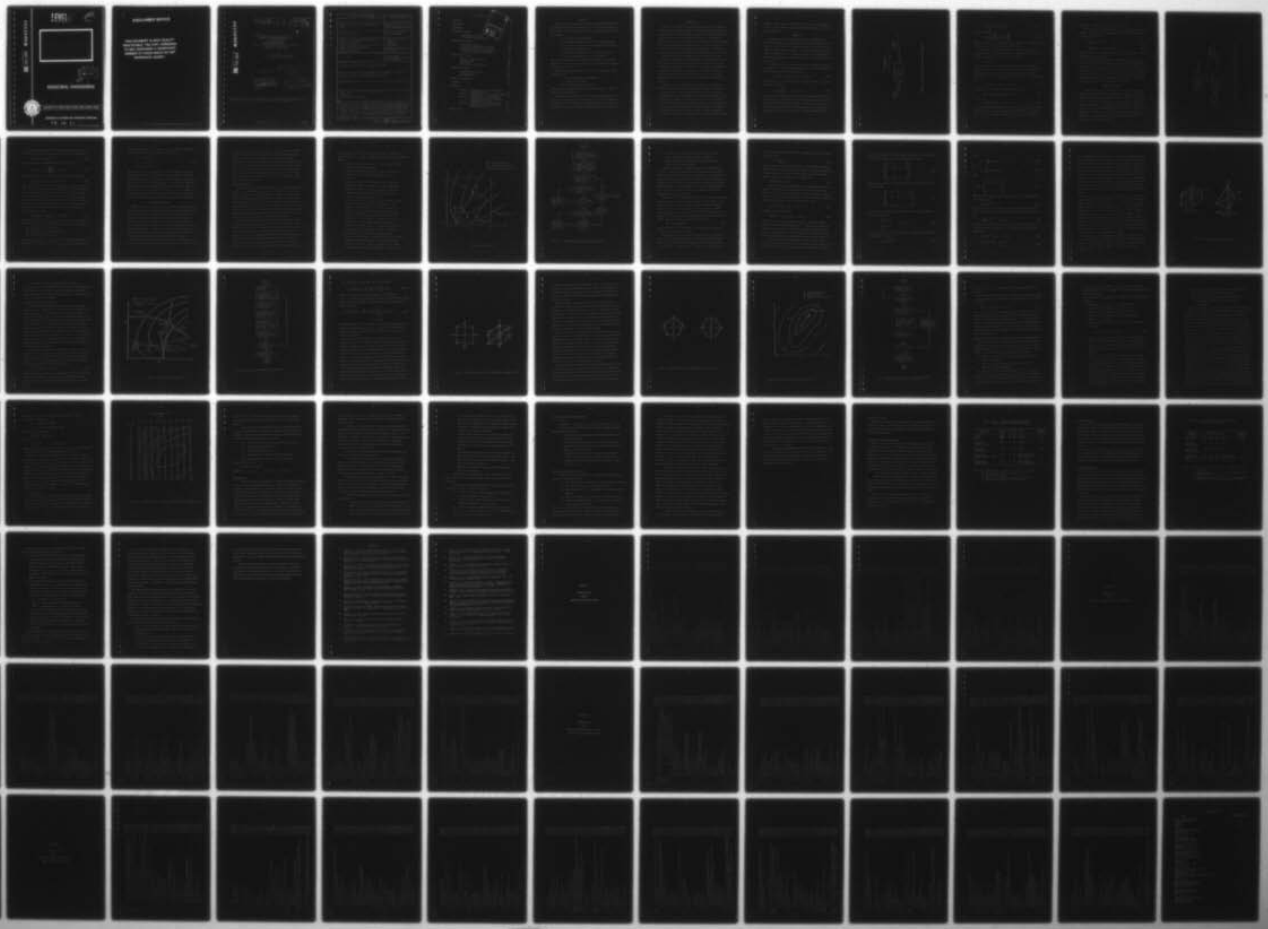
AD-A062 389 NOTRE DAME UNIV IND DEPT OF AEROSPACE AND MECHANICA--ETC F/G 12/2
OPTIMIZATION OF MULTIPLE-RESPONSE SIMULATION MODELS.(U)
OCT 78 W E BILES N00014-76-C-1021

UNCLASSIFIED NL




OF
DA
2389

DDC FILE COPY AD A062389

LEVEL II

12

DDC
RECEIVED
DEC 14 1978
F

INDUSTRIAL ENGINEERING

UNIVERSITY OF NOTRE DAME, NOTRE DAME, INDIANA 46556

Department of Aerospace and Mechanical Engineering

78 12 11 160

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DDC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

AD A062389

DDC FILE COPY

DDC
DEC 14 1978
F

(9) FINAL REPORT
1 Sep 76-31 Oct 78

(6) OPTIMIZATION OF MULTIPLE-RESPONSE SIMULATION MODELS

(10) William E. Biles
Department of Aerospace and Mechanical Engineering
University of Notre Dame
Notre Dame, Indiana 46556

This document has been approved for public release and sale; its distribution is unlimited

(11) 31 Oct 78

October 31, 1978

(12) 96p.

This research was sponsored by
Naval Analysis Programs
Office of Naval Research
Contract N00014-76-C-1021
(15) NR 277-237

(16) RR01411
(17) RR01411 01

Reproduction in whole or in part is permitted for any purpose of the United States Government. Available for public release; distribution unlimited.

401 829 JOB

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) OPTIMIZATION OF MULTIPLE RESPONSE SIMULATION MODELS.		5. TYPE OF REPORT & PERIOD COVERED Final Report September 1,1976-Oct. 31,1978 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Dr. William E. Biles		8. CONTRACT OR GRANT NUMBER(s) N0014-76-C-1021
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Aerospace and Mechanical Engr. University of Notre Dame Notre Dame, IN 46556		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61153 N RR014-11-01 NR 277-237
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Analysis Programs (Code 431) Office of Naval Research Arlington, VA 22217		12. REPORT DATE October 31, 1978 13. NUMBER OF PAGES 91
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) N/A		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Available for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Available for public release; distribution unlimited.

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Optimization
Simulation
Response Surface Methodology

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

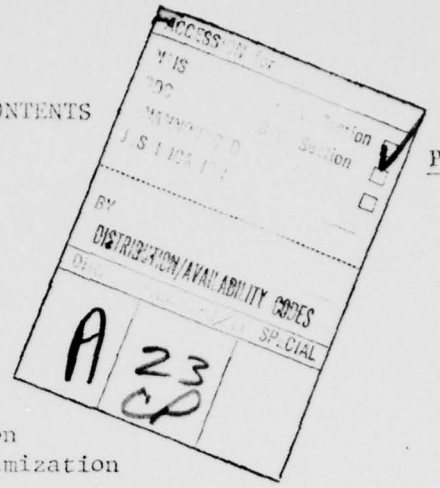
This report describes several computerized multiple-variable, multiple-response optimization procedures for use in connection with computer simulation models. These procedures include (1) "complex" search, (2) a sequential first-order response surface approach, (3) a second-order response surface method, and (4) "complex" search followed by second-order response surface analysis. These four optimization procedures are evaluated with (1) a stochastic inventory simulation model, (2) a tank duel simulation, and (3) a naval minefield evaluation model.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

78 12 11 160

TABLE OF CONTENTS



	Page
ABSTRACT	ii
INTRODUCTION	1
BACKGROUND	2
PROBLEM FORMULATIONS	5
Constrained Optimization	7
Multiple-Objective Optimization	7
OPTIMIZATION TECHNIQUES	10
Direct Search Methods	11
First-Order Response Surface Methods	15
Second-Order Response Surface Methods	23
Second-Order Response Surface Analysis of Box's Complex Search Experiments	32
SIMULATION TEST MODELS	36
Stochastic Inventory System	36
Tank Duel Model	37
Mine Hunting	39
Air Attack on a Surface Fleet	42
DISCUSSION OF RESULTS	45
Stochastic Inventory Model	45
Tank Duel Model	47
Mine Hunting Model	47
SUMMARY AND CONCLUSIONS	50
REFERENCES	54
APPENDICES	
Appendix A - Program Listing for an Experimental Box Complex Search	56
Appendix B - Program Listing for a Sequential First-Order Response Surface Method	61
Appendix C - Program Listing for a Second-Order Response Surface Method Based on a Central Composite Design	70
Appendix D - Program Listing for a Second-Order Response Surface Method Based on Box's Complex Search	79
DISTRIBUTION	90

ABSTRACT

This report describes several computerized multiple-variable, multiple-response optimization procedures developed for use in connection with simulation models. The four optimization procedures evaluated in this research were as follows:

1. Box's complex search method [8];
2. A sequential first-order response surface approach [2,3];
3. A second-order response-surface experimental design followed by mathematical optimization;
4. A Box complex search followed by a second-order response surface optimization (methods 1 and 3 combined).

Each of these optimization methods involved an adaptation of the original procedure on which it was based, in order to accommodate multiple simulation responses η_j, j=1, ..., m.

These four optimization procedures were evaluated with three computer simulation models:

i. A stochastic inventory model [15,16];

ii. A tank duel model [21];

iii. A minefield evaluation model [1].

Exploratory work was also performed with a simulation model of a missile attack on a surface fleet [18].

This research has produced workable optimization procedures for interfacing with simulation models of naval operations, but it is concluded that this interface must be managed on a "customized" basis for each simulation model due to the variety of approaches employed in naval systems modeling.

INTRODUCTION

The design and analysis of complex systems, especially those involving probabilistic or stochastic elements, often necessitates the use of digital computer simulation. A problem-solving procedure for defining and analyzing a model of a system, simulation affords the analyst the opportunity to evaluate complex systems, (a) without constructing them if they are proposed systems, (b) without interrupting their operation if they are actual systems, or (c) without damaging or destroying them if the purpose of simulation is to test the effectiveness of the system in a hazardous operating environment.

Naval operations present especially challenging problems for the simulationist. The purpose in simulating naval operations would very likely be to evaluate the effectiveness of certain offensive or defensive fleet configurations and tactics. Large numbers of controllable input variables, uncontrollable input variables, and measures of system effectiveness are often involved in realistic models of naval operations. This not only complicates model development, but hampers experimentation with the model as well.

Simulation can be defined as the establishment of a mathematical-logical model of a system and the experimental manipulation of that model on a digital computer. This paper concentrates on the latter aspect, particularly upon means of determining an "optimal" design through experimentation with a digital computer simulation model. (The word "optimal" is used guardedly here, since the randomness inherent in computer simulation contradicts the classical notion of an optimal solution.) This research has explored procedures for combining computer simulation, response surface methodology and mathematical programming to seek an estimated "optimal"

solution. These techniques are demonstrated and compared using simulation models of (1) a stochastic inventory system (2) a task duel, and (3) minefield evaluation.

BACKGROUND

A computer simulation model can be regarded as a "black-box", as illustrated in Figure 1, in which values for n controllable input variables x_i, $i=1,\ldots,n$ are combined in some manner to produce values for a set of m output or response variables $\eta_j, j=1,\ldots,m$. These responses are measures of merit or effectiveness for the system being modeled. It is typically true that the system responses are affected by a set of p uncontrollable factors, z_k, $k=1,\ldots,p$.

The execution of the computer simulation model, either for some fixed simulated time t or until r_j realizations of the j-th system response have been obtained, produces m simulated responses.

$$y_j = g_j(x_1,\ldots,x_n|z_1,\ldots,z_p) + \varepsilon_j, \quad j=1,\ldots,m \tag{1}$$

The quantity y_j is an estimate of the true system response η_j, and is found from the relation

$$y_j = \frac{1}{r_j} \sum_{\ell=1}^{r_j} \xi_{j\ell} \quad , \quad j=1,\ldots,m \tag{2}$$

where $\xi_{j\ell}$ is the individual value of the j-th system response at the ℓ-th realization and r_j is the number of realizations of this response. Equation (2) suggests that, for a given execution, or simulation trial, the r_j, $j=1,\ldots,m$ need not be the same. The simulation trial also produces an unbiased estimate of the variance of the j-th system response; that is,

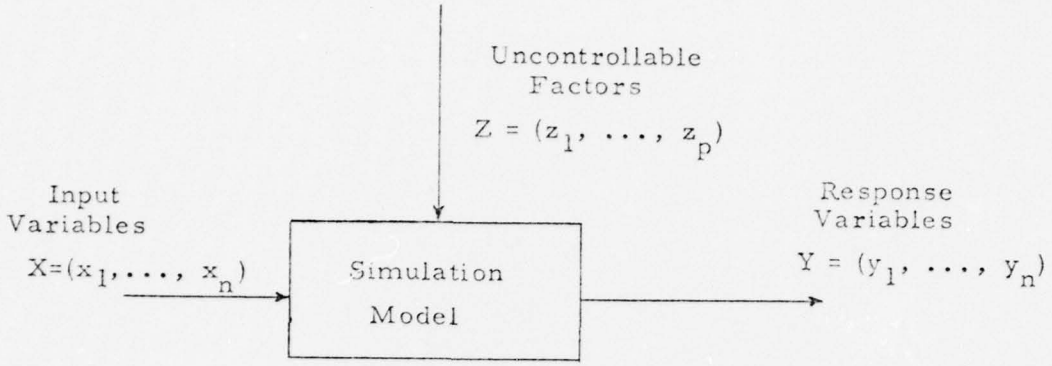


Figure 1. A "black-box" view of computer simulation

$$s_j^2 = \frac{1}{(r_j - 1)} \sum_{\ell=1}^{r_j} (\xi_{j\ell} - y_j)^2 \tag{3}$$

or more conveniently

$$s_j^2 = \frac{1}{(r_j - 1)} \left[\sum_{\ell=1}^{r_j} \xi_{j\ell}^2 - r_j y_j^2 \right] \tag{3a}$$

Fishman [12] and Kleijnen [19] describe other procedures for computing s_j^2 in the case of serial correlation of the relationships $\xi_{j\ell}$.

The structure of equation (1) must be carefully examined in order to develop the basic framework for the procedures presented here. It is stated that y_j is an estimate of the true response η_j ; hence,

$$y_j = \eta_j + \varepsilon_j \tag{4}$$

The ε_j term reflects the random variation inherent in the probabilistic system under study, and can be regarded as sampling error. The uncontrollable factors, the z_k, k=1,...,p terms in equation (1), can be viewed as contributing to the random variation ε_j. Thus equation (1) can be simplified to

$$y_j = g_j(x_1,\ldots,x_n) + \varepsilon_j, \qquad j=1,\ldots,m \tag{5}$$

The estimate y_j is therefore a random variable with mean

$$E(y_j) = \eta_j = g_j(x_1,\ldots,x_n), \quad j=1,\ldots,m \tag{6}$$

and variance

$$Var(y_j) = Var(\varepsilon_j), \qquad j=1,\ldots,m \tag{7}$$

This statement suggests that the true response η_j is a point on an unknown (n+1) - dimensional hypersurface $g_j(X)$ at a given point $X = (x_1,\ldots,x_n)$ and that the simulation-generated response y_j is a variate from a probability

distribution about the true value η_j at this point X.

The response surface methods employed in this methodology require the assumption and the random error ε_j be normally distributed with mean zero and variance σ_j^2; that is,

$$E(y_j|x_1,\ldots,x_n) = \eta_j \qquad j=1,\ldots,m \tag{8}$$

$$\text{Var}(y_j|x_1,\ldots,x_n) = \sigma_j^2, \qquad j=1,\ldots,m \tag{9}$$

It is also required that $\text{Var}(y_j|x_1,\ldots,x_n)$ be the same at any point on the (n+1)-dimensional hypersurface.

Note that there are m distinct (n+1)-dimensional hypersurfaces, each having its own characteristic random variation. The problem at hand is to have the computer simulation model which produces these m responses be either automatically or interactively controlled by an "optimization" procedure, as illustrated in Figure 2. In general, the "optimization" procedure must combine the experimental features of response surface methodology with the logical/numerical procedures of mathematical programming.

PROBLEM FORMULATIONS

We shall consider two basic approaches to formulating the problem of optimizing a multiple-variable, multiple-response simulation model. One approach is the familiar constrained optimization formulation in which one of the simulation responses, say η_1, is to be maximized or minimized, subject to maintaining the remaining m-1 responses within prescribed bounds. The second approach is the multiple-objective formulation, in which the m responses are either weighted to form a single objective or treated in a manner akin to goal programming. Each of these two formulations is described in the following sections.

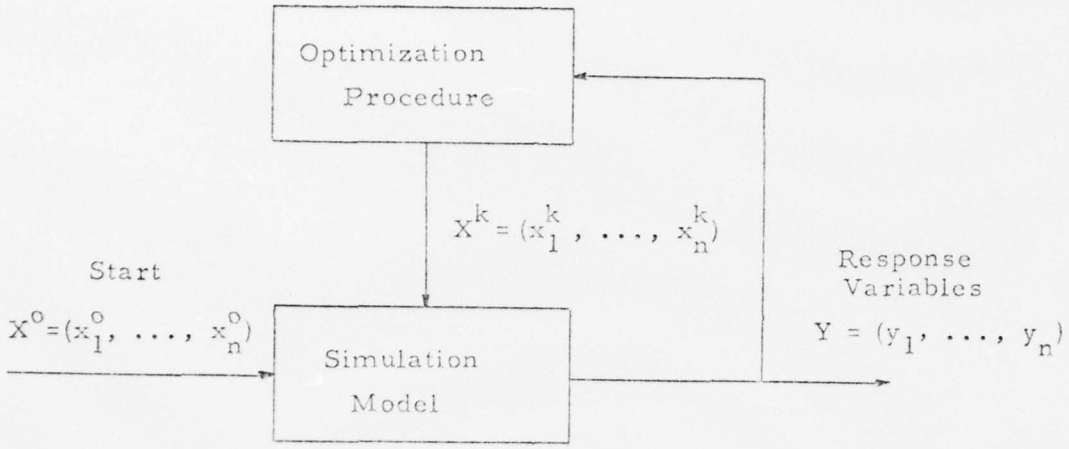


Figure 2. The interface between optimization and simulation

Constrained Optimization

Under the constrained optimization approach, the problem is stated as

$$\text{Max(or Min)} \quad \eta_1 = g_1(x_1,\ldots,x_n) \tag{10}$$

subject to the constraints

$$a_i \leq x_i \leq c_i, \quad i=1,\ldots,n \tag{11}$$

$$\eta_j = g_j(x_1,\ldots,x_n) \begin{Bmatrix} \geq \\ = \\ \leq \end{Bmatrix} d_j, \quad j=2,\ldots,m \tag{12}$$

The constraints expressed in (11) are bounds on the controllable input variables $x_1,\ldots,x_n$ and are typically known *a priori*. The bounds (11) generally form the known experimental region prior to conducting simulation trials. In contrast to that, the response functions $g_j(x_1,\ldots,x_n)$ in (12) are not usually known *a priori* and hence the responses η_j must be estimated experimentally via simulation. Thus, simulation trials performed at points satisfying (11) may yield responses violating (12).

To complicate matters even more, the random error ε_j can lead to erroneous decisions relative to feasibility with respect to the constraints in (12). The same is true relative to the objective function in (10). That is, one simulation trial can appear to represent an improvement over another when the true response at this particular set of values $x_1,\ldots,x_n$ does not. These difficulties can be countered through variance reduction techniques, as discussed by Fishman [12] and Kleijnen [19].

Multiple-Objective Optimization

One approach to a multiple-objective formulation is to assign weights $w_j, j=1,\ldots,m$ to the m responses and form a single objective function

$$\text{Max (or Min) } W = \sum_{j=1}^{m} w_j g_j(x_1,\ldots,x_n) \tag{13}$$

The bounds (11) still apply, so that the problem remains one of constrained optimization, but one in which the entire feasible region is known *a priori*. The weight $w_j, j=1,\ldots,m$ are typically assigned through the subjective judgment of the decision-maker in the system being simulated. These weights are usually normalized, so that

$$\sum_{j=1}^{m} w_j = 1 \tag{14}$$

One frequently encounters the situation in which certain of the responses η_j are to be maximized and other minimized. This case is handled by maximizing the negative of those functions which are to be minimized, so that the objective function in (13) is rearranged to the form

$$\text{Max } W = \sum_{j=1}^{s} w_j g_j(x_1,\ldots,x_n) - \sum_{j=s+1}^{m} w_j g_j(x_1,\ldots,x_n) \tag{15}$$

where s functions are maximized and m-s functions are minimized.

A second approach to the multiple-objective formulation is one which casts the objective function as a "utility function"; that is,

$$\text{Max } U[g_1(x_1,\ldots,x_n),\ldots,g_m(x_1\ldots,x_n)] \tag{16}$$

subject to the bounds in (11). The formulation in (15) is a special case of that in (16), in which $U[g_j(x_1,\ldots,x_n)]$ is a linear additive function. Montgomery and Bettencourt [21] discuss various formulations of the multiple objective optimization problem, as well as several approaches to its solution, and demonstrate its application to multiple-response simulation.

Another multiple-objective optimization formulation is that called

goal programming. This procedure is initiated by establishing a set of goals in terms of the $\underline{m}$ system responses. These goals are expressed as

$$G_j = g_j(x_1,\ldots,x_n),\ j=1,\ldots,m \tag{17}$$

Each goal must have an associated right-side value d_j; that is,

$$G_j = g_j(x_1,\ldots,x_n) \left\{ \begin{matrix} \leq \\ = \\ \geq \end{matrix} \right\} d_j,\ j=1,\ldots,m \tag{18}$$

With a slight modification, each goal can be expressed as an equality

$$G_j = g_j(x_1,\ldots,x_n) + n_j - p_j = d_j, j=1,\ldots,m \tag{19}$$

where n_j is a negative deviation from d_j, and p_j is a positive deviation. Either n_j or p_j must be zero in any given solution, and both could be zero. Next, each of the $\underline{m}$ goals G_j is assigned to a priority level P_k, $k=1,\ldots,\ell$, where P_1 represents the highest priority and P_ℓ the lowest. For any goal falling within a given priority level P_k, the decision-maker can weigh one goal relative to another. The final step in problem formulation is to combine these several levels of goals into an achievement function which has the form

$$A = \left\{ P_1(\bar{n}_1,\bar{p}_1),\ P_2(\bar{n}_2,\bar{p}_2),\ldots,P_\ell(\bar{n}_\ell,\bar{p}_\ell) \right\} \tag{20}$$

This achievement function is simply an ordered ℓ-vector. Its structure is predicated on one of the following procedures for achieving the j-th goal:

(a) To equal or exceed d_j, minimize n_j

(b) To equal or be less than d_j, minimize p_j

(c) To equal d_j, minimize $(n_j + p_j)$

A solution $(x_1^*,\ldots,x_n^*)$ is considered optimal if the corresponding value A^* is the same as or preferred to any other value A. Therefore, the general

goal programming problem is to find $x_1,\ldots,x_n$ so as to minimize the ordered vector (20) such that the goals (19) are satisfied and

$$\begin{aligned} x_i &\geq 0, \quad i=1,\ldots,n \\ n_j &\geq 0, \quad j=1,\ldots,m \\ p_j &\geq 0, \quad j=1,\ldots,m \end{aligned} \tag{21}$$

The functions $g_j(x_1,\ldots,x_n)$ in (19) are generally unknown, but are usually assumed to be nonlinear. Any technique proposed for solving this problem in the simulation domain must provide experimental estimates of these unknown functions, as well as a mathematical procedure for optimization. Moreover, the experimental observations are produced via simulation - each simulation trial at a point $x_1,\ldots,x_n$ produces m responses $y_j=1,\ldots,m$. Biles [4] has described the application of nonlinear goal programming to the multiple-response simulation problem, based on techniques proposed by Ignizio [17].

OPTIMIZATION TECHNIQUES

Various procedures have been applied in combining optimization and simulation to seek the "optimum" solution to systems possessing a single response η. The multiple-response problem described here is complicated by the necessity to observe several responses at once, and to incorporate these values into the optimization technique. But many of the same techniques that have been applied successfully to the single-response problem can, with appropriate modifications, be extended to accommodate multiple responses. Moreover, these modified procedures are often applicable to more than one of the aforementioned formulations of the multiple-response problem.

The optimization procedures described below fall into three categories: (1) direct search techniques, (2) first-order response surface methods, and

(3) second-order response surface procedures. Although numerous techniques will be cited, only a few broadly stated procedures will be outlined here. It should be remembered that, although we may refer to "optimization" techniques, the classical notion of an "optimum" solution is inapplicable due to the presence of the sampling error ε_j associated with each response variable η_j. Rather we shall seek a solution which hopefully lies close to the true solution. In a more formal sense, we might state that we are to some degree confident, say 90%, that the true solution lies within some interval about our estimated solution.

Direct Search Methods

Direct search methods are those which, applied in a purely computational manner, do not require the use of derivatives. These methods progress through a sequence of points according to some algorithm. Typical of this class of optimization techniques are the pattern search algorithm by Hooke and Jeeves [14], sequential simplex search by Spendley, Hext and Himsworth [26], and the so-called "complex" search method by M. J. Box [8]. In general, these direct search procedures make rapid early progress toward an "optimum", but iterate laboriously as a solution is neared. This is particularly true in the presence of random error, as encountered in simulation.

