Figure 4.21: ANOVA table in fit summary panel

ANOVA is calculated using the sequential sum of squares method explained in Section 3.5. The Significant column in the ANOVA table tells whether the corresponding model's additional terms added to the sequential model contribute significantly to the observed variability in the response. From Figure 4.21, it is clear that the pure quadratic terms contribute significantly to the regression model. The linear and two-variable interaction models are not significant. The response surface must exhibit second-order characteristics, which are best modeled with quadratic regression polynomial terms.

A second table of model statistics presents further information to help the user choose the appropriate regression model. Figure 4.22 lists useful statistics for each of the models presented in the ANOVA table in Figure 4.21. The Adjusted R^2 column is the most important. The table exhibits the fact that the R^2 value always increases as terms are added to the regression model, but the adjusted R^2 can actually decrease. A negative adjusted R^2 for the linear and two-variable interaction models indicates that they are not adequate models of the response. The quadratic model explains 74.59% of the response's observed variability, according to the adjusted R^2 value.

In both example fit summary tables, the cubic model is *aliased*. The concept of an aliased model in response surface methodology is similar to that in factorial designs. An aliased model is one in which there have not been enough experiments performed to estimate all of the effects in the model. In order to *de-alias* a response surface model, it is necessary to add more experiments in the design panel, as explained in Section 4.6.

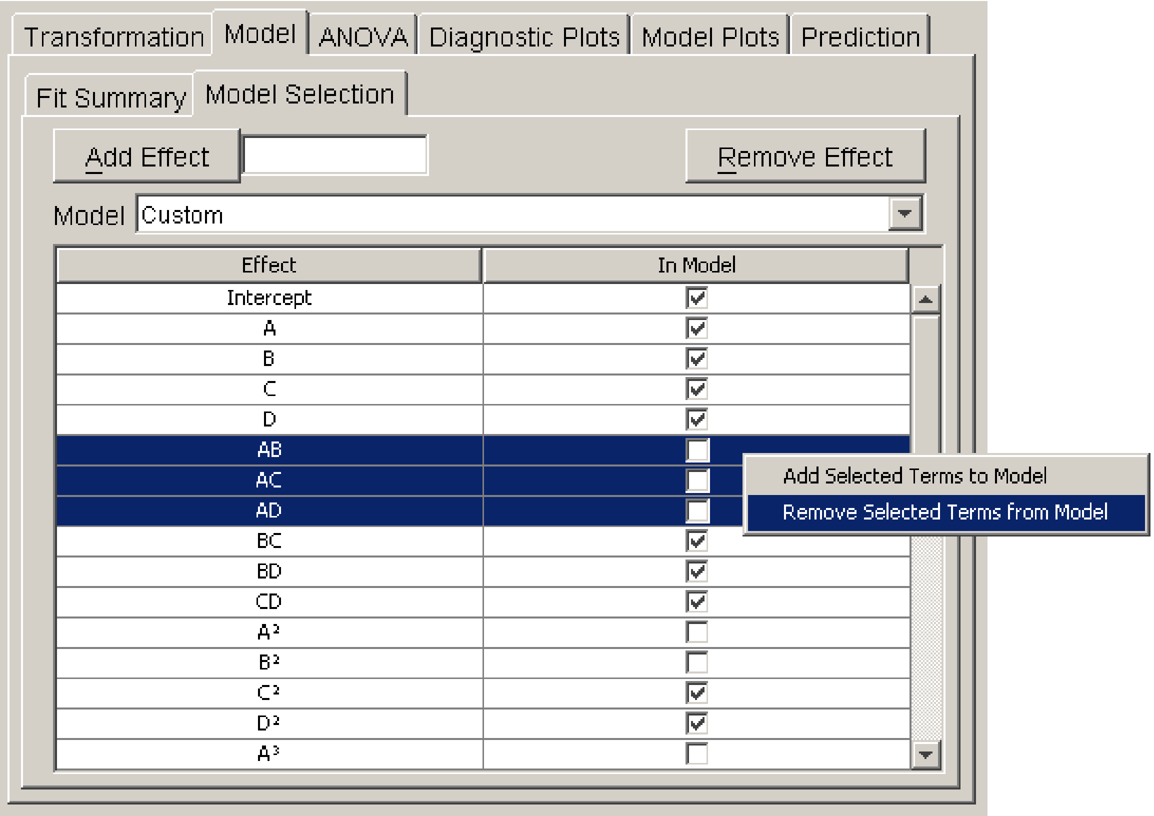
The last piece of information presented in the fit summary panel is the sum of squares details for the ANOVA table. Each effect added to the sequential model has a corre-



Model Statistics

Model	Standard Deviation	Coefficient of Variation	R²	Adjusted R²	Aliased?
Linear	2.695	58.082	0.0428	-0.1486	
2 Var Int	2.9064	62.6381	0.2207	-0.3359	
Quadratic	1.2676	27.3198	0.8941	0.7459	
Cubic	0.5	10.7759	0.9967	0.9605	Aliased

Figure 4.22: Fit summary model statistics table



Model Custom

Effect	In Model
Intercept	☑
A	☑
B	☑
C	☑
D	☑
AB	□
AC	□
AD	□
BC	☑
BD	☑
CD	☑
A^2	□
B^2	□
C^2	☑
D^2	☑
A^3	□

Figure 4.23: Model selection effects table

sponding sum of squares term. There are two separate sum of squares lists. The first is sorted by effect name; the second is sorted in descending order by sum of squares magnitude. The most influential effects will be placed at the top of the second list. It is important to remember that for sequential sum of squares, the order in which effects are added to the model may affect the magnitude of the sum of squares. The sorted sum of squares list is useful for selecting potentially highly significant effects for inclusion in the regression model.

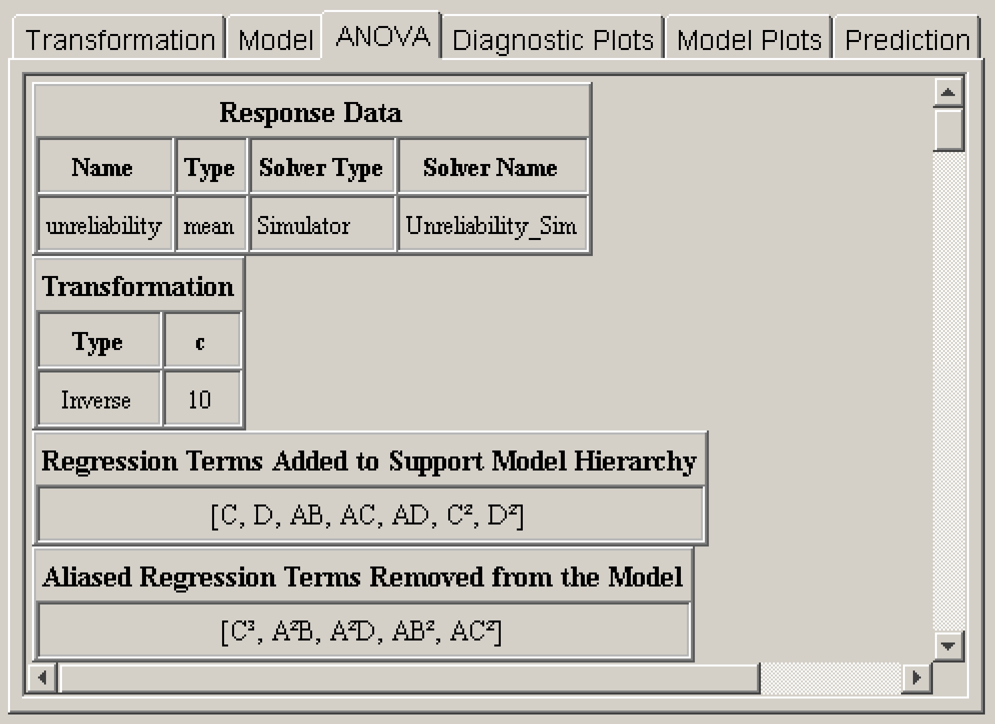
Effects are added to the regression model in the model selection panel. The table in Figure 4.23 lists each effect in the first column and a corresponding checkbox in the second column. A single mouse click on the checkbox toggles the effect's inclusion in the model. When any row is selected, the user can activate a popup menu by right-clicking anywhere on the table. The popup menu, shown in Figure 4.23, contains items for removing all selected effects from or adding all selected effects to the regression model. An alternative to clicking directly on the table to select effects is to use the Model pull-down menu above the table. From the pull-down menu, the mean, linear, quadratic, cubic, or other model may be selected.

By default, the highest-degree effect listed in the model selection panel is three. The user can alter this maximum degree by selecting a different maximum degree from the preferences dialog. The choices for the maximum degree are two through six. Another way to add a specific higher-order effect to the table is to type the desired effect in the text field at the top of the model selection panel and click the Add Effect button. For example, if effect A^5B is desired, the user may type any of "A^5B", "a^5b", "A5B", or "a5B" and push the Add Effect button to place A^5B in the table. All parent effects required to support hierarchy are automatically added to the table. Therefore, adding A^5B will automatically add A^4, A^3B, A^5, and A^4B to the table if they are not already there. The maximum degree effect allowed is six. For example, AB^2C^3, ABCDEF, and A^6 are all sixth-degree effects. To remove any user-added effects, the user can press the Remove Effect button; a dialog will prompt him or her to select from a list of added effects. The selected effect and all parent effects required for hierarchy are then removed from the table.

By default, no exponents of degree four or higher are displayed in superscript font. If the computer system running the Möbius interface has the superscript fonts installed, then the user can activate their display by clicking the "Use Exponent Font in Effects" checkbox in the preferences dialog. Selecting this option also affects the display of exponents in the fit summary and ANOVA panels.

4.10 ANOVA

After the effects for the regression model have been selected, a statistical analysis can be performed to evaluate the quality of the model. Clicking on the ANOVA tab reveals a wide variety of statistical output about the regression model. Figure 4.24 shows the first four pieces of information from an example response surface design. The first two items summarize the response and transformation information. The next two items do not always appear. If effects had to be automatically added to the regression model in order to make it hierarchical, then a list of those effects will be presented below the transformation data, as presented in Figure 4.24. The final piece of data applies only to response surface designs. It



Response Data			
Name	Type	Solver Type	Solver Name
unreliability	mean	Simulator	Unreliability_Sim

Transformation	
Type	c
Inverse	10

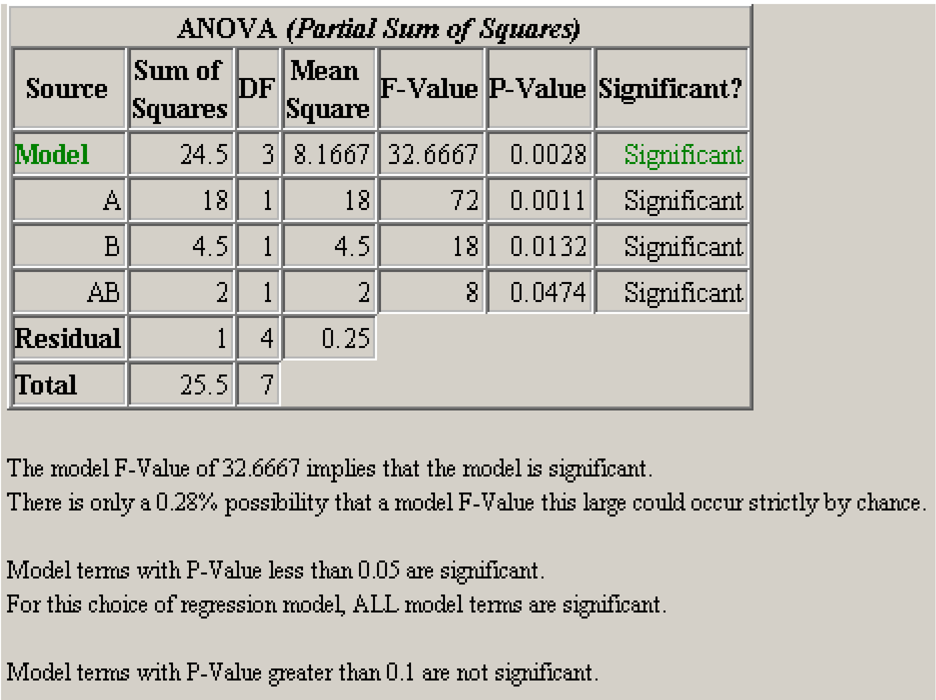
Regression Terms Added to Support Model Hierarchy
[C, D, AB, AC, AD, C^2, D^2]

Aliased Regression Terms Removed from the Model
[C^3, A^2B, A^2D, AB^2, AC^2]

Figure 4.24: Initial data in ANOVA panel

is possible in the model selection panel to select more effects than can be independently estimated given the available degrees of freedom. If this happens, some of the selected effects will be aliased to other effects in the model. As effects are sequentially added to the regression model from lower to higher order, any effect that is aliased to an effect already in the model is not actually added to the model. Instead, it is placed in the list of aliased terms presented in the ANOVA panel. The alias structure is dependent on which effects are already in the model. Therefore, the complete alias structure for the selected model is presented in the ANOVA panel *after* the entire model is selected.

The next item in the ANOVA panel is the analysis of variance table. Figure 4.25 shows a screen capture of the ANOVA table for the workstation model presented in Table 3.7. The significance of the model and each effect should be analyzed. If an effect is not significant and it is not required to support model hierarchy, then it should be removed from the model. The significance and insignificance thresholds are defined in the preferences dialog. The default thresholds are 0.05 for significant and 0.1 for insignificant. Any effect with a P-Value less than the significance threshold is significant. Any effect with a P-Value



ANOVA *(Partial Sum of Squares)*

Source	Sum of Squares	DF	Mean Square	F-Value	P-Value	Significant?
Model	24.5	3	8.1667	32.6667	0.0028	Significant
A	18	1	18	72	0.0011	Significant
B	4.5	1	4.5	18	0.0132	Significant
AB	2	1	2	8	0.0474	Significant
Residual	1	4	0.25			
Total	25.5	7				

The model F-Value of 32.6667 implies that the model is significant.
There is only a 0.28% possibility that a model F-Value this large could occur strictly by chance.

Model terms with P-Value less than 0.05 are significant.
For this choice of regression model, ALL model terms are significant.

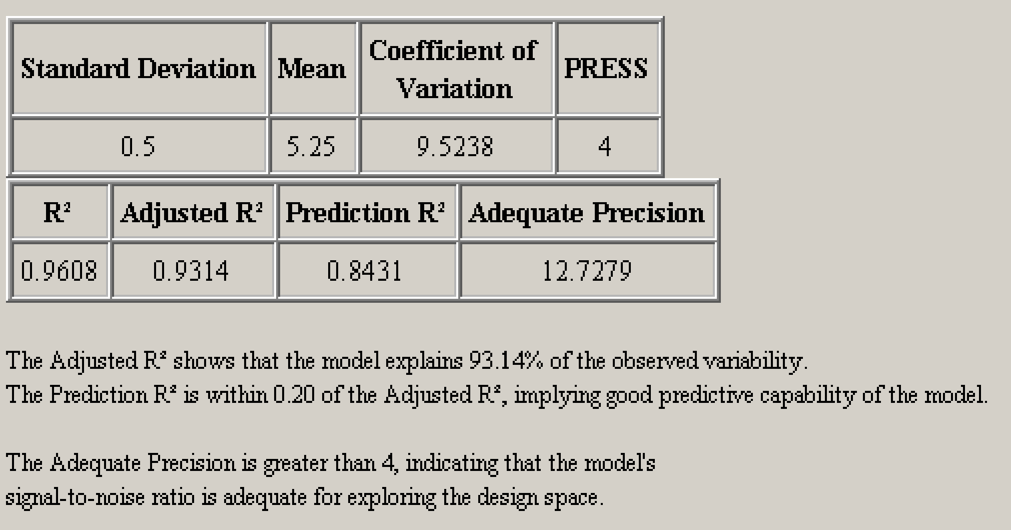
Model terms with P-Value greater than 0.1 are not significant.

Figure 4.25: ANOVA table with summary

greater than the insignificance threshold is not significant. The significance of an effect with a P-Value between the two thresholds is left to the experimenter to decide.

A summary of the ANOVA results is presented after the ANOVA table. The summary in Figure 4.25 gives an interpretation of the model P-Value. Multiplying the model P-Value by 100 gives the percent possibility that the given significance level could occur strictly by chance. A summary of which terms are significant and not significant is also given.

