

OPTIMIZATION OF INDUSTRIAL MANAGEMENT SYSTEMS

BY

THE CONJUGATE GRADIENT METHOD

by

1264

ASHOK KUMAR ARORA

B. Tech. (Hons), Indian Institute of Technology,
Kharagpur, India, 1965

A MASTER'S THESIS

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1970

APPROVED BY:


Major Professor

LD
2668
T4
1970
A75

TABLE OF CONTENTS

	PAGE
CHAPTER 1. INTRODUCTION	1
CHAPTER 2. LITERATURE REVIEW	3
2.1 Conjugate Gradient Methods for Unconstrained Problems	4
2.2 Methods for Constrained Problems	7
CHAPTER 3. THE CONJUGATE GRADIENT METHOD	9
3.1 Formulation of the Problem	10
3.2 Constrained Maximum	11
3.3 Algorithm	15
CHAPTER 4. PROGRAM FLOW	
4.1 Limits and Tolerances	18
4.2 Main Program	20
4.3 Subroutines	22
CHAPTER 5. APPLICATION OF THE CONJUGATE GRADIENT METHOD TO MANAGEMENT PROBLEMS	23
5.1 An Inventory Model	23
5.2 An Inventory and Advertisement Model	36
5.3 A Production and Advertisement Model	52
CHAPTER 6. DISCUSSION	70
ACKNOWLEDGEMENT	72
REFERENCES	73
APPENDIX A: NOMENCLATURE	76
APPENDIX B: CUBIC INTERPOLATION SCHEME	79

**THIS BOOK
CONTAINS
NUMEROUS PAGES
WITH THE ORIGINAL
PRINTING BEING
SKEWED
DIFFERENTLY FROM
THE TOP OF THE
PAGE TO THE
BOTTOM.**

**THIS IS AS RECEIVED
FROM THE
CUSTOMER.**

**THIS BOOK
CONTAINS
NUMEROUS PAGES
WITH DIAGRAMS
THAT ARE CROOKED
COMPARED TO THE
REST OF THE
INFORMATION ON
THE PAGE.**

**THIS IS AS
RECEIVED FROM
CUSTOMER.**

APPENDIX C: RECURSION RELATIONS	82
APPENDIX D: GRADIENT CALCULATIONS	85
APPENDIX E: FLOWCHARTS	107
APPENDIX F: COMPUTER PROGRAM	124

CHAPTER 1

INTRODUCTION

In recent years there has been a remarkable growth of interest in the problems of management systems optimization. It, perhaps, has been motivated by the economic benefits associated with the optimal decisions regarding the allocation of resources. Moreover, it seems natural that in treating decision problems an approach should be to make the best of the given situation. The fundamental approach to solve decision problems is to optimize a performance criterion by the proper selection of decision parameters.

Numerous mathematical techniques have been proposed to solve linear and non-linear systems. Linear programming has been used successfully to solve linear problems. Unfortunately, nonlinear problems have always presented greater difficulties than linear ones. Several methods like the calculus of variation, maximum principle and dynamic programming have been developed for solving non-linear systems. Of all the analytical approaches, only dynamic programming has been used fairly successfully. This technique, however, has a drawback in that it cannot be used for problems involving large number of state variables.

In this decade some new techniques have been proposed to solve a class of non-linear problems characterized by non-linear objective function and linear constraints. Among these techniques, Rosen's gradient projection method [20] and Goldfarb's conjugate gradient method [7] are noticeable. The two techniques are analogous to each other in many ways. The gradient projection method, however,

has the drawback of slow convergence rate near the optimum. For the purpose of this study, the conjugate gradient method developed by Goldfarb is considered.

A brief review of a class of gradient methods proposed by different authors for solving unconstrained and constrained nonlinear programming problems is given in chapter 2. The conjugate gradient method, along with its algorithm, is described in chapter 3. Formulation of the mathematical model is also given in this chapter. Chapter 4 deals with the program flow together with the limits and tolerances required for the successful running of the program. Three problems have been solved using the technique. The problems are discussed along with their results in chapter 5.

CHAPTER 2

LITERATURE REVIEW

In general non-linear programming problems are concerned with minimization or maximization of a non-linear objective function, $f(x_1, x_2, \dots, x_m)$, subject to certain constraints. The constraints may appear as linear or non-linear constraints. The functions may also be grouped into two main classes, according to whether the gradient vector $g^i = \frac{\partial f}{\partial x^i}$ can be obtained analytically at each point or must be estimated from the differences of values of the function, f .