Among the direct search techniques, Box's "complex" method [8] has been found to be the easily adapted to a multiple-response environment. It also performs better than any of the other direct search techniques in the face of random error and constraints. In fact, "complex" search is not at all complex, but derives its name from a contraction of the words "constrained simplex": it evolved from the sequential simplex method [26] and the necessity to deal with constraints. This report describes a modification of Box's method which makes it especially suitable for the multiple-response

simulation problem. The following procedure describes a generalized "complex" procedure as it might be applied to the multiple-response simulation problem:

1. Randomly generate a set of $N \geq n + 2$ search points $X^1,\ldots,X^N$ satisfying the known bounds (11).
2. Perform a simulation trial at each of these N search points and record the mN estimated responses y_j^ℓ, $j=1,\ldots,m$, $\ell=1,\ldots,N$.
3. Where a given point X^k is observed to violate one or more constraints, if such constraints apply with the particular problem formulation being employed, generate a replacement search point $X^{k'}$, perform a simulation trial at $X^{k'}$, and record the m estimated response at $X^{k'}$.
4. After N feasible search points have been established, evaluate the objective function for each of these N points. This "objective function" might be η_1 in (10), W in (15), U in (16), or A in (20). Among these N search points, find the worst point X^W; that is, the search point giving the least desirable value of the objective function. Define X^C as the centroid of the N-1 remaining points. Project from X^W through X^C to the image point $X^{W'}$. If the known bounds (11) are violated by this move, shorten the step to $X^{W'}$ until no violation occurs. Perform a simulation trial at $X^{W'}$.
5. Repeat steps 3 and 4 until a solution (X^*,Y^*) is obtained which represents the best solution that can be achieved within the available computer time. Figure 3 illustrates Box's complex search as applied to a constrained problem. Figure 4 gives a

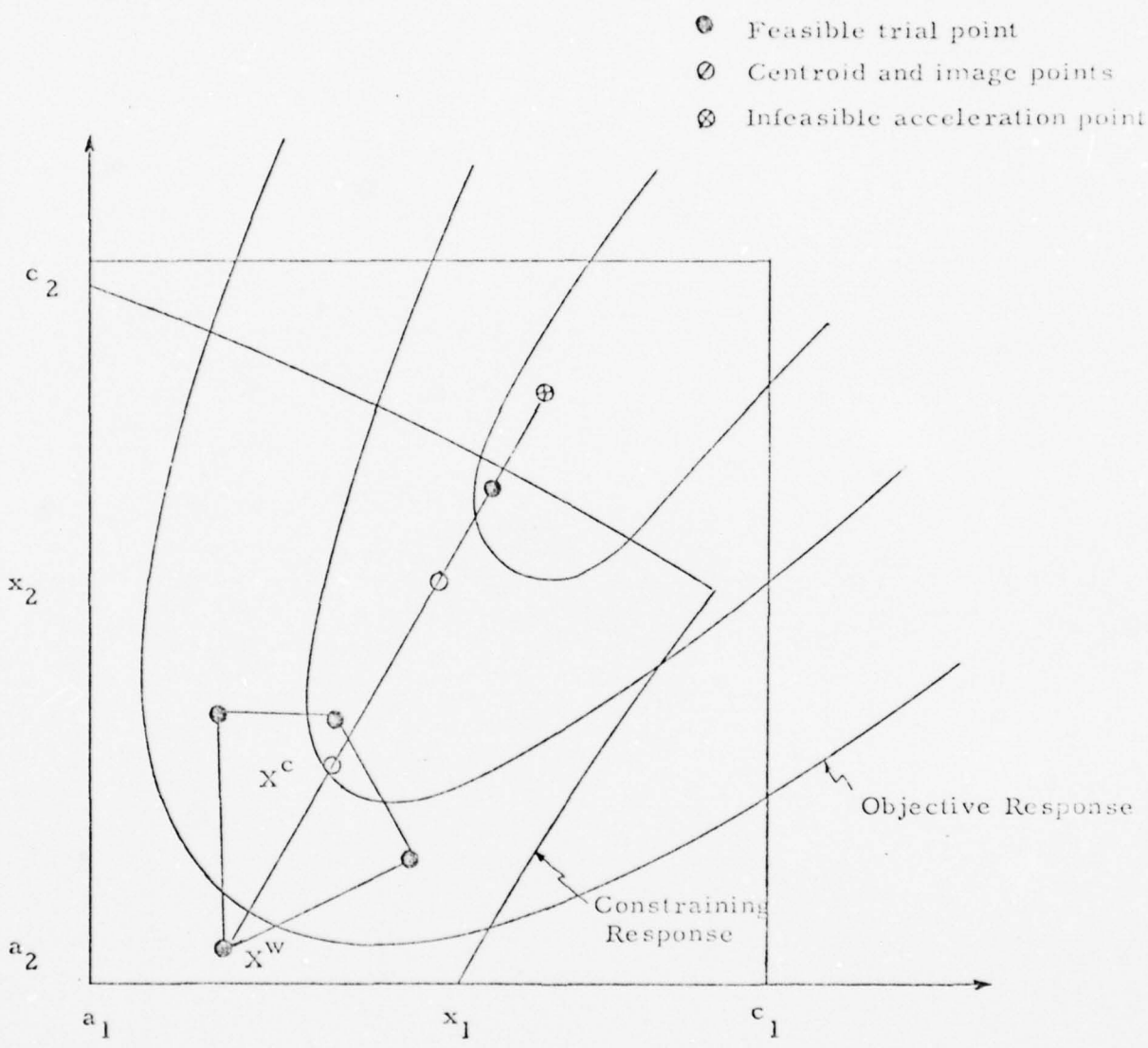


Figure 3. Complex Search

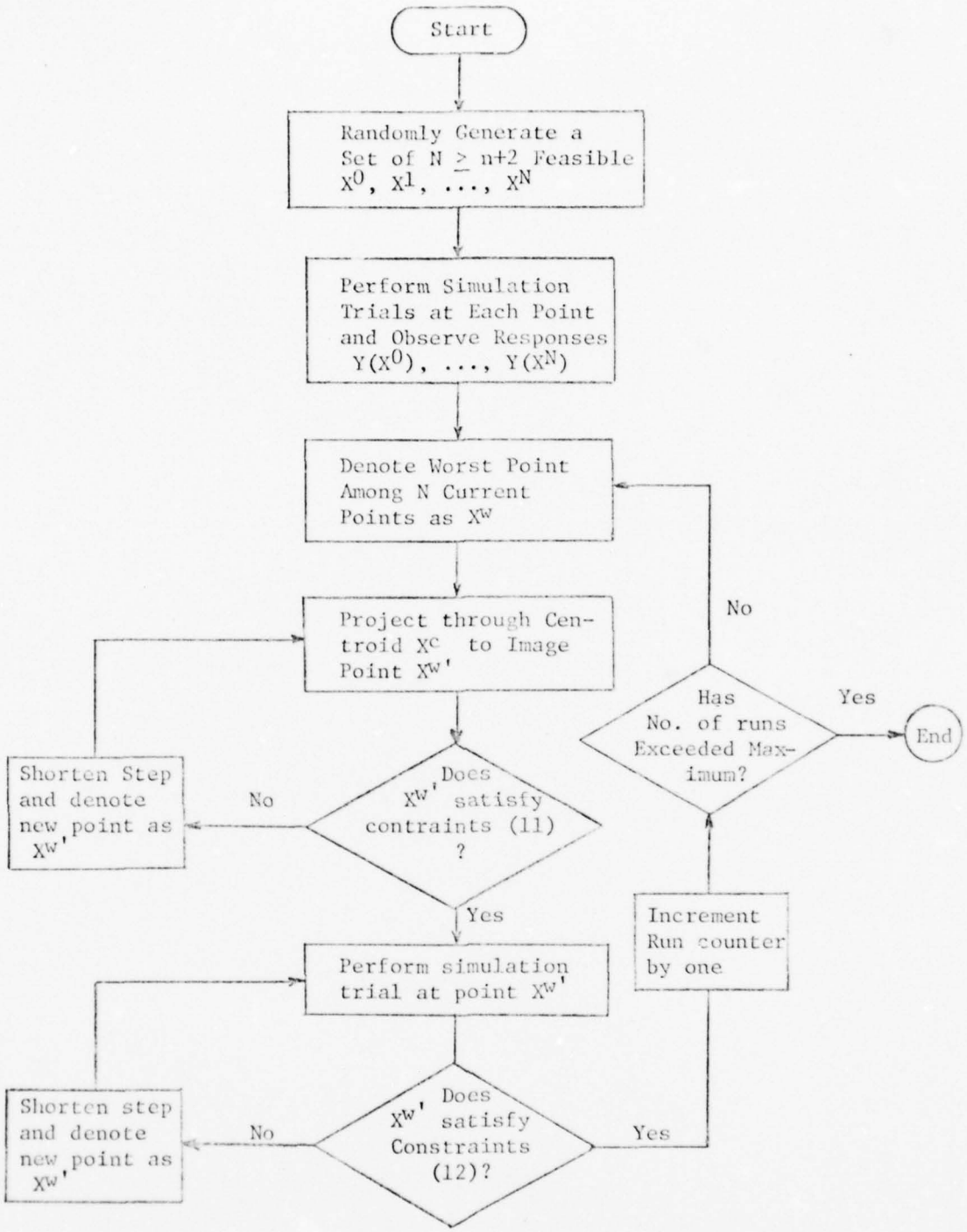


Figure 4. A multiple-response complex search procedure

a flow chart for a simulation/optimization method based on the procedure outlined above. Appendix A contains the program listing for a constrained optimization formulation of the Box complex search method.

A significant advantage of complex search is that, once an initial "complex" of N feasible simulation trials (and perhaps several infeasible trials) have been performed, trials are conducted one at a time thereafter. The search can be continued as long as improved solutions are obtained. If several successive simulations are performed at scattered points around the known experimental region without achieving an improved solution, however, the search can be terminated and the best available solution adopted.

First-Order Response Surface Methods

First-order response surface methods attempt to accomplish experimentally what the "method of steepest ascent" accomplishes computationally. From a current point X^k, a designed experiment is conducted (with a simulation trial at each design point) to estimate the gradient direction $\nabla g(X^k)$. Simulation trials are then conducted at points along this direction to a new point X^{k+1} which represents the best solution obtained along the direction $\nabla g(X^k)$. This process is an experimental approximation of

$$X^{k+1} = X^k + \lambda^k[\nabla g(X^k)] \tag{22}$$

The step length λ^k can be estimated by a line search or by a regression procedure as described by Biles [2,3].

The gradient direction $\nabla g(X^k)$ is estimated by placing an appropriate first-order experimental design, such as a 2^n factorial, 2^{n-p} fractional factorial, or n-dimensional simplex design (see Myers [22]) around the current point X^k. A simulation trial is performed at each point in the

selected experimental design. From these N observations the multiple linear regression model

$$\hat{y} = b_o + \sum_{i=1}^{n} b_i x_i \tag{23}$$

can be estimated (see Draper and Smith 10]). Since the gradient direction $\nabla g(X^k)$ is mathematically defined as the n-vector of first partial derivatives of $g(X)$ evaluated at X^k, it is clear that $\nabla g(X^k)$ is simply the n-vector of regression coefficients exclusive of the b_o term; that is,

$$\nabla g(X^k) = (b_1,\ldots,b_n)' \tag{24}$$

In the multiple-response simulation problem, a simulation trial is conducted at each design point in the selected first-order design and the $\underline{m}$ observations y_j^{ℓ}, $j=1,\ldots,m$ are recorded at each design point. Multiple linear regression is applied separately to each set of observations (assuming independence among the $\underline{m}$ responses), producing the $\underline{m}$ models

$$\hat{y}_j = b_{j,0} + \sum_{i=1}^{n} b_{j,i} x_i, \qquad j=1,\ldots,m \tag{25}$$

and hence the $\underline{m}$ gradient vectors

$$\nabla g_j(X^k) = (b_{j,1},\ldots,b_{j,n})' , \quad j=1,\ldots,m \tag{26}$$

These estimates can then be employed in any one of several optimization schemes to produce an improved solution X^{k+1}. A generalized procedure for accomplishing this improved solution, and an estimated "optimum", will be described later. But first it is necessary to give attention to the experimental designs employed to estimate the gradient vectors $\nabla g_j(X^k)$, $j=1,\ldots,m$.

In selecting a first-order response surface design, it is usually desirable to minimize the variances of the regression coefficients $b_i, i=1,\ldots,n$. To accomplish this the first-order experimental design should be orthogonal.

That is, the placement of the N experimental points (in our case, simulation trials) is described by the N by n design matrix D, where

$$D = \begin{vmatrix} x_{11} & x_{21} & \cdots & x_{n1} \\ x_{12} & x_{22} & \cdots & x_{n2} \\ . & & & \\ . & & & \\ . & & & \\ x_{1N} & x_{2N} & \cdots & x_{nN} \end{vmatrix} \tag{27}$$

Then an N by n+1 matrix X is constructed by placing a unit vector to the left of D. Thus,

$$X = \begin{vmatrix} 1 & x_{11} & x_{21} & \cdots & x_{n1} \\ 1 & x_{12} & x_{22} & \cdots & x_{n2} \\ . & & & & \\ . & & & & \\ . & & & & \\ 1 & x_{1N} & x_{2N} & \cdots & x_{nM} \end{vmatrix} \tag{28}$$

It is usually convenient to code the design levels, so that the following conditions are achieved:

$$\sum_{u=1}^{N} x_{iu}^2 = N$$

$$i=1,\ldots,n \tag{29}$$

$$\sum_{u=1}^{N} x_{iu} = 0$$

If the actual value of the u-th level of the i-th variable is z_{iu}, then the corresponding coded values is

$$x_{iu} = \frac{z_{iu} - z_{iu}}{S_i} \tag{30}$$

where

$$\bar{z}_i = \left(\sum_{u=1}^{N} z_{iu} \right)/N \tag{31}$$

and

$$S_i = \sum_{u=1}^{N} (z_{iu} - \bar{z}_i)^2/N \tag{32}$$

Then

$$X'X = \begin{vmatrix} N & 0 & 0 & \dots & 0 \\ 0 & N & 0 & \dots & 0 \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ 0 & 0 & 0 & \dots & 0 \end{vmatrix} \tag{33}$$

Since the (n+1) - vector of regression coefficients $\bar{b}$ is estimated by the least squares relation

$$\bar{b} = (X'X)^{-1} X' \bar{y} \tag{34}$$

where $\bar{y}$ is the N-vector of response estimates obtained from N simulation trials, the variance of the regression coefficients b_i, i=1,...,n is given by

$$\mathrm{Var}(b_i) = \sigma^2/N, \qquad i=1,\dots,n \tag{35}$$

where σ^2 is the variance of the error term ε. Since we are interested in m separate system response y_j, j=1,...,m, equations (34) and (35) can be generalized to

$$\bar{b}_j = (X'X)^{-1} X' \bar{y}_j, \qquad j=1,\dots,m \tag{36}$$

$$\mathrm{Var}(b_{ji}) = \sigma_j^2/N, \qquad \begin{matrix} i=1,\dots,n \\ j=1,\dots,m \end{matrix} \tag{37}$$

For an orthogonal first-order design, the results in (33)-(37) hold, giving a so-called "minimum-variance" design. The 2^n factorial and 2^{n-p} fractional factorial designs are orthogonal and hence minimum variance. Orthogonal n-simplex designs can be easily constructed (see Myers [22]). Since n-simplex designs provide the minimum number of design points needed to estimate the multiple-linear regression models in (23) or (25), and are hence the most "economical" of the first-order response surface designs, they are especially attractive for the purpose proposed here. Brooks and Mickey [9] concluded that n-simplex designs offer the most efficient approach to estimating the gradient direction $\nabla g_j(X)$. Figure 5 illustrates 2^n factorial and n-simplex designs.

Biles [2,3] has described a first-order response surface procedure for approaching the constrained formulation of the multiple-response simulation problem. This procedure involves performing a first-order design around a current point X^k to estimate the gradient direction $\nabla g(X^k)$ according to relation (24). A line search is then performed along $\nabla g(X^k)$ to estimate an optimal step λ in (22). As long as the search remains interior to the region bounded by the constraints (11) or (12), the procedure is basically the same as that proposed by Box and Wilson [6]. If one or more constraints (11) or (12) are encountered, however, Biles [2,3] proposes that the gradient projection direction be followed (see Rosen [23]). The procedure for estimating the gradient projection direction is as follows.

Suppose that at an estimated boundary point X^k, $\underline{q}$ constraints are satisfied as equalities. These can be either the (11) or (12) constraints, or both. Let B_q be the n x q matrix of first partial derivatives of these active constraints. Thus, B_q consists of the $\underline{q}$ gradient vectors $\nabla g_j(X^k)$, $j=1,\ldots,q$. That is,

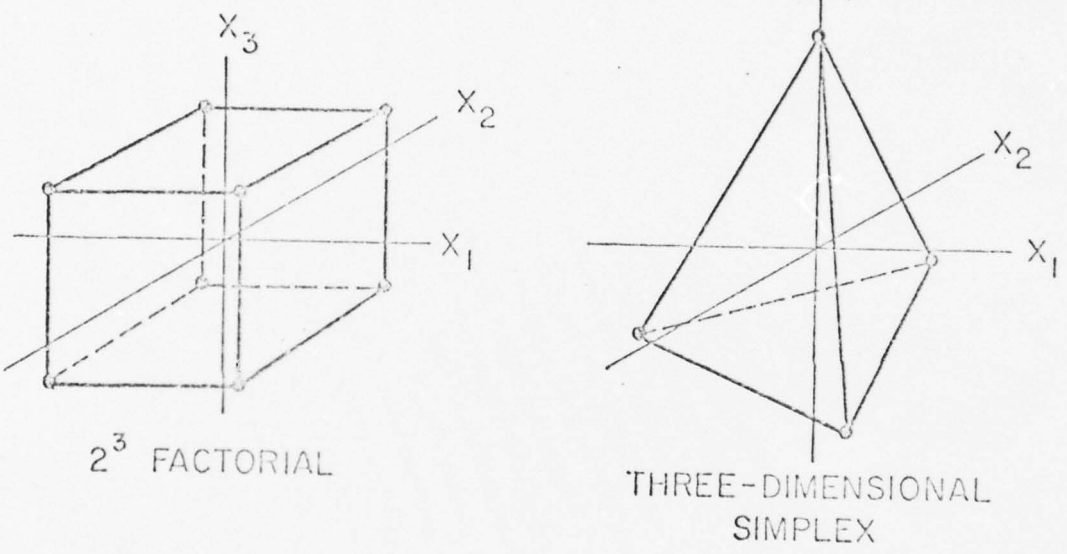


Figure 5. First-Order Response Surface Designs.

$$B_q = \begin{vmatrix} \partial g_1/\partial x_1 & \cdots & \partial g_q/\partial x_1 \\ \vdots & & \vdots \\ \partial q_1/\partial x_n & & \partial g_q/\partial x_n \end{vmatrix} \tag{38}$$

Since $g_j(X)$, $j=1,\ldots,q$ denotes the set of binding constraint functions (a constraint (11) or (12) is said to be "binding" if it is satisfied at the equality), for the moment let $f(X)$ represent the objective function. Then $\nabla f(X^k)$ and $\nabla g_i(X^k)$, $j=1,\ldots,q$ represent the gradient vectors of the objective and constraint functions, respectively, evaluated at the boundary point X^k.

Performing a first-order response surface experiment about the boundary point X^k yields estimates of the gradient vectors $\nabla f(X^k)$ and $\nabla g_i(X^k)$, $j=1,\ldots,q$ in the form of the vectors of regression coefficients. (If a constraint of type (11) is included in the set of binding constraints, the gradient vector has the form $(0,0,\ldots,1,\ldots,0)'$, where all elements are zero except the i-th element which is one). Following the procedure outlined by Rosen [23], the gradient projection direction is given by

$$S^k = [\nabla f(X^k)] - B_q (B'_q B_q)^{-1} B'_q [\nabla f(X^k)] \tag{39}$$

A "golden section" line search is performed along direction S^k until either (a) a local "optimum" is found, or (b) other constraints are encountered. This new point is denoted X^{k+1}. This procedure is repeated until the gradient projection direction S^k is approximately zero. This point X^* is taken as a "constrained optimal" solution. Figure 5 illustrates the application of the gradient projection procedure to a constrained optimization problem.

Swain [27] has compared other first-order response surface techniques, including those of Klingman and Himmelblau [20] and Zoutendijk [28], to

Rosen's gradient projection method [23]. He found little difference among these techniques in terms of experimental requirements, and hence computer simulation time, but saw significant variability in computational requirements in order to use these algorithms. The Zoutendijk methods of feasible directions [28] require greater computational effort than the other procedures.

Biles [4] demonstrated both first-order and second-order approaches to a nonlinear goal programming formulation of the multiple-response simulation problem. These approaches are based on Ignizio's adaptation [17] of the method of Griffith and Steward [13] to goal programming and, like the constrained procedures mentioned previously, combine simulation, experimental design and mathematical programming.

The following generalized procedure is followed in employing a first-order response surface approach to the multiple-response simulation problem. The particular problem formulation and optimization procedure will govern the precise sequence of steps in implementing this procedure.

1. Identify the known experimental region $a_i \leq x_i \leq c_i$, i=1,...,n. Selecting a starting point X^0 within this region. With X^0 as its center, array an orthogonal first-order response surface design within a selected design radius. Place $n_c = n/2 \geq 2$ points at the design center X^0 (coded as the $\bar{0}$ - vector).
2. Perform simulation trials at each of the N experimental design points and record the responses y_j^{ℓ}, j=1,...,m; ℓ=1,...,N. Using multiple linear regression, fit linear models of the form (23) and (25).
3. Apply the appropriate mathematical programming technique to locate the next center point in the search.
4. Repeat steps 1-3 until an "optimum" solution is located. It may be appropriate to add design points to complete a second-order response surface design to test this optimum solution. The procedure for

accomplishing this is described in the next section.

Figure 6 illustrates a sequential first-order response surface method applied to a constrained system. Figure 7 gives a flow chart for a simulation/optimization method based on the procedure outlined above. Appendix B gives a program listing for this procedure.

Second-Order Response Surface Methods

A second-order response surface approach to the multiple-variable, multiple-response simulation problem consists of one or more repetitions of a two-stage procedure: (a) the execution of a computer simulation trial at each point in a second-order response surface experimental design covering the known region given by (11) and the use of multiple linear regression to fit second-order regression models to the resulting data; and (b) the application of a suitable mathematical programming procedure to obtain a solution to the problem formulated in (10)-(12), in (15) together with (11), in (16) together with (11), or in (19)-(21). In contrast to the first-order methods, in which the optimization procedure was part and parcel with the experimental procedure, these procedures are distinct and sequential in the proposed second-order approaches.

The first step in the second-order approach is to identify the range of each input variable. A safe strategy is to cover the entire known region $a_i \leq x_i \leq c_i$, i=1,...,n with the first (and possibly only) experimental design. If we let α_i denote the radius of the n-dimensional hypersphere within which the design points are contained, then

$$\alpha_i = (c_i - a_i)/2, \qquad i=1,\ldots,n \tag{40}$$

is effectively the maximum radius we could construct. It is convenient to adopt the coding convention expressed in (29) - (30), but choosing x_{iu} in such a way that α_i satisfies (40). Myers [22] describes this coding process.

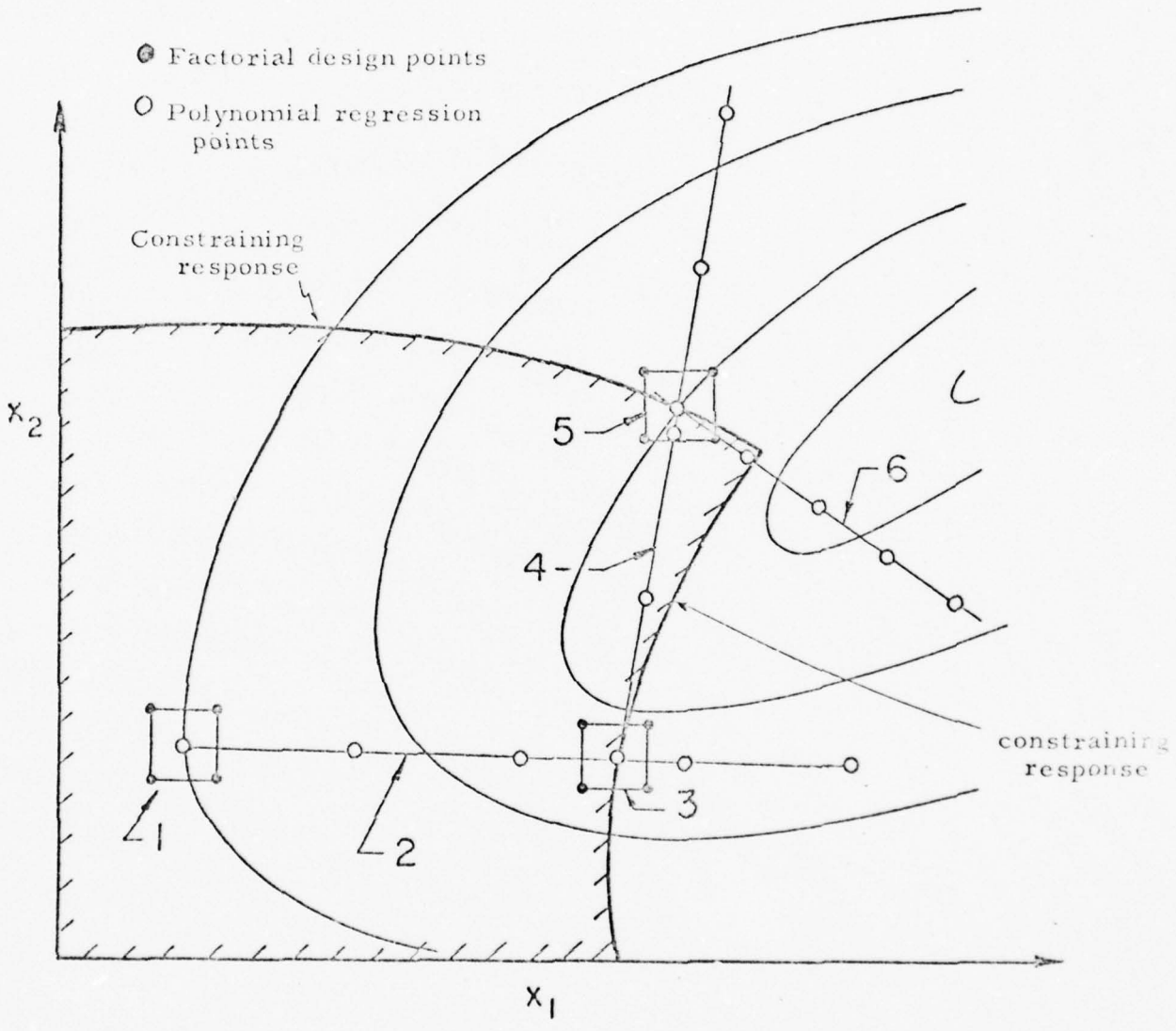


Figure 6. Gradient Projection Search.