After the ANOVA summary, there are two tables of regression model statistics. The theory behind these statistics was presented in Section 3.6. Figure 4.26 shows a sample with a summary of the data below the tables. The adjusted R^2 should be as close to one as possible. The model should explain as much variability as possible without insignificant effects. The prediction capability of the regression model is best when the adjusted R^2 and prediction R^2 are as close to each other as possible. If the two are within 0.20 of one another, then the predictive quality is acceptable.



Standard Deviation	Mean	Coefficient of Variation	PRESS
0.5	5.25	9.5238	4

R²	Adjusted R²	Prediction R²	Adequate Precision
0.9608	0.9314	0.8431	12.7279

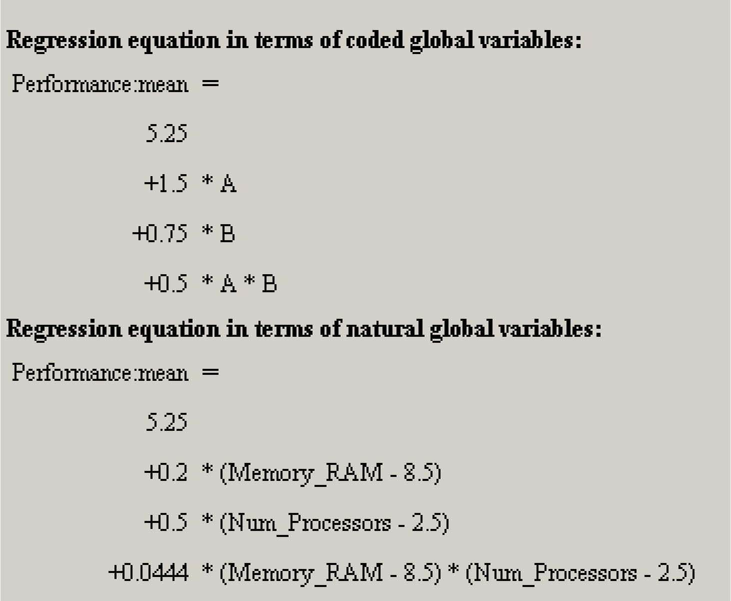
The Adjusted R² shows that the model explains 93.14% of the observed variability.
The Prediction R² is within 0.20 of the Adjusted R², implying good predictive capability of the model.

The Adequate Precision is greater than 4, indicating that the model's
signal-to-noise ratio is adequate for exploring the design space.

Figure 4.26: Regression model statistics table

A table of diagnostic statistics for each experiment is presented next. The data in each column is presented graphically in the Diagnostic Plots panel. The tabular presentation is an alternative to the scatter plots and is preferable for some users. For experiments with leverage values of one, the student's residuals, Cook's distance, and outlier t are undefined.

Below the diagnostic statistics table are two representations of the regression model polynomial. Figure 4.27 shows both representations. Each polynomial is written with one



Regression equation in terms of coded global variables:

```
Performance:mean  =
             5.25
             +1.5  * A
            +0.75  * B
             +0.5  * A * B
```

Regression equation in terms of natural global variables:

```
Performance:mean  =
             5.25
             +0.2  * (Memory_RAM - 8.5)
             +0.5  * (Num_Processors - 2.5)
          +0.0444  * (Memory_RAM - 8.5) * (Num_Processors - 2.5)
```

Figure 4.27: Regression equations in coded and natural formats

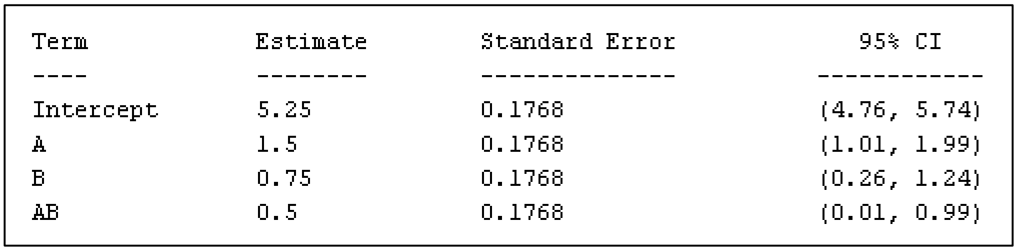
term per line to facilitate formatting. For example, the first equation in Figure 4.27 is

$$\textit{Performance:mean} = 5.25 + 1.5\mathrm{A} + 0.75\mathrm{B} + 0.5\mathrm{AB}$$

The first equation is presented in *coded* global variable terms. As explained in Section 2.4.5, the coded variables are the natural values scaled from –1 to +1. They are the values used for internal calculations. The equation in terms of natural global variable values is the one used to predict response values. The first item in the Design menu allows the natural form of the regression equation to be exported to a file. The file may be in either plain text or *Mathematica* notebook format.

The final piece of data presented in the ANOVA panel is a table of standard errors and confidence intervals for each regression model coefficient. Figure 4.28 shows an example table. The standard error is the standard deviation of the coefficient estimate. The 95% confidence intervals represent the range in which the coefficients should lie 95% of the time. If the range contains zero, then the coefficient is not significant.

The area of the ANOVA panel containing the table of standard errors is an editable text area. Comments can be entered in the text area. The entire ANOVA panel can be exported to an HTML file using the Design menu's Export Text item. A plain-text version of the ANOVA table can also be appended to the ANOVA panel output if the appropriate checkbox is selected in the preferences dialog. The plain text version is useful for copying and pasting text into the notes panel or to another application.

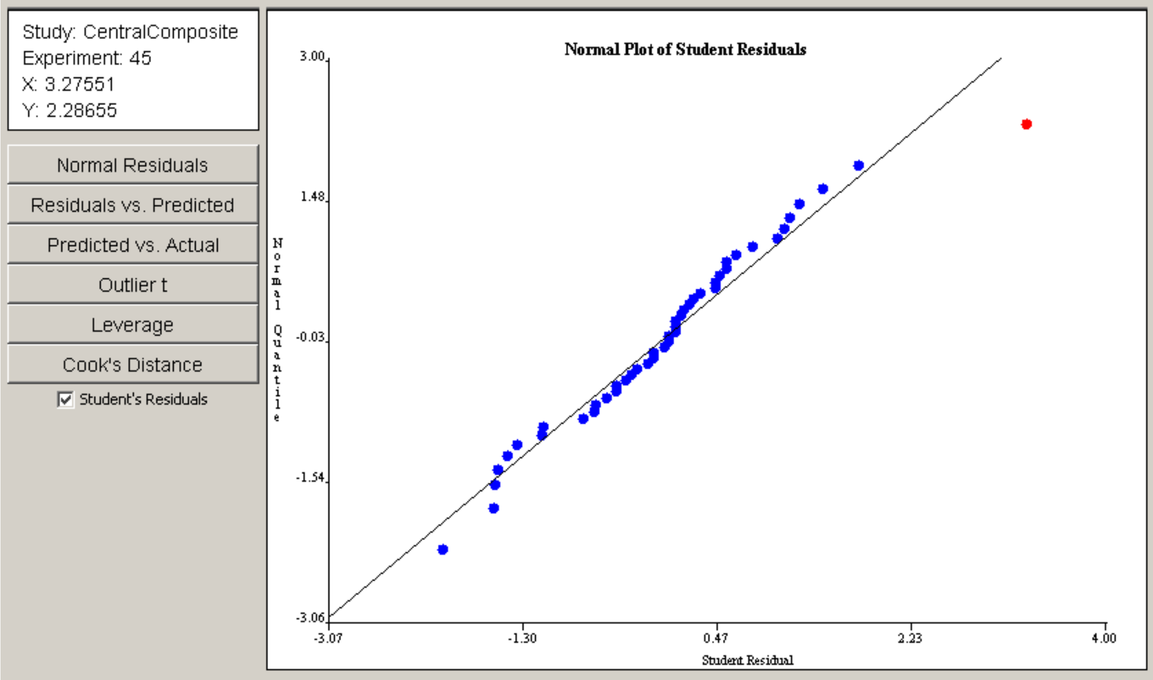


```
Term            Estimate        Standard Error            95% CI
----            --------        --------------          ------------
Intercept       5.25            0.1768                  (4.76, 5.74)
A               1.5             0.1768                  (1.01, 1.99)
B               0.75            0.1768                  (0.26, 1.24)
AB              0.5             0.1768                  (0.01, 0.99)
```

Figure 4.28: Standard errors and confidence intervals for regression coefficients

4.11 Diagnostic Plots

If the regression model produced satisfactory results in the ANOVA panel, the analysis may proceed to the Diagnostic Plots panel. The diagnostic plots consist of six scatter plots of the data from the Diagnostic Statistics table in the ANOVA panel. Six buttons on


Study: CentralComposite
Experiment: 45
X: 3.27551
Y: 2.28655

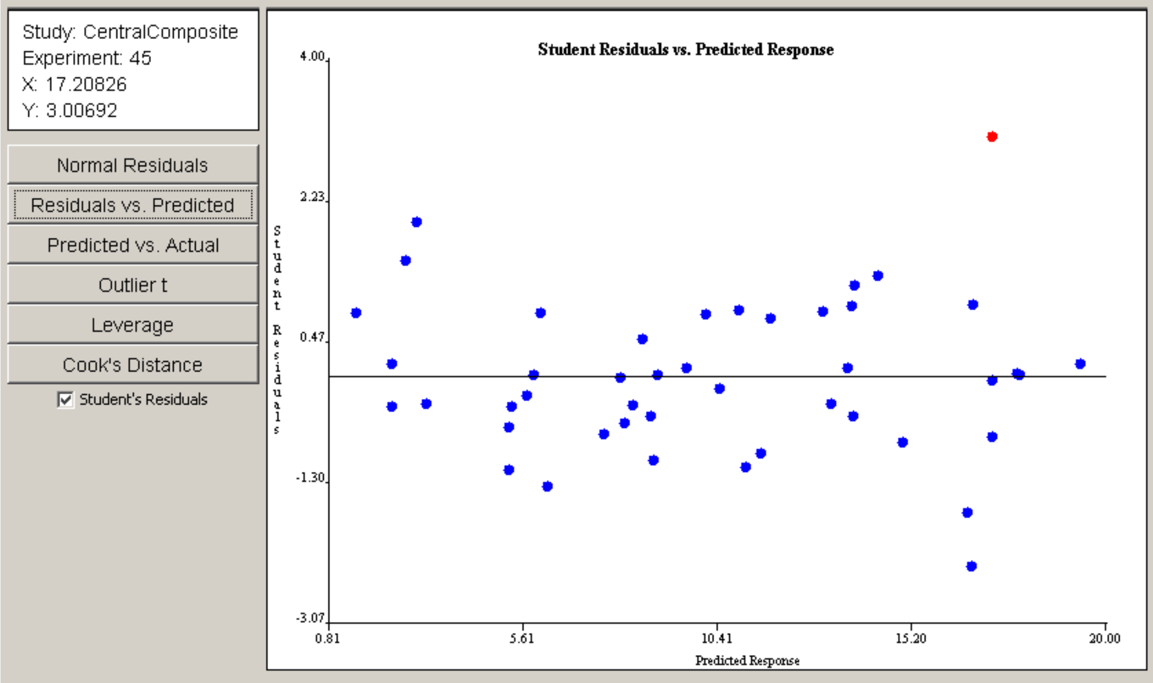
☑ Student's Residuals

Figure 4.29: Normal plot of studentized residuals

the left side of the Diagnostic Plots panel control which of the six plots is displayed. Figure 4.29 shows the panel with an example normal plot of studentized residuals. Each point in the diagnostic plots can be selected with a single mouse click. An information panel above the buttons gives details about the selected point, such as the coordinates and the corresponding experiment number.

The normal plot of studentized residuals in Figure 4.29 shows that the response residuals generally follow a straight line throughout the central portion of the plot. Therefore, it can be assumed that the residuals are normally distributed. The outliers at the lower and upper extremes are not sufficient grounds for dismissal of the normality assumption. The outlier in the upper right portion of the plot is selected in red. The information panel reveals that the selected outlier corresponds to experiment 45.

Clicking on the Residuals vs. Predicted button reveals the plot in Figure 4.30. With the exception of the selected outlier, no visible trend is present. Therefore, the assumption of constant variance of residuals is valid. As in the previous plot, the outlier corresponds to experiment 45. A checkbox below the six plot selection buttons toggles the display of studentized residuals versus ordinary least squares residuals. The studentized form of residuals

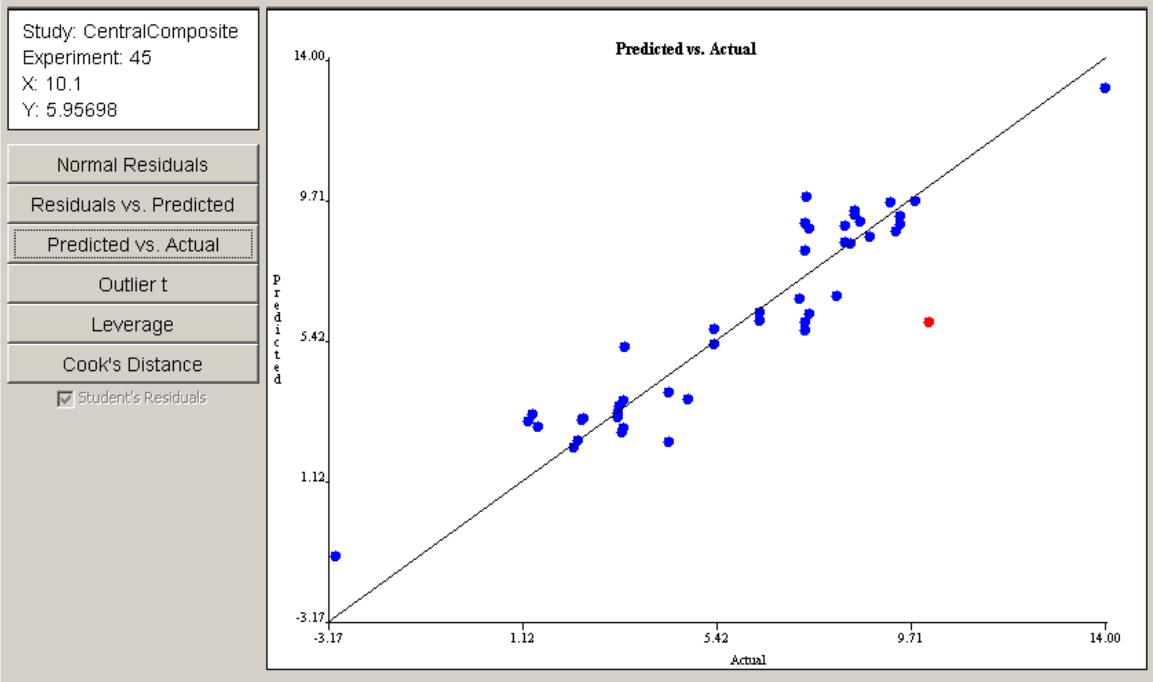


Study: CentralComposite
Experiment: 45
X: 17.20826
Y: 3.00692

Figure 4.30: Student residuals vs. predicted response diagnostic plot

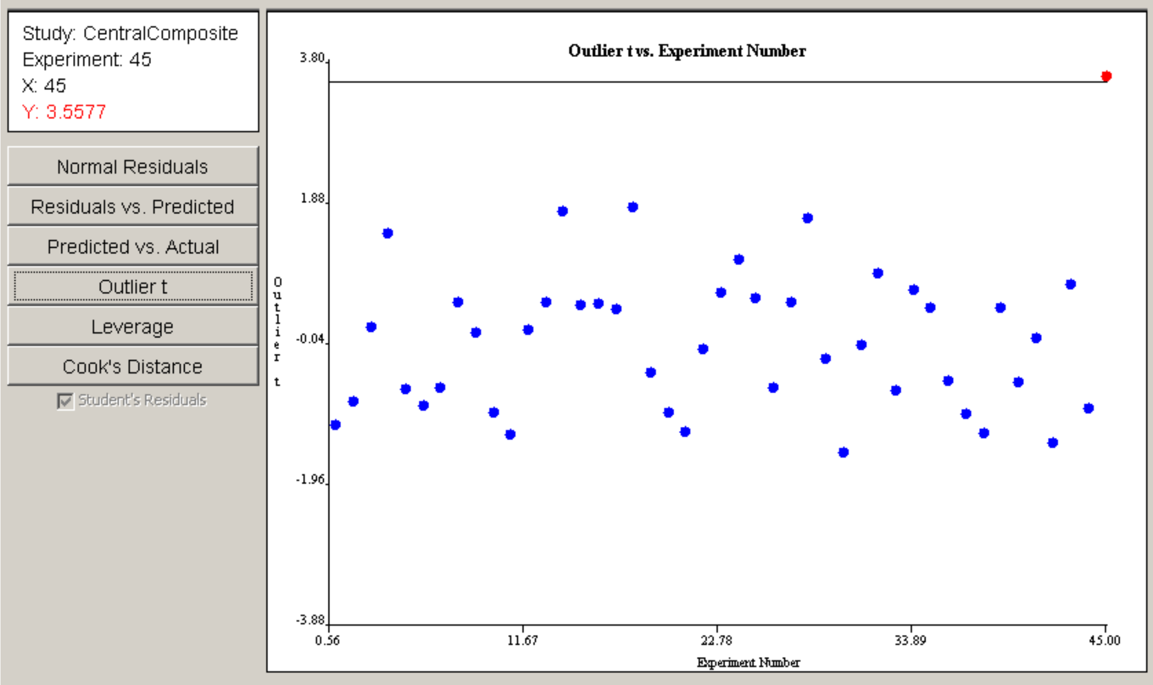
is preferred. However, in some cases, the studentized residuals are not defined. In those cases, the ordinary residuals must be used to validate the assumptions.