In the neighborhood of the optimum of a general non-linear function, the function can be approximated by the Taylor series expansion of the form

$$f(x) = f(x_0) + \frac{1}{2} (x - x_0)^T A (x - x_0) + \text{higher terms} \quad (2-1)$$

where x_0 is the optimum value of x and A is the matrix of second partial derivatives. Since near the optimum, the second order terms dominate, it is generally argued that the methods which can optimize a quadratic function efficiently will probably be good on more general functions. Furthermore, even in the regions remote from the optimum, such methods, by taking account of the curvature of the function, are best able to deal with complex situations, such as the presence of a long curving valley, thereby avoiding the deficiency of oscillatory behaviour, that is present in methods like steepest ascent. This provides a motivation for the development of second order convergence methods, and as such a number of methods have been proposed in recent

years for solving unconstrained as well as constrained problems. Such methods have the property that they develop quadratic convergence, without requiring the calculation of second partial derivatives, once a neighborhood of the optimum is reached. A brief review of some of such methods along with recently developed gradient projection methods follows.

2.1 Conjugate Gradient Methods for Unconstrained Problems.

The methods considered here are called variously conjugate gradient, quasi-Newton and variable metric methods. Before describing them, it would be worthwhile to define a conjugate gradient process. Consider the case of a quadratic function f having a Hessian matrix A , which is positive definite. Vectors $s^0, s^1, s^2, \dots, s^{n-1} \neq 0$ with the property

$$(s^i)^T A s^j = 0 \quad i \neq j \quad (2-2)$$

are said to be conjugate with respect to A . Any method, that generates conjugate directions $s^0, s^1, s^2, \dots, s^{n-1}$ satisfying (2-2), is said to be a conjugate direction process. Since the Hessian matrix A is positive definite, the vectors $s^0, s^1, s^2, \dots, s^{n-1}$ are linearly independent and form a basis in an n -dimensional Euclidean space E_n . For a quadratic f , successive minimizations in the conjugate directions minimize the function in, at most, n steps with no round off errors. Kelley [11] has summarized some of the algorithms that generate conjugate directions.

Algorithm 1:

$$s^i = - \gamma^i s^i \quad i \geq 0 \quad (2-3)$$

where

$$s^0 = g^0 \quad (2-4)$$

$$\text{and } s^{i+1} = g^{i+1} + \frac{(g^{i+1})^2}{(g^i)^2} s^i \quad i \geq 0 \quad (2-5)$$

Here and elsewhere, x^0 is chosen arbitrarily. $\gamma^i \geq 0$ is selected by a one-dimensional search for a minimum of the function, $f(x^i + \gamma^i s^i)$, along s^i , so as to obtain a new point $x^{i+1} = x^i + \sigma^i$. Thus, σ^i gives the change in the variable vector x during i^{th} iteration.

Algorithm 2:

$$\sigma^i = -\gamma^i s^i \quad i \geq 0 \quad (2-6)$$

$$\text{where } s^0 = g^0 \quad (2-7)$$

$$s^{i+1} = g^{i+1} - \left[\sum_{j=0}^i \frac{\sigma^j (y^j)^T}{(\sigma^j)^T y^j} \right] g^{i+1} \quad 0 \leq i \leq n-1 \quad (2-8)$$

$$= g^{i+1} - \left(\sum_{j=i-n+1}^i \frac{\sigma^j (y^j)^T}{(\sigma^j)^T y^j} \right) g^{i+1} \quad i \geq n \quad (2-9)$$

$$\text{and } y^i = g^{i+1} - g^i \quad (2-10)$$

Algorithm 3:

$$\sigma^i = -\gamma^i s^i \quad i \geq 0 \quad (2-11)$$

$$\text{where } s^0 = g^0 \quad (2-12)$$

$$s^{i+1} = g^{i+1} - \frac{\sigma^i (y^i)^T}{(\sigma^i)^T y^i} g^{i+1} \quad i \geq 0 \quad (2-13)$$

$$\text{and } y^i = g^{i+1} - g^i \quad (2-14)$$

The above three algorithms are the three different versions of the conjugate gradient method. They become identical for a quadratic f .

There are several variable metric methods proposed for generating conjugate directions. They consist of choosing an $n \times n$ matrix H^i (that approximates A^{-1}) and generating A -conjugate directions s^i given by

$$s^i = -H^i g^i \quad (2-15)$$

where $H^0, H^1, \dots$ is a sequence of symmetric positive definite matrices. H^0 is arbitrary, but is usually taken as an identity matrix. A new point is defined by $x^{i+1} = x^i + \gamma^i s^i$, where γ^i is calculated to minimize $f(x^{i+1})$. The matrix H^i is updated at each iteration. Several methods have been proposed which essentially differ in the manner in which the matrix H^i is updated. Probably the most powerful of the variable metric methods is Fletcher and Powell's modification [5] of Davidon's variable metric method [4]. The method employed by Fletcher and Powell in updating the matrix H_i may be stated as

$$H^{i+1} = H^i + \frac{\sigma^i (\sigma^i)^T}{(y^i)^T \sigma^i} - \frac{H^i y^i (y^i)^T H^i}{(y^i)^T H^i y^i} \quad i \geq 0 \quad (2-16)$$