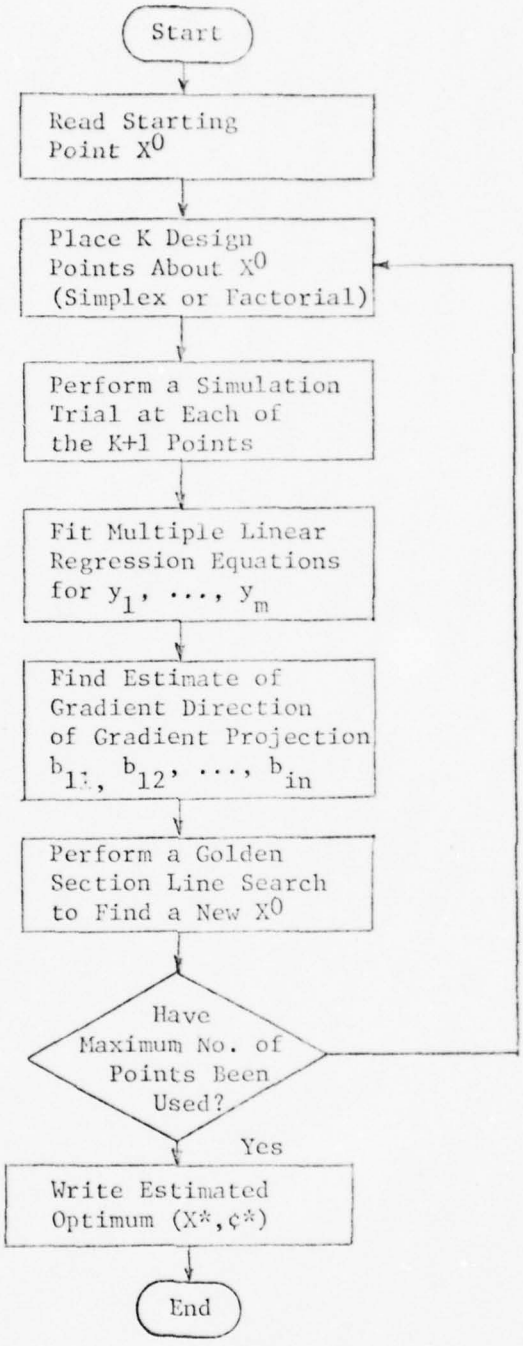


Figure 7. Flow Chart of Gradient Projection Search

The second-order fitted response surface has the form

$$\hat{y} - b_o + \sum_{i=1}^{n} b_i x_i + \sum_{i=1}^{n} b_{ii} x_i^2 + \sum_{i=1}^{n} \sum_{j=1}^{n} b_{ij} x_i x_j \qquad (41)$$

where $\hat{y}$ is the estimate of the true response η at a given value $X = (x_1,\ldots,x_n)$ and the b_i and b_{ij} are regression coefficients in the fitted model. Since we must estimate $\underline{m}$ separate response relationships, equation (41) is modified to

$$\hat{y}_k = b_{k,o} + \sum_{i=1}^{n} b_{k,i} x_i + \sum_{i=1}^{n} b_{k,ii} x_i^2 + \sum_{i=1}^{n} \sum_{\substack{j=1 \\ i \neq j}}^{n} b_{k,ij} x_i x_j \qquad (42)$$

$$k = 1,\ldots,m$$

Given the independence of the $\underline{m}$ responses, these m regression equations can be estimated independently from a set of $N \geq (n+1)(n+2)/2$ data points obtained by performing a simulation trial at each point in a second-order response surface design.

An experimental design employed for the purpose of estimating the regression coefficients in (42) must contain at least as many design points as there are coefficients b_i and b_{ij} in the fitted model, of which there are $(n+1)(n+2)/2$. Because of the non-linearity of (42), the experimental design must also have at least three levels of each controllable variable $x_i, i=1,\ldots,n$. It is also desirable to have a design which is rotatable; that is, the predicted response $\hat{y}$ at some point X is a function only of the distance from the design center to X and not a function of the direction.

The most widely used design for fitting a second-order model is the central composite design, shown in Figure 8 for n=2 and n=3. These designs consist of a 2^n factorial (or suitable fraction thereof), augmented by 2n axial points and k center points. A central composite design can be made

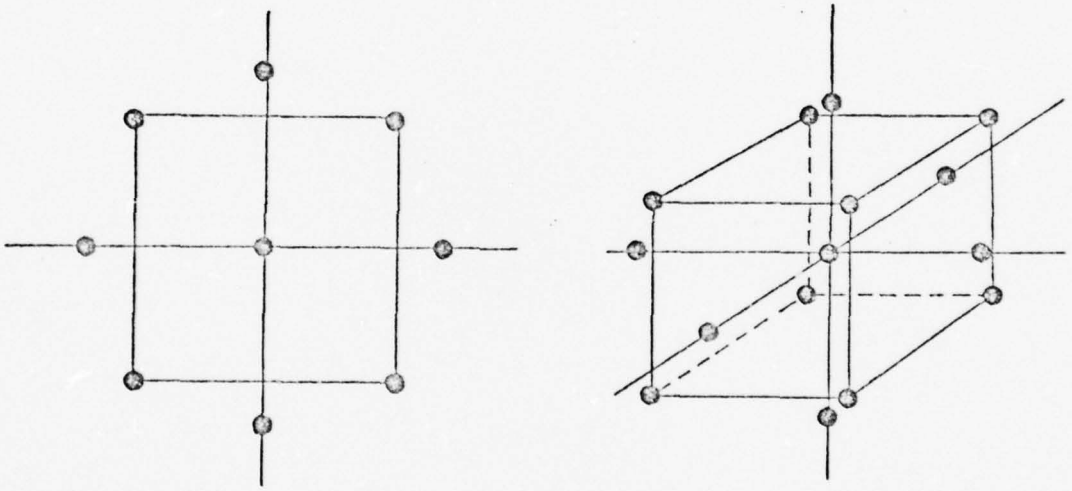


Figure 8. Central Composite Designs for Second-Order Response Surfaces.

rotatable by proper choice of α, the distance of the axial points from the design center. With the proper choice of the number of center points k, the central composite design can be made either orthogonal or uniform precision. Box and Hunter [7] give the characteristics of these designs for various sizes of n.

Another important class of second-order response surface designs is the equiradial designs. That is, $N \geq (n+1)(n+2)/2$ design points are placed at points on an n-dimensional hypersphere. Figure 9 illustrates two equiradial designs for two variables. It is important to note that each of these designs is also equiangular, in that the N points are placed in such a way as to form N equal angles from the design center. Biles and Swain [5] have demonstrated the construction of such designs based on orthogonal n-simplex designs, thus achieving an efficient rotatable design.

Having estimated the m second-order regression equations (42) and formulated the appropriate optimization problem, it remains to apply mathematical programming to obtain a solution. For the constrained formulation, any of the following procedures could be employed: (a) Box's complex search [8]; (b) Rosen's gradient projection method [23]; or (c) one of Zoutendijk's methods of feasible directions [28]. For the weighted objective function formulation, these same three procedures are applicable. For the goal programming formulation, Ignizio's [17] procedure based on the method of Griffith and Stewart [13] is computationally efficient.

Figure 10 illustrates a complex search applied computationally to second-order response surfaces. Figure 11 gives a flow chart for a simulation/optimization method which employs the computational version of Box's complex search applied to second-order regression equations estimated from data obtained from simulation trials performed at each point in a second-order

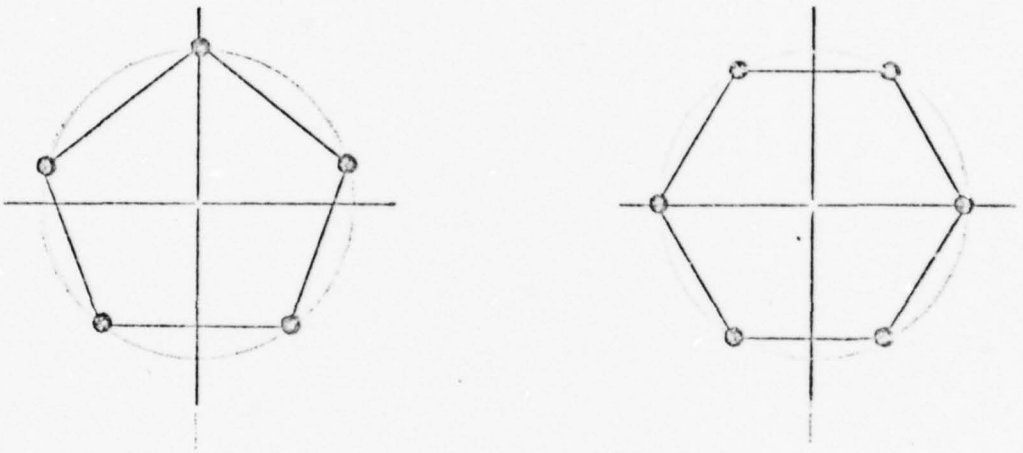


Figure 9. Equiradial Designs for Second-Order Response Surfaces.

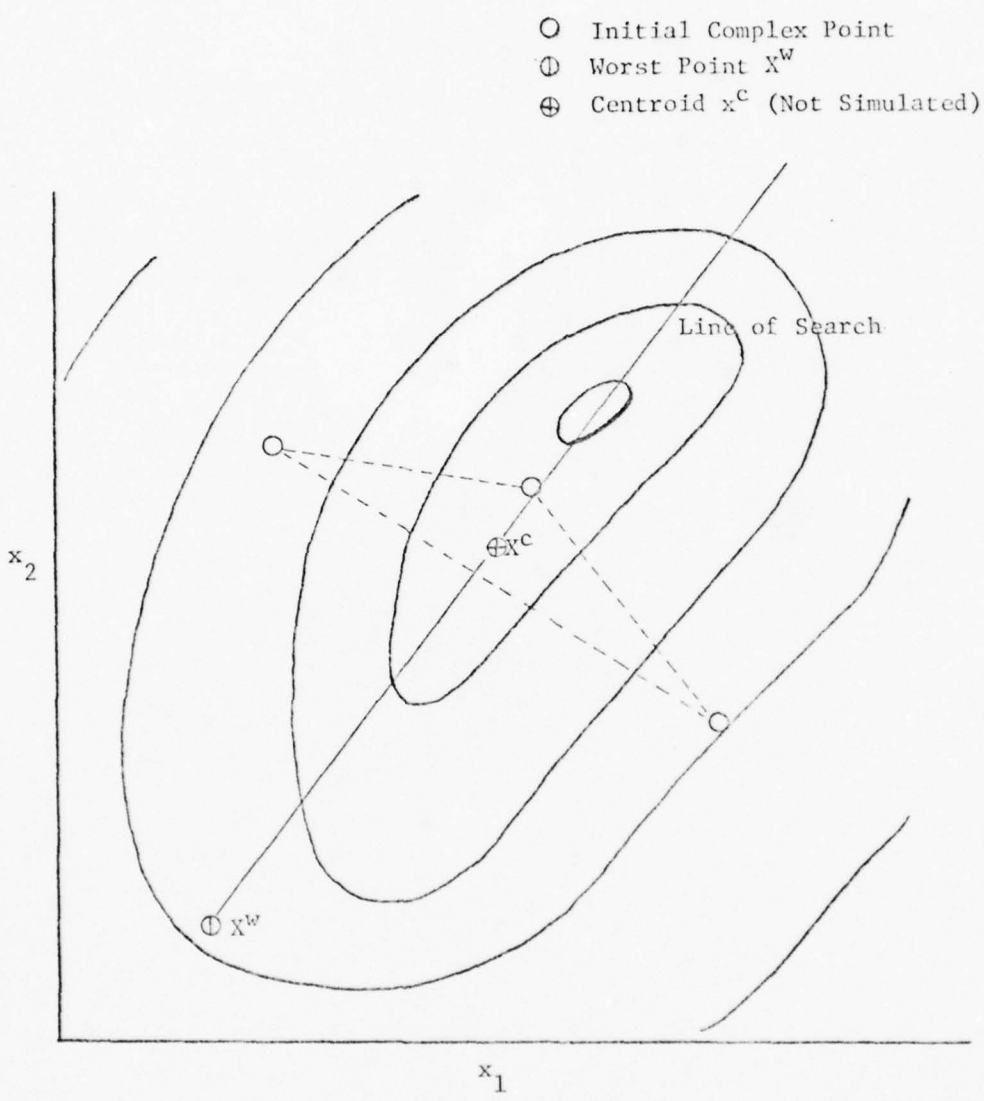


Figure 10. Complex Search with a Second-Order Surface.

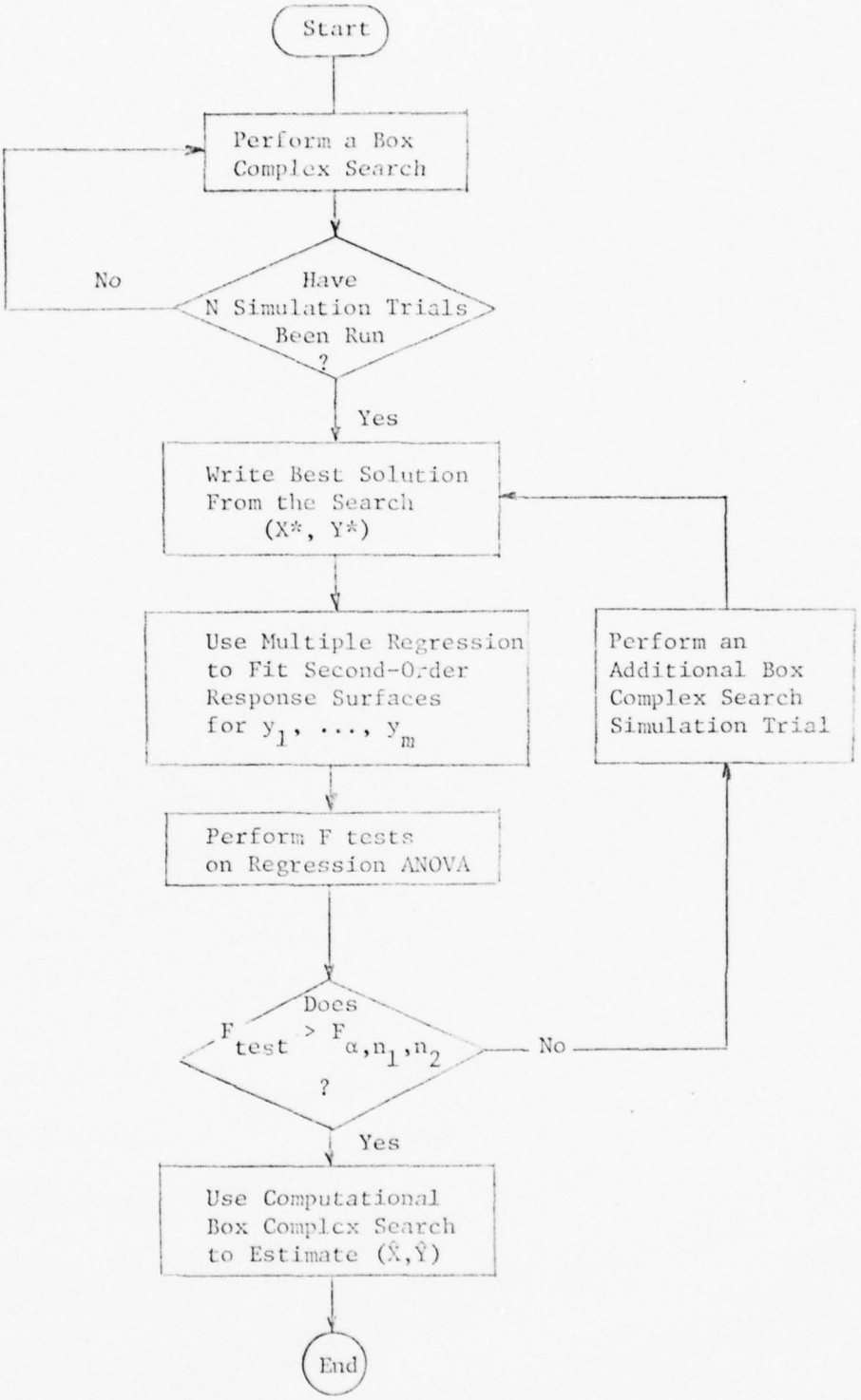


Figure 11. Flow-Chart of a Technique for Fitting Second-Order Response Surfaces to Complex Search Points

response surface design. Appendix C gives the program listing for this procedure.

Once an "optimal" solution has been obtained, it is necessary to compute a confidence interval, say at 90%, about the predicted "optimum." That is, we say that

$$P[y_{j\ell} \leq \eta_j^* \leq y_{ju}] = 0.9, \quad j=1,\ldots,m \tag{43}$$

where $y_{j\ell}$ and y_{ju} are lower and upper bounds, respectively, on the 90% confidence interval for response η_j^* at X^*. Draper and Smith [10] give the procedure for computing this confidence interval. If the range $(y_{ju}-y_{j\ell})$ is found to be excessively large, it may be necessary to perform confirmation simulations in the vicinity of the predicted "optimum," or even an entire repetition of the second-order design in a reduced experimental region about the predicted "optimum."

Second-Order Response Surface Analysis of Box's Complex Search Experiments

Another approach to the simulation/optimization problem is to combine second-order response surface analysis with Box's complex search. Such a procedure would be applicable to any of the problem formulations described earlier. The general procedure for this method is as follows:

1. Perform a set of Box's complex search experiments as described in the first procedure.
2. Using the simulation responses at each search point, fit second-order regression equations.
3. Apply Box's complex search computationally to these equations.

The advantages of this two-phase procedure over either of its "component" techniques are that (1) it provides an objective stopping mechanism for the experimental Box complex search, and (2) it extracts latent information

about the unknown response functions $\eta_j(X)$, $j=1,\ldots,m$ that the search alone would not discover. The following is a description of this simulation/optimization procedure.

The Box's "complex" search phase of simulation experimentation proceeds as follows:

1. Randomly generate a set of $N \geq n + 2$ search points satisfying the known bounds $a_i \leq x_i \leq c_i$, $i=1,\ldots,n$.
2. Perform a simulation trial at each of these N search points X^k, $k=1,\ldots,N$ and record the mN simulated responses y_j^k, $j=1,\ldots,m$, $k=1,\ldots,N$.
3. Where a given search point X^k is found to violate a constraint of the form

$$y_j^k \gtreqless \eta_j(X^k) \quad \begin{matrix} \geq \\ = \\ \leq \end{matrix} \quad d_j, \; j=1,\ldots,m \tag{6}$$

generate a replacement point $X^{k'}$, perform a simulation trial at $X^{k'}$, and assess the feasibility of these new responses $y_j^{k'}$, $j=1,\ldots,m$. Repeat this step until exact N search points satisfying both the known bounds and any constrained responses (6) are obtained.

4. Identify as X^w that search point yielding the least desirable set of responses y_j^w, $j=1,\ldots,m$. Compute the centroid X^c of the N=1 remaining search points. Project X^w through the centroid X^c to a new point. If this point satisfies the known bounds, perform a simulation trial to obtain the response $y_j^{w'}$, $j=1,\ldots,m$ at $X^{w'}$. If any responses fail to satisfy the constrained responses (6), shorten the step from X^w through X^c, and repeat this step until a feasible new point $X^{w'}$ is obtained. This point replaces X^w in the "complex".

5. Repeat step 4 until either of the following conditions occurs:

   (a) an estimated solution is obtained at $(\hat{X},\hat{Y})$; or

   (b) a predetermined number of search points K (both those satisfying (6) and those not), is obtained, where $K \geq (n+1)(n+2)/2$.

When the complex search phase of simulation experimentation has been terminated, a best available, feasible solution $(\hat{X},\hat{Y})$ is retained for future reference. An examination is made of the arrangement of search points X^k, k=1,...,K in the known experimental region $a_i \leq x_i \leq c_i$, i=1,...,n to detect any regions that have a sparse density of search points. Any such region will have a search point placed at or near its center and a simulation trial performed there. This process guarantees coverage of the entire experimental region. It is also worthwhile to compute the centroid of the T total search points and to perform R replicates of a simulation trial (with different sets of initial random number seeds) at this centroid point. This total set of W=T+R simulation trials comprises a random experimental design in the factor space $a_i \leq x_i \leq c_i$, i=1,...,n with something approaching the minimum number of points for estimating second-order response surfaces. This minimum number of points is (n+1)(n+2)/2, corresponding to the number of regression coefficients in the model (42). Multiple linear regression is then employed to fit each of the m second-order surfaces to the W sets of responses y_j^k, j=1,...,m, k=1,...,W. A Box's complex search procedure is applied computationally to this system of m second-order response surfaces according to the particular problem formulation. The solution obtained by this procedure (X^*,Y^*) is compared to that obtained by complex search $(\hat{X},\hat{Y})$. If (X^*,Y^*) is preferred to $(\hat{X},\hat{Y})$ and X^* and $\hat{X}$ differ significantly, one or more confirmation simulation trials must be performed at (X^*,Y^*) and possibly



Figure 12. Second-Order Surface Fitted to a Sequence of Complex Search Points.

at $(\hat{X},\hat{Y})$ as well to enable a choice of the "optimum" solution.

Figure 12 illustrates a second-order response surface fitted to a set of points obtained via an experimental complex search, and then a computational complex search applied to this surface. Figure 11 gives a flow chart for this simulation/optimization procedure. Appendix D is a program listing of this procedure.

SIMULATION TEST MODELS

Three simulation test models were used to evaluate each of the four optimization methods. These simulation models are as follows:

1. A stochastic inventory system as described in Ignall [16] and Hunter and Naylor [15].
2. A tank duel model as described in Montgomery and Bettencourt [21].
3. A naval minefield evaluation model as described by Bailey and Weedon [1].

Exploratory work was performed but not completed on a fourth simulation model, the SPEARS anti-aircraft Naval defense model by Kaplan et al [18]. These models are discussed in detail in the following sections, but test results are discussed later in this report.

Stochastic Inventory System

Ignall [16] and Naylor and Hunter [15] describe the application of experimental design to the optimization of computer simulation responses. They employed as a simulation test model a discrete-event simulation of a stochastic inventory system in which mean daily demand and order lead time are random variables with known probability density (mass) functions. The lone simulation response was

y = mean daily cost, $

which was the sum of carrying, setup and shortage costs. The two controllable variables were

x_1 = reorder point (ROP),

x_2 = economic order quantity (EOQ),

The optimization problem was

minimize $y = \eta(x_1, x_2)$

subject to

$$-5 \leq x_1 \leq 90 \qquad 50 \leq x_2 \leq 250$$

Ignall [16] found as a solution y = \$76 at $x_1 = 45$ and $x_2 = 175$ units, for a given set of values for the several constants in the model.

The manner in which this simulation test model was utilized for the current research did not involve an actual simulation program. Rather, the 20 x 9 response table was used as the simulator. This response table is shown in Figure 13. For a given (x_1, x_2), an interpolation was performed in this response table to find y. This enabled a comparison of the four optimization procedures without introducing a confounding influence from the simulation model. It should also be noted that, since the "simulation" produced a single response y, this simulation test model did not fully test the optimization procedure but principally evaluated the interface between the simulator and optimization modules.