The Predicted vs. Actual plot in Figure 4.31 shows that most points are gathered near



Study: CentralComposite
Experiment: 45
X: 10.1
Y: 5.95698

Figure 4.31: Predicted vs. actual diagnostic plot



Study: CentralComposite
Experiment: 45
X: 45
Y: 3.5577

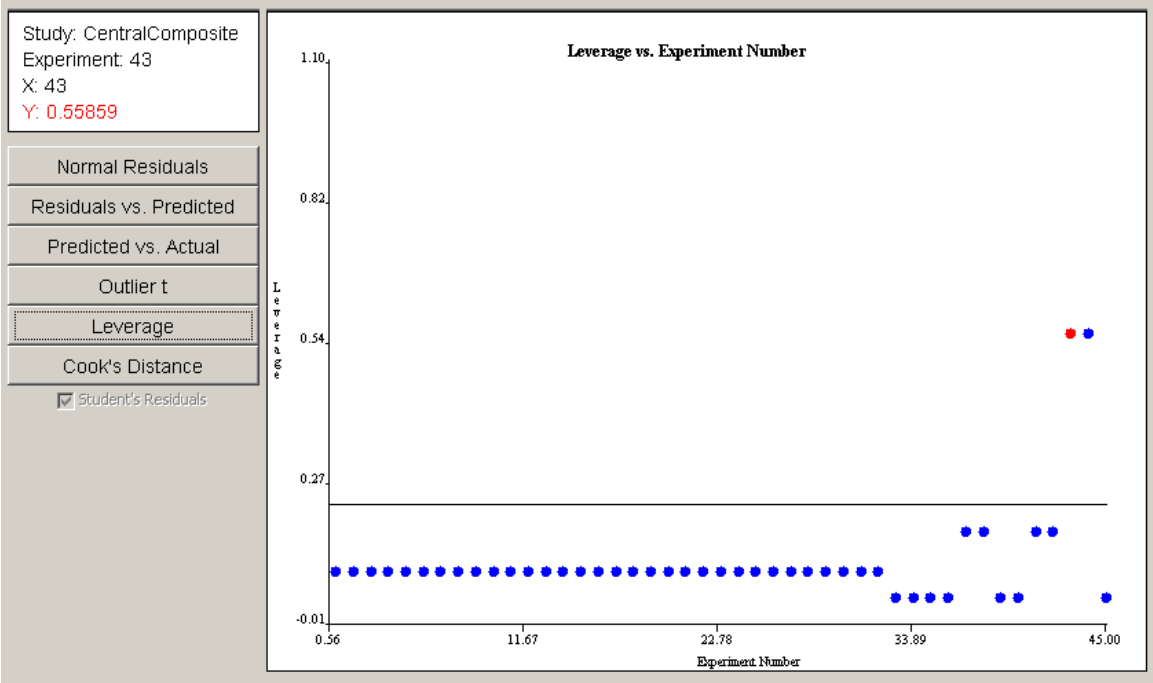
Figure 4.32: Outlier t diagnostic plot

the 45° line, implying a fairly good least squares fit. The point farthest from the line is selected in red and, again, corresponds to experiment 45. The data gathered from the first three plots indicate that experiment 45 may have a problem.

The outlier t plot in Figure 4.32 shows that experiment 45 is indeed an outlier. The information panel indicates in red type that the value of the outlier t for experiment 45 is barely larger than 3.5, making it an outlier. Further investigation by the experimenter would be required in order to either justify or discount the data from the outlying experiment.

The leverage plot in Figure 4.33 shows a pair of points with leverage values greater than twice the average leverage value. These two points exert a disproportionately large influence on the model fit. The Cook's distance plot in Figure 4.34 does not reveal any evidence of disproportionately large influence among the experimental points. Any points that cross the threshold lines in the leverage or Cook's distance plots should be investigated.

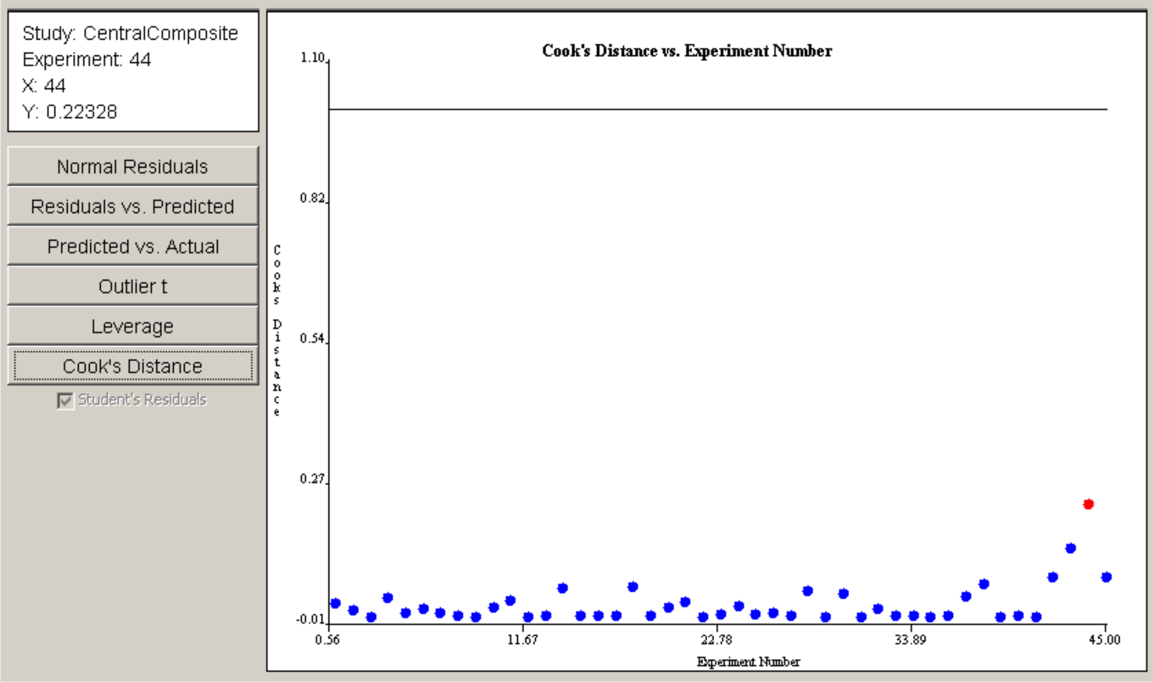
If all diagnostic plots look good, then the analysis may proceed to the inspection of the regression model graphs. If any of the diagnostic plots reveal a problem, then the analysis should not proceed to the next step. Instead, transformations, model reduction, or addi-



Study: CentralComposite
Experiment: 43
X: 43
Y: 0.55859

Figure 4.33: Leverage diagnostic plot

tional experiments should be considered. Conclusions from the Model Plots and Prediction phases of the analysis are valid *only* if all diagnostic tests show satisfactory results.

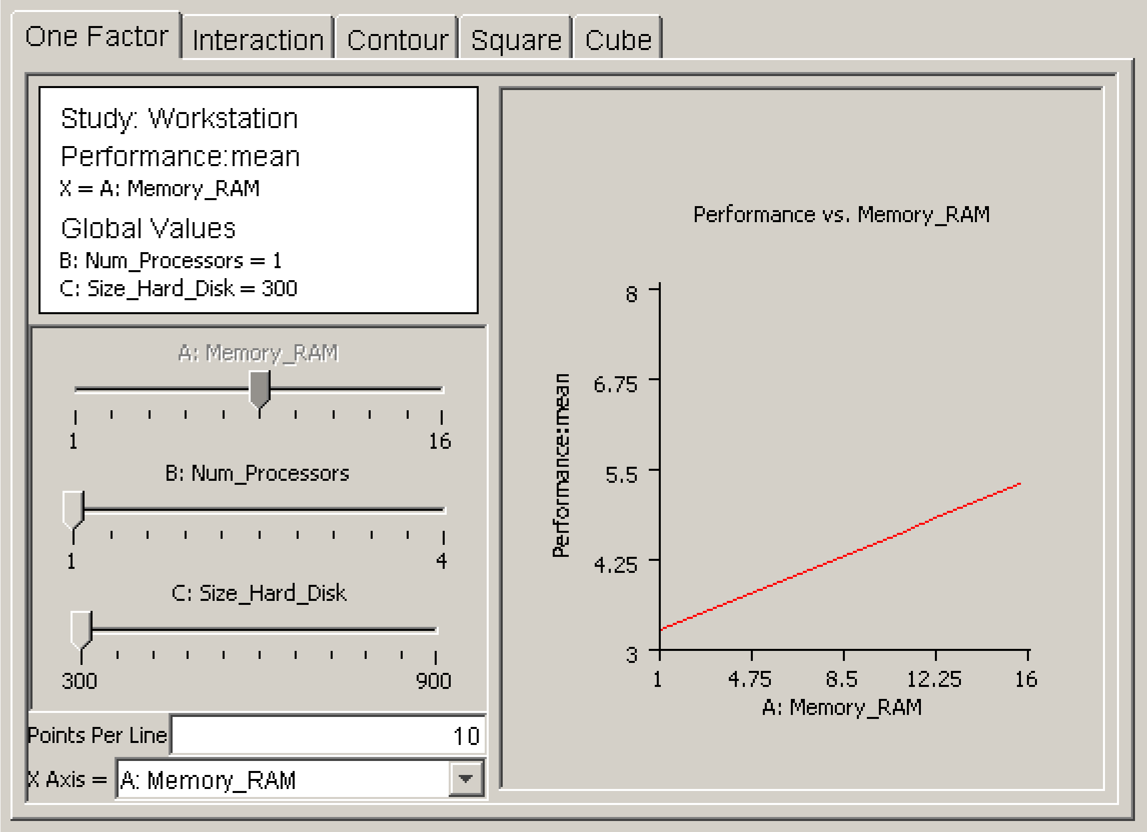


Study: CentralComposite
Experiment: 44
X: 44
Y: 0.22328

Figure 4.34: Cook's distance diagnostic plot

4.12 Model Plots

After verifying that the regression model is adequate for prediction, the user can use plots of the regression model to analyze response behavior. There are five different types of plots available in the model plots panel. The user selects the type of plot by clicking on the appropriate tab within the model plots panel. All of the plot panels have the same layout. Figure 4.35 shows the layout of the one-factor plot panel. The plot occupies the right side of the panel. The left middle portion of the panel contains slider controls for each global variable in the regression model. A slider can be enabled or disabled. If a slider is disabled, it will appear *dimmed*. In Figure 4.35, the slider for global variable A:Memory_RAM is disabled, but the other two sliders are enabled. A slider becomes disabled when its corresponding global variable is selected as an axis label. In Figure 4.35, the horizontal, or x-axis, is labeled as A:Memory_RAM. Axes are selected from pull-down menus at the bottom left



Study: Workstation
Performance:mean
X = A: Memory_RAM
Global Values
B: Num_Processors = 1
C: Size_Hard_Disk = 300

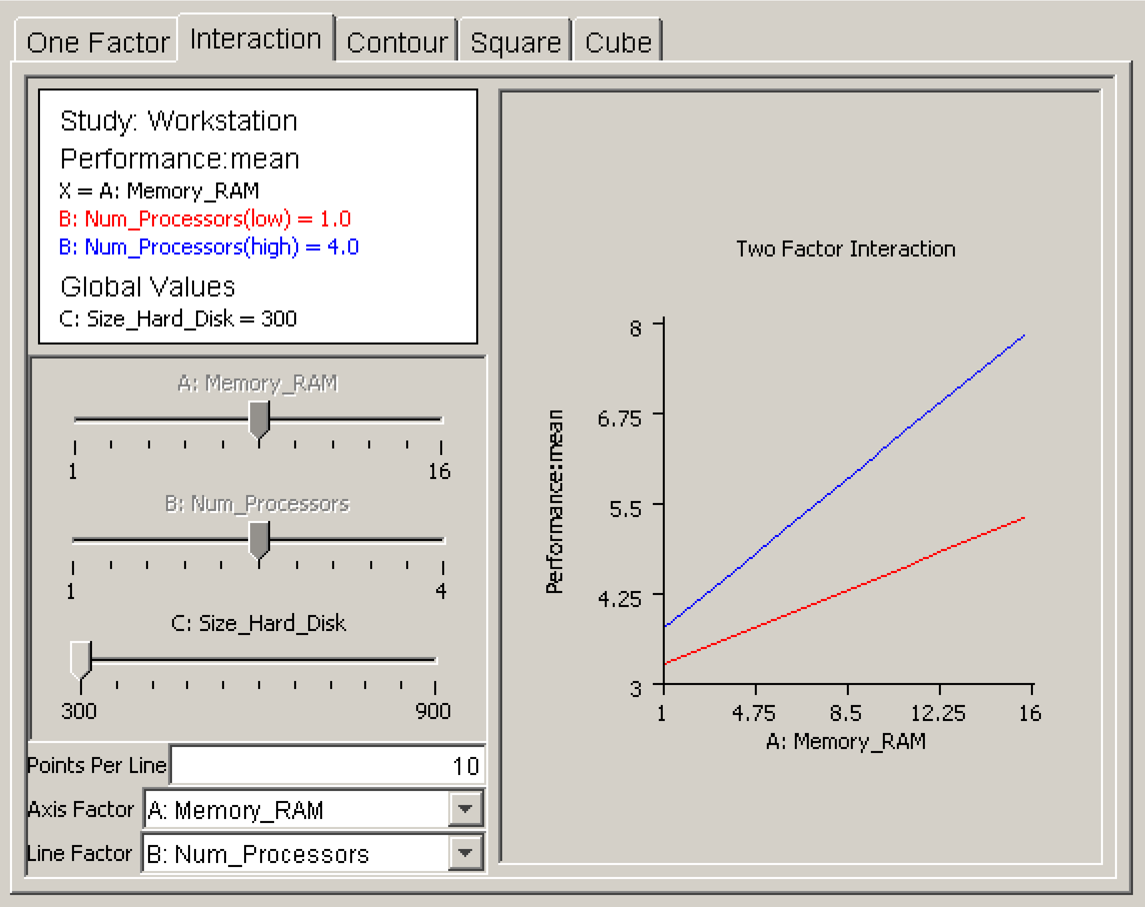
Points Per Line 10
X Axis = A: Memory_RAM

Figure 4.35: One-factor model plot

corner of the main panel. An information panel above the sliders displays the axes labels and the current values of all factors corresponding to all enabled sliders. In Figure 4.35, the values for global variables B and C are listed in the information panel at their lowest values of 1 and 300, which correspond to the selected values of each enabled slider. As the sliders are adjusted, the plot and information panels update immediately. This allows for "real time" feedback of response behavior as the global variable values are varied. The decimal format for the numbers in all model plots can be changed in the preferences dialog. The default number format is #.##, which limits the number of digits to the right of the decimal point to two.

The first type of model plot is the *one-factor* plot, presented in Figure 4.35. The one-factor plot shows how the response behaves as a single global variable is varied from its low to high value. The example plots in this section are from the workstation example presented earlier. Figure 4.35 shows how workstation performance increases as the amount of RAM is increased from 1 to 16 GB, with 1 processor and 300 GB of hard disk space. The Points Per Line text field below the sliders allows the resolution of the plot to be adjusted. All plots are drawn by drawing straight lines between a given number of sample plot points. Increasing the points per line makes the graph *smoother*. For factorial designs, all one-factor plots will be straight lines, so increasing the points per line does not affect the plot. In response surface regression models, global variables can be quadratic and higher-degree, which can add curvature to the one-factor plot.

