1978

SBOPT: A Simulation Based Optimization Algorithm

B. A. Dendrou

S. A. Dendrou

Elias N. Houstis
Purdue University, enh@cs.purdue.edu

Report Number:
78-274

SBOPT: A SIMULATION BASED OPTIMIZATION ALGORITHM

S. A. Dendrou*, B. A. Dendrou*, E. N. Houstis**

CSD-TR 274
July 1978

Abstract

A common problem encountered in engineering practice, management science and systems analysis is to find the values of input variables which optimize some function of system performance. When systems are too complicated to be described analytically, simulation is the appropriate tool for modeling purposes. Methods of optimization through simulation have recently become a steadily growing discipline.

The present report describes a constrained optimization algorithm for a class of problems where part of the constraints consist of functional relationships between variables, determined through simulation models. This algorithm (SBOPT) utilizes sequential successive simulation approximations until the intermediary optimal solution reaches acceptable tolerance levels of discrepancy. In an example of application, the algorithm proves to converge rapidly (5-10 iterations) requiring a limited number of actual simulation performances.

*School of Civil Engineering, Purdue University, W. Lafayette, IN 47907

**Department of Computer Science, Purdue University, W. Lafayette, IN 47907

TABLE OF CONTENTS

		Page
1.	Introduction	1
2.	Mathematical Formulation of a Class of Optimization Problems	3
3.	Successive Approximation Algorithm	7
4.	Simulation Approximation	10
5.	Convergence Considerations	13
6.	Numerical Experiment	15
List of References		24
Appendix A		25
Appendix B		31

SBOPT: A Simulation Based Optimization Algorithm

1. Introduction

Optimization problems in their general form can be described by a set of variables (decision type variables) whose values have to be determined so as to optimize (maximize or minimize) a set of functions called objective functions, while satisfying a given set of equalities and inequalities, termed as constraints. A general condition for avoiding degeneracy of the problem, is that the number of variables (dimension of the decision space) be larger than the number of constraints. This in general excludes the possibility of a uniquely defined set of decision variables or, worse yet, the emptiness of the feasible space, (incompatible set of constraints). Often, the set of decision variables can be partitioned into the subsets of independent and dependent variables. The latter are also called state variables, denoting that within the set of constraints of the optimization problem are included state equations describing the relationship between independent and state variables.

The present report describes a constrained optimization algorithm for the case where the state equations are not given analytically (in closed form), but rather consist of functional relationships, determined from the output response of simulation models. Most of the recent literature focuses on the problem of the direct optimization of a simulation response. Farrell, McCall and Russel, present a state-of-the-art review in a report to the Office of Naval Research, 1975 (2). All methods developed to date are approximate and iterative in nature. A few are adaptations of gradient optimization techniques. For example,

Billes, 1974 (3), proposes a gradient technique augmented by a regression scheme. Most techniques assume unimodality of the response surfaces. Smith and Storck, 1973 (4), have performed some research in this area. Others have explicitly taken into consideration the possibility of multimodality of the response surfaces, for example Smith, 1973 (5). Yet others have used direct search methods. The fundamental problem in this optique is the reduction of the size of the search region as reported by Luss and Jaakola, 1973 (7). Another interesting aspect reported in the literature is the treatment of the experimental optimization of statistical simulation, Elridge, 1974 (6). This approach leads to yet another important aspect of simulation-based optimization, namely that of discrete set optimization. Kleijner, Naylor and Seaks, 1972 (8), used multiple ranking procedures to analyze such situations.

However, in the perspective of the present paper, the simulation-response is considered only as an implicit constraint. The situation contemplated here is often encountered in engineering practice, especially in the steadily developing field of real-time optimal control. The mathematical formulation of this class of problems is given in the following section. The core of the proposed algorithm consists of sequential successive simulation approximations, as presented in sections 3 and 4. Considerations on the convergence and stability of the proposed algorithm and a numerical experiment conclude the report.

2. Mathematical Formulation of a Class of Optimization Problems

Denoting by (X_i) the set of independent variables and by (Y_j) the set of dependent or state variables, a class of simulation-based constrained optimization problems can be formulated as follows:

$$\underset{X_i}{\text{Opt}} \quad Z = Z(X_i, Y_j) \tag{2-1}$$

$$\text{s.t.} \quad Y_j = \Gamma_j(X_i) \qquad \forall\, j \tag{2-2}$$

$$g_\ell(X_i, Y_j) \lesseqgtr 0 \; ; \qquad \forall\, \ell \tag{2-3}$$

$$i=1,\ldots,n \; ; \; j=1,\ldots,m \; ; \; \ell=1,\ldots,k$$

Z in Eq. (2-1) is the objective function which is a function of both sets of variables (X_i) and (Y_j). It is to be optimized over both sets of variables (X_i) and (Y_j). Γ_j in Eq. (2-2) denotes the 'm' functional relationships between input and output response of the state-variable simulation. Finally g_ℓ, Eq. (2-3), denotes the set of 'k' equality and inequality constraints that define the feasible space for the decision variables. g_ℓ depend on both sets of variables (X_i) and (Y_j).

Were the functional relationships Γ_j, Eq. (2-2), to be known, a direct substitution into Eqs. (2-1) and (2-3) would transform the problem to a standard constrained optimization one, over the set of variables (X_i). A natural alternative then would be to find approximations to the Γ_j's. However, the simulation-response surfaces usually display high non-linearities. Satisfactory approximations could prove costly in such situations.

At the other end of the spectrum, relaxing the functional constraints (2-2) is equivalent to solving an alternate constrained optimization problem over both sets of variables (X_i) and (Y_j). Solving this latter problem has the advantage of detecting the vicinity of potentially useful

(X_i)-points. The proposed algorithm SBOPT takes advantage of both above situations by iteratively constraining the feasible space through successive approximations of the functional relationships $\Gamma_j(X_i)$, Eq. (2-2).

Schematically, the above discussion can be illustrated as shown in fig. 2.1. Denoting by Ω_X the feasible decision space as defined by Eq. (2-3), Fig. 2.1.a, its image Ω_X^{-1} in the state space can be determined through the mapping, Fig. 2.1.b.

$$\Gamma_j : X_i \rightarrow Y_j \tag{2-4}$$

Likewise, denoting by Ω_y the feasible state-space as defined by Eq. (2-3), Fig. 2.1.b, its image by the inverse mapping Γ_j^{-1}, denoted Ω_y^{-1} produces an additional restriction to the decision space, Fig. 2.1.a. The intersection of the above domains, define the overall feasible space, respectively in the decision and state space, Fig. 2.1.a,b. However, the mapping Γ_j is defined by the simulation model only in the direction $X_i \rightarrow Y_j$. Thus, the inverse image Ω_y^{-1} cannot be readily defined. This is in essence the difficulty of the simulation-based optimization situations.

On the other hand, optimizing Z in the X, Y-space, Fig. 2.1.c, over both sets of variables considered as independent (neglecting Eq. 2-2), produces the point (X_i^*, Y_j^*). The actual simulation-response corresponding to X_i^* being $Y_j(X_i^*)$, a natural measure of the discrepancy is given by the following norm:

$$||Y_j^* - Y_j(X_i^*)|| \tag{2-5}$$

The decrease of the above norm constitutes a natural criterion of convergence of an iterative scheme for the solution of the optimization problem. Moreover, the actual simulation-response points $Y_j(X_i^*)$ could be used to successively improve a simulation-response approximation. Such a scheme could have the potential advantage of reducing the number of