Tank Duel Model

In describing the application of multiple response surface methods in computer simulation, Montgomery and Bettencourt [21] employed a stochastic simulation model of a tank duel. The model simulates brief fire engagements between two armored vehicles. A stationary defending vehicle (Blue Tank)

y, AVERAGE DAILY COST, $

x_2, EOQ

x_1, ROP	50	75	100	125	150	175	200	225	250
-5	312	221	168	143	123	118	107	104	100
0	288	202	155	130	112	109	101	97	96
5	267	185	143	120	105	101	96	82	92
10	249	170	132	110	99	97	91	88	88
15	232	159	125	105	95	91	89	88	87
20	215	147	117	97	90	87	86	83	85
25	198	136	109	92	85	83	84	81	84
30	180	124	100	88	80	80	81	79	83
35	167	117	93	85	78	78	79	80	83
40	160	111	91	84	77	77	79	79	82
45	154	107	90	81	78	76	79	81	83
50	145	102	87	80	77	77	79	81	83
55	142	101	86	79	78	79	80	82	85
60	138	100	86	79	79	79	80	85	87
65	135	99	85	79	80	80	81	85	88
70	134	100	86	81	81	81	83	87	90
75	134	101	88	83	83	84	85	89	93
80	135	101	89	85	84	86	87	90	95
85	134	103	91	87	86	88	88	92	97
90	135	105	93	88	88	89	91	95	99

Figure 13. Response Table for Stochastic Inventory Simulation (Ignall [16]).

fires first at a fully-exposed attacking vehicle (Red Tank). The engagement ends when a kill occurs or a predetermined time limit of 120 seconds expires.

The input variables to the tank duel model are presented in detail in [21]. But for purposes of evaluating and comparing the four simulation/optimization procedures described in this report, the following independent variables x_i and response variables y_j were chosen:

x_1 = mean time to fire first round for the Blue crew (sec.),

x_2 = mean time between rounds (sec.),

y_1 = probability of Blue victory,

y_2 = expected number of rounds fired by the Blue tank.

The optimization problem was framed as one of constrained optimization, as follows:

$$\text{maximize} \quad y_1 = g_1(x_1, x_2)$$

$$\text{subject to} \quad 5 \leq x_1 \leq 25 \qquad 5 \leq x_2 \leq 25 \qquad 0 \leq y_2 \leq 2$$

Mine Hunting [1]

This model, written in FORTRAN IV, is a Monte Carlo computer simulation which is used to evaluate mine hunting tactics. This program may be used either to evaluate minefields or, in conjunction with a sustained attrition minefield evaluation model, to compare the effectiveness of minesweeping to that of mine hunting. The basic operation is to move a mine hunting ship through a minefield. Each time the ship passes a mine or minelike object, its distance from that object is calculated, and a random number in the (0,1) range is drawn and compared with the probability of the ship's sonar detecting the object. If the object is detected, a similar procedure determines if the object is correctly classified and, in the case of mines, neutralized.

The ship's course is altered to avoid mines it has detected. Actuation and detonation of mines and damage of ships is simulated for both mine hunters and traffic ships.

Inputs include the number of mines, shape of minefields, density of bottom clutter, number of hunters, length of operation, effectiveness of sonar, condition of bottom, number of traffic ships, and method of hunting. Geographically, the model is limited to an area 20 miles square (a function of the number of mines) while the force limits are 100 hunters, 5 traffic ship types, 60 types of mines or minelike objects (e.g., bottom clutter), and 750 individual mines or minelike objects.

Outputs include mines detected, mines neutralized, ship damage, and threat of the minefield as a function of time.

Two separate problems were used to evaluate the four optimization procedures with the mine hunting model. In each of these problems, the optimization was sought from the standpoint of the force which deploys the minefield. That is, the objective was to maximize minefield effectiveness in terms of damage to ships which attempt to traverse the field.

Problem 1 used two mine types, arming delays and a total of 58 mines. Problem 2 used only one mine type, no arming delays and 61 mines. In Problem 1, all traffic ships (i.e., the target ships as opposed to the mine sweepers) attempted to transit the mine field during the period 119-120 hours, while in Problem 2 traffic was uniform over the entire 120 hour period.

Additional information about the problem that was common to both plans is as follows:

- The minefield was to cover an area one nautical mile wide and ten nautical miles long. (The input provided was for a 1800 ft. x .0 nm. field; this area required fewer mines, hence reduced running time. It was assumed that mine requirements were proportional to the area

of the minefield so that the plan for a 1nm. x 10 nm. area used (6076.1/1800) x the number of mines in the 1800' x 10 nm. area).

- Water depth was constant (hence for a given mine type/target type/target speed combination, the characteristics of a mine-target interaction were the same everywhere).
- The target ships travelled at three knots with a navigational error of 100 yards (i.e., the perpendicular distance from the target's actual location to the intended location was normally distributed with a mean of zero and a standard deviation of 100 yards).
- Two minesweepers were available and had a combined capability of making ten transits per day at a safe speed of five knots. They were assumed to have actuation characteristics identical to the target but were safe from damage.
- The mines were assumed to be uniformly randomly distributed throughout the minefield.

The objective of each plan was to provide a minefield with a 90 percent chance of damaging (mission about level) at least two of the ten target ships.

The simulation/optimization problem for each of these two minefield evaluation problems was formulated as follows:

x_1 = mean of the arming time distribution, where arming time was Poisson distributed; $6 \leq x_1 \leq 12$ hrs.

x_2 = mean of the mine count distribution, where mine count was Poisson distributed; $3 \leq x_3 \leq 5$ counts.

y_1 = average ship damage, ships (summed over all ship types)

y_j, $j = 2, \ldots, 6$ = average "threat" in period j-1, where period 1 is 0 - 24 hrs., period 2 is 24-48 hrs., etc.

A constrained optimization problem was

$$\max y_1$$

subject to $6 \leq x_1 \leq 12$; $3 \leq x_2 \leq 5$; $0 \leq y_j \leq 0.25$, $j = 2, \ldots, 6$.

The simulation/optimization interface for the minefield evaluation model involved the following:

- The main program was modified to become Subroutine SIMUL in the optimization program.
- In Subroutine SIMUL, each mine was assigned random arming time and a random ship count from Poisson distributions having parameters x_1 and x_2, respectively.
- In Subroutine AVGTHR, in which the damage to traffic ships is tallied and minefield "threat" is computed, special program segments were written to give values to the response variables y_j, $j = 1, \ldots, 6$.

Air Attack on a Surface Fleet [18]

This model, written in FORTRAN IV, evaluates antiair warfare tactics related to the following factors:

i. Force disposition - e.g., dispersed or integral formation concepts;

ii. Fire control doctrine - e.g., shoot, look, shoot;

iii. Command and control - e.g., coordinated or autonomous assignment doctrine;

iv. Cover and deception - e.g.,"look alike" ships and decoy forces;

v. Deployment of remote sensors - e.g., antielectronic warfare aircraft and radar picket ships;

The effects of enemy force factics evaluated include multidirectional, multialtitudinal attacks, use of electronic countermeasures and penetration aids, "roll back' attack tactics, and weapon loading and launch criteria.

The model may be described as a one-sided, event-store, Monte Carlo computer simulation. Enemy attack tactics are fully defined at the outset of each game situation by inputs to the raid generator, and the attack is conducted as defined regardless of the evolution of the battle. Thus, enemy attack tactics fall into the category of the "uncontrollable factors" z_k, $k = 1, \ldots, p$ in equation (1). The defense force, typically a carrier task force, includes a surveillance radar network (shipborne and airborne air search radars, both conventional periodic scan types, and multimode aray radars), a communications network for track passing and firepower coordination, and defensive weapon systems (primarily SAM batteries and their associated tracking radars and fire-control computers). Task force runs may be made to establish sensitivity to changes in ship configurations (alternate radars or weapon systems) or in ship dispositions and assignment doctrine. The model was designed to be, to the greatest extent practicable, machine and installation independent.

Input data to this model define the size, composition, disposition, and fire-control tactics for both the offense and the defense units. Enemy forces are most often composed of aircraft and submarines that launch antiship cruise missiles from stand-off ranges, although other forms of attack are possible (e.g., gravity bombs). The model has the capability of handling up to 31 defense units (ships and AEW aircraft) and up to 255 offense units (launch vehicles, stand-off jammers, cruise missiles, etc.). Numerous other limits define the composition of each major unit (e.g., radars pership, missile launchers per ship, magazine capacity per launcher, cruise missiles per launch platform, flight path legs per target). Certain input parameters select tactical options (e.g., single or dual missile salvos) and simulation algorithm options (e.g., simple or complex radar detection models).

A summary of the processed input values is printed at the outset of a game situation. This may optionally be followed by a detailed listing of

event occurences (of all types or of a selected subset and over the entire battle or during a specified time interval). A summary listing at the completion of play tabulates the performance and status of individual units at the end of the game; this listing includes factors such as average range of detection by surveillance radars, intercept ranges and kill probabilities achieved by defensive missile systems, number of enemy weapon penetrations (hits) suffered by each defense unit, and damage status of major ship components at the end of game play.

Although no simulation/optimization trials were conducted with this model, trial simulations were executed to characterize its performance. As with the minefield evaluation model, the "air attack on a surface fleet" model would require that the optimization interface be "custom-fitted" by developing suitable FORTRAN code.

DISCUSSION OF RESULTS

This section discusses the results obtained from simulation/optimization runs with the four optimization procedures in combination with each of the three simulation models. The discussion is presented in three segments, one for each simulation model.

Stochastic Inventory Model [15,16]

Table 1 presents the results of several optimization attempts with the stochastic inventory model. The complex search procedure and the second-order reponse surface approach of employing multiple regression to fit a quadratic model to a set of complex search points gave very similar results, yielding near-optimal solutions with about 7n simulation trials. The second-order response surface method employing a central composite design produced a predicted optimum at a point for which the actual solution was 4 percent removed from the true optimum, using only 9 simulation trials. The sequential first-order response surface approach, employing either a 2^2 factorial design or 3-vertex simplex design (each augmented by a centroid point), with a golden section search along the gradient direction, required about 20n simulation trials to yield near-optimal solutions. Among the four optimization methods evaluated with this model, only the first-order response surface method fared poorly.

It is notable that, when a second-order response surface is fitted to the 14 search points obtained from a Box complex search, the predicted optimum solution is actually slightly worse than the best solution obtained via the search itself.

Table 1. Summary of Simulation/Optimization Results for the Stochastic Inventory Model [15,16]

Optimization Method	Starting Seed	Estimated Solution x_1	x_2	y	Number of Trials
Complex	12471	50	177	\$76.20	13
Search	21437	43	245	81.00	14
First-Order Factorial Design	17332	70	125	\$79.00	40
First-Order Simplex Design	17332	42	159	\$76.78	41
Second-Order Central Composite Design	35188	53	183	\$70.32 (Predicted) \$79.00 (Actual)	9
Second-Order Complex Search	14271	41	163	\$76.78 (Search) 77.52 (Predicted)	14

Notes: (1) Known Solution $x_1 = 45$ $x_2 = 175$ $y = \$76.00$

(2) CPU times less than 5 sec. per optimization run on an IBM 370/168 computer.

(3) Complex search terminated because $2n = 4$ search points were conducted without an improved solution.

Tank Duel Model [21]

Table 2 presents the results of four simulation/optimization runs with the tank duel model. An optimum solution lies near ($x_1 = 8.2$, $x_2 = 12.5$), Producing a probability of Blue Victory of 0.61 with an expected firing of 2 Blue rounds. Thus, the "expected rounds fired" response appears to bound the solution. As with the stochastic inventory model, only the sequential first-order response surface approach produces unacceptably costly experimentation.

In contrast to the result with the stochastic inventory model, the predicted solution obtained after fitting second-order response surfaces to the two sets of responses obtained in the Box complex search represents a significant improvement over the best point observed in the search.

Mine Hunting Model [1]

As seen in Table 3, only three optimization methods were evaluated with the mine hunting simulation model. The complex search method, as with the previous models, was terminated after 2n simulation trials had been performed without obtaining an improved solution. For example, with problem 1, the estimated optimum solution was observed with search point 12, and since four more search points failed to improve on that result the search was terminated after point 16.

Focusing on the results of the second-order response surface method, employing a central composite design with 16 simulation trials, a solution $x_1 = 10.6$ hours mean arming delay per mine and $x_2 = 3.1$ ships mean ship count per mine results in an expected kill of 0.51 ships. The threat profile is essentially zero during the first 72 hours of minefield operation, and increases to levels between 0.15 and 0.2 during the last 48 hours of operation when enemy ship traffic actually attempts to traverse the field.

Table 2. Summary of Simulation/Optimization Results for the Tank Duel Model [21]

Optimization Method	Estimated Solution x_1	x_2	y_1	y_2	Number of Trials
Complex Search	12.9	10.1	0.59	1.97	20
First-Order Simplex Design	8.2	12.4	0.60	1.97	55
Second-Order Central Composite	8.0	12.5	0.61	2.00	9
Second-Order	13.5	11.3	0.57	1.86 (Search)	15
Complex Search	8.2	12.2	0.61	1.99 (Predicted)	

Notes: (1) Known optimum at $x_1 = 8.2$ sec., $x_2 = 12.5$ sec., $y_1 = 0.61$, $y_2 = 2$ rds.

(2) CPU times less than 5 sec. per optimization run on an IBM 370/168 computer.

(3) Complex search was terminated because $2n = 4$ search points were evaluated without obtaining an improved solution.

Table 3. Summary of Simulation/Optimization Results for the Minefield Evaluation Model [1]

Optimization Method	Problem Number	Estimated Solution								Number of Trials
		x_1	x_2	y_1	y_2	y_3	y_4	y_5	y_6	
Complex Search	1	11.4	3.9	0.43	0.000	0.000	0.012	0.014	0.030	16
	2	6.2	3.5	0.32	0.000	0.027	0.027	0.044	0.058	9
Second-Order Central Composite	1	10.6	3.1	0.51	0.000	0.000	0.000	0.183	0.163	16
	2	6.0	3.0	0.37	0.000	0.005	0.044	0.091	0.128	16
Second-Order Complex Search	1 (Search)	8.6	3.6	0.55	0.000	0.000	0.000	0.183	0.140	25
	1 (Predicted)	8.8	4.0	0.26	0.000	0.000	0.000	0.153	0.129	

Notes: (1) True optimum solution is unknown.

(2) In each simulation trial, 5 replicates of the simulation were averaged.

(3) Optimization runs required about 10 minutes of CPU time each on an IBM 370/168 computer.

(4) The sequential first-order response surface method was not evaluated due to excessive computer time.

(5) The second-order complex search procedure did not coverge in a 10-minute run.

This result points out a difficulty in interfacing an optimization procedure with a simulation model when a gaming strategy is involved. In a sense, the optimization procedure "learns" the enemy tactic as the series of simulation runs progresses. In this instance, the optimization procedure "sees" that traffic ships traverse the field late in the game. To counter this "learning" behavior, it is necessary to level the threat over the entire period of minefield operation. Since such a narrow feasible region would cause complex search to yield a high proportion of infeasible responses, the preferred optimization method would be a second-order response surface approach, either by a designed experiment or by fitting a quadratic surface to Box's complex search results.

SUMMARY AND CONCLUSIONS

This research has investigated four alternative approaches to optimization of simulated systems having multiple independent variables x_i, i=1, ..., n and multiple simulation responses y_j, j=1, ..., m. Of these four methods, Box's complex search [8], a second-order response surface approach employing a central composite experimental design [6,7] followed by a computational complex search, and fitting a second-order response surface to a succession of complex search points, all yield "good" solutions in an "economical" number of computer simulation trials. The sequential first-order response surface method converged to near-optional solutions, but required an excessive number of simulation trials. This excessive number of trials derived from the golden section line search along the estimated gradient direction. Had a polynomial regression line search employing, say, five trials been conducted as in Biles [2,3], two complete cycles of gradient-determining blocks and line searches could have been performed in about 8n to 10n trials.

The most crucial aspects of the experimental Box complex search approach to simulation/optimization are twofold:

1. A termination criterion based on either (a) a maximum number of simulation trials or (b) a certain number of simulation trials without obtaining an improved solution. The typical criterion used to terminate a computational Box complex search is convergence; such as $y_{k+1} - y_k \leq \varepsilon$. Such a criterion can lead to excessive simulation trials in the experimental mode, however, unless ε is coarse.

2. In an experimental region in which the implicit constraints on simulation responses $y_2, \ldots, y_m$ describe a small feasible region, a number of infeasible simulation trials will be conducted. In fact, the probability that an early simulation trial is feasible with respect to implicit constraints is $P_f = R(Y_f)/R(E)$, where

   P_f = probability of a feasible trial

   $R(Y_f)$ = region subtended by the implicit constraints

   $R(E)$ = region subtended by the explicit constraints

   Once n+2 feasible simulation trials have been found, and complex search begins to seek improved solutions, there is a higher probability of obtaining a feasible simulation trial. As the "complex" becomes tighter, this probability approaches one. But the possibility of a large number of infeasible trials early in the search is perplexing.

But the capability of fitting quadratic response surfaces to a set of at least (n+1)(n+2)/2 complex search points (this being the number of regression coefficients in the quadratic model) in an "incomplete" complex search overcomes this difficulty.

The second-order response surface approach has the advantage of requiring relatively few simulation trials, but the accuracy of the predicted optimum solution $(\hat{X},\hat{Y})$ depends on how well the true surfaces are represented with quadratic estimators. In those cases where there is some knowledge of the nature of the true surfaces, the modeler is better able to hypothesize the fitted model (which may not be quadratic). Although the central composite design has superior statistical properties to the purely random design, such as obtained from a set of complex search points, the complex search approach will usually place more than one simulation trial in the neighborhood of the predicted optimum. This affords a smaller confidence region about the predicted optimum.

The computer programs listed in Appendices A through D offer a means of interfacing these optimization procedures with simulation models of interest. Minor alterations in the optimization program would be necessary for a particular simulation model, but major modifications in the simulation model itself could be entailed. In general, one could expect to have to "custom fit" the optimization program to a given simulation model. this is especially true of the typical naval simulation model, because modelers tend to "start from scratch" in developing models for their own needs.

This research has provoked two recommendations relative to naval simulation modeling activities in the several naval laboratories:

1. There is a definite need to have naval simulationists apply up-to-date statistical methodology, particularly variance reduction, in their modeling efforts.
2. There is a need to have naval simulationists become familiar with general simulation languages such as GASP-IV and SIMSCRIPT 2.5. It is also apparent that certain specialized modeling techniques, such as Q-GERT, is vastly under utilized in naval operations modeling.

Developing FORTRAN simulation models from "scratch" is sometimes necessary, but very often considerable time, effort and expense is spent in duplicating the kinds of functional capability contained in available and tested simulation soltware.

The most pressing need in terms of continuing research in the area of optimization of naval simulation models is to incorporate the techniques of game theory into the optimization approach. So many naval models embody a "friendly force/enemy force" focus, with decision-making capability on both sides, that optimization must entail game theory considerations.

REFERENCES

1. Bailey, J. L., and M. A. Weedon, "The Minefield Analysis with Hunting Evaluation Model (MAYHEM)," NWL Technical Report TR-2742, U. S. Naval Weapons Laboratory, Dahlgren, Virginia, May 1972.

2. Biles, W. E., "A Gradient-Regression Search Procedure for Simulation Experimentation," Proceedings of the 1974 Winter Simulation Conference, Washington, D.C., 1974, p. 491-497.

3. Biles, W. E., "Response Surface Methods for Optimization of Constrained Multivariable Processes," Industrial and Engineering Chemistry: Process Design and Development, Vol. 14, No. 2, 1975, p. 152-158.

4. Biles, W. E., "Optimization of Multiple-Objective Computer Simulations: A Nonlinear Goal Programming Approach," Proceedings of the 1977 AIDS Conference, Chicago, 1977.

5. Biles, W. E., and J. J. Swain, "Strategies for Optimization of Multiple Response Simulation Models," Proceedings of the 1977 Winter Simulation Conference, Gaithersburg, Maryland, December 1977.

6. Box, G. E. P., and K. B. Wilson, "On the Experimental Attainment of Optimum Conditions," Journal of the Royal Statistical Society, B, Vol. 13, 1951, p. 1-45.

7. Box, G. E. P., and J. S. Hunter, "Multifactor Experimental Designs for Exploring Response Surfaces," Annals of Mathematical Statistics, Vol. 28, 1957, p. 192-242.

8. Box, M. J., "A New Method of Constrained Optimization and a Comparison with Other Methods," Computer Journal, Vol. 8, 1965.

9. Brooks, S. H., and M. R. Mickey, "Optimization Estimation of Gradient Direction in Steepest Ascent Experiments," Biometrics, Vol. 17, No. 1, 1961.

10. Draper, N. R., and H. Smith, Applied Regression Analysis, John Wiley and Sons, New York, 1966.

11. Dyer, J. S., "Interactive Goal Programming," Management Science, Vol. 19, No. 1, 1973.

12. Fishman, G. S., Concepts and Methods in Discrete Event Digital Simulation, John Wiley and Sons, New York, 1973.

13. Griffith, R. E. and R. A. Stewart, "A Nonlinear Programming Technique for the Optimization of Continuous Processing Systems, Management Science, Vol. 7, 1961.

14. Hooke, R. and T. A. Jeeves, "Direct Search Solution of Numerical and Statistical Problems," Journal of the Association of Computing Machinery, Vol. 8, 1961.

15. Hunter, J. S., and T. H. Naylor, "Experimental Designs for Computer Simulation Experiments," Management Science, Vol. 16, No. , 1970, pp. 422-434.

16. Ignall, E. J., "On Experimental Designs for Computer Simulation Experiments," Management Science, Vol. 18, No. 7, March 1972, pp. 384-388.

17. Ignizio, J. P., Goal Programming and Extensions, Lexington Books, D.C. Heath and Company, Lexington, Mass, 1976.

18. Kaplan, D. J., L. C. Davis, M. E. B. Owens II, O. F. Forsyth and J.J. Penick, "SPEARS, an AAW Performance Simulation," NRL Report 7958, Naval Research Laboratory, Washington, D. C., June 1976.

19. Kleijnen, J. P. C., Statistical Techniques in Simulation, Parts I and II, Marcel Dekker, New York, 1975.

20. Klingman, W. R., and D. M. Himmelblau, "Nonlinear Programming with the Aid of the Multiple Gradient Summation Technique," Journal of the Association of Computing Machinery, Vol. 11, 1964.

21. Montgomery, D. C., and V. M. Bettencourt, "Multiple Response Surface Methods in Computer Simulation," Simulation, October, 1977, pp. 113-121.

22. Myers, R. H., Response Surface Methodology, Allyn and Bacon, Boston, Mass., 1971.

23. Rosen, J. B., "The Gradient Projection Method for Nonlinear Programming: Part II - Nonlinear Constraints," Journal of the Society of Nonlinear and Applied Mathematics, Vol. 8, 1961.

24. Smith, D. E., "Requirements of an Optimizer for Computer Simulations," Naval Research Logistics Quarterly, Vol. 20, No. 1, 1973.

25. Smith, D. E., "An Empirical Investigation of Optimum-Seeking in the Computer Simulation Situation," Operations Research, Vol. 21, 1973, pp. 475-497.

26. Spendley, W., and G. R. Hext, and R. F., Himsworth, "Sequential Application of Simplex Designs in Optimization and Evolutionary Operations," Technometrics, Vol. 4, 1962.

27. Swain, J. J., "Efficient Strategies for Constrained Experimental Optimization," unpublished Master's thesis, University of Notre Dame, 1977.