The *interaction* plot is useful for revealing how regression model global variables act *together* on a response. Figure 4.36 shows an example interaction plot. Two lines of workstation performance are plotted versus the amount of RAM: one line with one processor, the other with four processors. If the number of processors and the amount of RAM do not affect each other, then the lines will be parallel. However, Figure 4.36 reveals that the lines are not parallel, indicating that an interaction is present between the two global variables. Performance increases more sharply with four processors than with one as the RAM is increased from 1 to 16 GB. The hard disk size can be varied by adjusting its slider to see if it affects the interaction.

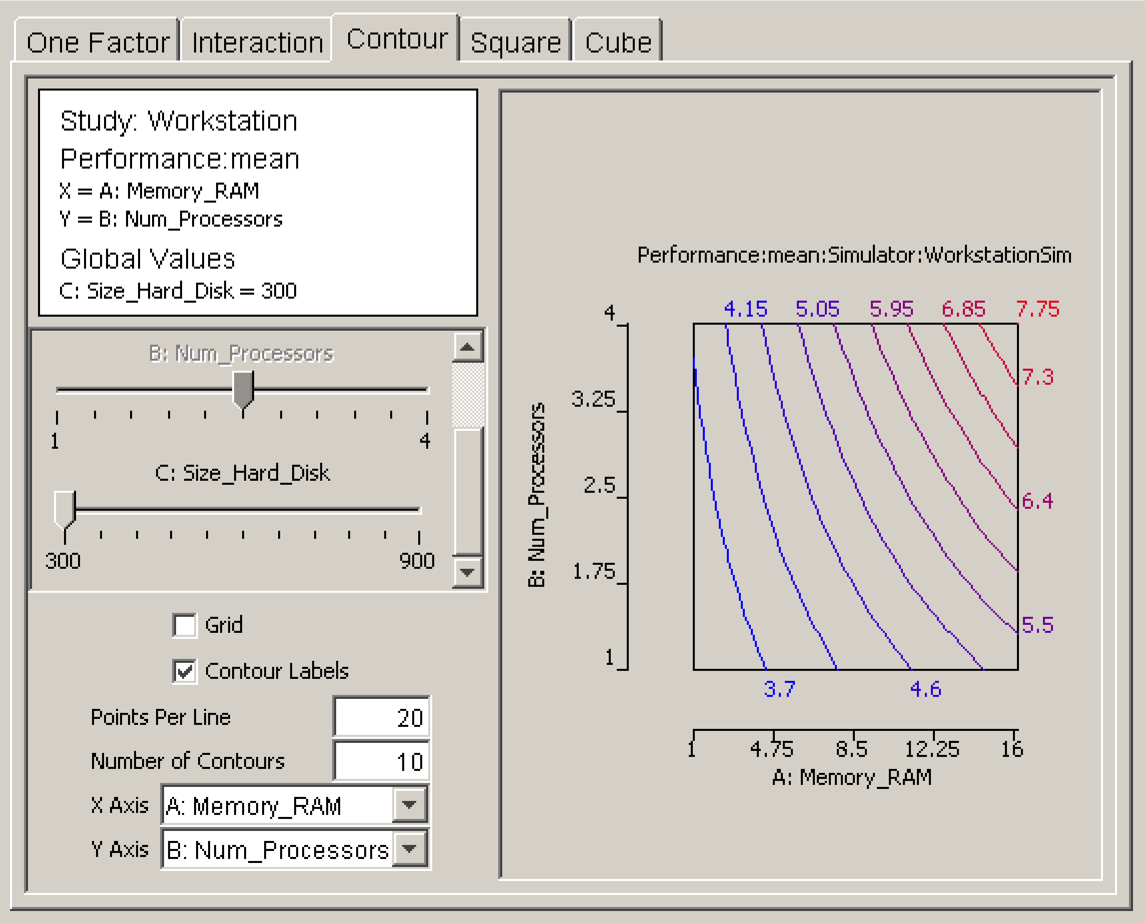


Study: Workstation
Performance:mean
X = A: Memory_RAM
B: Num_Processors(low) = 1.0
B: Num_Processors(high) = 4.0
Global Values
C: Size_Hard_Disk = 300

Points Per Line 10
Axis Factor A: Memory_RAM
Line Factor B: Num_Processors

Figure 4.36: Interaction model plot

If there are *m* factors in the regression model, then the response can be represented as a surface in *m*-space. Any two of the *m* factors can be selected as the axes for a graph of a two-dimensional projection of the response. The *contour plot* helps the user visualize the shape of the response using a two-dimensional projection. Figure 4.37 shows a contour plot of workstation performance versus the number of processors and the amount of RAM. Contours mark lines of constant performance. Each contour line is a different color. A label with the same color indicating the value of the response is placed near each contour line. Figure 4.37 shows how performance increases from 3.7 on the left side of the plot to 7.75 at the upper right corner. Checkboxes for toggling the display of the contour labels and the display of a grid are available. The grid size is determined by the number of points per line. The number of contours can be adjusted by the text field above the axes' pull-down menus.



Study: Workstation
Performance:mean
X = A: Memory_RAM
Y = B: Num_Processors
Global Values
C: Size_Hard_Disk = 300

☐ Grid
☑ Contour Labels

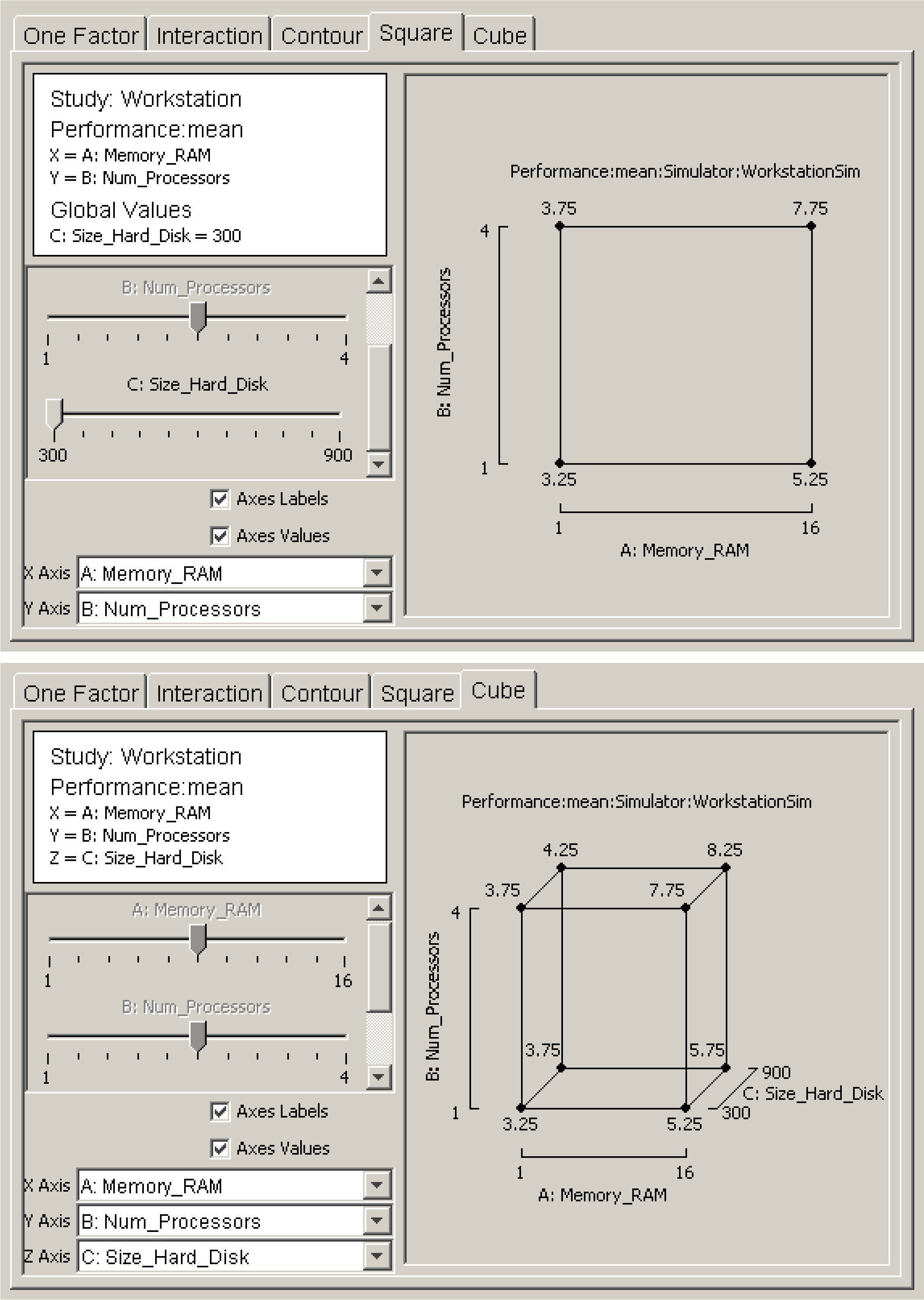
Points Per Line 20
Number of Contours 10
X Axis A: Memory_RAM
Y Axis B: Num_Processors

Figure 4.37: Contour model plot

The last two types of plots are useful visuals for revealing how the response varies as two or three global variables vary from their low to high values. The *square plot* and *cube plot* in Figure 4.38 show how workstation performance varies as each global variable is varied from one extreme to the other. The cube plot indicates that the size of the hard disk has a much smaller effect on performance than the other two global variables do.

4.13 Prediction

If the model plots show that the regression model is acceptable, the model can be used to predict the response at exact points in the design space. The prediction panel provides a method for selecting a point in the design space and calculating the response value, standard errors, confidence intervals, and prediction intervals for the selected point. Figure



Study: Workstation
Performance:mean
X = A: Memory_RAM
Y = B: Num_Processors
Global Values
C: Size_Hard_Disk = 300

☑ Axes Labels
☑ Axes Values

X Axis A: Memory_RAM
Y Axis B: Num_Processors

Study: Workstation
Performance:mean
X = A: Memory_RAM
Y = B: Num_Processors
Z = C: Size_Hard_Disk

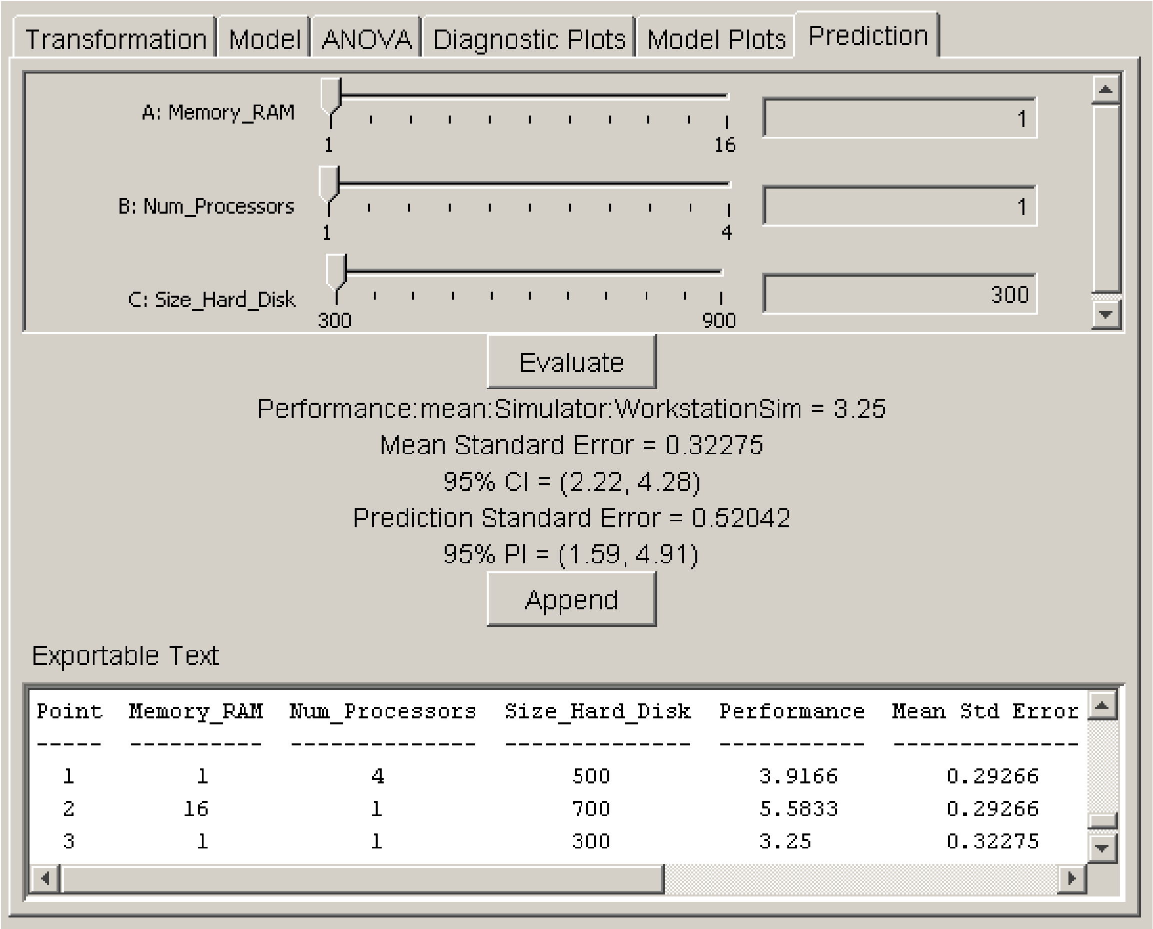
☑ Axes Labels
☑ Axes Values

X Axis A: Memory_RAM
Y Axis B: Num_Processors
Z Axis C: Size_Hard_Disk

Figure 4.38: Square and cube model plots

4.39 shows the interface for selecting the prediction point. Each global variable in the regression model is listed at the top of the prediction panel with a corresponding slider control and text field. The values in the text fields represent the coordinates in the design space. The user can alter the text field values by adjusting the slider controls from the low to high values. Values may also be typed directly into each text field. After entering a value in the text field, the user pushes the Evaluate button to update the labels for the prediction point data.

A text area at the bottom of the prediction panel displays a summary of information about the global variables in the study. Whenever the Append button is pushed, the data for the predicted point is appended to the text area. The text area is editable. The Export Text item in the Design menu allows the entire contents of the text area to be exported to a text file.



A: Memory_RAM 1

B: Num_Processors 1

C: Size_Hard_Disk 300

Performance:mean:Simulator:WorkstationSim = 3.25

Mean Standard Error = 0.32275

95% CI = (2.22, 4.28)

Prediction Standard Error = 0.52042

95% PI = (1.59, 4.91)

```
Point  Memory_RAM  Num_Processors  Size_Hard_Disk  Performance  Mean Std Error
-----  ----------  --------------  --------------  -----------  --------------
  1        1             4               500          3.9166        0.29266
  2       16             1               700          5.5833        0.29266
  3        1             1               300          3.25          0.32275
```

Figure 4.39: Prediction panel

The decimal format can be changed in the preferences dialog. The default format is to display five digits to the right of the decimal point. The confidence levels are based on the significance threshold defined in the preferences dialog.

As stated in Section 3.8, the user should be cautious about predicting response values that are far from the design center. Any global variable value that is outside the original design space will be highlighted in red in the corresponding text field, and any appended text in the text area will be marked with an asterisk.

5. MULTICOMPUTER SYSTEM EXAMPLE

5.1 Problem Description

In this chapter, the DOE study editor will be used to determine the most reliable configuration of a fault-tolerant computer system. The system was first presented in [23]. The complete documentation of the model in Möbius is presented in the Appendix. The system consists of a variable number of redundant computers. Figure 5.1 shows a block diagram of one computer. The four main components of each computer are the CPU, memory, input/output, and error-handling units. Each CPU contains three CPU modules of which one is a spare. There are six nonredundant chips in each CPU module. The memory unit consists of a variable number of memory modules of which one is a spare. Each memory module in the memory unit consists of 41 RAM chips and two interface chips. Two of the 41 RAM chips are redundant. The input/output unit consists of two I/O ports of which one is a spare. There are six nonredundant chips in each I/O port. Neither of the two channels in the error handler component is redundant.