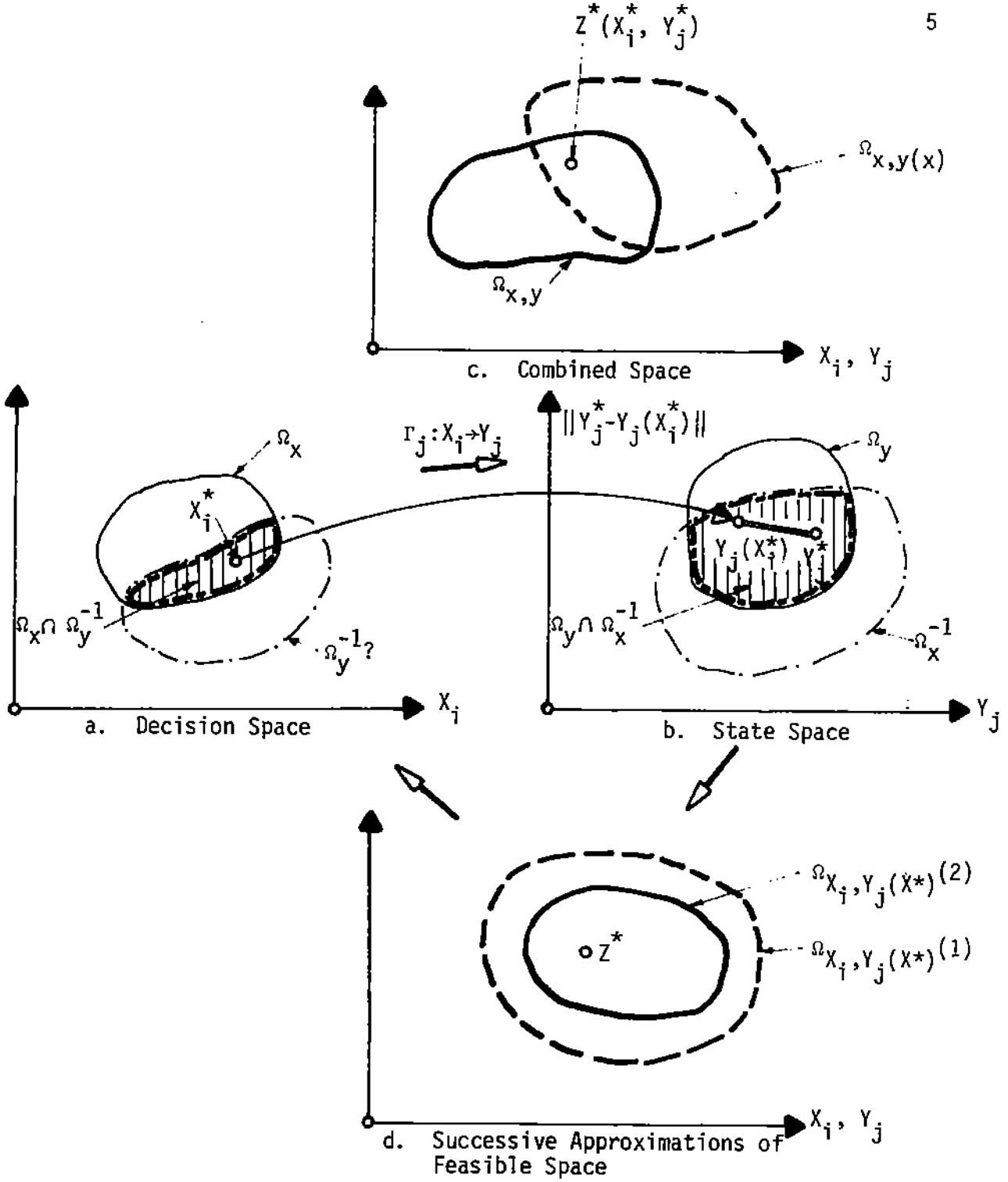

FIGURE 2.1 ILLUSTRATION OF OPTIMIZATION IN DECISION SPACE AND STATE SPACE

simulation runs, strictly to the number of iterations, if the algorithm is proven to converge. This advantage can be measured by comparing the above scheme with the 'brutal force' heuristic-search method, for example using Box's Complex algorithm, (1). This algorithm proceeds through a series of 'complexes' (sets of feasible points), formed by substituting successively the worst among the points of the 'complex', by an improving point. Each point evaluation requiring one simulation run, the total number of such runs would be equal to the total number of point evaluations. It usually runs in the prohibitive order of hundreds.

A pertinent question at this point, is that of the topologic relationship between the different feasible spaces illustrated in Figure 2.1. In fact it can be seen that the following relationship of inclusion holds:

$$\Omega_{x,y(x)} \subset \Omega_{x,y} \tag{2-6}$$

since the approximate space $\Omega_{x,y}$ disregards one of the conditions of $\Omega_{x,y(x)}$, namely Eq. (2-2). Thus, a temporary solution to the problem of iteration 'i' may prove to be infeasible. A condition for the convergence of the algorithm would be that the space $\Omega_{x,y}$ "shrinks" between successive iterations until it coincides with $\Omega_{x,y(x)}$ within acceptable tolerance limits.

3. Successive Approximation Algorithm

In the light of the above discussion, the following algorithm is proposed. It takes advantage of the comparatively easy initial determination of a suboptimal set of points (X_i^*) and (Y_j^*), by neglecting the simulation-response conditions, Eq. (2-2). Such a suboptimization can be performed by any standard Mathematical Programming technique, since all the remaining relations, Eqs. (2-1) and (2-3) are analytic. The simulation model can be called for a patterned set of points in the neighborhood of the above initial suboptimal point (X_i^*) and (Y_j^*). Alternatively, the simulation-response can be approximated locally, within the above suboptimal neighborhood, on the basis of the patterned simulation runs. Subsequently, a new optimization problem can be exactly solved, consisting of the initial problem augmented by the approximation to the simulation-response. The thus generated refined suboptimal point can be used in turn to refine the simulation response, and so on recursively until an acceptable convergence is achieved. Such a scheme is illustrated in the flowchart of Fig. 3.1.

In the scheme of the algorithm SBOPT, the standard constrained optimization method 'COMPLEX' of Box, (1), is used in the suboptimization for illustrative purposes. Between successive iterations, the following feature is introduced in addition to the scheme of the above discussion. All intermediary outcomes X* and Y* are saved along with the corresponding real simulation responses in arrays Y and YST (see Appendix A). All intermediary entries to array YST are sorted in an increasing order of the corresponding discrepancies, so that only the best among the known points are retained for the simulation approximation in the following iteration. However, in order to avoid cycling of the algorithm, new

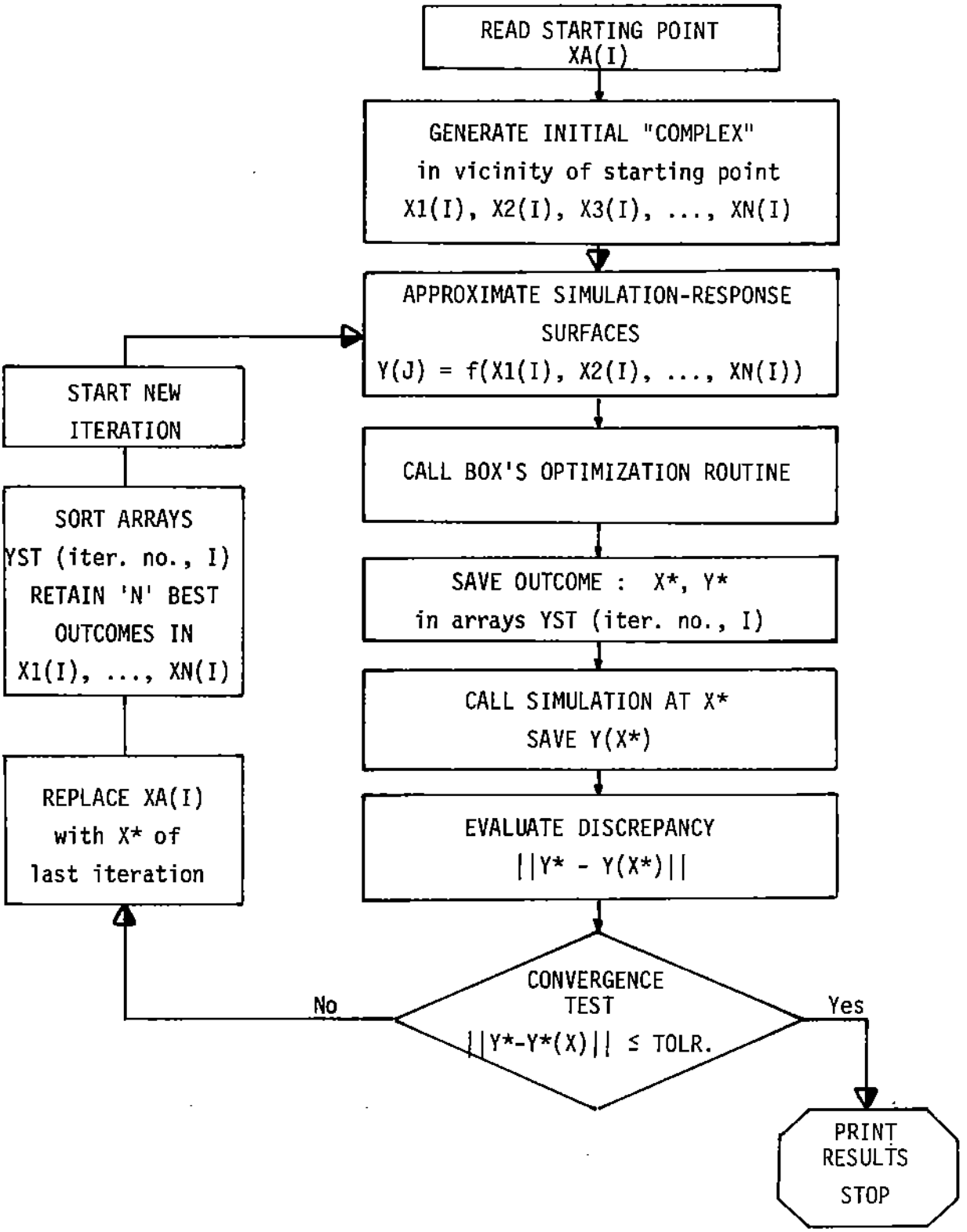

FIGURE 3.1 FLOWCHART OF ALGORITHM SBOPT

point repeats as worst point, the outcome of the latest iteration is always included in the set of retained points.

The above additional feature is believed to significantly accelerate the convergence of the algorithm. The convergence considerations are formally addressed in a later section. In the following section, the nature of and possible alternatives for the simulation approximation are discussed.

4. Simulation Approximation

Function approximation, commonly known as curve fitting techniques, have been developed in recent years, in an art in its own right. More and more sophisticated linear and non-linear regression techniques have been proposed, depicting trends and functional relationships between variables as observed by a wealth of experimental data. The situation described in the present report however, departs from the general case in that many successive approximations are foreseen, each based on a limited number of known points.

Since the operation is repetitive in nature in the proposed algorithm, a simple approximating scheme was sought, namely the one of linear regression. On the other hand, as was mentioned earlier, it appears logical to include in every successive approximation all additional information generated after each new iteration. However the above two arguments run against each other, as illustrated in Fig. 4.1. Indeed while a tangent to a curve (considered as a regression line) represents a very good approximation in the neighborhood of the point of tangence, a linear regression over a larger range of known points quickly becomes an unacceptable approximation. A trade-off is sought between the above two extreme possibilities, by adopting a straight-line approximation and accepting the burden of constraining the domain of its validity to an acceptable size.