28. Zoutendijk, G., Methods of Feasible Directions, Elsevier, New York, 1960.

APPENDIX A

PROGRAM LISTING

FOR AN

EXPERIMENTAL BOX COMPLEX SEARCH

```
C
C*****TEST PROGRAM FOR COMPLEX METHOD SUBROUTINE*****
C
      DIMENSION X(30,40),XP(30),L(30),WP(30),Z(20,30),Y(20),Z1(20)
      COMMON N,NP,NR
      COMMON/BLEE1/XLP(10),XUP(10)
      COMMON/BLEE2/ISEED1,ISEED2
      COMMON/BLEE3/XL(10),XU(10),YL,YU
      COMMON/BLEE4/ITER
C
      READ (9,*) N,NP
      READ (9,*) (XL(I), I=1,N), (XU(I), I=1,N)
      READ (9,*) (XLP(I),I=1,N), (XUP(I), I=1,N)
      READ (9,*) YL,YU
      READ (9,*) SMALL
      READ (9,*) ICHECK
      READ (9,*) IX
      READ (9,*) ISEED1,ISEED2
C
      WRITE (12,1000)
      WRITE (12,*) N,NP
      WRITE (12,*) (XL(I), I=1,N), (XU(I), I=1,N)
      WRITE (12,*) (XLP(I), I=1,N), (XUP(I),I=1,N)
      WRITE (12,*) YL,YU
      WRITE (12,*) SMALL
      WRITE (12,*) ICHECK
      WRITE (12,*) IX
      WRITE (12,*) ISEED1,ISEED2
C
      ITER=0
      M=N+1
      AV=N
      AMM1=N-1
C
      WRITE (12,160)
      DO 2 J=1,N
   10 DO 1 I=1,N
      CALL RANDU(IX,IY,RNUM)
      IX=IY
      DA=XUP(I)-XLP(I)
      X(I,J)=XLP(I)+RNUM*DA
    1 XP(I)=X(I,J)
      CALL FSUBT(XP,IOUT,Y)
      ITER=ITER+1
      IF (IOUT.EQ.1) GO TO 105
      DO 101 I=1,N
  101 WRITE (12,170) I,XP(I)
      DO 102 I=1,NR
  102 WRITE (12,180) I,Y(I)
      GO TO 115
  105 DO 111 I=1,N
  111 WRITE (12,175) I,XP(I)
      DO 112 I=1,NR
  112 WRITE (12,185) I,Y(I)
      WRITE (12,190) ITER
      GO TO 10
  115 DO 15 L=1,NR
      Z(L,J)=Y(L)
   15 Z1(J)=Y(L)
```

```
   12 WRITE (12,120) ITER
      CALL EXIT
C
      CALL ZOPT(KP,ZMAX,ZI)
   20 CALL WORST(K,ZMIN,ZI)
      DO 33 I=1,N
      SUM=-X(I,K)
      DO 30 L=1,M
   30 SUM=SUM+X(I,L)
   33 C(I)=SUM/ZMIN
C
      R=1.
   35 S=1.+R
      DO 37 II=1,N
   37 XP(II)=X(II,K)+S*(C(II)-X(II,K))
      CALL FSOPT(NP,IOUT,Y)
      R=R*.5
      IF (IOUT.EQ.2) GO TO 35
      ITER=ITER+1
      IF (IOUT.EQ.1) GO TO 385
      DO 371 I=1,N
  371 WRITE (12,170) I,XP(I)
      DO 372 I=1,NR
  372 WRITE (12,180) I,Y(I)
      GO TO 400
  385 DO 391 I=1,N
  391 WRITE (12,175) I,XP(I)
      DO 392 I=1,NR
  392 WRITE (12,185) I,Y(I)
      WRITE (12,135) ITER
      GO TO 35
C
   40 IF (Y(1).LE.ZMIN) WRITE (12,145)
      IF (Y(1).LE.ZMIN) GO TO 35
      WRITE (12,140) ITER
      DO 50 I=1,N
   50 X(I,K)=XP(I)
      DO 55 L=1,NR
   55 Z(I,L)=Y(L)
      ZI(K)=Y(1)
      IF (ZMAX.GE.ZI(K)) WRITE (12,155)
      IF (ZMAX.GE.ZI(K)) GO TO 20
      WRITE (12,150)
      KP=K
      ZMAX=Z(1,KP)
      DO 100 I=1,M
      A=ABS(ZMAX-ZI(I))
      IF (A.GT.SMALL) GO TO 20
  100 CONTINUE
CC
      IF (ICHECK.EQ.0) GO TO 60
      DO 550 I=1,N
      SUM=X(I,1)
      DO 530 L=2,M
  530 SUM=SUM+X(I,L)
  550 XP(I)=SUM/AM
C
      CALL SIMUL(XP,Y)
      GO TO 10
```

```
      IIII=IIII+1
      IF(IIII.GT.1) GO TO 95
      WRITE (12,170) I,XP(I)
      DO 95 I=1,NR
   95 WRITE (12,180) I,Y(I)
      WRITE (12,200) IIER
C
      DO 590 I=1,N
      A=Y(I)-Z1(I)
      IF (A.GT.SMALL) GO TO 600
  590 CONTINUE
      GO TO 60
C
  600 CALL WORST(K,ZMIN,ZI)
      IF (Y(I).LE.Z1(K)) GO TO 60
      DO 610 I=1,N
  610 X(I,K)=XP(I)
      DO 630 I=1,NR
  630 Z(I,K)=Y(I)
      Z1(K)=Y(I)
      IF (ZMAX.GE.Z1(K)) WRITE (12,155)
      IF (ZMAX.GE.Z1(K)) GO TO 20
      WRITE (12,150)
      KP=K
      ZMAX=Z1(K)
      GO TO 20
CC
   60 WRITE (12,160)
      WRITE (12,165)
      DO 70 I=1,N
   70 WRITE(6,170) I,X(I,KP)
      DO 80 I=1,NR
   80 WRITE(6,180) I,Z(I,KP)
C
  120 FORMAT (/1X,I5,'  TH  ITERATION'/5X,'*****  THE ABOVE POINT IS AN
     &INITIAL POINT  *****'/)
  130 FORMAT(/1X,I5,'  TH  ITERATION'/5X,'*****  THE ABOVE POINT IS NO
     &T AN INITIAL POINT  *****'/)
  135 FORMAT(/1X,I5,'  TH  ITERATION'/5X,'*****  THE ABOVE POINT IS NO
     &T A FEASIBLE POINT  *****'/)
  140 FORMAT (/1X,I5,'  TH  ITERATION'/5X,'*****  THE ABOVE POINT IS AN
     & IMPROVED POINT  *****'/)
  145 FORMAT (/1X,I5,'  TH  ITERATION'/5X,'*****  THE ABOVE POINT IS NOT
     & AN IMPROVED POINT  *****'/)
  150 FORMAT (5X,'*****  AND ALSO THE CURRENT OPTIMAL POINT  *****'/)
  155 FORMAT (5X,'*****  BUT NOT THE CURRENT OPTIMAL POINT  *****'/)
  160 FORMAT (1H1)
  165 FORMAT (/5X,'*****   OPTIMAL SOLUTION   *****'///)
  170 FORMAT (/20X,'X(',I1,') = ',F14.6)
  180 FORMAT (/20X,'Y(',I1,') = ',F14.6)
  175 FORMAT (/70X,'X(',I1,') = ',F14.6)
  185 FORMAT (/70X,'Y(',I1,') = ',F14.6)
  200 FORMAT (/1X,I5,'  TH  ITERATION'/5X,'*****  THE ABOVE POINT IS THRO
     &UGH CHECKING CENTROID  *****'/)
  100 FORMAT (/45X,'*****   LIST OF INPUT DATA   *****'///)
C
      STOP 5555
      END
      SUBROUTINE FSOPT(X,ICOI,Y)
```

```
C
C*****SUBROUTINES USED IN CONJUNCTION WITH SIMPLEX METHOD
C
      DIMENSION X(30),Y(20)
      COMMON N,M,NR
      COMMON/BLEE1/XLP(10),XUP(10)
      COMMON/BLEE3/XL(10),XU(10),YL,YU
      ICUT=2
      DO 1 I=1,N
      IF (X(I).LT.XLP(I)) GO TO 3
      IF (X(I).GT.XUP(I)) GO TO 3
    1 CONTINUE
      ICUT=ICUT-1
C
      CALL SIMUL(X,Y,NR)
      REWIND 10
      DO 10 J=2,NR
      IF (Y(J).LT.YL) GO TO 3
      IF (Y(J).GT.YU) GO TO 3
   10 CONTINUE
      ICUT=ICUT-1
C
    3 RETURN
      END
C
      SUBROUTINE WORST(K,ZMIN,Z1)
C
C*****THIS SUBROUTINE INDICATES THE CURRENT WORST POINT
C
      DIMENSION Z1(20)
      COMMON N,M,NR
      K=1
      ZMIN=Z1(1)
      DO 1 J=2,M
      IF (Z1(J).GE.ZMIN) GO TO 1
      K=J
      ZMIN=Z1(K)
    1 CONTINUE
      RETURN
      END
C
      SUBROUTINE ZOPT(KP,ZMAX,Z1)
C
C*****THIS SUBROUTINE INDICATES THE CURRENT BEST POINT
C
      DIMENSION Z1(20)
      COMMON N,M,NR
      KP=1
      ZMAX=Z1(1)
      DO 1 J=2,M
      IF (Z1(J).LE.ZMAX) GO TO 1
      KP=J
      ZMAX=Z1(KP)
    1 CONTINUE
      RETURN
      END
C
      SUBROUTINE SIMUL(XS,Y,IIPI)
      DIMENSION XS(10),Y(20)
```

APPENDIX B

PROGRAM LISTING

FOR A

SEQUENTIAL FIRST-ORDER RESPONSE SURFACE METHOD

```
C
      DIMENSION XBAR(30),STD(30),RY(30),
     &          BV(30),SB(30),IV(30),RX(30,30),RT(465),ANS(10),
     &          XP(10),XC(10)
      COMMON X(40,30),XV(1200),XL(20),XU(20),YL(10),YU(10),
     &       Y(10),GRAD(20),N,MVTEX,AVTEX,NV,M,MDB,NR,STEP,RAD,
     &       RRRA,CONF,GOLD,ISTOP,NIV
      COMMON/BLEE1/XLP(10),XUP(10)
      COMMON/BLEE2/ISEED1,ISEED2
      COMMON/BLEE3/XL(10),XU(10,YL,YU
      COMMON/BL4/IACT(30)
      COMMON/BL5/JV(30),ISAVE(30)
      COMMON/BLEE6/ITER
      WRITE (6,2000)
      READ (5,*) MDSIM
      READ (5,*) N,NR,MCP
      READ (5,*) (XL(J),J=1,N), (XU(J),J=1,N)
      READ (5,*) (XLP(I),I=1,N), (XUP(I),I=1,N)
      READ (5,*) (DB(J),J=1,9)
      READ (5,*) YL,YU
      READ (5,*) STEP,RAD,RRRA
      READ (5,*) CONF
      READ (5,*) IX
      READ (5,*) ISEED1,ISEED2
C
      WRITE (12,*) MDSIM
      WRITE (12,*) N,NR,MCP
      WRITE (12,*) (XL(J),J=1,N), (XU(J),J=1,N)
      WRITE (12,*) (XLP(I),I=1,N), (XUP(I),I=1,N)
      WRITE (12,*) (DB(J),J=1,9)
      WRITE (12,*) YL,YU
      WRITE (12,*) STEP,RAD,RRRA
      WRITE (12,*) CONF
      WRITE (12,*) IX
      WRITE (12,*) ISEED1,ISEED2
C
      GOLD=(-1.+SQRT(5.))/2.
      ITER=0
      GO TO (10,20), MODEV
   10 WRITE (6,2015)
      MVTEX=N+1
      GO TO 30
C
   20 WRITE (6,2025)
      MVTEX=2**N
C
   30 AVTEX=MVTEX
      M=MVTEX+1
      MP1=M+1
      MDB=MVTEX+MCP
      NV=N+1
      NIV=N+NR
      WRITE (6,2035)
      WRITE (6,2040) (YU(J),J=2,NR)
      DO 45 J=1,NIV
      ISAVE(J)=J
   45 CONTINUE
   50 DO 60 J=1,N
      CALL RANDU(IX,IY,RJNR)
```

```
00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000460
00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550
00000560
00000570
00000580
00000590
```

```
      IX=IY                                                             00000600
      DA=XU(J)-XL(J)                                                    00000610
      X(M,JJ)=XL(J)+RJMN*DA                                             00000620
      XC(I)=X(M,J)                                                      00000630
   60 CONTINUE                                                          00000640
C                                                                       00000650
      CALL FSUPT(IOUT,JACT,XC)                                          00000660
      IF (IOUT.EQ.1) GO TO 50                                           00000670
      DO 80 K=1,NR                                                      00000680
      J=N+K                                                             00000690
      X(M,J)=Y(K)                                                       00000700
      XC(J)=Y(K)                                                        00000710
   80 CONTINUE                                                          00000720
 1000 IF (MCP.EQ.1) GO TO 1050                                          00000730
      DO 90 I=MP1,MOB                                                   00000740
      IF (MDSIM.EQ.1) GO TO 83                                          00000750
      CALL SIMUL(XC,Y)                                                  00000760
   83 DO 85 K=1,NR                                                      00000770
      J=N+K                                                             00000780
      X(I,J)=Y(K)                                                       00000790
   85 CONTINUE                                                          00000800
   90 CONTINUE                                                          00000810
      GO TO (1050,1060), MDSIM                                          00000820
 1050 WRITE (6,2050) ITER                                               00000830
      GO TO 95                                                          00000840
 1060 WRITE (6,2060) ITER,MCP                                           00000850
   95 GO TO (100,150), MODEV                                            00000860
C                                                                       00000870
  100 CALL SIMPLX(XC)                                                   00000880
      GO TO 1100                                                        00000890
  150 CALL FACTOR(XC)                                                   00000900
 1100 WRITE (6,2100) ITER,MVTEX                                         00000910
      DO 350 J=1,NV                                                     00000920
      DO 350 I=1,MOB                                                    00000930
      K=MOB*(J-1)+I                                                     00000940
      XV(K)=X(I,J)                                                      00000950
  350 CONTINUE                                                          00000960
      CALL CURSOR(MOB,NV,1,XV,XBAR,STD,RX,RT,BV,DV,TV)                  00000970
      CALL ORDER(NV,KI,NV,1,ISAVE,RX,RY)                                00000980
      CALL MINV(RX,N,DET,DV,TV)                                         00000990
      WRITE (6,320) ((X(I,J),J=1,NTV),I=1,MOB)                          00001000
  320 FORMAT(/(6X,4F14.6))                                              00001010
      CALL MULTR(MOB,N,XBAR,STD,BV,RX,RY,ISAVE,DV,SB,TV,ANS)            00001020
      WRITE (6,2150) ITER,(XC(J),J=1,N), ANS(1), (DV(J),J=1,N)          00001030
      WRITE (6,2200) (DV(J),J=1,N)                                      00001040
      IF(IOUT.EQ.0) GO TO 380                                           00001050
      CALL PROJCT(JACT)                                                 00001060
      WRITE (6,2300) (DV(J),J=1,N)                                      00001070
  380 SUMSQ=0.                                                          00001080
      DO 400 J=1,N                                                      00001090
      SUMSQ=SUMSQ+DV(J)*DV(J)                                           00001100
  400 CONTINUE                                                          00001110
      DO 450 J=1,N                                                      00001120
      DV(J)=DV(J)/SQRT(SUMSQ)                                           00001130
  450 CONTINUE                                                          00001140
      CALL GOLDEN(IOUT,JACT,XC)                                         00001150
      IF (ISTOP.EQ.1) GO TO 500                                         00001160
      ITER=ITER+1                                                       00001170
      GO TO 1000                                                        00001180
```

```
  500 WRITE (6,3000) (XC(J),J=1,N)                                          00001190
      WRITE (6,3050) XC(NV)                                                 00001200
      WRITE (6,3100) STEP                                                   00001210
 2000 FORMAT (///5X,'***** THIS TANDUEL OPTIMIZATION USES :'///,            00001220
     &    15X,'FIRST ORDER REGRESSION FOR THE LOCAL RESPONSE SURVEY')       00001230
 2015 FORMAT (/15X,'SIMPLEX DESIGN FOR VERTICES')                           00001240
 2025 FORMAT (/15X,'FACTORIAL DESIGN FOR VERTICES')                         00001250
 2035 FORMAT (/15X,'LINE SEARCH WITH GOLDEN SECTION')                       00001260
 2040 FORMAT (/10X,'ROUNDS CARRIED BY BLUE IS',F7.1/)                       00001270
 2050 FORMAT (/5X,I5,' TH ITERATION'/10X,                                   00001280
     &    '***** THE ABOVE POINT IS THE CENTER POINT *****'/)               00001290
 2060 FORMAT (/5X,I5,' TH ITERATION'/10X,                                   00001300
     &    '***** THE ABOVE ',I4,2X,'POINTS ARE THE CENTER POINTS'/)         00001310
 2100 FORMAT (/5X,I5,' TH ITERATION'/10X,                                   00001320
     &    '***** THE ABOVE ',I4,2X,'POINTS ARE THE VERTICES'/)              00001330
 2150 FORMAT (/5X,I5,' TH ITERATION'/10X,'THE LOCAL RESPONSE SURFACE AO     00001340
     &BOUT ( X(1) =',F10.5,',  X(2) =',F10.5,' )  IS'//10X,F10.5,           00001350
     &' +',F10.5,' X(1) +',F10.5,' X(2)  =  Y(I)')                          00001360
 2200 FORMAT (/10X,'THE LOCAL STEEPEST ASCENT DIRECTION IS'//20X,           00001370
     &    '(  ',F10.5,'  ,  ',F10.5,'  )'/)                                 00001380
 2300 FORMAT (/10X,'THE LOCAL PROJECTION DIRECTION IS'//20X,                00001390
     &    '(  ',F10.5,'  ,  ',F10.5,'  )'/)                                 00001400
 3000 FORMAT (////5X,'OPTIMAL X(1) =',F10.5//5X,'OPTIMAL X(2) =',F10.5)     00001410
 3050 FORMAT (//5X,'OPTIMAL Y(1) =',F10.5)                                  00001420
 3100 FORMAT (///5X,'CONFIDENT RADIUS =',F10.5)                             00001430
      STOP                                                                  00001440
      END                                                                   00001450
C                                                                           00001460
      SUBROUTINE SIMPLX(XC)                                                 00001470
      DIMENSION XP(10),XC(10)                                               00001480
      COMMON X(40,30),XV(1200),XL(20),XU(20),YL(10),YU(10),                 00001490
     &       Y(10),GRAD(20),N,MVTEX,AVTEX,NV,M,MOB,NR,STEP,RAD,             00001500
     &       RRRA,CONF,GOLD,ISTOP,NTV                                       00001510
      ANG=2.*3.14159/AVTEX                                                  00001520
      ANGLE=0.                                                              00001530
      DO 300 I=1,MVTEX                                                      00001540
      X(I,1)=XC(1)+RAD*COS(ANGLE)                                           00001550
      X(I,2)=XC(2)+RAD*SIN(ANGLE)                                           00001560
      DO 80 J=1,N                                                           00001570
      XP(J)=X(I,J)                                                          00001580
   80 CONTINUE                                                              00001590
      CALL SIMUL(XP,Y)                                                      00001600
      DO 100 K=1,NR                                                         00001610
      J=N+K                                                                 00001620
      X(I,J)=Y(K)                                                           00001630
      XP(J)=Y(K)                                                            00001640
  100 CONTINUE                                                              00001650
      ANGLE=ANGLE+ANG                                                       00001660
  300 CONTINUE                                                              00001670
      RAD=RAD*RRRA                                                          00001680
      RETURN                                                                00001690
      END                                                                   00001700
C                                                                           00001710
      SUBROUTINE FACTOL(XC)                                                 00001720
      DIMENSION XP(10),XC(10)                                               00001730
      COMMON X(40,30),XV(1200),XL(20),XU(20),YL(10),YU(10),                 00001740
     &       Y(10),GRAD(20),N,MVTEX,AVTEX,NV,M,MOB,NR,STEP,RAD,             00001750
     &       RRRA,CONF,GOLD,ISTOP,NTV                                       00001760
      K1=1                                                                  00001770
```

```
      K2=1                                                              00001780
      DO 300 KK1=1,2                                                    00001790
      DO 200 KK2=1,2                                                    00001800
      I=2*(KK1-1)+KK2                                                   00001810
      X(I,1)=XC(1)+RAD*K1                                               00001820
      X(I,2)=XC(2)+RAD*K2                                               00001830
      DO 10 J=1,N                                                       00001840
      XP(J)=X(I,J)                                                      00001850
   10 CONTINUE                                                          00001860
      CALL SIMUL(XP,Y)                                                  00001870
      DO 100 K=1,NR                                                     00001880
      J=N+K                                                             00001890
      X(I,J)=Y(K)                                                       00001900
      XP(J)=Y(K)                                                        00001910
  100 CONTINUE                                                          00001920
      K2=-K2                                                            00001930
  200 CONTINUE                                                          00001940
      K1=-K1                                                            00001950
  300 CONTINUE                                                          00001960
      RAD=RAD*ERRA                                                      00001970
      RETURN                                                            00001980
      END                                                               00001990
C                                                                       00002000
      SUBROUTINE GOLDEN(IOUT,JACT,XC)                                   00002010
      DIMENSION XGL(10),XGU(10),XG1(10),XG2(10),XP(10),XC(10),KACT(20),  00002020
     &          JACT(50)                                                00002030
      COMMON X(40,30),XV(1200),XL(20),XU(20),YL(10),YU(10),             00002040
     &       Y(10),GRAD(20),N,MVTEX,AVTEX,NV,M,MDB,NR,STEP,RAD,         00002050
     &       ERRA,CONF,GOLD,ISTOP,NTV                                   00002060
      ISTOP=0                                                           00002070
      IND=1                                                             00002080
      IOUT=0                                                            00002090
      YIMAX=XC(NV)                                                      00002100
      DO 50 J=1,NTV                                                     00002110
      XGL(J)=XC(J)                                                      00002120
   50 CONTINUE                                                          00002130
   80 TSTEP=STEP                                                        00002140
  100 DO 150 J=1,N                                                      00002150
      XP(J)=XC(J)+TSTEP*GRAD(J)                                         00002160
  150 CONTINUE                                                          00002170
      CALL FSAPT(JOUT,JACT,XP)                                          00002180
      IOUT=IOUT+JOUT                                                    00002190
      IF(JOUT.EQ.1) GO TO 850                                           00002200
      GO TO (280,240,200), IND                                          00002210
  200 DO 220 J=1,NTV                                                    00002220
      XGL(J)=XG1(J)                                                     00002230
  220 CONTINUE                                                          00002240
  240 DO 260 J=1,NTV                                                    00002250
      XG1(J)=XGU(J)                                                     00002260
  260 CONTINUE                                                          00002270
  280 DO 300 J=1,N                                                      00002280
      XGU(J)=XP(J)                                                      00002290
  300 CONTINUE                                                          00002300
      DO 320 K=1,NR                                                     00002310
      J=N+K                                                             00002320
      XGU(J)=Y(K)                                                       00002330
  320 CONTINUE                                                          00002340
      IND=IND+1                                                         00002350
      IF (Y(1).LE.YIMAX) GO TO 400                                      00002360
```

```
      IF (IOUT.NE.0) GO TO 450                                          00002370
      YMAX=Y(I)                                                         00002380
      STEP=STEP/GOLD                                                    00002390
      TSTEP=TSTEP+STEP                                                  00002400
      GO TO 100                                                         00002410
C                                                                       00002420
  400 IF (IND.NE.2) GO TO 490                                           00002430
  430 DO 450 J=1,N                                                      00002440
      XG1(J)=XGL(J)+STEP*GOLD*GOLD*GRAD(J)                              00002450
  450 CONTINUE                                                          00002460
      CALL SIMUL(XG1,Y)                                                 00002470
      DO 480 K=1,NR                                                     00002480
      J=N+K                                                             00002490
      XG1(J)=Y(K)                                                       00002500
  480 CONTINUE                                                          00002510
      GO TO 500                                                         00002520
  490 STEP=STEP/GOLD                                                    00002530
  500 DO 550 J=1,N                                                      00002540
      XG2(J)=XGL(J)+STEP*GOLD*GRAD(J)                                   00002550
  550 CONTINUE                                                          00002560
      CALL SIMUL(XG2,Y)                                                 00002570
      DO 580 K=1,NR                                                     00002580
      J=N+K                                                             00002590
      XG2(J)=Y(K)                                                       00002600
  580 CONTINUE                                                          00002610
C                                                                       00002620
  600 STEP=STEP*GOLD                                                    00002630
      IF (XG1(NV).LT.XG2(NV)) GO TO 700                                 00002640
      DO 650 J=1,NTV                                                    00002650
      XG2(J)=XG1(J)                                                     00002660
      XG1(J)=XGL(J)+STEP*GOLD*GOLD*GRAD(J)                              00002670
  650 CONTINUE                                                          00002680
      CALL SIMUL(XG1,Y)                                                 00002690
      DO 680 K=1,NR                                                     00002700
      J=N+K                                                             00002710
      XG1(J)=Y(K)                                                       00002720
  680 CONTINUE                                                          00002730
      GO TO 800                                                         00002740
  700 DO 750 J=1,NTV                                                    00002750
      XGL(J)=XG1(J)                                                     00002760
      XG1(J)=XG2(J)                                                     00002770
      XG2(J)=XGL(J)+STEP*GOLD*GRAD(J)                                   00002780
  750 CONTINUE                                                          00002790
      CALL SIMUL(XG2,Y)                                                 00002800
      DO 780 K=1,NR                                                     00002810
      J=N+K                                                             00002820
      XG2(J)=Y(K)                                                       00002830
  780 CONTINUE                                                          00002840
C                                                                       00002850
  800 IF (STEP.LT.CONF) GO TO 950                                       00002860
      GO TO 600                                                         00002870
  850 STEP=STEP*GOLD                                                    00002880
      IF (IND.NE.1) GO TO 900                                           00002890
      IF (STEP.LT.CONF) GO TO 1200                                      00002900
      GO TO 80                                                          00002910
  900 IF (IND.NE.2) GO TO 490                                           00002920
      GO TO 430                                                         00002930
  950 IF (XG1(NV).LT.XG2(NV)) GO TO 1050                                00002940
      DO 1000 J=1,NTV                                                   00002950
```

```
      XC(J)=XG1(J)                                                      00002960
      X(M,J)=XC(J)                                                      00002970
 1000 CONTINUE                                                          00002980
      GO TO 1300                                                        00002990
 1050 DO 1100 J=1,NTV                                                   00003000
      XC(J)=XG2(J)                                                      00003010
      X(M,J)=XC(J)                                                      00003020
 1100 CONTINUE                                                          00003030
      GO TO 1300                                                        00003040
 1200 ISTOP=1                                                           00003050
 1300 RETURN                                                            00003060
      END                                                               00003070
C                                                                       00003080
      SUBROUTINE FSBPT(IOUT,JACT,XT)                                    00003090
      DIMENSION XT(2),JACT(30)                                          00003100
      COMMON X(40,30),XV(1200),XL(20),XU(20),YL(10),YU(10),             00003110
     &       Y(10),GRAD(20),N,MVTEX,AVTEX,NV,M,MCB,NK,STEP,RAD,         00003120
     &       RRRA,CONF,GOLD,ISTOP,NTV                                   00003130
      COMMON/BLEE1/XLP(10),XUP(10)                                      00003140
      COMMON/BLEE3/XL(10),XU(10),YL,YU                                  00003150
      IOUT=0                                                            00003160
      CALL SIMUL(XT,Y,NR)                                               00003170
      DO 50 J=1,NTV                                                     00003180
      JACT(J)=0                                                         00003190
   50 CONTINUE                                                          00003200
      DO 160 J=1,N                                                      00003210
      IF (XT(J).GE.XLP(J)) GO TO 120                                    00003220
      GO TO 140                                                         00003230
  120 IF (XT(J).LE.XUP(J)) GO TO 160                                    00003240
  140 JACT(J)=1                                                         00003250
      IOUT=1                                                            00003260
  160 CONTINUE                                                          00003270
      DO 260 L=2,NR                                                     00003280
      J=N+L                                                             00003290
      IF (Y(L).GE.YL) GO TO 220                                         00003300
      GO TO 240                                                         00003310
  220 IF (Y(L).LE.YU) GO TO 260                                         00003320
  240 JACT(J)=1                                                         00003330
      IOUT=1                                                            00003340
  260 CONTINUE                                                          00003350
      RETURN                                                            00003360
      END                                                               00003370
C                                                                       00003380
      SUBROUTINE PROJCT(JACT)                                           00003390
      DIMENSION G(4,4),GTRA(4,4),GPRO(4,4),GPRO1(4,4),GV(16),           00003400
     &          DV1(30),IV1(30),XBAR(30),STD(30),RX(30,30),RT(465),     00003410
     &          RY(30),BV(30),SB(30),ANS(10),JACT(30)                   00003420
      COMMON X(40,30),XV(1200),XL(20),XU(20),YL(10),YU(10),             00003430
     &       Y(10),GRAD(20),N,MVTEX,AVTEX,NV,M,MCB,NK,STEP,RAD,         00003440
     &       RRRA,CONF,GOLD,ISTOP,NTV                                   00003450
      COMMON/BLS/DV(30),ISAVE(30)                                       00003460
      ICONS=0                                                           00003470
      DO 100 J=1,N                                                      00003480
      IF (JACT(J).EQ.0) GO TO 100                                       00003490
      ICONS=ICONS+1                                                     00003500
      DO 50 K=1,N                                                       00003510
      G(ICONS,K)=0.                                                     00003520
   50 CONTINUE                                                          00003530
      G(ICONS,J)=1.                                                     00003540
```

```
  100 CONTINUE                                                          00003550
      DO 250 L=2,NR                                                     00003560
      J=N+L                                                             00003570
      IF (IACT(J).EQ.0) GO TO 250                                       00003580
      ICONS=ICONS+1                                                     00003590
      DO 150 I=1,MOB                                                    00003600
      K=MOBAN+I                                                         00003610
      XV(K)=X(I,J)                                                      00003620
  150 CONTINUE                                                          00003630
      CALL CORRB(MOB,NV,1,XV,XBAR,STD,RX,RT,BV,DV1,TV1)                 00003640
      CALL ORDER(NV,RT,NV,N,ISAVE,RX,RY)                                00003650
      CALL MINV(RX,N,DET,DV1,TV1)                                       00003660
      CALL MULTR(MOB,N,XBAR,STD,BV,RX,RY,ISAVE,DV1,SB,TV1,ANS)          00003670
      DO 200 K=1,N                                                      00003680
      G(ICONS,K)=DV1(K)                                                 00003690
  200 CONTINUE                                                          00003700
  250 CONTINUE                                                          00003710
      DO 500 J=1,N                                                      00003720
      DO 500 I=1,ICONS                                                  00003730
      K=ICONS*(J-1)+I                                                   00003740
      BV(K)=G(I,J)                                                      00003750
  500 CONTINUE                                                          00003760
      DO 300 I=1,ICONS                                                  00003770
      WRITE (6,1300) (G(I,J),J=1,N)                                     00003780
  300 CONTINUE                                                          00003790
 1300 FORMAT (5X,8F14.6)                                                00003800
      CALL GMTRA(BV,GTRA,ICONS,N)                                       00003810
      CALL GMPRD(BV,GTRA,GPRD,ICONS,N,ICONS)                            00003820
      CALL MINV(GPRD,ICONS,DET,DV1,TV1)                                 00003830
      CALL GMPRD(,TRA,GPRD,GPRDI,N,ICONS,ICONS)                         00003840
      CALL GMPRD(GPRDI,BV,GPRD,N,ICONS,N)                               00003850
      CALL GMPRD(GPRD,BV,GPRDI,N,N,1)                                   00003860
      CALL GMSUB(BV,GPRDI,BV,N,1)                                       00003870
      RETURN                                                            00003880
      END                                                               00003890
C                                                                       00003900
      SUBROUTINE DATA(NV,DV)                                            00003910
      DIMENSION DV(1)                                                   00003920
    1 FORMAT (12F6.0)                                                   00003930
      READ (3,1) (DV(J),J=1,NV)                                         00003940
      WRITE (1) (DV(J),J=1,NV)                                          00003950
      RETURN                                                            00003960
      END                                                               00003970
      SUBROUTINE SIMUL(XS,Y)                                            00003980
C                                                                       00003990
C*****THIS IS THE AMSAA TANK DULL SIMULATION MODEL                      00004000
C                                                                       00004010
      DIMENSION XS(30), Y(20)                                           00004020
      DIMENSION TMDB(45),TMDR(45),SDB(45),SDR(45),SKB(45),SKR(45)       00004030
      COMMON/BL1/BPHL,BPHH,BPHS,BPHL,BKHL,BKHH,BKHS,BKHL,BS             00004040
      COMMON/BL2/IR1,IRS,RPHL,RPHH,RPHS,RPHL,RKHL,RKHH,RKHS,RKHL,RS     00004050
      COMMON/BL3/ITER                                                   00004060
      REAL KH(45),KR(45),M(40,40),N(40,40),NDF,NODEC                    00004070
      INTEGER RANGE                                                     00004080
      DATA SIGMA,BLUE,RED,TCUT,ES,TFB,TTB,ID1,ID2,ID3,RANGE,            00004090
     1(ES,TFR,TTR,ID4,ID5,ID6,KILR/.5,4HBLUE,3HRED,120.0,0,             00004100
     2.35,7.,3HBLU,3HBLU,3HRED,1000,.35,1.17,7.,3HRED,3HRED,            00004110
     33HKE),.825/                                                       00004120
      TR1=XS(1)                                                         00004130
```

```
      END                                                                       00005320
      SUBROUTINE CONLOG(XI,SIGX,ETA,SIGY,ZETA,SIGZ)                             00005330
C                                                                               00005340
C*****THIS SUBROUTINE COMPUTES THE MEDIAN AND STANDARD DEVIATION FOR            00005350
C*****CONVOLUTION                                                              00005360
C                                                                               00005370
      XBAR=XI*EXP(.5*SIGX*SIGX)                                                 00005380
      YBAR=ETA*EXP(.5*SIGY*SIGY)                                                00005390
      SSX=XBAR*XBAR*(EXP(SIGX*SIGX)-1.0)                                        00005400
      SSY=YBAR*YBAR*(EXP(SIGY*SIGY)-1.0)                                        00005410
      ESIGZ=1.0+((SSX+SSY)/((XBAR+YBAR)**2))                                    00005420
      ZETA=(XBAR+YBAR)/SQRT(ESIGZ)                                              00005430
      SIGZ=SQRT(ALOG(ESIGZ))                                                    00005440
      RETURN                                                                    00005450
      END                                                                       00005460
      SUBROUTINE KASFT(K,SK,JOUT,KH1,KHH,KHS,KHL,PH1,PHH,PHS,PHL,S,R)           00005470
C                                                                               00005480
C*****THIS SUBROUTINE COMPUTES THE KILL AND SURVIVAL FOR THE TWO TANKS.         00005490
C                                                                               00005500
      DIMENSION SK(45)                                                          00005510
      REAL K(45),KH1,KHH,KHS,KHL,L                                              00005520
      DO 100 I=2,45                                                             00005530
      K(I)=0.0                                                                  00005540
100   SK(I)=0.0                                                                 00005550
      K(1)=PH1*KH1*R                                                            00005560
      SK(1)=1.0-K(1)                                                            00005570
      L=1.0-S                                                                   00005580
      X2=PH1*(1.0-KH1*R)                                                        00005590
      X3=(1.0-PH1)*S                                                            00005600
      X4=(1.0-PH1)*L                                                            00005610
      A12=PHH*KHH*R                                                             00005620
      A13=PHS*KHS*R                                                             00005630
      A14=PHL*KHL*R                                                             00005640
      A22=PHH*(1.0-KHH*R)                                                       00005650
      A23=PHS*(1.0-KHS*R)                                                       00005660
      A24=PHL*(1.0-KHL*R)                                                       00005670
      A32=(1.0-PHH)*S                                                           00005680
      A33=(1.0-PHS)*S                                                           00005690
      A34=(1.0-PHL)*S                                                           00005700
      A42=(1.0-PHH)*L                                                           00005710
      A43=(1.0-PHS)*L                                                           00005720
      A44=(1.0-PHL)*L                                                           00005730
      DO 130 I=2,45                                                             00005740
      K(I)=A12*X2+A13*X3+A14*X4                                                 00005750
      X3P=A32*X2+A33*X3+A34*X4                                                  00005760
      X2P=A22*X2+A23*X3+A24*X4                                                  00005770
      X4P=A42*X2+A43*X3+A44*X4                                                  00005780
      X2=X2P                                                                    00005790
      X3=X3P                                                                    00005800
      X4=X4P                                                                    00005810
      JOUT=I                                                                    00005820
      SK(I)=SK(I-1)-K(I)                                                        00005830
      IF (I.LT.11) GOTO 130                                                     00005840
      IF (SK(I-5).LT..0005) GOTO 135                                            00005850
130   CONTINUE                                                                  00005860
135   RETURN                                                                    00005870
      END                                                                       00005880
      FUNCTION PABAT(I,TA,TU,SA,SB)                                             00005890
C                                                                               00005900
```