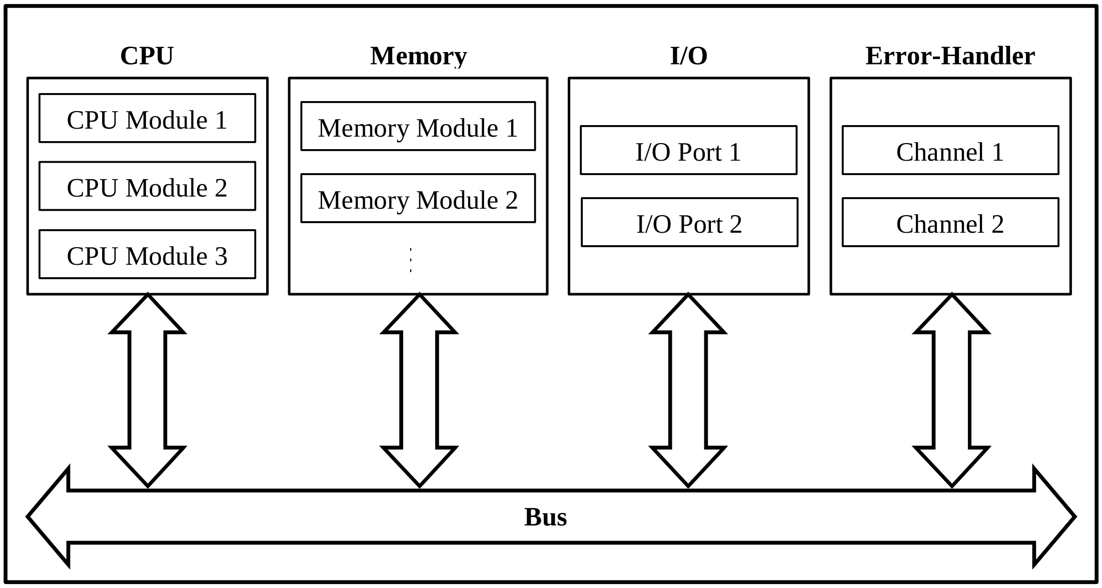


Figure 5.1: Block diagram for one computer in the mulitcomputer example

The multicomputer system is considered to operate successfully if at least one computer is operational. A computer is considered operational if at least two CPU modules, at least two memory modules, at least one I/O port, and both channels of the error-handler are functioning correctly. A memory module is operational if at least 39 of the 41 RAM chips and both interface chips are functioning. For each component that has redundancy, there is an associated coverage factor for that component's failure. For example, if the computer coverage level is 0.95, then a failed computer will be replaced successfully by a spare computer (if available) with probability 0.95. There is also a 0.05 probability that the fault recovery will fail, which may cause the entire multicomputer system to fail if there is not at least one operational computer remaining [23].

Ten global variables parameterize various system factors, such as the number of memory modules, the computer coverage level, and the CPU failure rate. A complete list of all global variables is in the Appendix. The DOE studies presented in this chapter will investigate how the global variables affect system reliability. Specifically, the response under investigation measures the *unreliability* of the system, which is equal to the probability that the system fails during a given time period.

5.2 Screening

First, a 12-experiment Plackett-Burman screening design is run. All global variables are varied, and any main effects that contribute significantly to the observed variability can be identified. Table 5.1 is taken directly from the Möbius Documenter. The table contains a complete listing of all global variables in the multicomputer model, along with the associated ID. In order to be concise, most of the data presented by the DOE study editor is in terms of global variable IDs rather than the full global variable names.

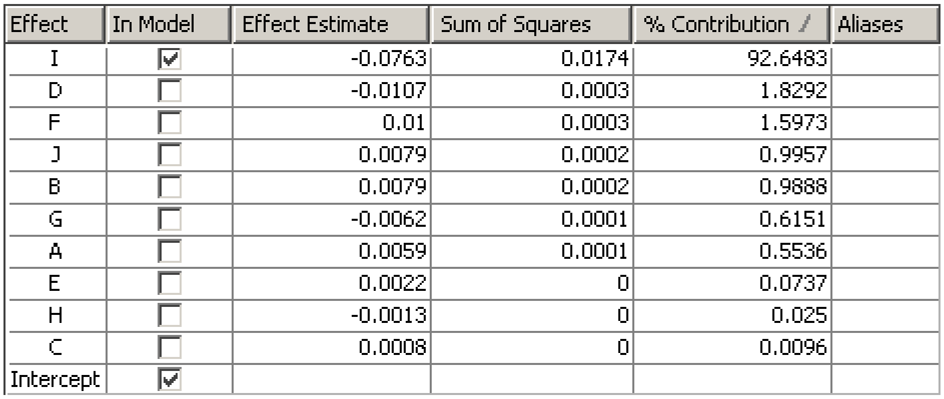
Table 5.2 shows the effects table sorted by percent contribution. 92.6% of the observed variability is explained by effect I, corresponding to global variable `num_comp`, which measures the number of computers. An analysis of variance reveals that all other global variables are insignificant when added to the regression model.

Table 5.1: Global variable data for initial Plackett-Burman design

Global Variable Data					
ID	Name	Type	Constant Value	Low Value	High Value
A	CPU_cov	double	-	0.9	0.999
B	IO_cov	double	-	0.9	0.999
C	RAM_cov	double	-	0.9	0.999
D	comp_cov	double	-	0.9	0.999
E	cpu_failure_rate	double	-	0.0001	0.001
F	failure_rate	double	-	0.0001	0.001
G	ioport_failure_rate	double	-	0.0001	0.001
H	mem_cov	double	-	0.9	0.999
I	num_comp	short	-	1	3
J	num_mem_mod	short	-	1	5

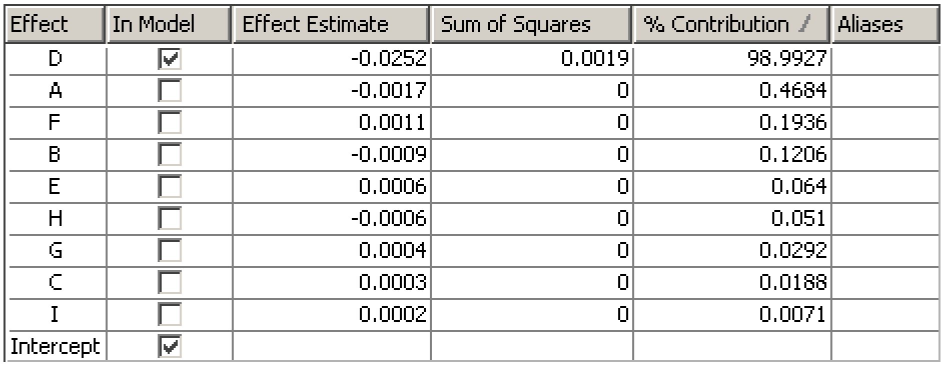
When one factor out of many exerts a large influence on the response, it is often useful to perform an additional screening design with the influential factor held constant. The subsequent analysis can reveal how the less-influential factors affect the response. An additional Plackett-Burman design is performed with the number of computers held constant at three. Before running the additional experiments, a results database query indicates that three experiments in the new Plackett-Burman study were already performed in the previous study; therefore, only nine new experiments need be performed. The resulting effects table

Table 5.2: Effects table for initial Plackett-Burman screening design



Effect	In Model	Effect Estimate	Sum of Squares	% Contribution	Aliases
I	☑	-0.0763	0.0174	92.6483	
D	☐	-0.0107	0.0003	1.8292	
F	☐	0.01	0.0003	1.5973	
J	☐	0.0079	0.0002	0.9957	
B	☐	0.0079	0.0002	0.9888	
G	☐	-0.0062	0.0001	0.6151	
A	☐	0.0059	0.0001	0.5536	
E	☐	0.0022	0	0.0737	
H	☐	-0.0013	0	0.025	
C	☐	0.0008	0	0.0096	
Intercept	☑				

Table 5.3: Plackett-Burman effects table with `num_comp` held constant



Effect	In Model	Effect Estimate	Sum of Squares	% Contribution	Aliases
D	☑	-0.0252	0.0019	98.9927	
A	□	-0.0017	0	0.4684	
F	□	0.0011	0	0.1936	
B	□	-0.0009	0	0.1206	
E	□	0.0006	0	0.064	
H	□	-0.0006	0	0.051	
G	□	0.0004	0	0.0292	
C	□	0.0003	0	0.0188	
I	□	0.0002	0	0.0071	
Intercept	☑				

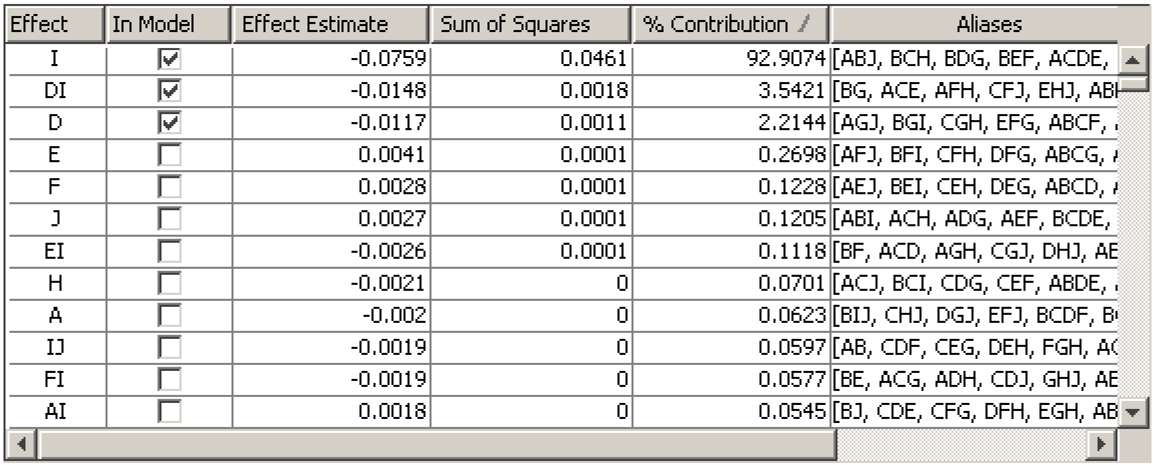
is presented in Table 5.3. Effect D, the computer coverage, explains 98.99% of the variability. This agrees with the data in Table 5.2, which showed that computer coverage was the second most influential factor in the system.

It is suspected that the number of computers and the computer coverage may interact to affect the response. The screening designs presented are not capable of estimating any interactions. Therefore, additional experimental design techniques must be employed to reveal any global variable interactions.

5.3 Two-Level Factorial

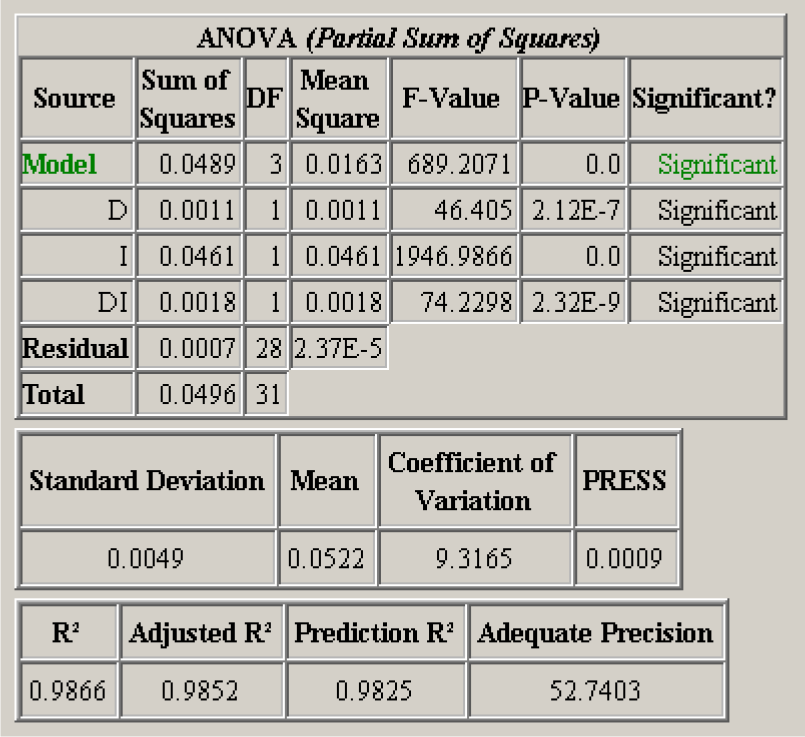
In order to identify any influential interactions among the ten global variables in the system, a 2_{IV}^{10-5} design is created. The results database indicates that two of the 32 experiments have already been performed in the screening designs. After completing the remaining 30 experiments, the effects table in Table 5.4 is created. As suspected, the most variability is explained by the number of computers, effect I, and the computer coverage, effect D. These two factors and their interaction explain 98.46% of the observed variability. The ANOVA and model statistics in Figure 5.2 show that the model is highly significant with excellent predictive capability. An analysis of the diagnostics (not shown) indicates that no assumptions have been violated.

Table 5.4: Two-level factorial effects table



Effect	In Model	Effect Estimate	Sum of Squares	% Contribution	Aliases
I	☑	-0.0759	0.0461	92.9074	[ABJ, BCH, BDG, BEF, ACDE,
DI	☑	-0.0148	0.0018	3.5421	[BG, ACE, AFH, CFJ, EHJ, AB
D	☑	-0.0117	0.0011	2.2144	[AGJ, BGI, CGH, EFG, ABCF,
E	☐	0.0041	0.0001	0.2698	[AFJ, BFI, CFH, DFG, ABCG,
F	☐	0.0028	0.0001	0.1228	[AEJ, BEI, CEH, DEG, ABCD,
J	☐	0.0027	0.0001	0.1205	[ABI, ACH, ADG, AEF, BCDE,
EI	☐	-0.0026	0.0001	0.1118	[BF, ACD, AGH, CGJ, DHJ, AE
H	☐	-0.0021	0	0.0701	[ACJ, BCI, CDG, CEF, ABDE,
A	☐	-0.002	0	0.0623	[BIJ, CHJ, DGJ, EFJ, BCDF, B
IJ	☐	-0.0019	0	0.0597	[AB, CDF, CEG, DEH, FGH, AC
FI	☐	-0.0019	0	0.0577	[BE, ACG, ADH, CDJ, GHJ, AE
AI	☐	0.0018	0	0.0545	[BJ, CDE, CFG, DFH, EGH, AB

The contour plot in Figure 5.3 displays the unreliability of the computer system as the number of computers and the computer coverage, `comp_cov`, are varied. It appears from the plot that the effect of computer coverage increases as the number of computers in-



ANOVA *(Partial Sum of Squares)*

Source	Sum of Squares	DF	Mean Square	F-Value	P-Value	Significant?
Model	0.0489	3	0.0163	689.2071	0.0	Significant
D	0.0011	1	0.0011	46.405	2.12E-7	Significant
I	0.0461	1	0.0461	1946.9866	0.0	Significant
DI	0.0018	1	0.0018	74.2298	2.32E-9	Significant
Residual	0.0007	28	2.37E-5			
Total	0.0496	31				

Standard Deviation	Mean	Coefficient of Variation	PRESS
0.0049	0.0522	9.3165	0.0009

R^2	Adjusted R^2	Prediction R^2	Adequate Precision
0.9866	0.9852	0.9825	52.7403

Figure 5.2: ANOVA and model statistics for two-level factorial model

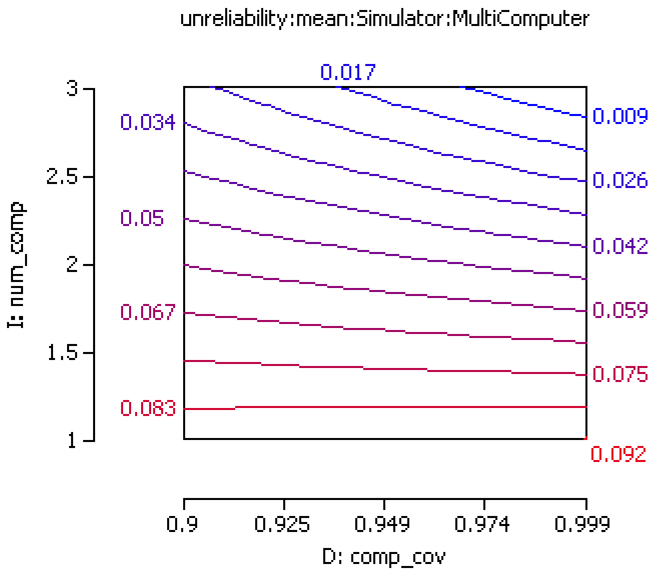


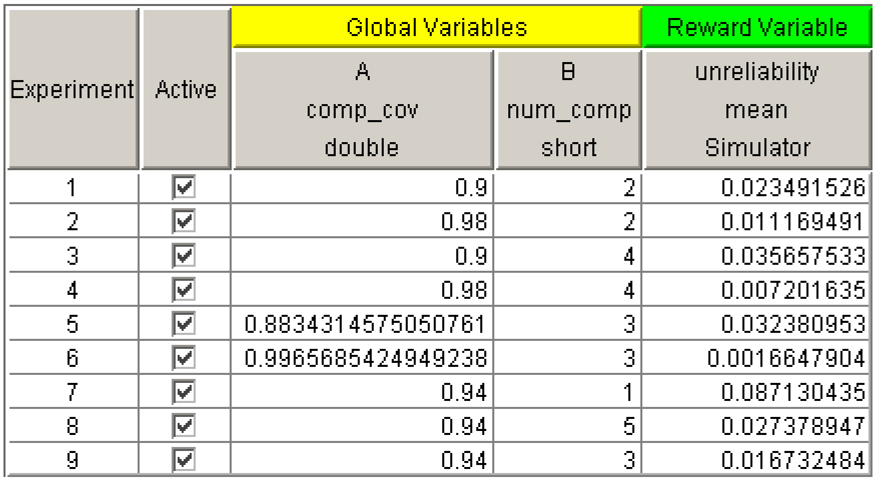
Figure 5.3: Contour plot of multicomputer unreliability for two-level factorial design

creases. The effect is most evident when the number of computers is at its high level of three. Further experimentation is necessary in order to investigate the response behavior in this region of the design space.