Assuming that each simulation output Y_j is an independent function of all the input variables X_i, it can easily be seen that a linear regression requires a minimum of (n+1) known points, where n is the dimension of the decision space

$$R^n = \{X_i : \ i = 1, \ldots, n\} \qquad (4\text{-}1)$$

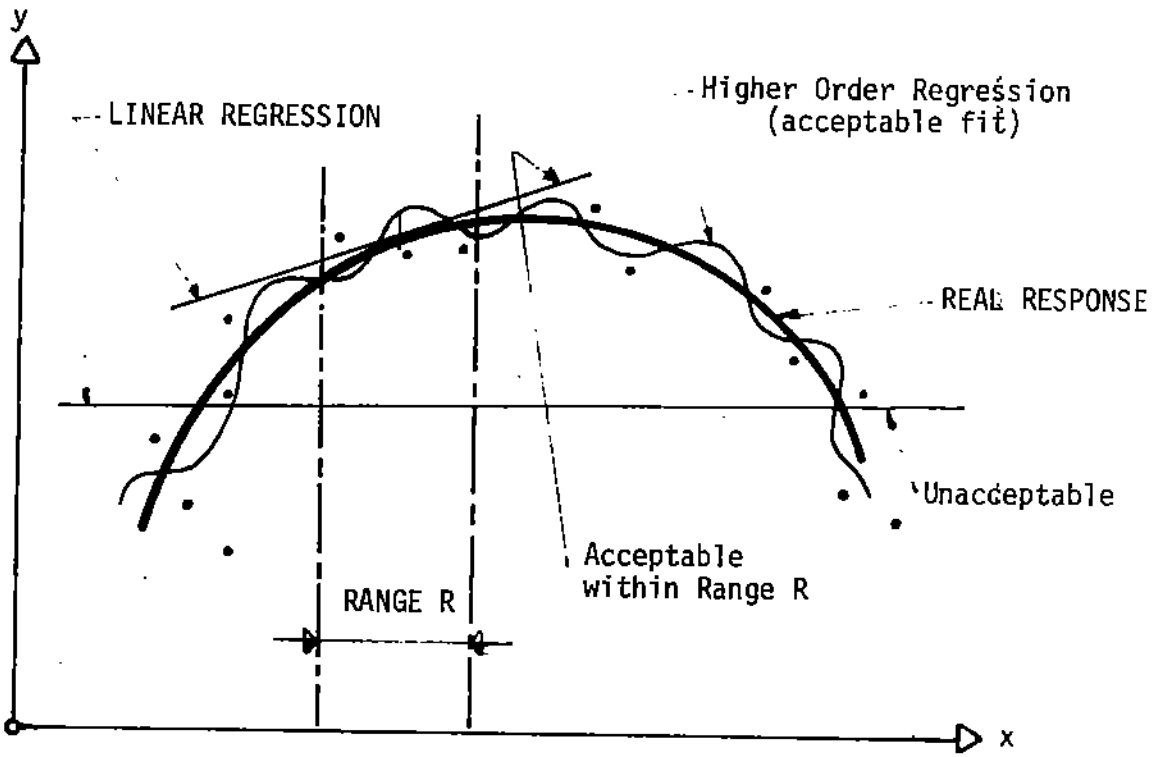

FIGURE 4.1 SIMULATION APPROXIMATION SCHEMES

Denoting by X_i^A, Y_j^A the coordinates of a known point A the linear approximation of Y_j can be written as

$$Y_j = Y_j^A + \alpha(X_1-X_1^A) + \beta(X_2-X_2^A) + \gamma(X_3-X_3^A) + \ldots + \eta(X_n-X_n^A) \qquad (4\text{-}2)$$

where α, β, γ, ..., η are the linear regression coefficients.

The determination of the 'n' regression coefficients requires the knowledge of an additional number of 'n' points. However care must be taken to avoid conditions of degeneracy. For example the (n+1) known points should be distinct, and better yet, uniformly distributed over the range of interest, so as to insure a small numerical error. The scheme of Eq. (4-2) is used in the algorithm SBOPT. The conditions of regularity are checked in the main-line routine, while the regression coefficients are determined in subroutine SIMAPRX. The listing of subroutine SIMAPRX is given in Appendix B. The following section deals with convergence considerations.

5. Convergence Considerations

The importance of the convergence considerations for an algorithm is very much a practical one. It constitutes a necessary condition for the algorithm to have any practical applicability. More important than the theoretical proof of convergence is the determination of the conditions required for a satisfactory rate of convergence. In this section, only a sketch of a proof is given, along with a list of necessary conditions for the existence and convergence of the solution.

The main feature of the iterative algorithm being the simulation approximation by regression hyperplanes, this algorithm reminds of the method of the secant encountered in classical numerical analysis. The obvious termination criterion is a tolerable discrepancy, at the solution point, between the approximated and the actual simulation response values. Moreover, the algorithm can be characterized as a fixed-point iterative scheme at the level of the simulation approximations, since the intermediary solution points are used to improve the approximations at the following step. A standard proof of convergence can be followed, for example, Wismer, 1971 (9).

However, the values of the independent variables X_i are determined at the intermediary steps of the algorithm such as to optimize the objective function while satisfying the set of constraints of the problem. Important assumptions will thus have to be made on the existence of such a solution. They include the requirement for convexity of the feasible space, and monotonicity or at least unimodality of the objective function within the feasible region.

While the above properties of convexity, and monotonicity or unimodality can be proved for the set of equations (2-1) and (2-3), they can

only be observed for the simulation response conditions (2-2). However, these properties define necessary conditions for the successive approximations to generate a stable solution. In particular, these approximations should not alter the convexity of the feasible space. In particular it is essential that the successive regressions do not degenerate. In a numerical example of the following section, an illustration of the instability resulting from degeneracy of the regression is shown.

Finally, an important consideration is that of the rate of convergence of the algorithm. Aside from the above mentioned properties and conditions, the rate of convergence also depends on the starting point. In the numerical experiment proposed in the following section convergence was achieved in 5-10 iterations.

6. Numerical Experiment

The example treated hereafter is inspired from an urban storm-drainage systems optimization scheme, reported in S. A. Dendrou, et al, (1978), (10). It is meant to provide a clear illustration of the implementation of the algorithm SBOPT.

An urban storm-drainage system can be defined, at the drainage basin level by three independent variables, denoted X_i, i=1,3, namely the detention storage capacity (volume), the contribution to a central treatment facility (rate) and a controlled overflow (rate). These measures are meant to alleviate the effects produced by storms and thundershowers. The effectiveness of the storm-drainage system can be monitored by such quantities as the quantity and quality of overflow, the frequency of street-flooding and others. These variables denoted by Y_j, j=1,...,8, are best generated by a digital model that processes a historical rainfall record and simulates the performance of the drainage system. The problem then consists of finding the least cost system configuration, while satisfying conditions on the acceptable pollution level and reliability of performance. The optimization problem equations are of the form

$$\underset{X_i}{\text{Min}} \quad C = C(X_i, Y_j) \tag{6-1}$$

$$\text{s.t.} \quad \Gamma_j : X_i \rightarrow Y_j \tag{6-2}$$

$$Y_j \leq t_{1j} \tag{6-3}$$

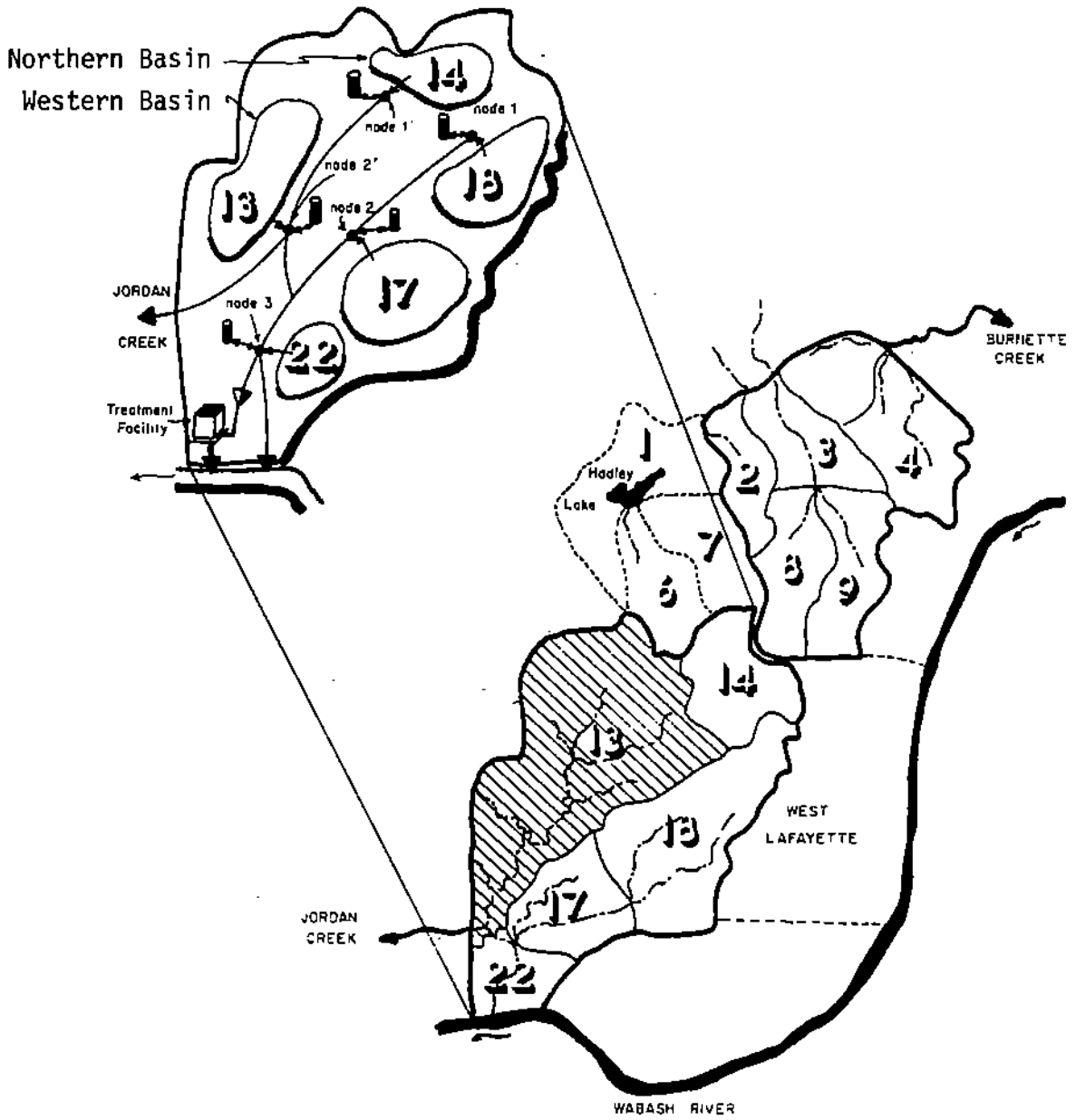

FIGURE 6.1 EXAMPLE OF A STORM-DRAINAGE SYSTEM

$$g_1(Y_j) \cdot \exp(g_2(Y_j)) \leq t_{2j} \tag{6-4}$$

$$[\theta_{ij}] \cdot [X_j] \leq t_{3i} \tag{6-5}$$

$$[\varepsilon_{ij}] \cdot [Y_j] \leq t_{4i} \tag{6-6}$$

$$h_1(X_i,Y_j) \cdot \exp(h_2(X_i,Y_j)) \leq t_5 \tag{6-7}$$

The simulation model used, Eq. (6-2), is the known hydrologic model STORM, (10).