APPENDIX C

PROGRAM LISTING

FOR A

SECOND-ORDER RESPONSE SURFACE METHOD

BASED ON A CENTRAL COMPOSITE DESIGN

```
C                                                                       00000010
C                                                                       00000020
C*****  THIS OPTIMIZATION TECHNIQUE USES :                              00000030
C*****     1.  CENTRAL COMPOSITE DESIGN FOR THE VERTICES               00000040
C*****     2.  SECOND ORDER REGRESSION FOR THE RESPONSE SURFACE SURVEY  00000050
C*****     3.  BOX'S COMPLEX SEARCH ON THE ESTIMATED RESPONSE SURFACES  00000060
C*****           N     :  NO. OF INDEPENDENT VARIABLE TO BE OPTIMIZED   00000070
C*****           NR    :  NO. OF DEPENDENT VARIABLE (RESPONSES)         00000080
C*****           MCP   :  WEIGHT OF CENTER POINT IN REGRESSION          00000090
C*****           M     :  NO. OF EXPERIMENTS FOR EACH REGRESSION        00000100
C*****           MOB   :  NO. OF OBSERVATIONS IN EACH REGRESSION        00000110
C                                                                       00000120
      DIMENSION X(10,30),XP(30),C(10),Y(20),Y1(10),YP(10,10),DA(10)     00000130
      COMMON N,AN,NM1,N2D,N2DP1,NR,M,MCP,MOB,MDSIM,NP                   00000140
      COMMON/BLEE1/XLP(10),XUP(10)                                      00000150
      COMMON/BLEE2/ISEED1,ISEED2                                        00000160
      COMMON/BLEE3/XL(10),XU(10),YL,YU                                  00000170
      COMMON/BLE5/RAD(10),XC(10)                                        00000180
      COMMON/BLEE6/ITER                                                 00000190
      COMMON/BLE15/KOUPK,OUPK                                           00000200
C                                                                       00000210
      READ (9,*) MDSIM                                                  00000220
      READ (9,*) N,MCP,MPBC                                             00000230
      READ (9,*) (XL(J),J=1,N),(XU(J),J=1,N)                            00000240
      READ (9,*) (XLP(J),J=1,N),(XUP(J),J=1,N)                          00000250
      READ (9,*) YL,YU                                                  00000260
      READ (9,*) KOUPK                                                  00000270
      READ (9,*) IX                                                     00000280
      READ (9,*) ISEED1,ISEED2                                          00000290
C                                                                       00000300
      WRITE (12,1050)                                                   00000310
      WRITE (12,*) MDSIM                                                00000320
      WRITE (12,*) N,MCP,MPBC                                           00000330
      WRITE (12,*) (XL(J),J=1,N),(XU(J),J=1,N)                          00000340
      WRITE (12,*) (XLP(J),J=1,N),(XUP(J),J=1,N)                        00000350
      WRITE (12,*) YL,YU                                                00000360
      WRITE (12,*) KOUPK                                                00000370
      WRITE (12,*) IX                                                   00000380
      WRITE (12,*) ISEED1,ISEED2                                        00000390
C                                                                       00000400
      OUPK=KOUPK                                                        00000410
      AN=N                                                              00000420
      NP=N+MPBC                                                         00000430
      NPM1=NP-1                                                         00000440
      ANPM1=NPM1                                                        00000450
      ITER=0                                                            00000460
      ITER2=0                                                           00000470
C                                                                       00000480
      DO 100 J=1,N                                                      00000490
      DA(J)=XUP(J)-XLP(J)                                               00000500
      RAD(J)=DA(J)/2.                                                   00000510
      XC(J)=(XUP(J)+XLP(J))/2.                                          00000520
  100 CONTINUE                                                          00000530
C                                                                       00000540
      WRITE (12,5000)                                                   00000550
      CALL CENCOM                                                       00000560
      CALL REGRES                                                       00000570
C                                                                       00000580
      MAX=1                                                             00000590
```

```
      DO 220 I=1,NP                                                     00000600
  120 DO 140 J=1,N                                                      00000610
      CALL RANDU(IX,IY,RUMN)                                            00000620
      IX=IY                                                             00000630
      XP(J)=XLP(J)+RUMN*DA(J)                                           00000640
      X(I,J)=XP(J)                                                      00000650
  140 CONTINUE                                                          00000660
      CALL SELIZE(XP)                                                   00000670
      CALL FSMPT(XP,Y,IOUT)                                             00000680
      IF (IOUT.EQ.2) GO TO 120                                          00000690
      ITER2=ITER2+1                                                     00000700
      WRITE (12,1450) ITER2                                             00000710
      CALL WRITE(XP,Y,N,NR,IOUT)                                        00000720
      IF (IOUT.NE.0) GO TO 120                                          00000730
      DO 200 J=1,NR                                                     00000740
      YP(I,J)=Y(J)                                                      00000750
  200 CONTINUE                                                          00000760
      Y1(I)=Y(1)                                                        00000770
      IF (Y(1).GT.Y1(MAX)) MAX=I                                        00000780
  220 CONTINUE                                                          00000790
      WRITE (12,1230) NP                                                00000800
C                                                                       00000810
  300 CALL WORST(Y1,MIN)                                                00000820
      DO 340 J=1,N                                                      00000830
      SUM=-X(MIN,J)                                                     00000840
      DO 320 I=1,NP                                                     00000850
      SUM=SUM+X(I,J)                                                    00000860
  320 CONTINUE                                                          00000870
      C(J)=SUM/ANPM1                                                    00000880
  340 CONTINUE                                                          00000890
C                                                                       00000900
      R=2.                                                              00000910
  400 R=R*0.5                                                           00000920
      IF (R.LT.1.E-2) GO TO 600                                         00000930
      DO 420 J=1,N                                                      00000940
      XP(J)=C(J)+R*(C(J)-X(MIN,J))                                      00000950
  420 CONTINUE                                                          00000960
      CALL SELIZE(XP)                                                   00000970
      CALL FSMPT(XP,Y,IOUT)                                             00000980
      IF (IOUT.EQ.2) GO TO 400                                          00000990
      ITER2=ITER2+1                                                     00001000
      WRITE (12,1450) ITER2                                             00001010
      CALL WRITE(XP,Y,N,NR,IOUT)                                        00001020
      IF (IOUT.NE.0) GO TO 400                                          00001030
      IF (Y(1).LE.Y1(MIN)) GO TO 400                                    00001040
      DO 440 J=1,NR                                                     00001050
      YP(MIN,J)=Y(J)                                                    00001060
  440 CONTINUE                                                          00001070
      DO 500 J=1,N                                                      00001080
      X(MIN,J)=XP(J)                                                    00001090
  500 CONTINUE                                                          00001100
      Y1(MIN)=Y(1)                                                      00001110
      IF (Y1(MIN).GT.Y1(MAX)) MAX=MIN                                   00001120
      GO TO 300                                                         00001130
C                                                                       00001140
  600 WRITE (12,1600)                                                   00001150
      DO 620 J=1,N                                                      00001160
      WRITE (12,1160) J,X(MAX,J)                                        00001170
  620 CONTINUE                                                          00001180
```

```
      DO 640 J=1,NR                                                     00001190
      WRITE (12,1180) J,YP(MAX,J)                                       00001200
  640 CONTINUE                                                          00001210
C                                                                       00001220
      DO 700 J=1,N                                                      00001230
      XP(J)=X(MAX,J)                                                    00001240
  700 CONTINUE                                                          00001250
      CALL AVGSIM(XP,Y,N,NR)                                            00001260
      WRITE (12,1600)                                                   00001270
      CALL WRITE(XP,Y,N,NR,0)                                           00001280
C                                                                       00001290
 1050 FORMAT (1H1/35X,'*****   LIST OF INPUT DATA   *****'//)           00001300
 1100 FORMAT (30X,'X(',I1,') = ',F14.6)                                 00001310
 1180 FORMAT (30X,'Y(',I1,') = ',F14.6)                                 00001320
 1230 FORMAT (//5X,'THE ABOVE ',I2,' POINTS ARE THE INITIAL POINTS')    00001330
 1450 FORMAT (///1X,I3,' TH ITERATION'//)                               00001340
 1600 FORMAT (////5X,'OPTIMAL SOLUTION'//)                              00001350
 5000 FORMAT (1H1)                                                      00001360
      STOP                                                              00001370
      END                                                               00001380
C                                                                       00001390
      SUBROUTINE CENCOM                                                 00001400
C                                                                       00001410
C*****  THIS SUBROUTINE GENERATES THE EXPERIMENT POINTS FOR THE         00001420
C*****  CENTRAL COMPOSITE DESIGN                                        00001430
C                                                                       00001440
      DIMENSION Y(20),ED(10),XP(30)                                     00001450
      COMMON N,AN,AN1,N2D,N2DP1,NR,M,MCP,MOB,MDSIM,NP                   00001460
      COMMON/BL5/RAD(10),XC(10)                                         00001470
      COMMON/BL7/XL(1000),YE(1000)                                      00001480
      COMMON/BL9/EX(1000),EY(1000)                                      00001490
C                                                                       00001500
      SQRTN=SQRT(AN)                                                    00001510
      MFAC=2**N                                                         00001520
      MAXIS=2*N                                                         00001530
      MVTEX=MFAC+MAXIS                                                  00001540
      M=MVTEX+1                                                         00001550
      MOB=MVTEX+MCP                                                     00001560
      AN1=N-1                                                           00001570
      N2D=N*(N+3)/2                                                     00001580
      N2DP1=N2D+1                                                       00001590
C                                                                       00001600
      DO 50 J=1,N                                                       00001610
      ED(J)=RAD(J)/SQRTN                                                00001620
   50 CONTINUE                                                          00001630
C                                                                       00001640
      KX=0                                                              00001650
      KY=0                                                              00001660
      DO 160 K=1,MFAC                                                   00001670
      KA=K                                                              00001680
      DO 120 J=1,N                                                      00001690
      KB=MOD(KA,2)                                                      00001700
      KA=KA/2                                                           00001710
      XP(J)=XC(J)+ED(J)*(-1)**KB                                        00001720
  120 CONTINUE                                                          00001730
      CALL SELIZE(XP)                                                   00001740
      CALL AVGSIM(XP,Y,N,NR)                                            00001750
      CALL RECORD(XP,Y,KX,KY)                                           00001760
  160 CONTINUE                                                          00001770
```

```
C                                                                               00001780
                                                                                00001790
      DO 280 I=1,N                                                              00001800
      DO 240 IA=1,2                                                             00001810
      DO 220 J=1,N                                                              00001820
      XP(J)=XC(J)                                                               00001830
      IF (I.EQ.J) XP(J)=XC(J)+RAD(J)*(-1)**IA                                   00001840
  220 CONTINUE                                                                  00001850
      CALL SECIZE(XP)                                                           00001860
      CALL AVGSIM(XP,Y,N,NR)                                                    00001870
      CALL RECORD(XP,Y,KX,KY)                                                   00001880
  240 CONTINUE                                                                  00001890
  280 CONTINUE                                                                  00001900
C                                                                               00001910
      CALL SECIZE(XC)                                                           00001920
      CALL AVGSIM(XC,Y,N,NR)                                                    00001930
      CALL RECORD(XC,Y,KX,KY)                                                   00001940
C                                                                               00001950
      IF (MCP.EQ.1) GO TO 500                                                   00001960
      DO 400 I=2,MCP                                                            00001970
      IF(MSIM.EQ.I) GO TO 300                                                   00001980
      CALL AVGSIM(XC,Y,N,NR)                                                    00001990
  300 CALL RECORD(XC,Y,KX,KY)                                                   00002000
  400 CONTINUE                                                                  00002010
C                                                                               00002020
      WRITE (12,1450)                                                           00002030
  500 DO 600 I=1,MOB                                                            00002040
      KX1=(I-1)*N2D+1                                                           00002050
      KX2=I*N2D                                                                 00002060
      KY1=(I-1)*NR+1                                                            00002070
      KY2=I*NR                                                                  00002080
      WRITE (12,1600) I,(EX(J),J=KX1,KX2),(EY(J),J=KY1,KY2)                     00002090
  600 CONTINUE                                                                  00002100
C                                                                               00002110
      CALL GMTRA(EX,XE,N2D,MOB)                                                 00002120
      CALL GMTRA(EY,YE,NR,MOB)                                                  00002130
C                                                                               00002140
 1450 FORMAT (////50X,'REGRESSION TABLEAU '//12X,'X1',12X,'X2',11X,             00002150
     &'X1SQ',10X,'X2SQ',10X,'X1X2',11X,'Y1',12X,'Y2',12X,'Y3',12X,'Y4'/         00002160
     &12X,'Y5',12X,'Y6')                                                        00002170
 1600 FORMAT(/1X,I3,3X,9(F12.6,2X)/7X,9(F12.6,2X)/)                             00002180
      RETURN                                                                    00002190
      END                                                                       00002200
C                                                                               00002210
      SUBROUTINE SECIZE(XS)                                                     00002220
C                                                                               00002230
C*****  THIS SUBROUTINE TRANSFORMS 1ST ORDER INNDEPENDENT VARIABLES             00002240
C*****  INTO 2ND ORDER INDEPENDENT VARIABLES FOR                                00002250
C*****  2ND ORDER REGRESSION                                                    00002260
C                                                                               00002270
      DIMENSION XS(30)                                                          00002280
      COMMON N,AN,NM1,N2D,N2DP1,NR,M,MCP,MOB,MOSIM,NP                           00002290
      K=N                                                                       00002300
      DO 100 J=1,N                                                              00002310
      K=K+1                                                                     00002320
      XS(K)=XS(J)*XS(J)                                                         00002330
  100 CONTINUE                                                                  00002340
C                                                                               00002350
      DO 300 I=1,NM1                                                            00002360
      IP1=I+1
```

```
      DO 200 J=IP1,N                                                    00002370
      K=K+1                                                             00002380
      XS(K)=XS(I)*XS(J)                                                 00002390
  200 CONTINUE                                                          00002400
  300 CONTINUE                                                          00002410
      RETURN                                                            00002420
      END                                                               00002430
C                                                                       00002440
      SUBROUTINE REGRES                                                 00002450
C                                                                       00002460
C*****  THIS SUBROUTINE CALCULATES THE REGRESSION COEFFICIENTS OF THE    00002470
C*****  RESPONSE SURFACES                                               00002480
C                                                                       00002490
      DIMENSION XBAR(30),STD(30),RX(900),RT(465),BV(30),DV(30),TV(30),  00002500
     &          ISAVE(30),RY(30),SB(30),ANS(10)                         00002510
      COMMON N,AN,N1,N2D,N2DP1,NR,M,MCF,MDB,MDSIM,NP                    00002520
      COMMON/BL6/B(10,30)                                               00002530
      COMMON/BL7/XE(1000),YE(1000)                                      00002540
C                                                                       00002550
      NXF=MDB*N2D                                                       00002560
C                                                                       00002570
      DO 50 J=1,N2DP1                                                   00002580
      ISAVE(J)=J                                                        00002590
   50 CONTINUE                                                          00002600
C                                                                       00002610
C                                                                       00002620
      WRITE (12,1150)                                                   00002630
      DO 300 J=1,NR                                                     00002640
      KXF=NXE                                                           00002650
      KYE=MDB*(J-1)                                                     00002660
      DO 100 I=1,MDB                                                    00002670
      KXE=KXE+1                                                         00002680
      KYE=KYE+1                                                         00002690
      XE(KXE)=YE(KYE)                                                   00002700
  100 CONTINUE                                                          00002710
C                                                                       00002720
      CALL CORRE(MDB,N2DP1,1,XE,XBAR,STD,RX,RT,BV,DV,TV)                00002730
      CALL ORDER(N2DP1,RT,N2DP1,N2D,ISAVE,RX,RY)                        00002740
      CALL MINV(RX,N2D,DET,DV,TV)                                       00002750
      CALL MULTR(MDB,N2D,XBAR,STD,BV,RX,RY,ISAVE,DV,SB,TV,ANS)          00002760
      WRITE (12,1160)(DV(L),L=1,N2D),ANS(1),J                           00002770
      WRITE (12,1170) ANS(10)                                           00002780
C                                                                       00002790
      DO 200 JCOEF=1,N2D                                                00002800
      B(J,JCOEF)=DV(JCOEF)                                              00002810
  200 CONTINUE                                                          00002820
      B(J,N2DP1)=ANS(1)                                                 00002830
  300 CONTINUE                                                          00002840
C                                                                       00002850
 1150 FORMAT (////1X,'THE ESTIMATED RESPONSE SURFACES ARE :')           00002860
 1160 FORMAT (/5X,F12.6,' X(1) + ',F12.6,' X(2) + ',F12.6,              00002870
     &' X(1)SQ + ',F12.6,' X(2)SQ + ',F12.6,' X(1)*X(2) + ',           00002880
     &F12.6,' = Y(',I1,')')                                             00002890
 1170 FORMAT (//5X,'F VALUE  =  ',F14.6)                                00002900
      RETURN                                                            00002910
      END                                                               00002920
C                                                                       00002930
      SUBROUTINE PSBPT(XT,Y,IOUT)                                       00002940
C                                                                       00002950
```

```
C*****  THIS SUBROUTINE CALCULATES THE RESPONSES  AND                    00002960
C*****  DETECTS IF THEY ARE FEASIBLE                                     00002970
C                                                                        00002980
      DIMENSION XT(30),Y(20)                                             00002990
      COMMON N,AN,ANM1,N2D,N2DP1,NR,M,MCP,MCB,MDSIM,NP                   00003000
      COMMON/BLEE3/XL(10),XU(10),YL,YU                                   00003010
      COMMON/BLEE1/XLP(10),XUP(10)                                       00003020
      COMMON/BLG/B(10,30)                                                00003030
      IOUT=2                                                             00003040
C                                                                        00003050
      DO 100 J=1,N                                                       00003060
      IF (XT(J).LT.XLP(J)) GO TO 500                                     00003070
      IF (XT(J).GT.XUP(J)) GO TO 500                                     00003080
  100 CONTINUE                                                           00003090
      IOUT=1                                                             00003100
C                                                                        00003110
      DO 300 I=2,NR                                                      00003120
      Y(I)=B(I,N2DP1)                                                    00003130
      DO 200 J=1,N2D                                                     00003140
      Y(I)=Y(I)+B(I,J)*XT(J)                                             00003150
  200 CONTINUE                                                           00003160
      IF (Y(I).LT.YL) GO TO 500                                          00003170
      IF (Y(I).GT.YU) GO TO 500                                          00003180
  300 CONTINUE                                                           00003190
      Y(1)=B(1,N2DP1)                                                    00003200
      DO 350 J=1,N2D                                                     00003210
      Y(1)=Y(1)+B(1,J)*XT(J)                                             00003220
  350 CONTINUE                                                           00003230
      IOUT=0                                                             00003240
C                                                                        00003250
  500 RETURN                                                             00003260
      END                                                                00003270
C                                                                        00003280
      SUBROUTINE WORST(YT,MIN)                                           00003290
C                                                                        00003300
C*****  THIS SUBROUTINE FINDS OUT THE WORST POINT IN THE                 00003310
C*****  POLYGON                                                          00003320
C                                                                        00003330
      DIMENSION YT(30)                                                   00003340
      COMMON N,AN,ANM1,N2D,N2DP1,NR,M,MCP,MCB,MDSIM,NP                   00003350
C                                                                        00003360
      MIN=1                                                              00003370
      YMIN=YT(1)                                                         00003380
      DO 50 J=2,NP                                                       00003390
      IF (YT(J).GE.YMIN) GO TO 50                                        00003400
      MIN=J                                                              00003410
      YMIN=YT(J)                                                         00003420
   50 CONTINUE                                                           00003430
      RETURN                                                             00003440
      END                                                                00003450
C                                                                        00003460
      SUBROUTINE DATA(NV,DV)                                             00003470
C                                                                        00003480
C*****  THIS SUBROUTINE IS CALLED BY THE SSP SUBROUTINE 'CORRE'          00003490
C*****  TO READ IN THE EXPERIMENT DATA NEEDED FOR REGRESSION             00003500
C                                                                        00003510
      DIMENSION DV(1)                                                    00003520
    1 FORMAT (12F6.0)                                                    00003530
      READ (3,1) (DV(J),J=1,NV)                                          00003540
```

```
      WRITE (1) (DV(J),J=1,NV)                                          00003550
      RETURN                                                            00003560
      END                                                               00003570
C                                                                       00003580
      SUBROUTINE WRITE(XW,YW,N,NR,ICODE)                                00003590
      DIMENSION XW(30),YW(20)                                           00003600
C                                                                       00003610
      IF (ICODE.NE.0) GO TO 500                                         00003620
C                                                                       00003630
      DO 100 J=1,N                                                      00003640
      WRITE (12,1100) J,XW(J)                                           00003650
  100 CONTINUE                                                          00003660
      DO 200 J=1,NR                                                     00003670
      WRITE (12,1200) J,YW(J)                                           00003680
  200 CONTINUE                                                          00003690
      GO TO 1000                                                        00003700
C                                                                       00003710
  500 DO 600 J=1,N                                                      00003720
      WRITE (12,1600) J,XW(J)                                           00003730
  600 CONTINUE                                                          00003740
      DO 700 J=1,NR                                                     00003750
      WRITE (12,1700) J,YW(J)                                           00003760
  700 CONTINUE                                                          00003770
C                                                                       00003780
 1100 FORMAT (30X,'X(',I1,') = ',F14.6)                                 00003790
 1200 FORMAT (30X,'Y(',I1,') = ',F14.6)                                 00003800
 1600 FORMAT (80X,'X(',I1,') = ',F14.6)                                 00003810
 1700 FORMAT (80X,'Y(',I1,') = ',F14.6)                                 00003820
 1000 RETURN                                                            00003830
      END                                                               00003840
C                                                                       00003850
      SUBROUTINE RECORD(XP,Y,KX,KY)                                     00003860
      DIMENSION XP(30),Y(20)                                            00003870
      COMMON N,NN,NR1,N20,N20P1,NR,M,MCP,MCB,MDSIM,NP                   00003880
      COMMON/BL9/EX(1000),EY(1000)                                      00003890
C                                                                       00003900
      DO 100 J=1,N20                                                    00003910
      KX=KX+1                                                           00003920
      EX(KX)=XP(J)                                                      00003930
  100 CONTINUE                                                          00003940
      DO 200 J=1,NR                                                     00003950
      KY=KY+1                                                           00003960
      EY(KY)=Y(J)                                                       00003970
  200 CONTINUE                                                          00003980
      RETURN                                                            00003990
      END                                                               00004000
C                                                                       00004010
C                                                                       00004020
      SUBROUTINE AVGSIM(XA,Y,N,NR)                                      00004030
      DIMENSION XA(10),Y(20),YY(20)                                     00004040
      COMMON/BL15/KOUPK,OUPK                                            00004050
C                                                                       00004060
      I=1                                                               00004070
      IF (KOUPK.NE.1) WRITE (12,1120) I                                 00004080
      CALL SIMUL(XA,Y,NP,1)                                             00004090
      REWIND 10                                                         00004100
      IF (KOUPK.EQ.1) GO TO 1000                                        00004110
C                                                                       00004120
      DO 500 I=2,KOUPK                                                  00004130
```

```
      WRITE (12,1120) I                                                 00004140
      CALL SIMUL(XA,YY,NR,I)                                            00004150
      REWIND 10                                                         00004160
      DO 200 J=1,NR                                                     00004170
      Y(J)=Y(J)+YY(J)                                                   00004180
  200 CONTINUE                                                          00004190
  300 CONTINUE                                                          00004200
      DO 400 J=1,NR                                                     00004210
      Y(J)=Y(J)/DUPK                                                    00004220
  400 CONTINUE                                                          00004230
      WRITE (12,1400)                                                   00004240
 1000 CALL WRITE(XA,Y,N,NR,1)                                           00004250
C                                                                       00004260
 1120 FORMAT (/1X,'** SIMUL. **',I3/)                                   00004270
 1400 FORMAT (//1X,'*****  AVERAGE RESULT  *****'//)                   00004280
      RETURN                                                            00004290
      END                                                               00004300
C                                                                       00004310
C                                                                       00004320
      SUBROUTINE SIMUL(XS,Y,IIP1,KTER)                                  00004330
      DIMENSION XS(10),Y(20)                                            00004340
      COMMON /SUMMRY/ TKACT,TKDET,TITSNK,TITOUT,TNEUAT,TNESUC,TNUEAL,   00004350
     1    TLOST,TWAST,RUN                                               00004360
      COMMON /IPRNTX/ IPRINT(25)                                        00004370
     1       /ULIMIT/JPOS,ONEG                                          00004380
     2       /BLCKH/ENTIM(50),BTIME(300),CNTIM(300)                     00004390
     3       /BLCKJ/MINTYP(300),MINCNT(300)                             00004400
     4       /BLCKM/NSHTYP(50),TTIME,NHUNI,NSHIP,NTRAFF,JGROUP,NMINE    00004410
     5       /BLCKO/JET(10),DPATH(300),DEPTH(300),TOP(10),OMGSW,THRSW   00004420
      COMMON/BLCKT/RESTIM(6,10),STRTIM(6,10),IOTRAN(10),KOURR(50),      00004430
     1            NPER(10),PLATIM                                       00004440
     2       /BLCKUU/PMINX(300),PMINY(300)                              00004450
     3       /BLCKXX/TEMPOR,CASTIM,TIMSAV,TIMMAX                        00004460
      COMMON/BLCKWW/COUNTIM(300),NORMTH,CW2,,DXMAX(45),NAVMAX,NGP1,     00004470
     1            INFTYP,KUNTYP                                         00004480
     2       /BLCKZZ/MINLOK(300)                                        00004490
      COMMON/SAVE/PTORE(100,4),QTORE(100,4),TIIN,POORE(100,4),QPORE(100,00004500
     1,4)                                                               00004510
      COMMON/DAMAGE/SDMG(5,10),NRUN,KSMPLT(10),FREQ(16,4),PTOTA(4),QTOTA00004520
     1(4)                                                               00004530
      COMMON/FLEG/JFLCG                                                 00004540
      COMMON/BLEE8/ITER                                                 00004550
      COMMON/BLEE1/XLP(10),XUP(10)                                      00004560
      COMMON/BLEE2/ISEED1,ISEED2                                        00004570
      COMMON/BLEE3/XL(10),XU(10),YL,YU                                  00004580
      COMMON/BLEE16/LSEED                                               00004590
      REAL*8 SEED                                                       00004600
C                                                                       00004610
C INITIALIZING FOR TOTAL AVERAGES IN OUTPUT                             00004620
C                                                                       00004630
      TKACT=0                                                           00004640
      TKDET=0                                                           00004650
      TITSNK=0                                                          00004660
      TITOUT=0                                                          00004670
      TNEUAT=0                                                          00004680
      TNESUC=0                                                          00004690
      TNEUAL=0                                                          00004700
      TLOST=0                                                           00004710
      TWAST=0                                                           00004720
```