5.4 Response Surface

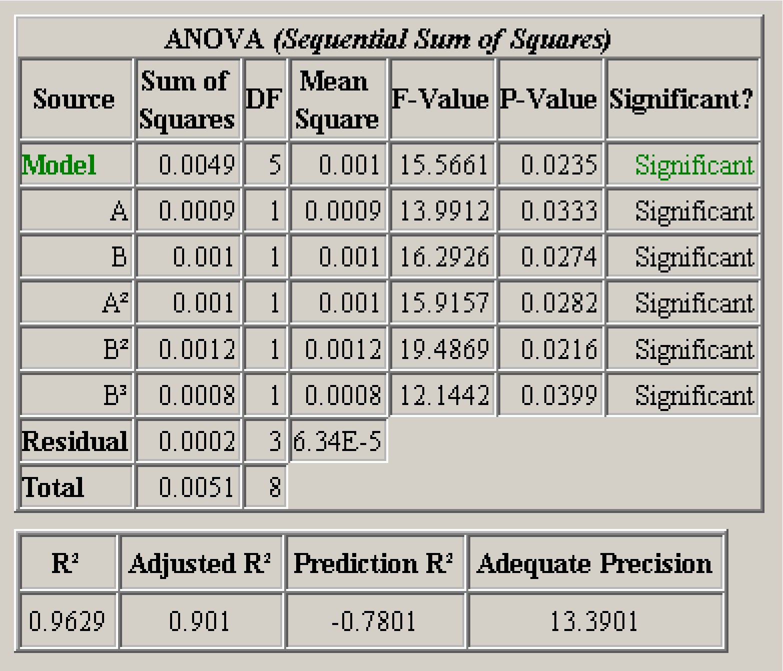
The previous factorial design indicated interesting response behavior near the boundary of the factorial's design region. In particular, the number of computers and the computer coverage appeared to interact when the number of computers equaled three. In order to investigate the unreliability of the system near the boundary of the factorial's design region, a central composite design is created with a design center at `num_comp` = 3. All global variables other than the number of computers and the computer coverage are assigned constant values. The model documentation in the Appendix lists all global variables for all studies in this chapter. The design table is presented in Table 5.5. The values for `num_comp` had to be rounded to the nearest `short` for the axial points in experiments 7 and 8. For that reason, the design is not perfectly rotatable. Therefore, close attention should be paid to the standard errors when predicting response values near those axial points.

Table 5.5: Central composite design with modified axial points



Experiment	Active	Global Variables		Reward Variable
		A comp_cov double	B num_comp short	unreliability mean Simulator
1	☑	0.9	2	0.023491526
2	☑	0.98	2	0.011169491
3	☑	0.9	4	0.035657533
4	☑	0.98	4	0.007201635
5	☑	0.8834314575050761	3	0.032380953
6	☑	0.9965685424949238	3	0.0016647904
7	☑	0.94	1	0.087130435
8	☑	0.94	5	0.027378947
9	☑	0.94	3	0.016732484

The fit summary ANOVA indicates that none of the models other than the mean is significant. Selecting effects A, B, A^2, B^2, and B^3 produces a significant model according to the ANOVA in Figure 5.4. However, the model statistics show that the prediction R^2 is negative. Further investigation with transformations and different regression models also leads to a negative prediction R^2. The only way to create a good predictive model is to augment the design.



ANOVA *(Sequential Sum of Squares)*

Source	Sum of Squares	DF	Mean Square	F-Value	P-Value	Significant?
Model	0.0049	5	0.001	15.5661	0.0235	Significant
A	0.0009	1	0.0009	13.9912	0.0333	Significant
B	0.001	1	0.001	16.2926	0.0274	Significant
A^2	0.001	1	0.001	15.9157	0.0282	Significant
B^2	0.0012	1	0.0012	19.4869	0.0216	Significant
B^3	0.0008	1	0.0008	12.1442	0.0399	Significant
Residual	0.0002	3	6.34E-5			
Total	0.0051	8				

R^2	Adjusted R^2	Prediction R^2	Adequate Precision
0.9629	0.901	-0.7801	13.3901

Figure 5.4: ANOVA and model statistics for central composite design

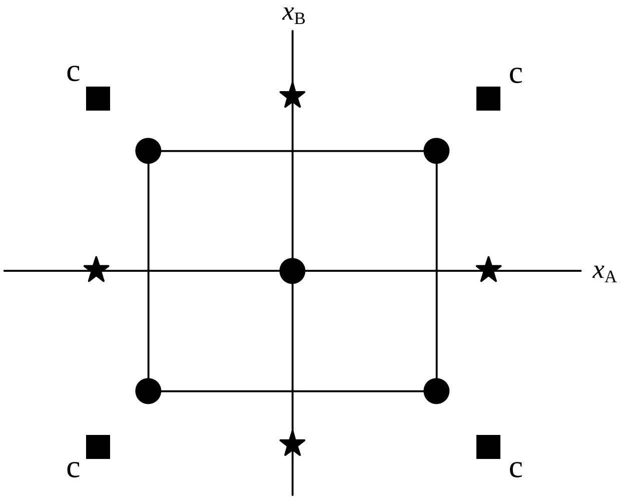
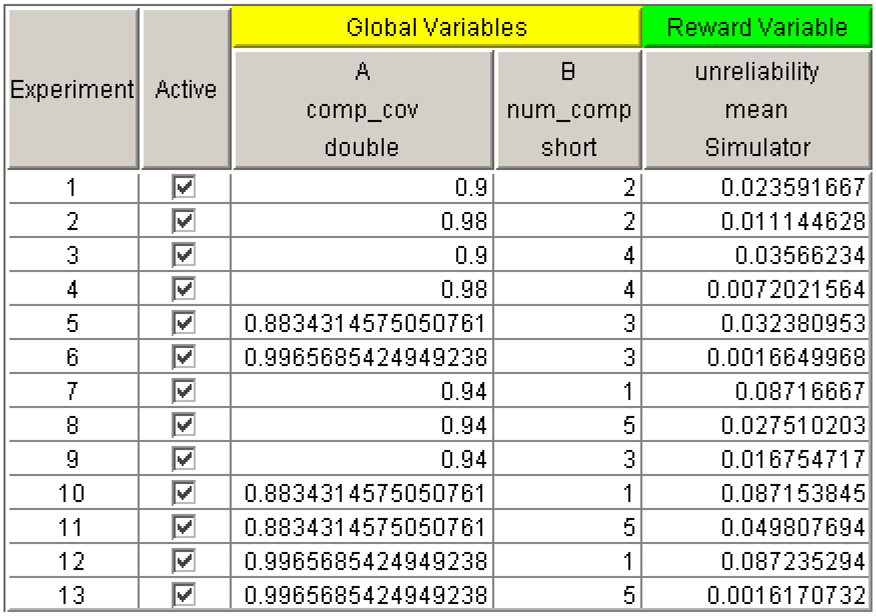


Figure 5.5: Two-factor central composite design augmented with corner points

Figure 5.5 shows a standard two-factor central composite design with four additional *corner points*, all labeled *c*. We placed the corner points symmetrically about the design center so that the symmetry of the standard error would not be distorted. The updated design table with corner points as the last four experiments is presented in Table 5.6.

The fit summary ANOVA (not shown) indicates that the linear and quadratic models are significant. The additional four experiments also provide enough degrees of freedom to

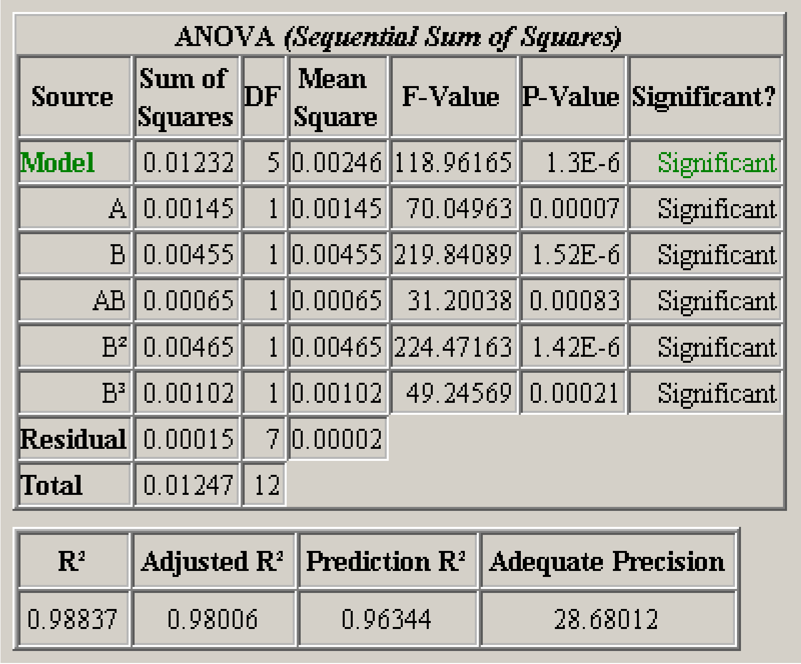
Table 5.6: Central composite design table augmented with corner points



Experiment	Active	Global Variables		Reward Variable
		A comp_cov double	B num_comp short	unreliability mean Simulator
1	☑	0.9	2	0.023591667
2	☑	0.98	2	0.011144628
3	☑	0.9	4	0.03566234
4	☑	0.98	4	0.0072021564
5	☑	0.8834314575050761	3	0.032380953
6	☑	0.9965685424949238	3	0.0016649968
7	☑	0.94	1	0.08716667
8	☑	0.94	5	0.027510203
9	☑	0.94	3	0.016754717
10	☑	0.8834314575050761	1	0.087153845
11	☑	0.8834314575050761	5	0.049807694
12	☑	0.9965685424949238	1	0.087235294
13	☑	0.9965685424949238	5	0.0016170732

de-alias the cubic model. Effects A, B, AB, B^2, and B^3 are selected for the regression model. The ANOVA table in Figure 5.6 indicates that all model terms are significant. The figure also shows that the prediction R^2 is not only positive, but very close to the adjusted R^2. The additional four corner experiments helped prediction tremendously. An analysis of the diagnostic statistics for all of the experiments indicates that none of the assumptions has been violated.

After verifying that the model is acceptable for prediction, we can investigate the response behavior using the model plots. Figure 5.7 shows a contour plot of the system unreliability. For any fixed coverage value, increasing the number of computers from two to three decreases system unreliability. Increasing from three to four computers *increases* unreliability. The optimal unreliability for the lowest required coverage occurs on the line between `num_comp` equal to 2.5 and 2.8, as indicated in Figure 5.7. The line is drawn on the plot manually and is not part of the software output. Only integral values of `num_comp` are physically possible. Additional studies can be performed on the remaining global variables in order to determine the values that bring the optimal line closer to an integral value of `num_comp`.



ANOVA *(Sequential Sum of Squares)*

Source	Sum of Squares	DF	Mean Square	F-Value	P-Value	Significant?
Model	0.01232	5	0.00246	118.96165	1.3E-6	Significant
A	0.00145	1	0.00145	70.04963	0.00007	Significant
B	0.00455	1	0.00455	219.84089	1.52E-6	Significant
AB	0.00065	1	0.00065	31.20038	0.00083	Significant
B^2	0.00465	1	0.00465	224.47163	1.42E-6	Significant
B^3	0.00102	1	0.00102	49.24569	0.00021	Significant
Residual	0.00015	7	0.00002			
Total	0.01247	12				

R^2	Adjusted R^2	Prediction R^2	Adequate Precision
0.98837	0.98006	0.96344	28.68012

Figure 5.6: Augmented central composite design ANOVA and model statistics

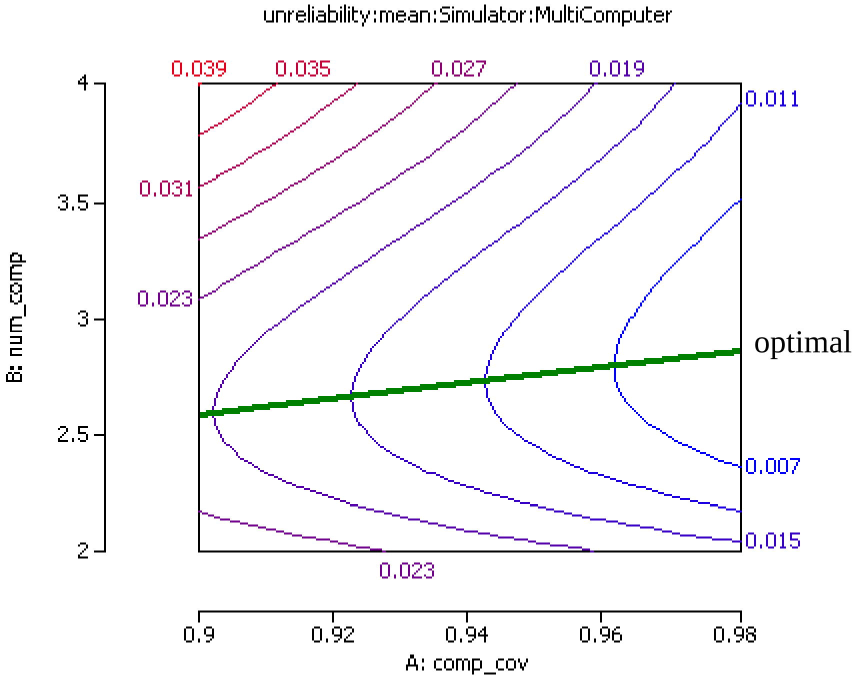


Figure 5.7: Contour plot of central composite design augmented with corner points

5.5 Prediction

The final step of a designed experimental study is to predict the response value within the design region. The prediction panel provides the ability to estimate response values for different global variable settings. Standard errors and confidence intervals are given for each estimation.

Suppose, for example, that the minimum computer coverage value that keeps system unreliability under 0.02 is desired. The second column of Table 5.7 lists the minimum coverage value for each number of computers. The coverage requirement for three computers is much less than for two or four computers. Table 5.7 shows that in order to meet the desired unreliability, a system with four computers requires a *higher* coverage level than a system with only two computers.

Table 5.7: Minimum coverage levels required for unreliability of 0.02

num_comp	comp_cov	Unreliability	Mean Std Error	95% CI	Prediction Std Error	95% PI
2	0.952	0.01996	0.00276	(0.0134, 0.0265)	0.005325	(0.00737, 0.0326)
3	0.910	0.01990	0.00218	(0.0147, 0.0250)	0.005046	(0.00797, 0.0318)
4	0.957	0.01976	0.00279	(0.0132, 0.0264)	0.005341	(0.00713, 0.0324)

Alternatively, one might wish to find the coverage levels that bring either the 95% confidence or prediction interval below 0.02. Table 5.8 shows that for two computers, even a perfect coverage value cannot bring either interval below 0.02. For three computers, a coverage level of 0.96 forces the prediction interval below 0.02, but also introduces negative values into the interval. Care should be taken when interpreting predicted values that fall in an interval with physically impossible values. For four computers, a coverage value outside the design space is required in order to move the prediction interval below 0.02. An asterisk in the table indicates a value outside the design space.