The sets of runs are performed for two different drainage basins, the Northern and Western basins, Figure 6.1. First, the effect of instability by degeneracy of the regression scheme is illustrated in Figure 6.2. In the case of the upper figure, the regression scheme is allowed to consider redundant points among the set of regression points. Thus, the algorithm enters an unstable phase between iterations 5 and 15. When this possibility is eliminated, the algorithm proceeds to convergence in 5 steps.

In two other examples of application, respectively Figures 6.3 and 6.4, use is made of data from the Northern and Western basins, Figure 6.1. A slight oscillation is observed in both instances around the final value of convergence. This denotes that the boundary of the feasible region oscillates around a stable geometry for the successive simulation approximations (overshooting effect). Since the optimal point lies <u>on</u> the boundary, (see reference (10)), the above effect is repercuted on the values of the objective function.

The trajectory of the iterations for both the Northern and Western basins is shown in Table 6.1. In both cases it takes four iterations to converge to a solution within 5% of the acceptable response. This corresponds to eight (8) simulation calls altogether (four initial calls,

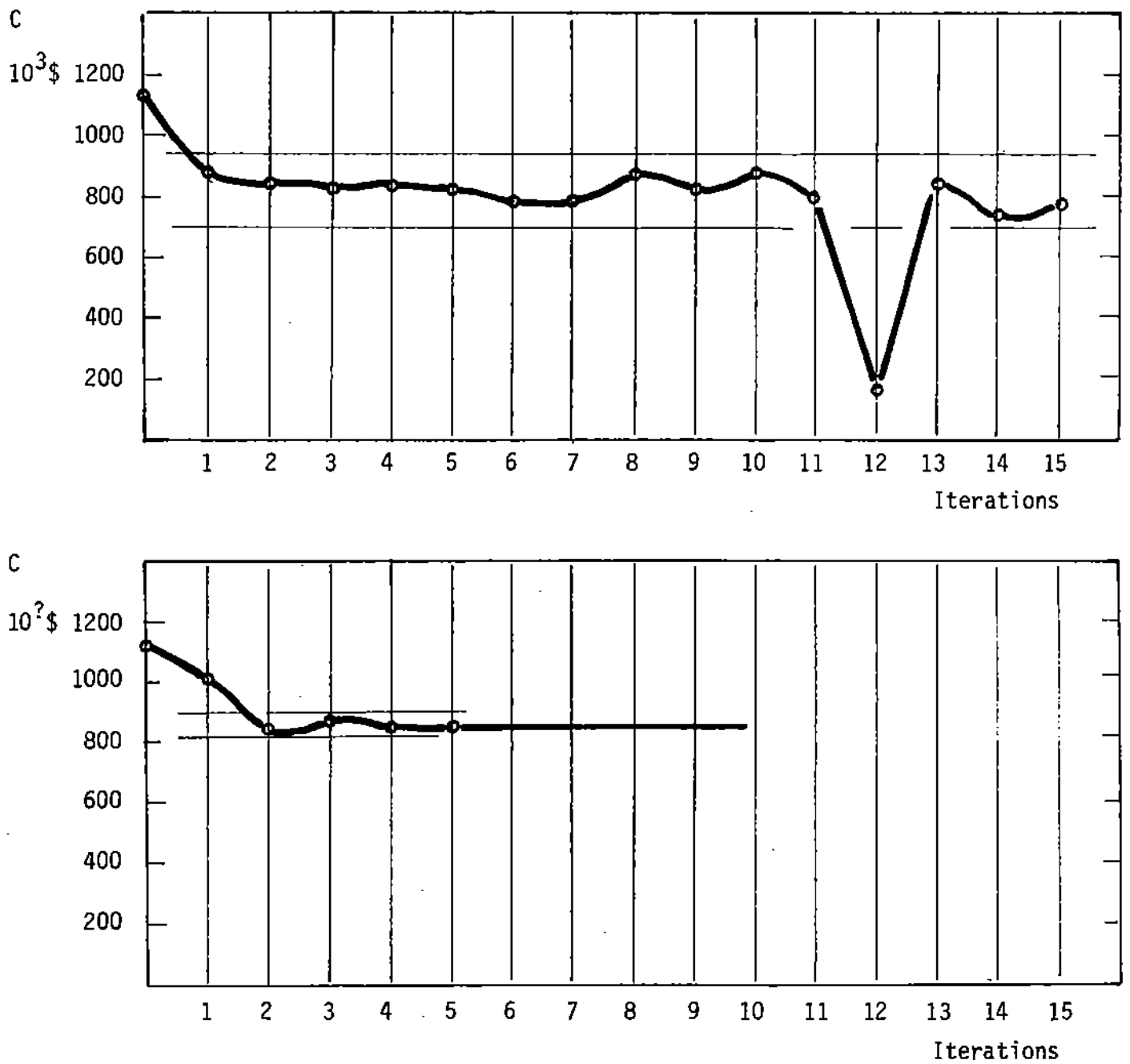

FIGURE 6.2 ILLUSTRATION OF INSTABILITY PRODUCED BY REGRESSION DEGENERATION

plus one per iteration). For comparison purposes, the "brutal force" approach used in (10), with the same starting point, requires 55 steps to convergence. Using an average of 10 simulation calls per step, this results in 550 simulation calls. The gain by the successive approximations method is thus seen to be appreciable.

The discrepancy between the approximated and the actual simulation response values in the first and last iterations for the case of the Northern basin is shown in Table 6.2. Often times the various simulation responses vary greatly in units and orders of magnitude. This is the reason for introducing the standard discrepancy as the r.m.s. of the actual discrepancies. Finally, in Table 6.3, the successive regression points and corresponding regression coefficients are shown for the case of the Northern basin.

In conclusion, more work has to be performed on the sensitivity of the model, as well as comparison with other techniques. One obvious direction of potential improvement is the approximation scheme for the response surfaces. For situations where the optimal point is expected to lie <u>on</u> the boundary of the feasible region it could be advantageous to approximate the simulation response using points on <u>either</u> side of the boundary and not in the feasible side alone.

TABLE 6.1 TRAJECTORIES OF ITERATIONS

	ITERATIONS	X1 (inches)	X2 (inches/hr)	X3 (inches/hr)	C $X10^3$\$	DISCREPANCY	EXACT SOLUTION	STEPS TO CONVERGENCE (re.10)
NORTHERN BASSIN	0	0.250	0.040	0.010	629.		X1 = .237	
	1	0.224	0.001	0.027	310.	4553.	X2 = .000	55
	2	0.222	0.101	0.022	313.	253.	X3 = .010	
	3	0.227	0.001	0.010	301.	2304.		
	4	0.225	0.001	0.020	305.	69.	C = 328 x 10^3 \$	
WESTERN BASSIN	0	0.250	0.040	0.010	1057.			
	1	0.232	0.001	0.005	619.	76.	X1 = .110	
	2	0.222	0.001	0.005	642.	64.	X2 = .001	56
	3	0.224	0.001	0.005	638.	68.	X3 = .009	
	4	0.190	0.001	0.007	585.	43.		
							C = 509 x 10^3 \$	

TABLE 6.2 DISCREPANCY IN SIMULATION RESPONSE

SIMULATION RESPONSE	FIRST ITERATION			LAST ITERATION		
	Y^*	$Y(X^*)$	$Y^*-Y(X^*)$	Y^*	$Y(X^*)$	$Y^*-Y(X^*)$
Y_1	.614	.963	- .3483	.955	1.089	- .133
Y_2	.033	.103	- .0698	.102	.117	- .015
Y_3	.971	2.583	- 1.6118	2.582	2.631	- .048
Y_4	319.079	4773.141	-4454.0622	4788.8	4740.5	48.3
Y_5	68.055	578.097	- 510.0423	580.00	551.69	28.3
Y_6	44.819	819.579	- 774.7601	822.66	784.02	38.6
Y_7	2.665	186.270	- 183.6045	186.83	176.78	10.1
Y_8	1.546	50.256	- 48.7098	50.42	47.95	2.4