APPENDIX D

PROGRAM LISTING

FOR A

SECOND-ORDER RESPONSE SURFACE METHOD

BASED ON BOX'S COMPLEX SEARCH

```
C                                                                       00000010
C*****  THIS PROGRAM USES THE FOLLOWING SCHEMES:                        00000020
C       1. USE BOX'S COMPLEX SEARCH TO FIND THE OPTIMAL SOLUTION        00000030
C       2. CHECK IF THERE IS ANY SUBREGION WHICH HAS EXPERIMENT(S) LESS 00000040
C          THAN REQUIRED. DO COMPENSATE EXPERIMENT(S) IN IT.            00000050
C       3. FROM THE DATA RECORDED IN STEP 1 & 2 , CALCULATE THE         00000060
C          REGRESSION FUNCTIONS                                         00000070
C       4. USE THE REGRESSION FUNCTIONS TO FIND THE OPTIMAL SOLUTION WITH00000080
C          BOX'S COMPLES SEARCH                                         00000090
C***  NOTATIONS  ***                                                    00000100
C              N :  NO. OF INPUT VARIABLES                              00000110
C             NR :  NO. OF RESPONSES                                    00000120
C             NP :  NO. OF VERTICES OF THE POLYGON IN THE COMPLEX SEARCH 00000130
C          SMALL :  TERMINATING CRITERIA                                00000140
C          MIEXR :  MIN. NO. OF EXPERIMENTS IN EACH SUBREGION           00000150
C       JSECT(J) :  NO. OF SUBSECTIONS THE (J)TH VARIABLE IS DEVIDED INTO00000160
C          MAEXP :  MAX. NO. OF EXPERIMENTS IN THE FIRST COMPLEX SEARCH 00000170
C                                                                       00000180
      DIMENSION DA(10),XLR(10),XUR(10)                                  00000190
      COMMON N,AN,NP1,M2D,N2DP1,NR,M,MEXP,NP,ANP,NPM1,ANPM1             00000200
      COMMON/BLEE1/XLP(10),XUP(10)                                      00000210
      COMMON/BLEE2/ISEED1,ISEED2                                        00000220
      COMMON/BLEE3/XL(10),XU(10),YL,YU                                  00000230
      COMMON/BL5/ICHK                                                   00000240
      COMMON/BLEE8/ITER                                                 00000250
      COMMON/BL10/MAEXP,SMALL(2)                                        00000260
      COMMON/BL11/JSECT(10),JAP(10)                                     00000270
      COMMON/BL12/KEXPG(1000)                                           00000280
      COMMON/BL13/IFEGN,MIEXR                                           00000290
      COMMON/BL14/ITER2                                                 00000300
      COMMON/BL15/KOUPK,OUPK                                            00000310
C                                                                       00000320
      READ (9,*) N,MPBC                                                 00000330
      READ (9,*) (XL(J),J=1,N),(XU(J),J=1,N)                            00000340
      READ (9,*) (XLP(J),J=1,N),(XUP(J),J=1,N)                          00000350
      READ (9,*) YL,YU                                                  00000360
      READ (9,*) MAEXP,MAEXR,SMALL(1),SMALL(2)                          00000370
      READ (9,*) (JSECT(J),J=1,N)                                       00000380
      READ (9,*) MIEXR                                                  00000390
      READ (9,*) ICHK                                                   00000400
      READ (9,*) IX                                                     00000410
      READ (9,*) ISEED1,ISEED2                                          00000420
      READ (9,*) KOUPK                                                  00000430
C                                                                       00000440
      WRITE (12,1050)                                                   00000450
      WRITE (12,*) N,MPBC                                               00000460
      WRITE (12,*) (XL(J),J=1,N),(XU(J),J=1,N)                          00000470
      WRITE (12,*) (XLP(J),J=1,N),(XUP(J),J=1,N)                        00000480
      WRITE (12,*) YL,YU                                                00000490
      WRITE (12,*) MAEXP,MAEXR,SMALL(1),SMALL(2)                        00000500
      WRITE (12,*) (JSECT(J),J=1,N)                                     00000510
      WRITE (12,*) MIEXR                                                00000520
      WRITE (12,*) ICHK                                                 00000530
      WRITE (12,*) IX                                                   00000540
      WRITE (12,*) ISEED1,ISEED2                                        00000550
      WRITE (12,*) KOUPK                                                00000560
C                                                                       00000570
      OUPK=KOUPK                                                        00000580
      KX=0                                                              00000590
```

```
      KY=0                                                                      00000600
C                                                                               00000610
      DO 100 J=1,N                                                              00000620
      DA(J)=XUP(J)-XLP(J)                                                       00000630
      SECT=JSECT(J)                                                             00000640
      DAP(J)=DA(J)/SECT                                                         00000650
  100 CONTINUE                                                                  00000660
C                                                                               00000670
      IREGN=1                                                                   00000680
      DO 300 J=1,N                                                              00000690
      IREGN=IREGN*JSECT(J)-                                                     00000700
  300 CONTINUE                                                                  00000710
C                                                                               00000720
      DO 500 K=1,IREGN                                                          00000730
      KEXR(K)=0                                                                 00000740
  500 CONTINUE                                                                  00000750
C                                                                               00000760
C                                                                               00000770
      AN=N                                                                      00000780
      NP=N+MPBC                                                                 00000790
      NPM1=NP-1                                                                 00000800
      ANP=NP                                                                    00000810
      ANPM1=NPM1                                                                00000820
      N2D=NP*(N+3)/2                                                            00000830
C                                                                               00000840
      MEXP=0                                                                    00000850
      ITER=0                                                                    00000860
      ITER2=10000                                                               00000870
      IBOX=1                                                                    00000880
      WRITE (12,5001)                                                           00000890
      CALL BOXCOM(IBOX,IX,KX,KY,XLP,XUP,DA,YIMAX)                               00000900
C                                                                               00000910
      IF(MIEXR.EQ.0) GO TO 1000                                                 00000920
      WRITE (12,1400)                                                           00000930
      CALL COMPEN(KX,KY,ITRY,XLR,XUR,YIMAX)                                     00000940
C                                                                               00000950
      IF (ITRY.EQ.0) GO TO 1000                                                 00000960
      ITER2=30000                                                               00000970
      MAEXP=MEXP+MAEXR                                                          00000980
      WRITE (12,5001)                                                           00000990
      CALL BOXCOM(IBOX,IX,KX,KY,XLR,XUR,DAP,YIMAX)                              00001000
C                                                                               00001010
 1000 WRITE (12,5001)                                                           00001020
      CALL REGRES                                                               00001030
C                                                                               00001040
      WRITE (12,5001)                                                           00001050
      IBOX=2                                                                    00001060
      ITER2=20000                                                               00001070
      CALL BOXCOM(IBOX,IX,KX,KY,XLP,XUP,DA,YIMAX)                               00001080
C                                                                               00001090
 1050 FORMAT (//45X,'*****   LIST OF INPUT DATA   *****'////)                   00001100
 1400 FORMAT (1H1//45X,'*****   COMPENSATE EXPERIMENTS   *****'////)            00001110
 5001 FORMAT (1H1)                                                              00001120
      STOP                                                                      00001130
      END                                                                       00001140
C                                                                               00001150
      SUBROUTINE BOXCOM(IBOX,IX,KX,KY,XL,XU,DA,YIMAX)                           00001160
      DIMENSION X(10,30),XP(30),C(10),Y(10),YP(10,10),XL(10),XU(10),DA(10 00001170
     &          0),YI(10)                                                      00001180
```

```
      COMMON N,AN,NM1,N2D,N2DP1,NR,M,MEXP,NP,ANP,NPM1,ANPM1                00001190
      COMMON/BLEE1/XLP(10),XUP(10)                                        00001200
      COMMON/BLEE2/ISEED1,ISEED2                                          00001210
      COMMON/BLS/ICMK                                                     00001220
      COMMON/BLEEH/ITER                                                   00001230
      COMMON/BL10/MAEXP,SMALL(2)                                          00001240
      COMMON/BL14/ITER2                                                   00001250
C                                                                         00001260
      WRITE (12,1700)                                                     00001270
      MAX=1                                                               00001280
C                                                                         00001290
      DO 220 I=1,NP                                                       00001300
  120 DO 140 J=1,N                                                        00001310
      CALL RANDU(IX,IY,RNUM)                                              00001320
      IX=IY                                                               00001330
      XP(J)=XL(J)+RNUM*DA(J)                                              00001340
      X(I,J)=XP(J)                                                        00001350
  140 CONTINUE                                                            00001360
      CALL SECIZE(XP)                                                     00001370
      IF(IBOX.EQ.1) CALL FSBPT1(XP,Y,IOUT)                                00001380
      IF(IBOX.EQ.2) CALL FSBPT2(XP,Y,IOUT)                                00001390
      IF(IOUT.EQ.2) GO TO 120                                             00001400
      ITER2=ITER2+1                                                       00001410
      WRITE (12,1450) ITER2                                               00001420
      CALL WRITE(XP,Y,N,NR,IOUT)                                          00001430
      IF(IBOX.EQ.2) GO TO 160                                             00001440
      CALL RECORD(XP,Y,KX,KY)                                             00001450
      MEXP=MEXP+1                                                         00001460
      CALL REGION(XP)                                                     00001470
  160 IF (IOUT.EQ.1) GO TO 120                                            00001480
      DO 200 J=1,NR                                                       00001490
      X(I,J)=Y(J)                                                         00001500
  200 CONTINUE                                                            00001510
      Y1(I)=Y(1)                                                          00001520
      IF (Y(1).GT.Y1(MAX)) MAX=I                                          00001530
  220 CONTINUE                                                            00001540
      WRITE (12,1230) NP                                                  00001550
C                                                                         00001560
  300 CALL WORST(Y1,MIN)                                                  00001570
      DO 340 J=1,N                                                        00001580
      SUM=-X(MIN,J)                                                       00001590
      DO 320 I=1,NP                                                       00001600
      SUM=SUM+X(I,J)                                                      00001610
  320 CONTINUE                                                            00001620
      C(J)=SUM/ANPM1                                                      00001630
  340 CONTINUE                                                            00001640
C                                                                         00001650
      R=2.                                                                00001660
  400 R=R*0.5                                                             00001670
      DO 420 J=1,N                                                        00001680
      XP(J)=C(J)+R*(C(J)-X(MIN,J))                                        00001690
  420 CONTINUE                                                            00001700
      CALL SECIZE(XP)                                                     00001710
      IF(IBOX.EQ.1) CALL FSBPT1(XP,Y,IOUT)                                00001720
      IF(IBOX.EQ.2) CALL FSBPT2(XP,Y,IOUT)                                00001730
      IF(IOUT.EQ.2) GO TO 400                                             00001740
      IF(IBOX.EQ.2) GO TO 430                                             00001750
      ITER2=ITER2+1                                                       00001760
      WRITE (12,1450) ITER2                                               00001770
```

```
      CALL WRITE(XP,Y,N,NR,IOUT)                                        00001780
      CALL RECORD(XP,Y,KX,KY)                                           00001790
      MEXP=MEXP+1                                                       00001800
      CALL REGION(XP)                                                   00001810
      IF(MEXP.GE.MAEXP.AND.IOUT.EQ.1) GO TO 600                         00001820
  430 IF (IOUT.EQ.1) GO TO 400                                          00001830
      IF (Y(I).LE.YI(MIN)) GO TO 400                                    00001840
      DO 500 J=1,N                                                      00001850
      X(MIN,J)=XP(J)                                                    00001860
  500 CONTINUE                                                          00001870
      DO 440 J=1,NR                                                     00001880
      YP(MIN,J)=Y(J)                                                    00001890
  440 CONTINUE                                                          00001900
      YI(MIN)=Y(I)                                                      00001910
      IF (YI(MIN).GT.YI(MAX)) MAX=MIN                                   00001920
      IF(IBOX.EQ.1.AND.MEXP.GE.MAEXP) GO TO 600                         00001930
C                                                                       00001940
      DO 510 I=1,NP                                                     00001950
      A=YI(MAX)-YI(I)                                                   00001960
      IF (A.GT.SMALL(IBOX)) GO TO 300                                   00001970
  510 CONTINUE                                                          00001980
C                                                                       00001990
      IF (ICHK.EQ.0) GO TO 600                                          00002000
      DO 520 J=1,N                                                      00002010
      SUM=X(1,J)                                                        00002020
      DO 520 I=2,NP                                                     00002030
      SUM=SUM+X(I,J)                                                    00002040
  520 CONTINUE                                                          00002050
      XP(J)=SUM/ANP                                                     00002060
  520 CONTINUE                                                          00002070
C                                                                       00002080
      CALL SEGIZE(XP)                                                   00002090
      IF(IBOX.EQ.1) CALL FSBPT1(XP,Y,IOUT)                              00002100
      IF(IBOX.EQ.2) CALL FSBPT2(XP,Y,IOUT)                              00002110
      IF (IBOX.EQ.2) GO TO 540                                          00002120
      CALL RECORD(XP,Y,KX,KY)                                           00002130
      MEXP=MEXP+1                                                       00002140
      CALL REGION(XP)                                                   00002150
  540 ITER2=ITER2+1                                                     00002160
      WRITE (12,1450) ITER2                                             00002170
      WRITE (12,1500)                                                   00002180
      CALL WRITE(XP,Y,N,NR,IOUT)                                        00002190
C                                                                       00002200
      DO 550 I=1,NP                                                     00002210
      A=Y(I)-YI(I)                                                      00002220
      IF (A.GT.SMALL(IBOX)) GO TO 560                                   00002230
  550 CONTINUE                                                          00002240
      GO TO 600                                                         00002250
C                                                                       00002260
  560 CALL WORST(YI,MIN)                                                00002270
      IF (Y(I).LE.YI(MIN)) GO TO 600                                    00002280
      DO 570 J=1,N                                                      00002290
      X(MIN,J)=XP(J)                                                    00002300
  570 CONTINUE                                                          00002310
      DO 580 J=1,NR                                                     00002320
      YP(MIN,J)=Y(J)                                                    00002330
  580 CONTINUE                                                          00002340
      YI(MIN)=Y(I)                                                      00002350
      IF (YI(MAX).GT.YI(MIN)) GO TO 300                                 00002360
```

```
      MAX=MIN                                                           00002370
      GO TO 300                                                         00002380
C                                                                       00002390
  600 WRITE (12,1600)                                                   00002400
      DO 620 J=1,N                                                      00002410
      WRITE (12,1160) J,X(MAX,J)                                        00002420
  620 CONTINUE                                                          00002430
      DO 640 J=1,NR                                                     00002440
      WRITE (12,1180) J,YP(MAX,J)                                       00002450
  640 CONTINUE                                                          00002460
C                                                                       00002470
      YMAX=Y1(MAX)                                                      00002480
      IF (IBOX.EQ.1) RETURN                                             00002490
      DO 700 J=1,N                                                      00002500
      XP(J)=X(MAX,J)                                                    00002510
  700 CONTINUE                                                          00002520
      CALL FSORTI(XP,Y,IOUT)                                            00002530
      WRITE (12,1600)                                                   00002540
      CALL WRITE(XP,Y,N,NR,IOUT)                                        00002550
C                                                                       00002560
 1160 FORMAT (30X,'X(',I1,') = ',F14.6)                                 00002570
 1180 FORMAT (30X,'Y(',I1,') = ',F14.6)                                 00002580
 1230 FORMAT (//5X,'THE ABOVE ',I2,' POINTS ARE THE INITIAL POINTS'//)  00002590
 1450 FORMAT (///1X,I5,' TH ITERATION'/)                                00002600
 1500 FORMAT (7X,'**  CENTROID CHECKING POINT  **'/)                    00002610
 1600 FORMAT (////5X,'OPTIMAL SOLUTION'//)                              00002620
 1700 FORMAT (//27X,'*****   FEASIBLE POINT   *****',15X,               00002630
     &       '*****   INFEASIBLE POINT   *****')                      00002640
C                                                                       00002650
 5000 FORMAT (//)                                                       00002660
      RETURN                                                            00002670
      END                                                               00002680
C                                                                       00002690
C                                                                       00002700
      SUBROUTINE SECIZE(XS)                                             00002710
C                                                                       00002720
C*****  THIS SUBROUTINE TRANSFORMS 1ST ORDER INNDEPENDENT VARIABLES     00002730
C*****  INTO 2ND ORDER INDEPENDENT VARIABLES FOR                        00002740
C*****  2ND ORDER REGRESSION                                            00002750
C                                                                       00002760
      DIMENSION XS(30)                                                  00002770
      COMMON N,AN,NM1,N2D,N2DP1,NR,M,MEXP,NP,ANP,NPM1,ANPM1             00002780
      K=N                                                               00002790
      DO 100 J=1,N                                                      00002800
      K=K+1                                                             00002810
      XS(K)=XS(J)*XS(J)                                                 00002820
  100 CONTINUE                                                          00002830
C                                                                       00002840
      DO 300 I=1,NM1                                                    00002850
      IP1=I+1                                                           00002860
      DO 200 J=IP1,N                                                    00002870
      K=K+1                                                             00002880
      XS(K)=XS(I)*XS(J)                                                 00002890
  200 CONTINUE                                                          00002900
  300 CONTINUE                                                          00002910
      RETURN                                                            00002920
      END                                                               00002930
C                                                                       00002940
      SUBROUTINE REGRES                                                 00002950
```

```
C                                                                       00002960
C*****  THIS SUBROUTINE CALCULATES THE REGRESSION COEFFICIENTS OF THE    00002970
C*****  RESPONSE SURFACES                                               00002980
C                                                                       00002990
      DIMENSION XBAR(30),STD(30),RX(900),RT(465),BV(30),DV(30),TV(30),  00003000
     &          ISAVE(30),RY(30),SB(30),ANS(10),XE(1000),YE(1000)       00003010
      COMMON N,AN,ANM1,N2D,N2DP1,NR,M,MEXP,NP,ANP,NPM1,ANPM1            00003020
      COMMON/BL6/B(10,30)                                               00003030
      COMMON/BL9/EX(1000),EY(1000)                                      00003040
C                                                                       00003050
      NXE=MEXP*N2D                                                      00003060
      N2DP1=N2D+1                                                       00003070