After determining the optimal global variable values for the given response, we can perform confirmation experiments to verify the predicted values. These additional runs can either confirm that the optimal values have been achieved at the desired confidence level, or serve as additional design points upon which a revised regression model can be created. As long as the computing resources are available, the predictive model can be revised until the desired confidence levels are achieved.

Table 5.8: Coverage levels that guarantee that intervals fall below 0.02

Num_comp	Comp_cov	Unreliability	Mean Std Error	95% CI	Prediction Std Error	95% PI
2	1.000*	0.01362	0.00342	(0.0055, 0.0217)	0.005692	(0.00016, 0.0271)
3	0.930	0.01514	0.00202	(0.0104, 0.0199)	0.004981	(0.00336, 0.0269)
3	0.960	0.00799	0.00208	(0.0031, 0.0129)	0.005006	(-0.0038, 0.0198)
4	0.977	0.01281	0.00301	(0.0057, 0.0199)	0.005459	(-0.0001, 0.0257)
4	0.995*	0.00669	0.00332	(0.0012, 0.0145)	0.005632	(-0.0066, 0.0199)

6. CONCLUSION

We have presented a Design of Experiments interface integrated into the Möbius modeling tool. Some of the benefits of DOE were presented, with emphasis placed on the increased efficiency of the experimentation process. The techniques of DOE are most valuable when modeling sophisticated systems that require large amounts of computer resources for solution. DOE uses the results from a select number of simulation or numerical solutions to build an empirical model of the reward variable. Evaluation of the empirical model takes much less time than model solution. The empirical model, when found to be satisfactory, can effectively replace simulation or numerical solution within specific areas of the design space. DOE techniques attempt to maximize the amount of information per experimental run by reducing the number of experimental points required.

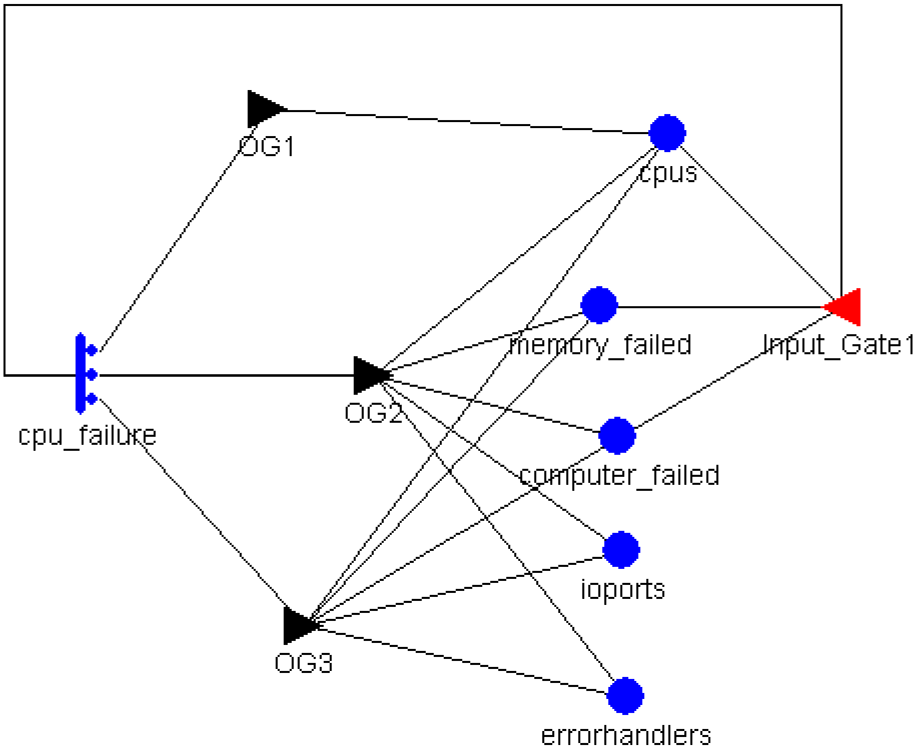
The user interface for developing the empirical model of the reward variable was presented in detail. Both graphical and numerical methods were presented for ranking the sensitivity of the reward variables to each model parameter. The interface includes methods for screening a large number of model parameters in order to determine which ones contribute significantly to the reward variable. The ability to perturb system parameters with immediate graphical or numerical predictions of the reward values is available. Graphical methods are available for revealing the optimal reward value within the design region. Integration of the DOE interface into the Möbius tool allows experimental results to be imported directly from the results database without the need to copy data manually. We presented an example illustrating the key features of the DOE study editor.

Future research projects will take advantage of the DOE study editor's efficient experimentation strategies and graphical presentation of results. Additional types of experimental designs can be added to the interface. A GUI-based browser for viewing and exporting Möbius solution results will be developed. This results browser will share graphical and numerical results with the DOE study editor.

APPENDIX MULTICOMPUTER SYSTEM DOCUMENTATION

This appendix contains a description of the multicomputer system described in Chapter 5. The following is an example of the automatically generated documentation created from the project's "Document Project" function. The model was first published in [23].

Atomic Model: SAN: cpu_module



Place Attributes	
Name	**Initial Marking**
cpus	3
ioports	2
errorhandlers	2
memory_failed	0
computer_failed	0

Timed Activity Attributes: cpu_failure

Field Name	Field Value
Exponential Distribution Parameters	*Rate* 6.0 * cpu_failure_rate * cpus->Mark()
Activation Predicate	1
Reactivation Predicate	1

Case Distributions	***case 1***
	if (cpus->Mark() == 3)
	  return(CPU_cov);
	else
	  return(0.0);
	
	case 2
	if (cpus->Mark() == 3)
	  return((1.0 - CPU_cov) * comp_cov);
	else
	  return(comp_cov);
	
	case 3
	if (cpus->Mark() == 3)
	  return((1.0 - CPU_cov) * (1.0 - comp_cov));
	else
	  return(1.0 - comp_cov);

Input Gate Attributes: Input_Gate1

Field Name	Field Value
Predicate	(cpus->Mark() > 1) && (memory_failed->Mark() < 2) && (computer_failed->Mark() < num_comp)
Function	;

Output Gate Attributes: OG3

Field Name	Field Value

Function	cpus->Mark() = 0;
	ioports->Mark() = 0;
	errorhandlers->Mark() = 0;
	memory_failed->Mark() = 2;
	computer_failed->Mark() = num_comp;

Output Gate Attributes: OG2

Field Name	Field Value

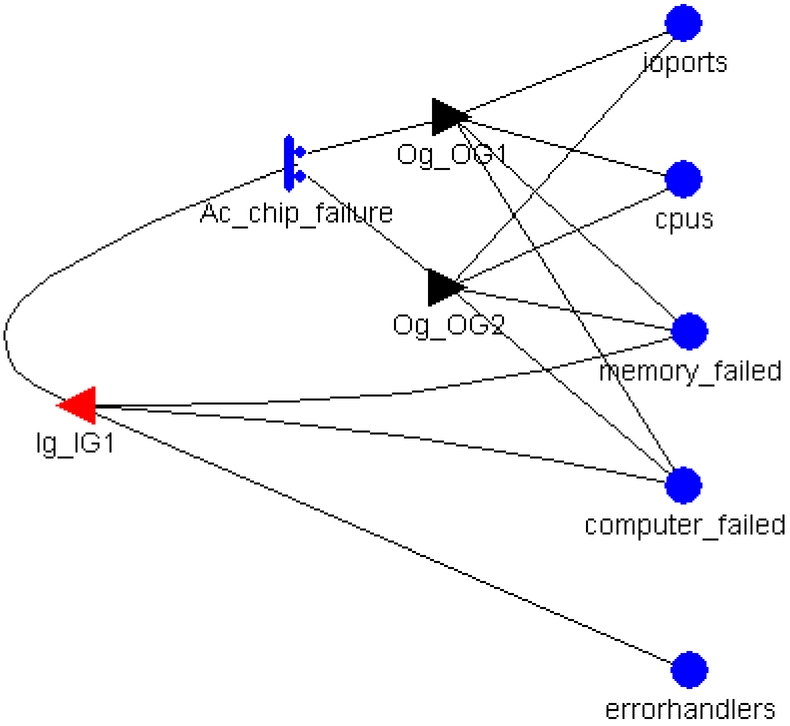
Function	`cpus->Mark() = 0;`
	`ioports->Mark() = 0;`
	`errorhandlers->Mark() = 0;`
	`memory_failed->Mark() = 2;`
	`computer_failed->Mark()++;`

Output Gate Attributes: OG1

Field Name	Field Value

Function	`if (cpus->Mark() == 3)`
	`        cpus->Mark()--;`

Atomic Model: SAN: errorhandlers



Place Attributes

Name	Initial Marking
cpus	3
computer_failed	0
memory_failed	0
errorhandlers	2
ioports	2

Timed Activity Attributes: Ac_chip_failure

Field Name	Field Value
Exponential Distribution Parameters	***Rate*** `0.008766  * errorhandlers->Mark()`
Activation Predicate	`1`
Reactivation Predicate	`1`

Case Distributions	***case 1***
	`comp_cov`
	case 2
	`1.0 – comp_cov`

Input Gate Attributes: Ig_IG1

Field Name	Field Value
Predicate	`(errorhandlers->Mark() == 2) && (memory_failed->Mark() < 2) && (computer_failed->Mark() < num_comp)`
Function	`errorhandlers->Mark() = 0;`

Output Gate Attributes: Og_OG2

Field Name	Field Value

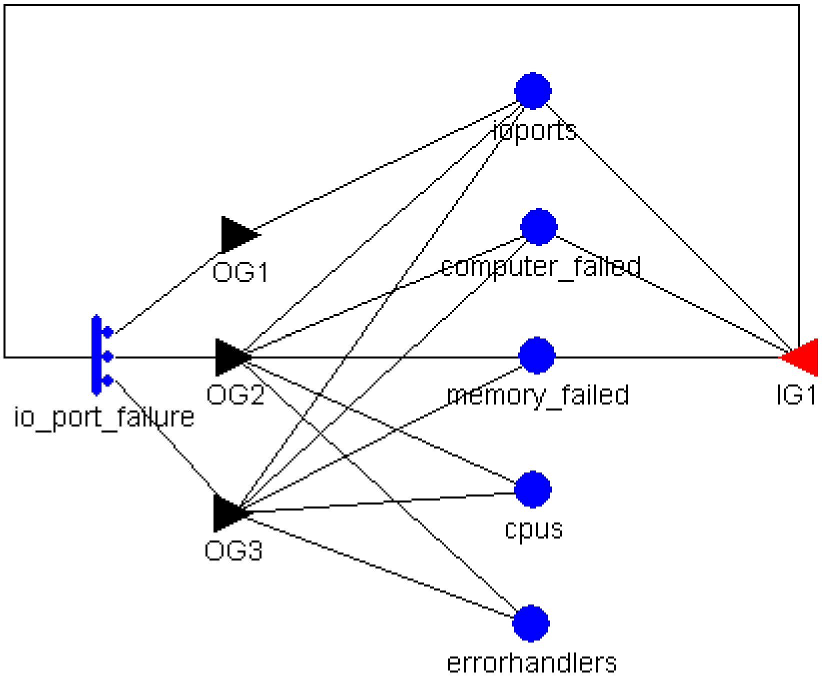
Function	`cpus->Mark() = 0;`
	`ioports->Mark() = 0;`
	`memory_failed->Mark() = 2;`
	`computer_failed->Mark() = num_comp;`

Output Gate Attributes: Og_OG1

Field Name	Field Value

Function	`cpus->Mark() = 0;`
	`ioports->Mark() = 0;`
	`memory_failed->Mark() = 2;`
	`computer_failed->Mark()++;`

Atomic Model: SAN: io_port_module



Place Attributes

Name	Initial Marking
cpus	3
computer_failed	0
memory_failed	0
errorhandlers	2
ioports	2

Timed Activity Attributes: io_port_failure	
Field Name	**Field Value**
Exponential Distribution Parameters	***Rate*** `ioport_failure_rate * ioports->Mark()`
Activation Predicate	`1`
Reactivation Predicate	`1`

Case Distributions	***case 1***
	`if (ioports->Mark() == 2)`
	`  return (IO_cov);`
	`else`
	`  return(0.0);`
	case 2
	`if (ioports->Mark() == 2)`
	`  return((1.0 – IO_cov) * comp_cov);`
	`else`
	`  return(comp_cov);`
	case 3
	`if (ioports->Mark() == 2)`
	`  return((1.0 – IO_cov) * (1.0 – comp_cov));`
	`else`
	`  return (1.0 – comp_cov);`

Input Gate Attributes: IG1	
Field Name	**Field Value**

Predicate	`(ioports->Mark() > 0) && (memory_failed->Mark() < 2) &&`
	`(computer_failed->Mark() < num_comp)`

Function	`;`

Output Gate Attributes: OG3	
Field Name	**Field Value**

Function	`cpus->Mark() = 0;`
	`ioports->Mark() = 0;`
	`errorhandlers->Mark() = 0;`
	`memory_failed->Mark() = 2;`
	`computer_failed->Mark() = num_comp;`

Output Gate Attributes: OG2

Field Name	Field Value

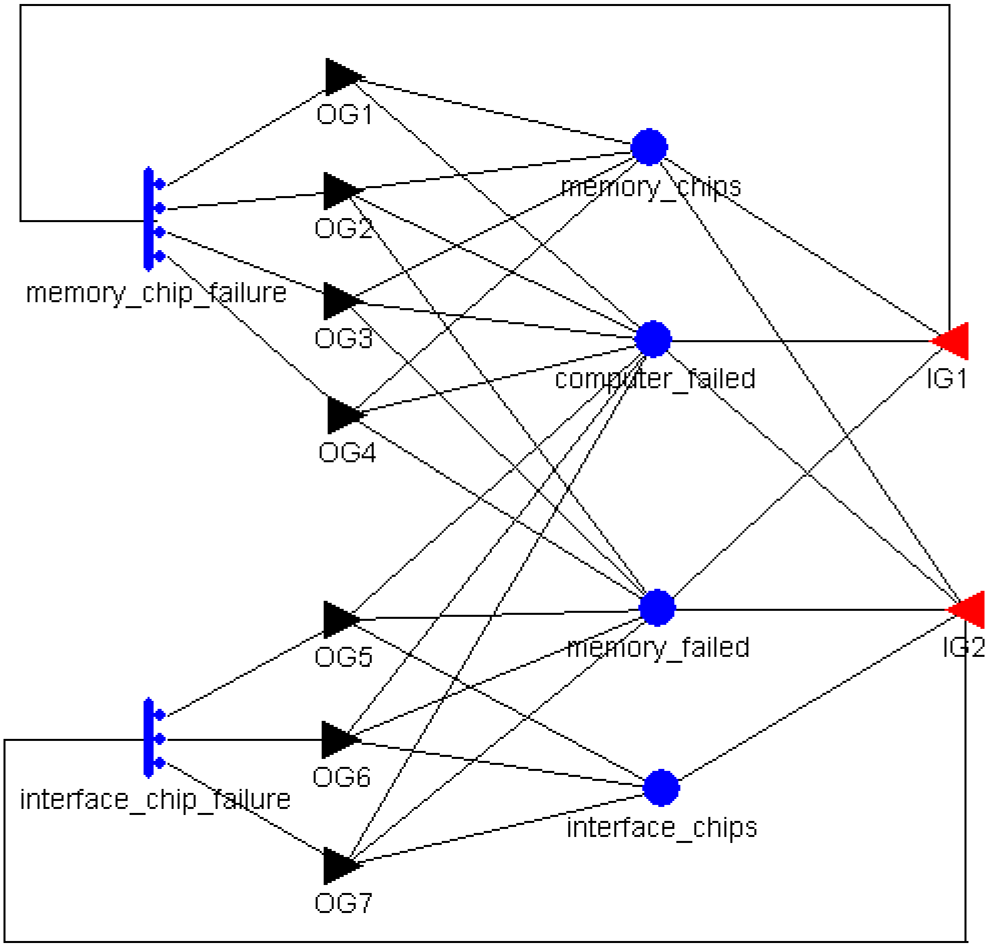
Function	`cpus->Mark() = 0;`
	`ioports->Mark() = 0;`
	`errorhandlers->Mark() = 0;`
	`memory_failed->Mark() = 2;`
	`computer_failed->Mark()++;`