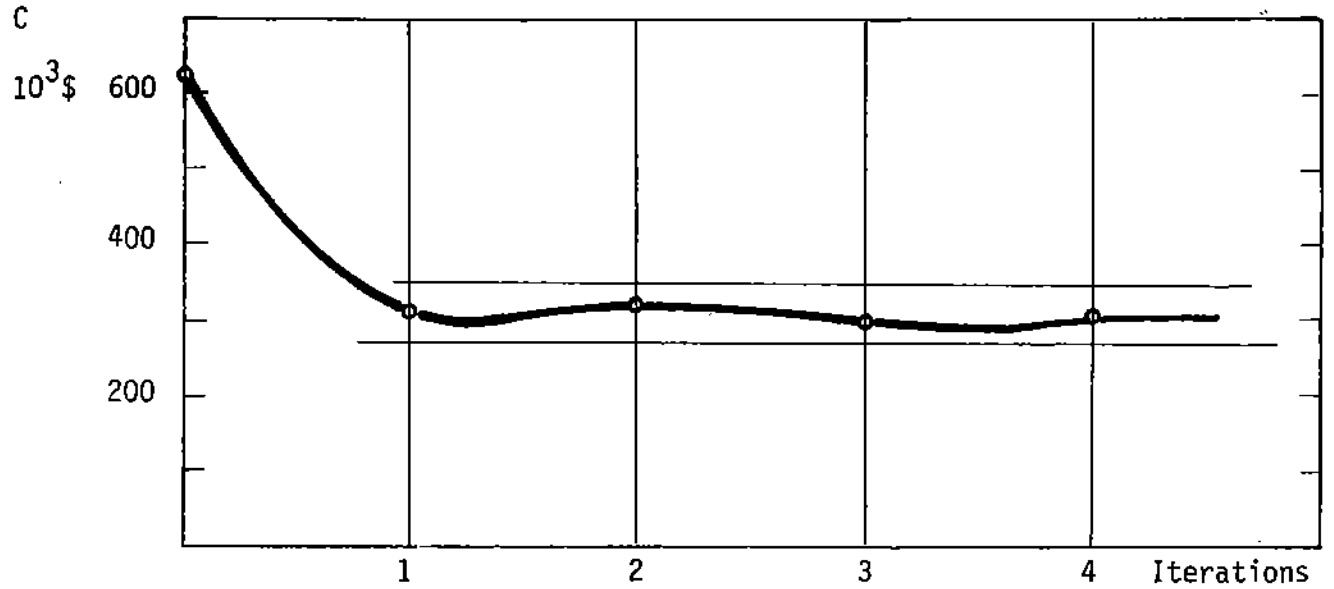

FIGURE 6.3 CONVERGENCE TRAJECTORY - NORTHERN BASIN

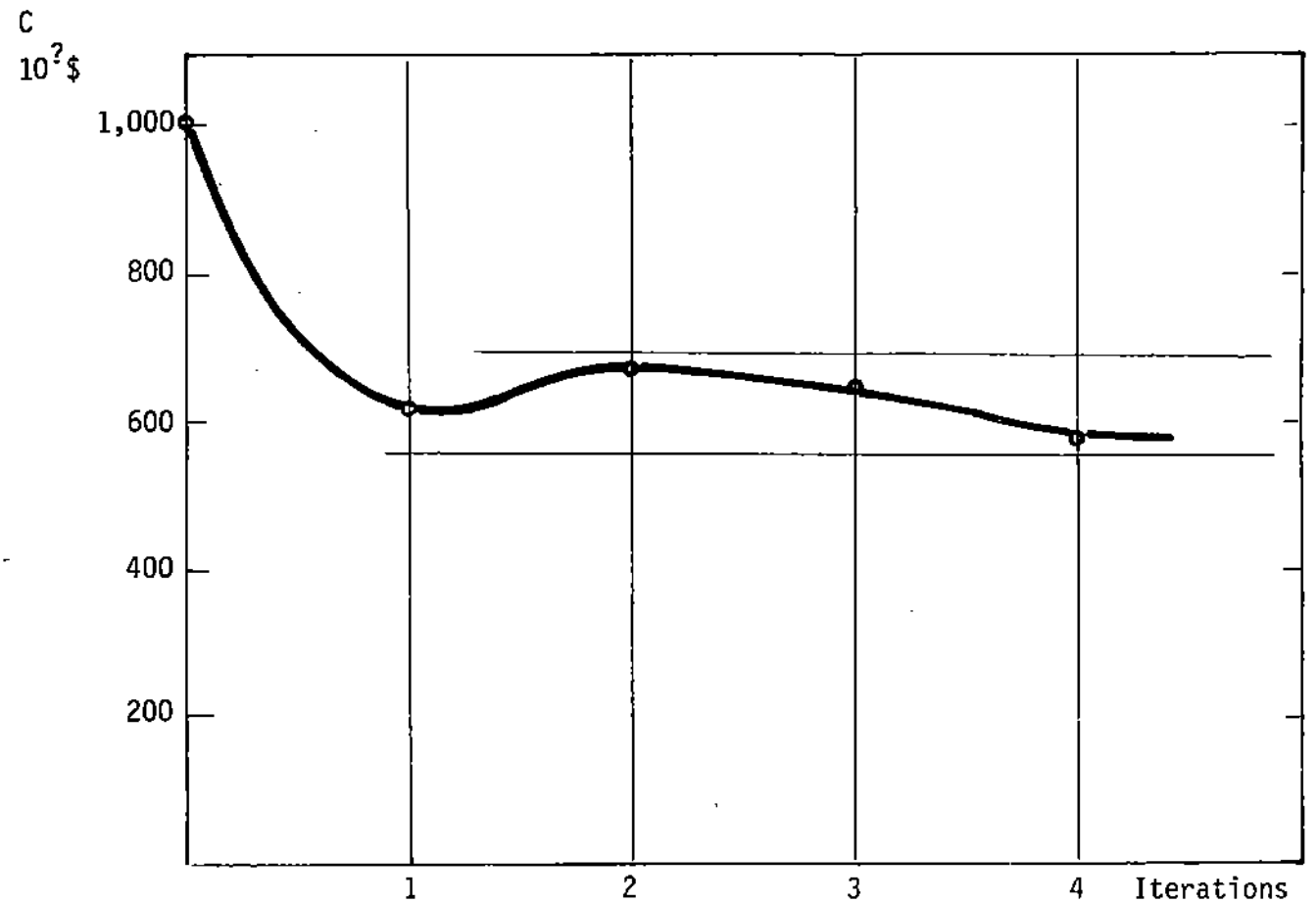

FIGURE 6.4 CONVERGENCE TRAJECTORY - WESTERN BASIN

TABLE 6.3 SUCCESSIVE SIMULATION APPROXIMATIONS

	VARIABLES	REGRESSION POINTS				REGRESSION COEFFICIENTS		
		$X1_i$	$X2_i$	$X3_i$	$X4_i$	CALPH	CBETA	CGAMMA
	X_1	.250	.400	.325	.287			
	X_2	.040	.048	.025	.055			
	X_3	.010	.012	.010	.008			
ITERATION 0	Y_1	.477	.093	.324	.293	- 3.1669	6.9458	19.4282
	Y_2	.046	.002	.031	.023	- .3384	.7667	.5211
	Y_3	.545	.148	.381	.324	- 3.8281	11.3120	45.8211
	Y_4	239.500	103.200	197.750	132.425	-1351.4863	5458.5424	15741.9355
	Y_5	31.500	18.000	27.000	18.450	- 190.9247	934.6237	4064.5161
	Y_6	31.500	14.400	26.750	18.037	- 181.2222	744.4444	2250.0000
	Y_7	9.000	3.000	7.750	5.100	- 44.5305	129.3190	- 145.1613
	Y_8	2.500	1.200	2.250	1.550	- 10.8053	38.4707	16.1290
	X_1	.224	.400	.325	.287			
	X_2	.001	.048	.025	.055			
	X_3	.027	.012	.010	.008			
ITERATION 1	Y_1	.963	.093	.324	.293	- 6.2668	6.0202	3.0880
	Y_2	.103	.002	.031	.023	- .7406	.7770	.4758
	Y_3	2.583	.148	.381	.324	- 18.9628	21.0244	4.9472
	Y_4	4773.141	103.200	197.750	132.425	7488.9253	43473.9354	6019.9884
	Y_5	578.097	18.00	27.000	18.450	-4507.6489	5255.7303	718.1408
	Y_6	819.579	14.400	26.750	18.037	-6468.4881	7488.8467	959.7831
	Y_7	186.270	3.000	7.750	5.100	-1467.7455	1695.9199	245.6747
	Y_8	50.256	1.200	2.250	1.550	- 393.6508	456.3060	63.9253
	X_1	.222	.400	.325	.287			
	X_2	.001	.048	.025	.055			
	X_3	.022	.012	.010	.008			
ITERATION 2	Y_1	1.089	.093	.324	.293	- 7.2179	7.0769	4.4592
	Y_2	.117	.002	.031	.023	- .8485	.8961	.6844
	Y_3	2.631	.148	.381	.324	- 19.1505	21.1447	7.0419
	Y_4	4540.543	103.200	197.750	132.425	5238.6595	40715.1759	8278.7999
	Y_5	551.694	18.000	27.000	18.450	-4249.4982	4938.1576	989.6060
	Y_6	784.019	14.400	26.750	18.037	-6116.1131	7054.2635	1323.9932
	Y_7	176.779	3.000	7.750	5.100	-1376.6489	1584,4687	337.5366
	Y_8	47.950	1.200	2.250	1.550	- 371.1159	428.5889	88.0955

List of References

1. Box, M. J., "A New Method of Constrained Optimization and a Comparison with Other Methods," Computer J., 8, 1965, pp. 42-52.

2. Farrell, W., McCall, C., and Russell, E., "Optimization Techniques for Computerized Simulation Models," Report to Office of Naval Research, NTIS #AD-A011 844/8G1, 1975.

3. Billes, W. E., "A Gradient-Regression Search Procedure for Simulation Experimentation," Proceedings of the 1974 Winter Simulation Conference, Vol. 2, pp. 491-497.

4. Smith, D. E., and Storck, C. E., "Research on an Optimizer Computer Program for Use in Simulation Studies," NTIS, (AD766-089/7WC), August 1973.

5. Smith, D. E., "An Empirical Investigation of Optimum-Seeking in the Computer Simulation Situation," Operations Research, Vol. 21, No. 2, March-April 1973, pp. 475-497.

6. Eldridge, D. L., "Experimental Optimization of Statistical Simulation," Proceedings of the 1974 Winter Simulation Conference, pp. 503-510.

7. Luss, R., and Jaakola, T. H. I., "Optimization by Direct Search and Systematic Reduction of the Size of the Search Region," American Institute of Chemical Engineers Journal, Vol. 19, No. 4, July 1973, pp. 760-766.

8. Kleijner, J. P. C., Naylor, T. H., and Seaks, T. G., "The Use of Multiple Ranking Procedures to Analyze Simulations of Management Systems: A Tutorial," Management Science, Vol. 18, February 1972, pp. B245-B257.

9. Wismer, D. A., Distributed Multilevel Systems, Chapter 6, pp. 252-253, "Optimization Methods for Large-Scale Systems ... with Applications," D. A. Wismer, Editor, McGraw-Hill, New York, 1971.

10. Dendrou, S. A., Talavage, J. J., and Delleur, J. W., "Urban Storm-Drainage Systems Planning," Technical Report No. 101, Purdue University Water Resources Research Center, May 1978.