C                                                                       00003080
      WRITE (12,1450)                                                   00003090
  500 DO 600 I=1,MEXP                                                   00003100
      KX1=(I-1)*N2D+1                                                   00003110
      KX2=I*N2D                                                         00003120
      KY1=(I-1)*NR+1                                                    00003130
      KY2=I*NR                                                          00003140
      WRITE (12,1600) I,(EX(J),J=KX1,KX2),(EY(J),J=KY1,KY2)             00003150
  600 CONTINUE                                                          00003160
C                                                                       00003170
      CALL GMTRA(EX,XE,N2D,MEXP)                                        00003180
      CALL GMTRA(EY,YE,NR,MEXP)                                         00003190
C                                                                       00003200
      DO 50 J=1,N2DP1                                                   00003210
      ISAVE(J)=J                                                        00003220
   50 CONTINUE                                                          00003230
C                                                                       00003240
C                                                                       00003250
      WRITE (12,1150)                                                   00003260
      DO 300 J=1,NR                                                     00003270
      KXE=NXE                                                           00003280
      KYE=MEXP*(J-1)                                                    00003290
      DO 100 I=1,MEXP                                                   00003300
      KXE=KXE+1                                                         00003310
      KYE=KYE+1                                                         00003320
      XE(KXE)=YE(KYE)                                                   00003330
  100 CONTINUE                                                          00003340
C                                                                       00003350
      CALL CORRE(MEXP,N2DP1,1,XL,XBAR,STD,RX,RT,BV,DV,TV)               00003360
      CALL ORDER(N2DP1,RT,N2DP1,N2D,ISAVE,RX,RY)                        00003370
      CALL MINV(RX,N2D,DET,DV,TV)                                       00003380
      CALL MULTR(MEXP,N2D,XBAR,STD,BV,RX,RY,ISAVE,DV,SB,TV,ANS)         00003390
      WRITE (12,1160)(DV(L),L=1,N2D),ANS(1),J                           00003400
      FVALU=ANS(10)                                                     00003410
      WRITE (12,1170) FVALU                                             00003420
C                                                                       00003430
      DO 200 JCOEF=1,N2D                                                00003440
      B(J,JCOEF)=DV(JCOEF)                                              00003450
  200 CONTINUE                                                          00003460
      B(J,N2DP1)=ANS(1)                                                 00003470
  300 CONTINUE                                                          00003480
C                                                                       00003490
 1450 FORMAT (//45X,'*****  REGRESSION TABLEAU  *****'//12X,'X1',12X,   00003500
     &'X2',11X,'X1SQ',10X,'X2SQ',10X,'X1X2',11X,'Y1',12X,'Y2',12X,'Y3',  00003510
     &12X,'Y4'/12X,'Y5',12X,'Y6')                                       00003520
 1600 FORMAT(/1X,I3,3X,9(F12.6,2X)/7X,9(F12.6,2X))                      00003530
 1150 FORMAT (////1X,'THE ESTIMATED RESPONSE SURFACES ARE :')           00003540
```

```
 1160 FORMAT (/5X,F12.6,' X(1) + ',F12.6,' X(2) + ',F12.6,                      00003550
     &' X(1)SQ + ',F12.6,' X(2)SQ + ',F12.6,' X(1)*X(2) + ',                    00003560
     &F12.6,' = Y(',I1,')')                                                     00003570
 1170 FORMAT (//5X,'F VALUE  =  ',F12.6)                                        00003580
      RETURN                                                                    00003590
      END                                                                       00003600
C                                                                               00003610
      SUBROUTINE FSBPT1(XT,Y,IOUT)                                              00003620
C                                                                               00003630
C*****  THIS SUBROUTINE INDICATES IF THE CURRENT POINT IS FEASIBLE              00003640
C*****          IOUT=1    :  VIOLATES THE IMPLICIT CONSTRAINT(S)               00003650
C*****          IOUT=2    :  VIOLATES THE EXPLICIT CONSTRAINT(S)               00003660
C                                                                               00003670
      DIMENSION XT(30),Y(10),YY(10)                                             00003680
      COMMON N,AN,N1,N2,N2P1,NR,M,MEXP,NP,ANP,NPM1,ANPM1                        00003690
      COMMON/BLEE1/XLP(10),XUP(10)                                              00003700
      COMMON/BLEE3/XL(10),XU(10),YL,YU                                          00003710
      COMMON/BL15/KOUPK,DUPK                                                    00003720
      IOUT=2                                                                    00003730
      DO 100 J=1,N                                                              00003740
      IF (XT(J).LT.XLP(J)) GO TO 500                                            00003750
      IF (XT(J).GT.XUP(J)) GO TO 500                                            00003760
  100 CONTINUE                                                                  00003770
      IOUT=1                                                                    00003780
C                                                                               00003790
      I=1                                                                       00003800
      IF (KOUPK.NE.1) WRITE (12,1120)  I                                        00003810
      CALL SIMUL(XT,Y,NR,1)                                                     00003820
      REWIND 10                                                                 00003830
      IF (KOUPK.EQ.1) GO TO 190                                                 00003840
      DO 160 I=2,KOUPK                                                          00003850
      WRITE (12,1120) I                                                         00003860
      CALL SIMUL(XT,YY,NR,2)                                                    00003870
      REWIND 10                                                                 00003880
      DO 140 J=1,NR                                                             00003890
      Y(J)=Y(J)+YY(J)                                                           00003900
  140 CONTINUE                                                                  00003910
  160 CONTINUE                                                                  00003920
      DO 180 J=1,NR                                                             00003930
      Y(J)=Y(J)/DUPK                                                            00003940
  180 CONTINUE                                                                  00003950
C                                                                               00003960
  190 DO 200 J=2,NR                                                             00003970
      IF (Y(J).LT.YL) GO TO 500                                                 00003980
      IF (Y(J).GT.YU) GO TO 500                                                 00003990
  200 CONTINUE                                                                  00004000
      IOUT=0                                                                    00004010
C                                                                               00004020
 1120 FORMAT (/1X,'** SIMUL.',I3,' **'/)                                        00004030
  500 RETURN                                                                    00004040
      END                                                                       00004050
C                                                                               00004060
      SUBROUTINE FSBPT2(XT,Y,IOUT)                                              00004070
C                                                                               00004080
C*****  THIS SUBROUTINE CALCULATES THE RESPONSES FROM THE REGRESSION            00004090
C*****  FUNCTIONS AND INDICATES IF THE CURRENT POINT IS FEASIBLE                00004100
C*****          IOUT=1    :  VIOLATES THE IMPLICIT CONSTRAINT(S)               00004110
C*****          IOUT=2    :  VIOLATES THE EXPLICIT CONSTRAINT(S)               00004120
C                                                                               00004130
```

```
      DIMENSION XT(30),Y(10)                                            00004140
      COMMON N,AN,NM1,N2D,N2DP1,NR,M,MEXP,NP,ANP,NPM1,ANPM1             00004150
      COMMON/BLCE3/XL(10),XU(10),YL,YU                                  00004160
      COMMON/BLCE1/XLP(10),XUP(10)                                      00004170
      COMMON/BL6/B(10,30)                                               00004180
      IOUT=2                                                            00004190
C                                                                       00004200
      DO 100 J=1,N                                                      00004210
      IF (XT(J).LT.XLP(J)) GO TO 500                                    00004220
      IF (XT(J).GT.XUP(J)) GO TO 500                                    00004230
  100 CONTINUE                                                          00004240
      IOUT=1                                                            00004250
C                                                                       00004260
      DO 300 I=2,NR                                                     00004270
      Y(I)=B(I,N2DP1)                                                   00004280
      DO 200 J=1,N2D                                                    00004290
      Y(I)=Y(I)+B(I,J)*XT(J)                                            00004300
  200 CONTINUE                                                          00004310
      IF (Y(I).LT.YL) GO TO 500                                         00004320
      IF (Y(I).GT.YU) GO TO 500                                         00004330
  300 CONTINUE                                                          00004340
      Y(1)=B(1,N2DP1)                                                   00004350
      DO 350 J=1,N2D                                                    00004360
      Y(1)=Y(1)+B(1,J)*XT(J)                                            00004370
  350 CONTINUE                                                          00004380
      IOUT=0                                                            00004390
C                                                                       00004400
  500 RETURN                                                            00004410
      END                                                               00004420
C                                                                       00004430
      SUBROUTINE WORST(YT,MIN)                                          00004440
C                                                                       00004450
C*****  THIS SUBROUTINE FINDS OUT THE WORST POINT IN THE                 00004460
C*****  POLYGON                                                         00004470
C                                                                       00004480
      DIMENSION YT(30)                                                  00004490
      COMMON N,AN,NM1,N2D,N2DP1,NR,M,MEXP,NP,ANP,NPM1,ANPM1             00004500
C                                                                       00004510
      MIN=1                                                             00004520
      YMIN=YT(1)                                                        00004530
      DO 50 J=2,NP                                                      00004540
      IF (YT(J).GE.YMIN) GO TO 50                                       00004550
      MIN=J                                                             00004560
      YMIN=YT(J)                                                        00004570
   50 CONTINUE                                                          00004580
      RETURN                                                            00004590
      END                                                               00004600
C                                                                       00004610
      SUBROUTINE DATA(NV,DV)                                            00004620
C                                                                       00004630
C*****  THIS SUBROUTINE IS CALLED BY THE SSP SUBROUTINE 'CORRE'         00004640
C*****  TO READ IN THE EXPERIMENT DATA NEEDED FOR REGRESSION            00004650
C                                                                       00004660
      DIMENSION DV(1)                                                   00004670
    1 FORMAT (12F6.0)                                                   00004680
      READ (3,1) (DV(J),J=1,NV)                                         00004690
      WRITE (1) (DV(J),J=1,NV)                                          00004700
      RETURN                                                            00004710
      END                                                               00004720
```

```
C
      SUBROUTINE WRITE(XW,YW,N,NR,IOUT)                                 00004730
      DIMENSION XW(30),YW(10)                                           00004740
C                                                                       00004750
      IF (IOUT.NE.0) GO TO 500                                          00004760
C                                                                       00004770
      DO 100 J=1,N                                                      00004780
      WRITE (12,1100) J,XW(J)                                           00004790
  100 CONTINUE                                                          00004800
      DO 200 J=1,NR                                                     00004810
      WRITE (12,1200) J,YW(J)                                           00004820
  200 CONTINUE                                                          00004830
      GO TO 1000                                                        00004840
C                                                                       00004850
  500 DO 600 J=1,N                                                      00004860
      WRITE (12,1600) J,XW(J)                                           00004870
  600 CONTINUE                                                          00004880
      DO 700 J=1,NR                                                     00004890
      WRITE (12,1700) J,YW(J)                                           00004900
  700 CONTINUE                                                          00004910
C                                                                       00004920
 1100 FORMAT (30X,'X(',I1,') = ',F14.6)                                 00004930
 1200 FORMAT (30X,'Y(',I1,') = ',F14.6)                                 00004940
 1600 FORMAT (80X,'X(',I1,') = ',F14.6)                                 00004950
 1700 FORMAT (80X,'Y(',I1,') = ',F14.6)                                 00004960
 1000 RETURN                                                            00004970
      END                                                               00004980
C                                                                       00004990
      SUBROUTINE RECORD(XP,Y,KX,KY)                                     00005000
      DIMENSION XP(30),Y(10)                                            00005010
      COMMON N,AN,NM1,N2D,N2DP1,NR,M,MEXP,NP,ANP,NPM1,ANPM1             00005020
      COMMON/BL9/EX(1000),EY(1000)                                      00005030
C                                                                       00005040
      DO 100 J=1,N2D                                                    00005050
      KX=KX+1                                                           00005060
      EX(KX)=XP(J)                                                      00005070
  100 CONTINUE                                                          00005080
      DO 200 J=1,NR                                                     00005090
      KY=KY+1                                                           00005100
      EY(KY)=Y(J)                                                       00005110
  200 CONTINUE                                                          00005120
      RETURN                                                            00005130
      END                                                               00005140
C                                                                       00005150
      SUBROUTINE REGION(XP)                                             00005160
      DIMENSION XP(10),JREGN(10)                                        00005170
      COMMON N,AN,NM1,N2D,N2DP1,NR,M,MEXP,NP,ANP,NPM1,ANPM1             00005180
      COMMON/BLEE1/XLP(10),XUP(10)                                      00005190
      COMMON/BLEE3/XL(10),XU(10),YL,YU                                  00005200
      COMMON/BL11/JSECT(10),DAP(10)                                     00005210
      COMMON/BL12/KEXRG(1000)                                           00005220
      COMMON/BL13/IREGN,MILAR                                           00005230
C                                                                       00005240
      DO 300 J=1,N                                                      00005250
      KREGN=(XP(J)-XLP(J))/DAP(J)                                       00005260
      JREGN(J)=KREGN+1                                                  00005270
      IF (KREGN.EQ.JSECT(J)) JREGN(J)=JSECT(J)                          00005280
  300 CONTINUE                                                          00005290
C                                                                       00005300
                                                                        00005310
```

```
      K=JREGN(N)                                                        00005320
      DO 500 JP=1,NM1                                                   00005330
      J=N-JP                                                            00005340
      K=(K-1)*JSECT(J)+JREGN(J)                                         00005350
  500 CONTINUE                                                          00005360
C                                                                       00005370
      KEXRG(K)=KEXRG(K)+1                                               00005380
      RETURN                                                            00005390
      END                                                               00005400
C                                                                       00005410
      SUBROUTINE COMPEN(KX,KY,ITRY,XLR,XUR,YMAX)                        00005420
      DIMENSION XP(10),Y(10),XLR(10),XUR(10),XLRG(10),XURG(10)          00005430
      COMMON N,AN,NM1,N2D,N2DP1,NR,M,MEXP,NP,ANP,NPM1,ANPM1             00005440
      COMMON/BLEE1/XLP(10),XUP(10)                                      00005450
      COMMON/BLEE3/XL(10),XU(10),YL,YU                                  00005460
      COMMON/BL11/JSECT(10),DAP(10)                                     00005470
      COMMON/BL12/KEXRG(1000)                                           00005480
      COMMON/BL13/IREGN,MIEXR                                           00005490
C                                                                       00005500
      ITRY=0                                                            00005510
      ICOMP=0                                                           00005520
      DO 500 K=1,IREGN                                                  00005530
      IF(KEXRG(K).GE.MIEXR) GO TO 500                                   00005540
      KA=K                                                              00005550
      DO 300 J=1,N                                                      00005560
      KB=MOD(KA,JSECT(J))                                               00005570
      IF(KB.EQ.0) KB=JSECT(J)                                           00005580
      KA=(KA-KB)/JSECT(J)+1                                             00005590
      AKB=KB                                                            00005600
      XLRG(J)=XLP(J)+DAP(J)*(AKB-1.)                                    00005610
      XURG(J)=XLP(J)+DAP(J)*AKB                                         00005620
      XP(J)=(XLRG(J)+XURG(J))/2.                                        00005630
  300 CONTINUE                                                          00005640
C                                                                       00005650
      MEXP=MEXP+1                                                       00005660
      ICOMP=ICOMP+1                                                     00005670
      WRITE (12,1400) ICOMP                                             00005680
      CALL SEIZE(XP)                                                    00005690
      CALL FSOPTI(XP,Y,IOUT)                                            00005700
      CALL RECORD(XP,Y,KX,KY)                                           00005710
      CALL WRITE(XP,Y,N,NR,IOUT)                                        00005720
      IF (IOUT.EQ.1) GO TO 500                                          00005730
      IF (Y(1).LE.YMAX) GO TO 500                                       00005740
      YMAX=Y(1)                                                         00005750
      DO 400 J=1,N                                                      00005760
      XLR(J)=XLRG(J)                                                    00005770
      XUR(J)=XURG(J)                                                    00005780
  400 CONTINUE                                                          00005790
      ITRY=1                                                            00005800
  500 CONTINUE                                                          00005810
C                                                                       00005820
 1400 FORMAT(/1X,I4,' TH COMPENSATE POINT'/)                            00005830
      RETURN                                                            00005840
      END                                                               00005850
C                                                                       00005860
      SUBROUTINE SIMUL(XS,Y,IIP1,KTER)                                  00005870
      DIMENSION XS(10),Y(20)                                            00005880
      COMMON /SUMRY/ TKACT,TKDET,TITSNK,TITOUT,TNEUAT,TNESUC,TNUEAL,    00005890
     1     TLOST,TWAST,RUN                                              00005900
```

DISTRIBUTION LIST

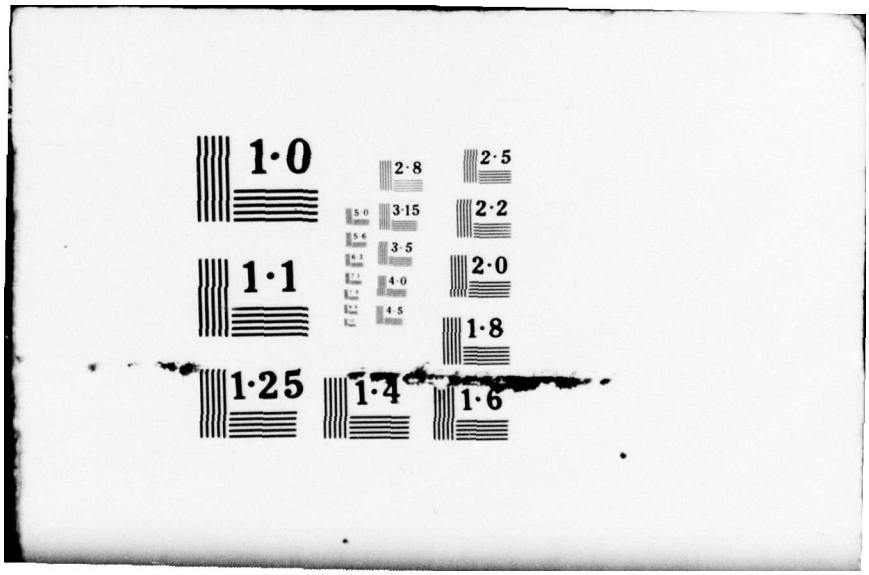
Name	Number of copies
Office of Naval Research Arlington, VA 22217 Code 431 Code 434 Code 436	 2 1 1
Defense Documentation Center Cameron Station Alexandria, VA 22314	12
ONR Branch Office 536 South Clark Street Chicago, Illinois 60605	1
Naval Research Laboratory Code 2627 Washington, DC 20375	6
ONR Resident Representative Purdue University Room 85 - Graduate House West Lafayette, Indiana 47907	1
Desmatics, Inc. PO Box 618 State College, PA 16801	1
George S. Fishman Curriculum in Operations Research and Systems Analysis Phillips Annex University of North Carolina at Chapel Hill Chapel Hill, NC 27514	1
Donald P. Gaver Naval Postgraduate School Operations Research Department Monterey, CA 93940	1
Toke Jayachandran Naval Postgraduate School Monterey, CA 93940	1
D. L. Iglehart Department of Operations Research Stanford University Stanford, CA 94305	1

AD-A062 389 NOTRE DAME UNIV IND DEPT OF AEROSPACE AND MECHANICA--ETC F/G 12/2
OPTIMIZATION OF MULTIPLE-RESPONSE SIMULATION MODELS.(U)
OCT 78 W E BILES N00014-76-C-1021
UNCLASSIFIED NL

2 OF 2
ADA
062389



END
DATE
FILMED
3 -79
DDC


1·0
1·1
1·25
1·4
1·6
1·8
2·0
2·2
2·5
2·8
3·15
3·5
4·0
4·5
5·0
5·6

DISTRIBUTION LIST

Name	Number of copies
Dr. Averill M. Law 1513 University Avenue Department of Industrial Engineering University of Wisconsin Madison, Wisconsin 53706	1
Dr. Peter A. W. Lewis Naval Postgraduate School Department of Administrative Sciences Monterey, CA 93940	1
Dr. Richard E. Nance Computer Science, Head Virginia Polytechnic Institute and State University Blacksburg, VA 24061	1
Dr. A. Alan B. Pritsker Department of Industrial Engineering Purdue University West Lafayette, Indiana 47907	1
J. William Schmidt Department of Industrial Engineering and Operations Research Virginia Polytechnic Institute and State University Blacksburg, VA 24061	1
Naval Surface Weapons Center Code K-30 (Kulp and Patton, and Miller) Dahlgren, VA 22448	3
David W. Taylor Naval Ship Research & Development Center Code 1806 (S. Goldstein) Bethesda, MD 20084	1
Naval Research Laboratory Code 5308 Washington, DC 20375	1
Naval Ocean Systems Center Code 16 San Diego, CA 92152	1