Output Gate Attributes: OG1

Field Name	Field Value

Function	`if (ioports->Mark() == 2)`
	`        ioports->Mark()--;`

Atomic Model: SAN: memory_module



Place Attributes

Name	Initial Marking
memory_failed	0
computer_failed	0
interface_chips	2
memory_chips	41

Timed Activity Attributes: interface_chip_failure

Field Name	Field Value
Exponential **Distribution Parameters**	***Rate*** `failure_rate * interface_chips->Mark()`
Activation Predicate	`1`
Reactivation Predicate	`1`

Case Distributions	***case 1***
	`mem_cov`
	case 2
	`((1.0 – mem_cov) * (comp_cov))`
	case 3
	`(1.0 – mem_cov) * (1.0 – comp_cov)`

Timed Activity Attributes: memory_chip_failure

Field Name	Field Value
Exponential Distribution Parameters	***Rate*** `failure_rate * memory_chips->Mark()`
Activation Predicate	`1`
Reactivation Predicate	`1`

Case Distributions	***case 1***
	`if (memory_chips->Mark() == 39)`
	`  return(0.0);`
	`else`
	`  return(RAM_cov);`
	case 2
	`if (memory_chips->Mark() == 39)`
	`  return(mem_cov);`
	`else`
	`  return((1.0 – RAM_cov) * mem_cov);`
	case 3
	`if (memory_chips->Mark() == 39)`
	`  return ((1.0 – mem_cov) * comp_cov);`
	`else`
	`  return ((1.0 – RAM_cov) * (1.0 – mem_cov) * (comp_cov));`
	case 4
	`if (memory_chips->Mark() == 39)`
	`  return((1.0 – mem_cov) * (1.0 – comp_cov));`
	`else`
	`  return((1.0 – RAM_cov) * (1.0 – mem_cov) * (1.0 – comp_cov));`

Input Gate Attributes: IG2

Field Name	Field Value

Predicate	`(interface_chips->Mark() > 1) && (memory_failed->Mark() < 2) &&`
	`(computer_failed->Mark() < num_comp)`

Function	`memory_chips->Mark() = 0;`

Input Gate Attributes: IG1

Field Name	Field Value

Predicate	`(memory_chips->Mark() > 38) &&`
	`(computer_failed->Mark() < num_comp) &&`
	`(memory_failed->Mark() < 2)`

Function	`;`

Output Gate Attributes: OG7	
Field Name	**Field Value**

Function	`interface_chips->Mark() = 0;`
	`memory_failed->Mark() = 2;`
	`computer_failed->Mark() = num_comp;`

Output Gate Attributes: OG6	
Field Name	**Field Value**

Function	`interface_chips->Mark() = 0;`
	`if ((memory_failed->Mark() == 1) && (computer_failed->Mark() <= num_comp – 2))`
	`          computer_failed->Mark() = num_comp;`
	`else`
	`          computer_failed->Mark()++;`
	`memory_failed->Mark() = 2;`

Output Gate Attributes: OG5	
Field Name	**Field Value**

Function	`interface_chips->Mark() = 0;`
	`memory_failed->Mark()++;`
	`if (memory_failed->Mark() > 1)`
	`          computer_failed->Mark()++;`

Output Gate Attributes: OG4	
Field Name	**Field Value**

Function	`memory_chips->Mark() = 0;`
	`interface_chips->Mark() = 0;`
	`memory_failed->Mark() = 2;`
	`computer_failed->Mark() = num_comp;`

Output Gate Attributes: OG3	
Field Name	**Field Value**

Function	`memory_chips->Mark() = 0;`
	`interface_chips->Mark() = 0;`
	`if ((memory_failed->Mark() == 1) && (computer_failed->Mark() <= num_comp – 2))`
	`          computer_failed->Mark() = num_comp;`
	`else`
	`          computer_failed->Mark()++;`
	`memory_failed->Mark() = 2;`

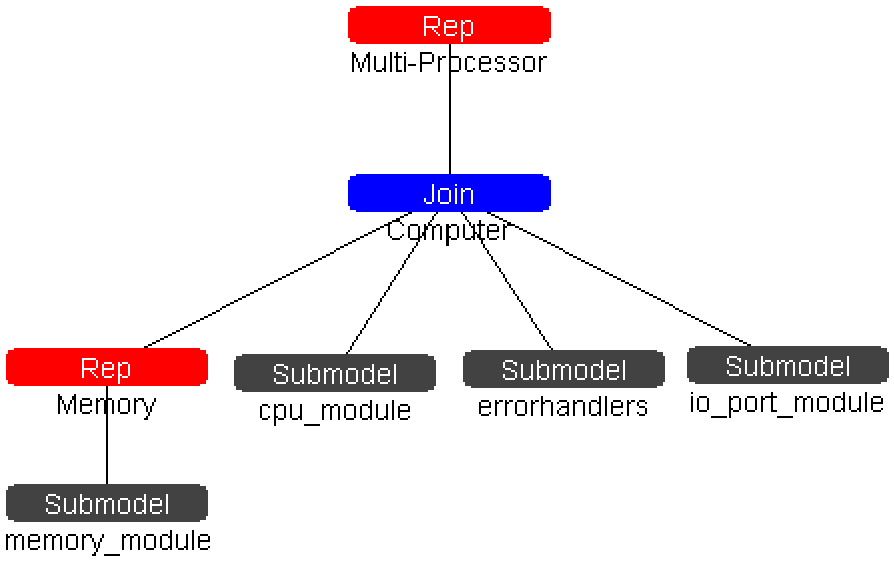
Output Gate Attributes: OG2	
Field Name	**Field Value**

Function	`memory_chips->Mark() = 0;`
	`interface_chips->Mark() = 0;`
	`memory_failed->Mark()++;`
	`if (memory_failed->Mark() > 1)`
	`        computer_failed->Mark()++;`

Output Gate Attributes: OG1	
Field Name	**Field Value**

Function	`if (memory_chips->Mark() > 39)`
	`        memory_chips->Mark()--;`

Composed Model: Rep/Join: multi_proc



Rep Node Definitions for Composed Model multi_proc in project Multi-Proc		
Node	**Reps**	**Shared State Variables**
Multi-Processor	num_comp	computer_failed
Memory	num_mem_mod	computer_failed
		memory_failed

Join Node Definitions for Computer

State Variable Name	Submodel Variables
computer_failed	Memory->computer_failed
	io_port_module->computer_failed
	errorhandlers->computer_failed
	cpu_module->computer_failed
cpus	cpu_module->cpus
	errorhandlers->cpus
	io_port_module->cpus
errorhandler	io_port_module->errorhandlers
	errorhandlers->errorhandlers
	cpu_module->errorhandlers
ioports	cpu_module->ioports
	errorhandlers->ioports
	io_port_module->ioports
memory_failed	Memory->memory_failed
	io_port_module->memory_failed
	errorhandlers->memory_failed
	cpu_module->memory_failed

Reward Model

Information for MultiProc_PV		
Top Level Model Information	Model	c:\MobiusProject\Multi-Proc\Composed\multi_proc\multi_proc.cmp
	Model Name	multi_proc
	Model Type	Rep/Join
Compilation Information	Output Directory	c:\MobiusProject\Multi-Proc\Reward\MultiProc_PV\
	Model File Name	MultiProc_PVPVModel
	Node File Name	MultiProc_PVPVNodes
	Archive Name	MultiProc_PVPV.a

Performance Variable: unreliability			
Affecting Models	cpu_module		
Impulse Functions			

Reward Function	*(Reward is over all available models)*
	`if (cpu_module->computer_failed->Mark() == num_comp)`
	`{`
	`        return 1.0/num_comp;`
	`}`		

Simulator Statistics	Type	Instant of Time	
	Options	Estimate Mean	
		Estimate Variance	
		Estimate Interval	
		Estimate out of Range Probabilities	
		Confidence Level is Relative	
	Parameters	Start Time	5.0
	Intervals	Lower Bound on Interval Estimate	0.0
		Upper Bound on Interval Estimate	1.0
	Confidence	Confidence Level	0.99
		Confidence Interval	0.05

Studies

Information for Study *Plackett-Burman_1* in Project *Multi-Proc*

Design Type	Number of Experiments	Number of User-Added Experiments
Plackett Burman	12	0

Global Variable Data

ID	Name	Type	Constant Value	Low Value	High Value
A	CPU_cov	double	-	0.9	0.999
B	IO_cov	double	-	0.9	0.999
C	RAM_cov	double	-	0.9	0.999
D	comp_cov	double	-	0.9	0.999
E	cpu_failure_rate	double	-	0.0001	0.001
F	failure_rate	double	-	0.0001	0.001
G	ioport_failure_rate	double	-	0.0001	0.001
H	mem_cov	double	-	0.9	0.999
I	num_comp	short	-	1	3
J	num_mem_mod	short	-	1	5

Response Data

Reward Variable	Reward Type	Solver Type	Solver Name
unreliability	mean	Simulator	PB1Sim

Information for Study *Plackett-Burman_2* in Project ***Multi-Proc***		
Design Type	**Number of Experiments**	**Number of User-Added Experiments**
Plackett Burman	12	0

Global Variable Data					
ID	**Name**	**Type**	**Constant Value**	**Low Value**	**High Value**
A	CPU_cov	double	-	0.9	0.999
B	IO_cov	double	-	0.9	0.999
C	RAM_cov	double	-	0.9	0.999
D	comp_cov	double	-	0.9	0.999
E	cpu_failure_rate	double	-	0.0001	0.001
F	failure_rate	double	-	0.0001	0.001
G	ioport_failure_rate	double	-	0.0001	0.001
H	mem_cov	double	-	0.9	0.999
	num_comp	short	3	-	-
I	num_mem_mod	short	-	1	5

Response Data			
Reward Variable	**Reward Type**	**Solver Type**	**Solver Name**
unreliability	mean	Simulator	PB2Sim

Information for Study *MultiComputerFactorial* in Project *Multi-Proc*

Design Type	Number of Experiments	Number of User-Added Experiments
2 Level Factorial	32	0

Global Variable Data

ID	Name	Type	Constant Value	Low Value	High Value
A	CPU_cov	double	-	0.9	0.999
B	IO_cov	double	-	0.9	0.999
C	RAM_cov	double	-	0.9	0.999
D	comp_cov	double	-	0.9	0.999
E	cpu_failure_rate	double	-	0.0001	0.001
F	failure_rate	double	-	0.0001	0.001
G	ioport_failure_rate	double	-	0.0001	0.001
H	mem_cov	double	-	0.9	0.999
I	num_comp	short	-	1	3
J	num_mem_mod	short	-	1	5

Response Data

Reward Variable	Reward Type	Solver Type	Solver Name
unreliability	mean	Simulator	FactorialSim

Information for Study *CentralComposite_1* in Project ***Multi-Proc***

Design Type	Number of Experiments	Number of User-Added Experiments
Central Composite	9	0

Global Variable Data

ID	Name	Type	Constant Value	Low Value	High Value
	CPU_cov	double	0.995	-	-
	IO_cov	double	0.99	-	-
	RAM_cov	double	0.998	-	-
A	comp_cov	double	-	0.9	0.98
	cpu_failure_rate	double	0.0008766	-	-
	failure_rate	double	0.0008766	-	-
	ioport_failure_rate	double	0.0052596	-	-
	mem_cov	double	0.95	-	-
B	num_comp	short	-	2	4
	num_mem_mod	short	3	-	-

Response Data

Reward Variable	Reward Type	Solver Type	Solver Name
unreliability	mean	Simulator	CentralComp_1Sim

Information for Study *CentralCompositeCorners* in Project *Multi-Proc*

Design Type	Number of Experiments	Number of User-Added Experiments
Central Composite	13	4

Global Variable Data

ID	Name	Type	Constant Value	Low Value	High Value
	CPU_cov	double	0.995	-	-
	IO_cov	double	0.99	-	-
	RAM_cov	double	0.998	-	-
A	comp_cov	double	-	0.9	0.98
	cpu_failure_rate	double	0.0008766	-	-
	failure_rate	double	0.0008766	-	-
	ioport_failure_rate	double	0.0052596	-	-
	mem_cov	double	0.95	-	-
B	num_comp	short	-	2	4
	num_mem_mod	short	3	-	-

Response Data

Reward Variable	Reward Type	Solver Type	Solver Name
unreliability	mean	Simulator	CentralComp_CornersSim

User Added Experiments

Experiment Number	A	B
10	0.8834314575050761	1
11	0.8834314575050761	5
12	0.9965685424949238	1
13	0.9965685424949238	5

HTML generated by Möbius Documenter

REFERENCES

[1] G. Clark, T. Courtney, D. Daly, D. Deavours, S. Derisavi, J. M. Doyle, W. H. Sanders, and P. Webster, "The Möbius modeling tool," in *Proceedings of the 9th International Workshop on Petri Nets and Performance Models*, Aachen, Germany, September 11-14, 2001, pp. 241-250.

[2] D. D. Deavours and W. H. Sanders, "Möbius: Framework and atomic models," in *Proceedings of the 9th International Workshop on Petri Nets and Performance Models*, Aachen, Germany, September 11-14, 2001, pp. 251-260.

[3] D. D. Deavours, G. Clark, T. Courtney, D. Daly, S. Derisavi, J. M. Doyle, W. H. Sanders, and P. G. Webster, "The Möbius framework and its implementation," *IEEE Transactions on Software Engineering special issue on PNPM 2001,* to appear.

[4] J. M. Doyle, "Abstract model specification using the Möbius modeling tool," M.S. thesis, University of Illinois at Urbana-Champaign, 2000.

[5] A. L. Christensen, "Result specification and model connection in the Möbius modeling framework," M.S. thesis, University of Illinois at Urbana-Champaign, 2000.

[6] A. J. Stillman, "Model composition within the Möbius modeling framework," M.S. thesis, University of Illinois at Urbana-Champaign, 1999.

[7] G. E. P. Box, W. G. Hunter, and J. S. Hunter, *Statistics for Experimenters: An Introduction to Design, Data Analysis, and Model Building*. New York, New York: John Wiley & Sons, 1978.

[8] D. C. Montgomery, *Design and Analysis of Experiments.* Fifth Edition, New York, New York: John Wiley & Sons, 2000.

[9] R. Jain, *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. New York, New York: John Wiley & Sons, 1991.

[10] G. E. P. Box and N. R. Draper, *Empirical Model Building and Response Surfaces*. New York, New York: John Wiley & Sons, 1987.

[11] R. L. Plackett and J. P. Burman, "The design of optimum multifactorial experiments," *Biometrika,* vol. 33, pp. 303-325, 1946.

[12] H. J. Ryser, *Combinatorial Mathematics*. New York, New York: John Wiley & Sons, 1963.

[13] R. H. Myers and D. C. Montgomery, *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. New York, New York: John Wiley & Sons, 1995.

[14] G. E. P. Box and D. W. Behnken, "Some new three level designs for the study of quantitative variables," *Technometrics*, vol. 2, pp. 455-476, 1960.

[15] H. R. Lindman, *Analysis of Variance in Experimental Design*. New York, New York: Springer-Verlag, 1991.

[16] D. R. Cook, “Influential observations in linear regression,” *Journal of the American Statistical Association*, vol. 74, pp. 169-174, 1979.

[17] P. G. Hoel, *Introduction to Mathematical Statistics*. New York, New York: John Wiley & Sons, 1970.

[18] R. H. Myers, *Classical and Modern Regression with Applications.* Second Edition, Boston: PWS-KENT, 1990.

[19] J. A. Nelder, “The selection of terms in response-surface models: How strong is the weak-heredity principle?” *The American Statistician*, vol. 52, no. 4, pp 315-318, November 1998.

[20] J. Peixoto, “A property of well-formulated polynomial regression models,” *The American Statistician*, vol. 44, no. 1, pp. 26-30, February 1990.

[21] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to Linear Regression Analysis.* 3rd Edition, New York, New York: John Wiley & Sons, 2001.

[22] C. Daniel, “Use of half-normal plots in interpreting factorial two-level experiments,” *Technometrics,* vol. 1, pp. 311-341, November 1959.

[23] W. H. Sanders and L. M. Malhis, “Dependability evaluation using composed SAN-based reward models,” *Journal of Parallel and Distributed Computing, Special Issue on Petri Net Models of Parallel and Distributed Computers*, vol. 15, no. 3, pp. 238-254, July 1992.