APPENDIX A

```
      SUBROUTINE SBOPT (N,M,K,ITMAX,ALPHA,BETA,GAMMA,DELTA,NIT,IEV2,ND,N   A   10
     1KOUNT,IPRINT)                                                       A   20
C                                                                         A   30
C     ***********************************************************       A   40
C                                                                         A   50
C     SIMULATION  BASED  OPTIMIZATION  SCHEME                             A   60
C                                                                         A   70
C     -----------------------------------------------------------       A   80
C     FOR DETAILED  DEFINITION  OF  VARIABLES  SEE REF. 10/               A   90
C                                                                         A  100
C     TRANSFERED THROUGH COMMON-BLOCK FROM MAIN-LINE ROUTINE =            A  110
C                                                                         A  120
C       S,T,AOR,AREA,LSUB,ICOORD,TF,VF,OV,TO,OP                           A  130
C                                                                         A  140
C     DEFINITION  OF  VARIABLES                                           A  150
C     -------------------------                                           A  160
C                                                                         A  170
C      G. LE. (X) . LE. H.       THRESHOLD  CONSTRAINTS                  A  180
C     X1,X2,X3,X4,  XA=X1,   = GIVEN  REGRESSION  POINTS                  A  190
C     CALPH,CBETA,CGAMMA,    = REGRESSION  COEFFICIENTS                   A  200
C     Y (..,..)              = ARRAY OF  INTERMEDIARY RESULTS             A  210
C                              ( UP TO 40 ITERATIONS )                    A  220
C     YST(..,..)             = ACTUAL SIMULATION RESPONSE                 A  230
C     DYST(..,..)            = DISCRIPENCY                                A  240
C     DNST(..,..)            = STD. DISCRIPENCY                           A  250
C     DSORT(.,.)             = SORTED ARRAY OF DNST                       A  260
C                                                                         A  270
C                                                                         A  280
C     -----------------------------------------------------------       A  290
C                                                                         A  300
C                                                                         A  310
      COMMON /SAVE/ Y(40,18),YST(40,8),DYST(40,8),DNST(40),ITENO(40)      A  320
      COMMON /ONE/ X(5,76),R(5,3),F(5),G(76),H(76),XC(76),NOCRD,ITER1(15  A  330
     1)                                                                   A  340
      COMMON /CNCTR/ CTHR(5)                                              A  350
      COMMON /APRXS/ XA(15),CALPH(8),CBETA(8),CGAMA(8)                    A  360
      DIMENSION DSORT(40)                                                 A  370
      COMMON /FORPP/ X1(15),X2(15),X3(15),X4(15),NPOINTS                  A  380
      INTEGER GAMMA                                                       A  390
C                                                                         A  400
C     INITIALIZATION                                                      A  410
C     ==============                                                      A  420
C                                                                         A  430
      NPOINTS=1                                                           A  440
      NIT=1                                                               A  450
      DO 101 L1J=1,M                                                      A  460
  101 WRITE (NO,149) L1J,G(L1J),L1J,H(L1J)                                A  470
      DO 102 I1L=1,M                                                      A  480
         IF (G(I1L).EQ.H(I1L)) WRITE (NOCRD,150) G(I1L),I1L,LSUB          A  490
  102 CONTINUE                                                            A  500
C                                                                         A  510
C     TRANSFER OF POINT  X1                                               A  520
C                                                                         A  530
C                                                                         A  540
      X1(1)=S(LSUB)                                                       A  550
      X1(2)=T(LSUB)                                                       A  560
      X1(3)=AOR(LSUB)                                                     A  570
      X1(4)=TF(LSUB)                                                      A  580
      X1(5)=VF(LSUB)                                                      A  590
      X1(6)=TO(LSUB)                                                      A  600
      DO 103 IP=1,5                                                       A  610
  103 X1(6+IP)=OP(IP,LSUB)                                                A  620
C                                                                         A  630
C     GENERATION OF POINTS X2,X3,X4                                       A  640
C                                                                         A  650
C                                                                         A  660
      R(2,1)=1.                                                           A  670
      R(2,2)=1.                                                           A  680
      R(2,3)=-1.                                                          A  690
      R(3,1)=0.                                                           A  700
      R(3,2)=-1.                                                          A  710
      R(3,3)=0.                                                           A  720
```

```
      R(4,1)=-1.                                                        A  730
      R(4,2)=1.                                                         A  740
      R(4,3)=1.                                                         A  750
      R(5,1)=-1.                                                        A  760
      R(5,2)=0.                                                         A  770
      R(5,3)=0.                                                         A  780
      DDS=0.15                                                          A  790
      DDT=0.015                                                         A  800
      DDAR=0.002                                                        A  810
      RS=S(LSUB)+DDS                                                    A  820
      RT=T(LSUB)+0.5*DDT                                                A  830
      RCT=RT+AOR(LSUB)+DDAR                                             A  840
C                                                                       A  850
C     SUBROUTINE KSTORM IS A SIMULATION ROUTINE                         A  860
C                                                                       A  870
      CALL KSTORM (RS,RT,RCT)                                           A  880
      X2(1)=RS                                                          A  890
      X2(2)=RT                                                          A  900
      X2(3)=RCT-RT                                                      A  910
      X2(4)=TF(LSUB)                                                    A  920
      X2(5)=VF(LSUB)                                                    A  930
      X2(6)=TO(LSUB)                                                    A  940
      DO 104 IDI=1,5                                                    A  950
  104 X2(6+IDI)=(OP(IDI,LSUB))                                          A  960
      RS=S(LSUB)+.5*DDS                                                 A  970
      RT=T(LSUB)-DDT                                                    A  980
      RCT=RT+AOR(LSUB)                                                  A  990
      IF (RT.LT.0.) RT=0.                                               A 1000
      CALL KSTORM (RS,RT,RCT)                                           A 1010
      X3(1)=RS                                                          A 1020
      X3(2)=RT                                                          A 1030
      X3(3)=RCT-RT                                                      A 1040
      X3(4)=TF(LSUB)                                                    A 1050
      X3(5)=VF(LSUB)                                                    A 1060
      X3(6)=TO(LSUB)                                                    A 1070
      DO 105 IDI=1,5                                                    A 1080
  105 X3(6+IDI)=(OP(IDI,LSUB))                                          A 1090
      RS=S(LSUB)+DDS/4.                                                 A 1100
      RT=T(LSUB)+DDT                                                    A 1110
      RCT=RT+AOR(LSUB)-DDAR                                             A 1120
      IF (RCT.LT.0.) RCT=0.                                             A 1130
      CALL KSTORM (RS,RT,RCT)                                           A 1140
      X4(1)=RS                                                          A 1150
      X4(2)=RT                                                          A 1160
      X4(3)=RCT-RT                                                      A 1170
      X4(4)=TF(LSUB)                                                    A 1180
      X4(5)=VF(LSUB)                                                    A 1190
      X4(6)=TO(LSUB)                                                    A 1200
      DO 106 IDI=1,5                                                    A 1210
  106 X4(6+IDI)=(OP(IDI,LSUB))                                          A 1220
C                                                                       A 1230
C     START  MAIN  LOOP  OF  SBOPT                                      A 1240
C                                                                       A 1250
C     ********************                                            A 1260
C                                                                       A 1270
C     EVALUATION  OF  REGRESSION  COEFFICIENTS                          A 1280
C                                                                       A 1290
C                                                                       A 1300
  107 CALL SIMAPRX                                                      A 1310
C                                                                       A 1320
      X(1,1)=X1(1)                                                      A 1330
      X(1,2)=X1(2)                                                      A 1340
      X(1,3)=X1(3)                                                      A 1350
C                                                                       A 1360
C      *COMPLEX*  PACKAGE OF BOX, REF,(1), CONTAINS                     A 1370
C                 SUBROUTINES= CONSX,CHECHK,CENTR,FUNC,CONST            A 1380
C                                                                       A 1390
C                                                                       A 1400
C     CHECK  FEASIBILITY  OF    STARTING  POINT  X1(.)                  A 1410
C                                                                       A 1420
      CALL CHECKK (N,M,K,NKOUNT,1,1,NO,DELTA,1,BETA)                    A 1430
      CALL FUNC (N,M,K,1,NO,NKOUNT)                                     A 1440
```

```
      DO 109 II=2,K                                                          A 1450
         DO 108 J=1,N                                                        A 1460
  108    X(II,J)=0.0                                                         A 1470
  109 CONTINUE                                                             A 1480
C                                                                          A 1490
C     PERTURBATION  GENERATED  ≠COMPLEX≠  REF.(1)                          A 1500
C                                                                          A 1510
      DO 111 II=2,K                                                        A 1520
         I=II                                                              A 1530
         K1=II-1                                                           A 1540
         DO 110 J=1,N                                                      A 1550
            X(II,J)=X(1,J)+R(II,J)*DELTA*.93                               A 1560
  110    CONTINUE                                                          A 1570
         CALL CHECKK (N,M,K,NKOUNT,I,KODE,NO,DELTA,K1,BETA)                A 1580
         CALL FUNC (N,M,K,I,NO,NKOUNT)                                     A 1590
  111 CONTINUE                                                             A 1600
C                                                                          A 1610
C      CALL  MAIN-LINE  ≠COMPLEX≠- ROUTINE                                 A 1620
C                                                                          A 1630
C                                                                          A 1640
      CALL CONSX (N,M,K,ITMAX,ALPHA,BETA,GAMMA,DELTA,IT,IEV2,NO,NKOUNT,I   A 1650
     1PRINT)                                                               A 1660
C                                                                          A 1670
C     SAVE  SUBOPTIMAL  POINT                                              A 1680
C                                                                          A 1690
C                                                                          A 1700
      NAB=11+NTOT                                                          A 1710
      Y(NIT,4)=TF(LSUB)                                                    A 1720
      Y(NIT,5)=VF(LSUB)                                                    A 1730
      Y(NIT,6)=TO(LSUB)                                                    A 1740
      DO 112 IQ=1,5                                                        A 1750
  112 Y(NIT,6+IQ)=OP(IQ,LSUB)                                              A 1760
      DO 113 IW=1,N                                                        A 1770
         Y(NIT,IW)=X(IEV2,IW)                                              A 1780
  113 X1(IW)=X(IEV2,IW)                                                    A 1790
C                                                                          A 1800
C                                                                          A 1810
      RS=X1(1)                                                             A 1820
      RT=X1(2)                                                             A 1830
      RCT=RT+X1(3)                                                         A 1840
C                                                                          A 1850
C      CALL ACTUAL  SIMULATION  RESPONCE                                   A 1860
C                                                                          A 1870
C                                                                          A 1880
      CALL KSTORM (RS,RT,RCT)                                              A 1890
C                                                                          A 1900
C                                                                          A 1910
      YST(NIT,1)=TF(LSUB)                                                  A 1920
      YST(NIT,2)=VF(LSUB)                                                  A 1930
      YST(NIT,3)=TO(LSUB)                                                  A 1940
      DO 114 LI=1,5                                                        A 1950
  114 YST(NIT,3+LI)=OP(LI,LSUB)                                            A 1960
      INFL=0                                                               A 1970
C                                                                          A 1980
C      EVALUATE DISCRIPENCY                                                A 1990
C                                                                          A 2000
C                                                                          A 2010
      DO 115 KK=1,8                                                        A 2020
         DYST(NIT,KK)=Y(NIT,3+KK)-YST(NIT,KK)                              A 2030
         ABA=ABS(DYST(NIT,KK))                                             A 2040
         IF (YST(NIT,KK).GE.1.) ABET=BETA*YST(NIT,KK)                      A 2050
C                                                                          A 2060
C      CHECK  CONVERGENCE                                                  A 2070
C                                                                          A 2080
         IF (YST(NIT,KK).LE.1.) ABET=BETA                                  A 2090
         IF (ABA.GE.ABET) INFL=10                                          A 2100
  115 CONTINUE                                                             A 2110
      DNST(NIT)=0.                                                         A 2120
      DO 116 KI=1,8                                                        A 2130
  116 DNST(NIT)=DNST(NIT)+DYST(NIT,KI)**2                                  A 2140
      DNST(NIT)=SQRT(DNST(NIT))                                            A 2150
C                                                                          A 2160
```

```
C
C       PRINT-OUT  OF  INTERMEDIARY  RESULTS                                A 2170
C                                                                           A 2180
C                                                                           A 2190
      WRITE (60,151) NIT,DNST(NIT)                                          A 2200
      WRITE (60,152) (Y(NIT,I),I=1,3)                                       A 2210
      WRITE (60,153)                                                        A 2220
      DO 117 I=1,8                                                          A 2230
  117 WRITE (60,154) I,Y(NIT,3+I),YST(NIT,I),DYST(NIT,I)                    A 2240
      WRITE (60,155) (I,QT(I),I=1,NTOT)                                     A 2250
      NA=NIT-1                                                              A 2260
      IF (NA.EQ.0) GO TO 118                                                A 2270
      WRITE (60,156) (ITENO(I),I=1,NA)                                      A 2280
      WRITE (60,157) (DSORT(I),I=1,NA)                                      A 2290
  118 CONTINUE                                                              A 2300
C                                                                           A 2310
C                                                                           A 2320
C                                                                           A 2330
                                                                            A 2340
C                                                                           A 2350
      IF (INFL.EQ.10) GO TO 119                                             A 2360
      RETURN                                                                A 2370
C                                                                           A 2380
C     START  NEW  ITERATION                                                 A 2390
C                                                                           A 2400
  119 NIT=NIT+1                                                             A 2410
      IF (NIT.GT.ITMAX) GO TO 148                                           A 2420
      NNIT=NIT-1                                                            A 2430
      DO 120 IK=1,8                                                         A 2440
         X1(3+IK)=(YST(NNIT,IK))                                            A 2450
  120 CONTINUE                                                              A 2460
      IF (NIT.GT.5) GO TO 121                                               A 2470
C                                                                           A 2480
C                                                                           A 2490
C                                                                           A 2500
C                                                                           A 2510
C                                                                           A 2520
C                                                                           A 2530
C                                                                           A 2540
C                                                                           A 2550
C                                                                           A 2560
C                                                                           A 2570
C                                                                           A 2580
C                                                                           A 2590
C                                                                           A 2600
C                                                                           A 2610
C                                                                           A 2620
C                                                                           A 2630
C                                                                           A 2640
C                                                                           A 2650
C                                                                           A 2660
C                                                                           A 2670
C                                                                           A 2680
      GO TO 107                                                             A 2690
  121 CONTINUE                                                              A 2700
C                                                                           A 2710
C     SORTING  PROCEDURE (DECREASING ORDER)                                 A 2720
C                                                                           A 2730
      NONE=NNIT-1                                                           A 2740
      DO 122 IKE=1,NONE                                                     A 2750
C                                                                           A 2760
C     PRINT,≠DNST(I)≠,IKE,DNST(IKE)                                         A 2770
C                                                                           A 2780
  122 DSORT(IKE)=DNST(IKE)                                                  A 2790
      MA1=NONE                                                              A 2800
  123 MA1=MA1/2                                                             A 2810
      IF (MA1.EQ.0) GO TO 127                                               A 2820
      KA1=NONE-MA1                                                          A 2830
      JA1=1                                                                 A 2840
  124 IA1=JA1                                                               A 2850
  125 IMA1=IA1+MA1                                                          A 2860
      IF (DSORT(IA1).GE.DSORT(IMA1)) GO TO 126                              A 2870
                                                                            A 2880
```

```
      TEMPA1=DSORT(IA1)                                                  A 2890
      DSORT(IA1)=DSORT(IMA1)                                             A 2900
      DSORT(IMA1)=TEMPA1                                                 A 2910
      IA1=IA1-MA1                                                        A 2920
      IF (IA1.GE.1) GO TO 125                                            A 2930
  126 JA1=JA1+1                                                          A 2940
      IF (JA1.GT.KA1) GO TO 123                                          A 2950
      GO TO 124                                                          A 2960
  127 CONTINUE                                                           A 2970
      DO 129 L1=1,NONE                                                   A 2980
         DO 128 JL1=1,NONE                                               A 2990
C                                                                        A 3000
C     PRINT,≠DNST(I),DSORT(I)≠,JL1,DNST(JL1),DSORT(L1)                   A 3010
C                                                                        A 3020
            IF (DNST(JL1).EQ.DSORT(L1)) GO TO 129                        A 3030
  128    CONTINUE                                                        A 3040
  129 ITEND(L1)=JL1                                                      A 3050
C                                                                        A 3060
C     END  OF  SORTING - DETERMINE  NEW  POINTS  X1,X2,X3,X4             A 3070
C                                                                        A 3080
      KK1=ITEND(NONE)                                                    A 3090
      KK2=ITEND(NONE-1)                                                  A 3100
      KK3=ITEND(NONE-2)                                                  A 3110
      DO 130 L=1,3                                                       A 3120
         X2(L)=(Y(KK1,L))                                                A 3130
         X3(L)=(Y(KK2,L))                                                A 3140
         X4(L)=(Y(KK3,L))                                                A 3150
  130 CONTINUE                                                           A 3160
      DO 131 LIK=1,8                                                     A 3170
         X2(3+LIK)=(YST(KK1,LIK))                                        A 3180
         X3(3+LIK)=(YST(KK2,LIK))                                        A 3190
         X4(3+LIK)=(YST(KK3,LIK))                                        A 3200
  131 CONTINUE                                                           A 3210
      NKI=3                                                              A 3220
  132 L21=0                                                              A 3230
      L31=0                                                              A 3240
      L41=0                                                              A 3250
      L32=0                                                              A 3260
      L43=0                                                              A 3270
      L32=0                                                              A 3280
      L43=0                                                              A 3270
C                                                                        A 3280
C                                                                        A 3290
C     CHECK  REDUNDANCY  OF  X1,X2,X3,X4                                 A 3300
C                                                                        A 3310
C                                                                        A 3320
      DO 133 I3=1,3                                                      A 3330
         IF (X3(I3).EQ.X2(I3)) L32=10                                    A 3340
         IF (X2(I3).EQ.X1(I3)) L21=10                                    A 3350
         IF (X3(I3).EQ.X1(I3)) L31=10                                    A 3360
         IF (X4(I3).EQ.X1(I3)) L41=10                                    A 3370
         IF (X4(I3).EQ.X3(I3)) L43=10                                    A 3380
         IF (X4(I3).EQ.X2(I3)) L42=10                                    A 3390
  133 CONTINUE                                                           A 3400
      IF (L41.NE.10) GO TO 134                                           A 3410
      GO TO 140                                                          A 3420
  134 IF (L31.NE.10) GO TO 135                                           A 3430
      GO TO 144                                                          A 3440
  135 IF (L21.NE.10) GO TO 138                                           A 3450
      KNONE=NONE-NKI                                                     A 3460
      IF (KNONE.LE.0) GO TO 147                                          A 3470
      KKI=ITEND(KNONE)                                                   A 3480
      DO 136 L=1,3                                                       A 3490
  136 X2(L)=Y(KKI,L)                                                     A 3500
      DO 137 L=1,8                                                       A 3510
  137 X2(3+L)=YST(KKI,L)                                                 A 3520
      NKI=NKI+1                                                          A 3530
      GO TO 132                                                          A 3540
  138 IF (L43.NE.10) GO TO 139                                           A 3550
      GO TO 140                                                          A 3560
  139 IF (L42.NE.10) GO TO 143                                           A 3570
  140 KNONE=NONE-NKI                                                     A 3580
```

APPENDIX B

```
      IF (KNONE.LE.0) GO TO 147                                          A 3590
      KKI=ITENO(KNONE)                                                   A 3600
      DO 141 L=1,3                                                       A 3610
  141 X4(L)=Y(KKI,L)                                                     A 3620
      DO 142 L=1,8                                                       A 3630
  142 X4(3+L)=YST(KKI,L)                                                 A 3640
      NKI=NKI+1                                                          A 3650
      GO TO 132                                                          A 3660
  143 IF (L32.NE.10) GO TO 147                                           A 3670
  144 KNONE=NONE-NKI                                                     A 3680
      IF (KNONE.LE.0) GO TO 147                                          A 3690
      KKI=ITENO(KNONE)                                                   A 3700
      DO 145 L=1,3                                                       A 3710
  145 X3(L)=Y(KKI,L)                                                     A 3720
      DO 146 L=1,8                                                       A 3730
  146 X3(3+L)=YST(KKI,L)                                                 A 3740
      NKI=NKI+1                                                          A 3750
      GO TO 132                                                          A 3760
  147 CONTINUE                                                           A 3770
      GO TO 107                                                          A 3780
  148 WRITE (60,158) LSUB                                                A 3790
      RETURN                                                             A 3800
C                                                                        A 3810
  149 FORMAT (/,5X,2HG(,I2,2H)=,F10.3,10X,2HH(,I2,2H)=,F10.3,/)          A 3820
  150 FORMAT (/,5X,22H CAUTION   G(I)=H(I) =,F10.3,4X,18H  AT  CONSTRAIN A 3830
     1T  ,I3,14H ,  SUBBASIN  ,I4,//)                                    A 3840
  151 FORMAT (///,5X, 15H ITERATION NO. ,I4, 37H STANDARD DISCRIPENCY     A 3850
     1       DNST =,F10.5,//)                                            A 3860
  152 FORMAT (/,5X,  6H X(1)=,F10.3,  6H X(2)=,F10.3,  6H X(3)=,F10.3,//  A 3870
     1/)                                                                 A 3880
  153 FORMAT (/,5X,  5HINDEX,2X,2X,  5HY*(I),2X,2X,2X,  5HY(X*),2X,2X, 1 A 3890
     11H DELTA   Y ,//)                                                  A 3900
  154 FORMAT (/,5X,2X,I1,2X,2X,F9.3,2X,F9.3,2X,F9.4,//)                  A 3910
  155 FORMAT (//,5X,7(2X,  2HQ(,I1,  2H)=,F9.2,2X))                      A 3920
  156 FORMAT (//,5X, 33HITERATIONS , FROM WORST TO BEST :,//,10X,15(2X,I A 3930
     13,2X),)                                                            A 3940
  157 FORMAT (//,5X, 33HCORESPONDING  NORM(DISCRIPENCY) .,//,10X,10(F8.2 A 3950
     1,2X),)                                                             A 3960
  158 FORMAT (/,5X, 53H FAILED TO CONVERGE IN ITMAX ITERATIONS           A 3970
     1    ,I4,//)                                                       A 3980
C                                                                        A 3990
      END                                                                A 4000
```

```
      SUBROUTINE SIMAPRX                                                B   10
C                                                                       B   20
C     ****************************************************************  B   30
C                                                                       B   40
C                                                                       B   50
C     DETERMINES  LINEAR  REGRESSION  COEFFICIENTS  FOR  SIMULATION    B   60
C                 RESPONSE  APPROXIMATION                               B   70
C                                                                       B   80
C                                                                       B   90
C     ----------------------------------------------------------------  B  100
C                                                                       B  110
C     INPUT  PARAMETERS                                                 B  120
C                                                                       B  130
C           X1,X2,X3,X4      =  GIVEN  POINTS                         B  140
C            XA=X1           =  CENTAL POINT OF  REGRESSION           B  150
C                                                                       B  160
C     OUTPUT  PARAMETERS                                                B  170
C                                                                       B  180
C           CALPH,CBETA,CGAMA  =  REGRESSION  COEFFICIENTS              B  190
C                                                                       B  200
C                                                                       B  210
C     ----------------------------------------------------------------  B  220
C                                                                       B  230
      COMMON /FORPP/ X1(15),X2(15),X3(15),X4(15),NPOINTS                B  240
      COMMON /APRXS/ XA(15),CALPH(8),CBETA(8),CGAMA(8)                  B  250
C                                                                       B  260
      DIMENSION DEDLT(12)                                               B  270
C                                                                       B  280
      EQUIVALENCE (DEDLT(1),D1BA)                                       B  290
      EQUIVALENCE (DEDLT(2),D2BA)                                       B  300
```

```
      EQUIVALENCE (DEDLT(3),D3BA)                                          B  310
      EQUIVALENCE (DEDLT(4),D1DA)                                          B  320
      EQUIVALENCE (DEDLT(5),D2DA)                                          B  330
      EQUIVALENCE (DEDLT(6),D3DA)                                          B  340
      EQUIVALENCE (DEDLT(7),D1CA)                                          B  350
      EQUIVALENCE (DEDLT(8),D2CA)                                          B  360
      EQUIVALENCE (DEDLT(9),D3CA)                                          B  370
      EQUIVALENCE (DEDLT(10),DYBA)                                         B  380
      EQUIVALENCE (DEDLT(11),DYCA)                                         B  390
      EQUIVALENCE (DEDLT(12),DYDA)                                         B  400
C                                                                          B  410
      DO 101 IL=1,11                                                       B  420
  101 XA(IL)=X1(IL)                                                        B  430
      WRITE (60,105)                                                       B  440
      DO 102 I=1,11                                                        B  450
  102 WRITE (60,106) I,XA(I),X1(I),X2(I),X3(I),X4(I)                       B  460
      D1BA=X2(1)-X1(1)                                                     B  470
      D1CA=X3(1)-X1(1)                                                     B  480
      D2BA=X2(2)-X1(2)                                                     B  490
      D2CA=X3(2)-X1(2)                                                     B  500
      D3BA=X2(3)-X1(3)                                                     B  510
      D3CA=X3(3)-X1(3)                                                     B  520
      D1DA=X4(1)-X1(1)                                                     B  530
      D2DA=X4(2)-X1(2)                                                     B  540
      D3DA=X4(3)-X1(3)                                                     B  550
      WRITE (60,107)                                                       B  560
      DO 104 K=1,8                                                         B  570
         DYBA=X2(K+3)-X1(3+K)                                              B  580
         DYCA=X3(3+K)-X1(3+K)                                              B  590
         DYDA=X4(3+K)-X1(3+K)                                              B  600
C                                                                          B  610
         DO 103 I=1,12                                                     B  620
            IF ((DEDLT(I)).EQ.0.0000) DEDLT(I)=.0001                       B  630
  103    CONTINUE                                                          B  640
C                                                                          B  650
         CGAMA(K)=(DYBA*D1BA*D1CA*D2DA+DYBA*D1CA*D1DA*D2BA-DYCA*D1BA*D1B   B  660
     1   A*D2DA-DYCA*D1BA*D1DA*D2BA-DYBA*D1BA*D1DA*D2CA-DYBA*D1CA*D1DA*D   B  670
     2   2BA+DYDA*D1BA*D1BA*D2CA+DYDA*D1CA*D1BA*D2BA)/(D1BA*D1BA*D2CA*D3   B  680
     3   DA+D1BA*D1CA*D2BA*D3DA+D1DA*D1BA*D2CA*D3BA+D1CA*D1DA*D2BA*D3BA+   B  690
     4   D1BA*D1BA*D2DA*D3CA+D1BA*D1DA*D2BA*D3CA+D2CA*D1BA*D2DA*D3BA+D1C   B  700
     5   A*D1DA*D2BA*D3BA)                                                 B  710
         CBETA(K)=(DYBA*D1DA-DYDA*D1BA-CGAMA(K)*(D1BA*D3DA+D1DA*D3BA))/(   B  720
     1   D1BA*D2DA+D1DA*D2BA)                                              B  730
         CALPH(K)=(DYBA-CBETA(K)*D2BA-CGAMA(K)*D3BA)/D1BA                  B  740
         WRITE (60,108) K,CALPH(K),CBETA(K),CGAMA(K)                       B  750
  104 CONTINUE                                                             B  760
      RETURN                                                               B  770
C                                                                          B  780
  105 FORMAT (///,12X,  1HI,15X,  5HXA(I),7X,  5HX1(I),7X,  5HX2(I),7X,    B  790
     1 5HX3(I),7X,  5HX4(I),/)                                             B  800
  106 FORMAT (/,10X,I4,10X,5(F12.3,3X),/)                                  B  810
  107 FORMAT (//,9X,  5HINDEX,2X,  8HCALPH(I),3X,  8HCBETA(I),3X,  8HCGA   B  620
     1MA(I),//)                                                            B  830
  108 FORMAT (//,10X,I3,2X,F10.4,2X,F10.4,2X,F10.4,/)                      B  840
C                                                                          B  850
      END                                                                  B  860
```

```
      SUBROUTINE LSTORM (X22)                                                  C   10
C                                                                              C   20
C     ***************************************************************          C   30
C                                                                              C   40
C      TRANSFERED  THROUGH  COMMON-BLOCK =                                     C   50
C                 S,T,AOR,AREA,LSUB,ICOORD,TF,VF,OV,TO,OP                      C   60
C                                                                              C   70
C                                                                              C   80
C     ---------------------------------------------------------------          C   90
C                                                                              C  100
      DIMENSION X22(76)                                                        C  110
      COMMON /APRXS/ XA(15),CALPH(8),CBETA(8),CGAMA(8)                         C  120
C                                                                              C  130
C     STORM  SIMULATION  RESPONSE  APROXIMATION                                C  140
C                                                                              C  150
      DO 101 KI=1,8                                                            C  160
         X22(3+KI)=XA(3+KI)+CALPH(KI)*(X22(1)-XA(1))+CBETA(KI)*(X22(2)-X       C  170
     1   A(2))+CGAMA(KI)*(X22(3)-XA(3))                                        C  180
  101 CONTINUE                                                                 C  190
      TF(LSUB)=X22(4)                                                          C  200
      VF(LSUB)=X22(5)                                                          C  210
      TO(LSUB)=X22(6)                                                          C  220
      DO 102 LM=1,5                                                            C  230
  102 OP(LM,LSUB)=X22(6+LM)                                                    C  240
      RETURN                                                                   C  250
C                                                                              C  260
      END                                                                      C  270
                                                                               C  280
```