$$\text{where } y^i = g^{i+1} - g^i \quad (2-17)$$

Another algorithm that bears a relationship to the above algorithm similar to that which batch processing of observations bears to sequential processing in orbit determination is given in [11]. This has been referred to as modified Davidon algorithm. According to this

$$H^{i+1} = H^i - \frac{H^i y^i (y^i)^T H^i}{(y^i)^T H^i y^i} \quad i \geq 0 \neq mn-1 \quad (2-18)$$

and

$$H^{mn} = \sum_{j=mn-n}^{mn-1} \frac{\sigma^j (\sigma^j)^T}{(\sigma^j)^T y^j} \quad (2-19)$$

where

$$y^i = g^{i+1} - g^i \quad (2-20)$$

The basic idea behind this algorithm to reduce the rank of the matrix H by one at each update and after rank annihilation, replace it by the estimate of A^{-1} given by equation (2-19). A similar approach has been suggested in the gradient projection method [19]. There, however, after rank annihilation, the matrix H is replaced by the identity matrix I . A more detailed comparative study of the various conjugate direction methods or variable metric methods can be found in [11,19,23].

2.2 Methods for Constrained Problems

Based on the knowledge gained from the work done in the field of unconstrained optimization, some new methods have been proposed to handle linear as well as non-linear constraints on the control variables. Rosen [20] developed a gradient projection method for solving a subclass of non-linear problems characterized by linear constraints. The constraints may consist of inequalities, equalities or both. The important features of the method are i) movement along the boundary of the polyhedron describing the feasible region, and ii) the generation of a sequence of improved objective function values. Since the method is based on the well known steepest ascent method, it has the same deficiencies as its precursor, namely, slow rate of convergence near

the optimum and oscillation along steep ridges. Analogous to the gradient projection method, Goldfarb [7] developed the conjugate gradient method which is the subject of this study.

Rosen has also extended his gradient projection method to handle non-linear constraints [21]. The basic approach of this method is similar to that with linear constraints. An advantage of the gradient projection method is that if a feasible starting point is known, all points obtained in the optimization procedure are feasible, so that the procedure may be stopped at anytime with an improved feasible point. This method is also advantageous over the cutting plane method developed by Kelley [13] in that there is no increase in the total number of constraints as the solution proceeds. Kelley and his coworkers [10] have proposed an accelerated gradient method for solving problems with non-linear constraints. Their approach is to augment the performance function by penalty terms which measure the constraint violations. The augmented function is minimized using Davidon's method. This results in an approximate solution with small violations of the constraints. Newton's method is then used to refine the solution exploiting the second order information accumulated in the converged metric. Very recently, an accelerated gradient projection method for use with non-linear constraints has been reported by Kelley and Speyer [12]. The method uses a one-dimensional search to a minimum of a performance index augmented by a correction for the constraint violation. A metric update, based on the changes in the projection gradient vector, provides quadratic terminal convergence.

CHAPTER 3

THE CONJUGATE GRADIENT METHOD

The conjugate gradient method developed by Goldfarb is a new method of solving a class of non-linear problems characterized by linear constraints. This technique is the modification and the extension of Davidon's variable metric method for unconstrained minimization, to take into account linear constraints which may consist of equalities, inequalities or both. It maximizes the value of a given objective function of the system of variables while satisfying the equations and constraints. The objective function may also be minimized by maximizing the negative of the actual objective function.

Briefly, the conjugate gradient method is designed to start from an initial specified point and to locate a feasible point using Rosen's technique, if the initial point is not feasible. The method, then, follows an iterative procedure locating new feasible points with increased value of the objective function. This is accomplished by moving in a direction conjugate to the previous directions of search. As the method proceeds, information gained about the local curvature of the objective function is incorporated into a matrix which determines the direction in which to move.

At each iteration, the initial step length is selected as the maximum length without violating the constraints, or a predetermined maximum step length. In the case of non-linear objective function, the cubic interpolation scheme proposed by Davidon is used so that the value of the objective function increases at each step. The

maximum is found when any step size that would increase the value of the objective function would violate the constraints, or when the gradient is within the specified tolerance.

3.1 Formulation of the Problem

Consider a problem with m variables, x_i , $i = 1, 2, \dots, m$, representing a point in an m -dimensional Euclidean space. It is required to find a local maximum (or minimum) of a function

$$f(x) = f(x_1, x_2, \dots, x_m) \quad (3-1)$$

subject to k linear equality and inequality constraints of the form

$$\sum_{j=1}^m n_{ij} x_j - b_i = 0 \quad i = 1, 2, \dots, e \quad (3-2)$$

$$\sum_{j=1}^m n_{ij} x_j - b_i \geq 0 \quad i = e+1, e+2, \dots, k \quad (3-3)$$

where n_{ij} have been normalized so that

$$\sum_{j=1}^m (n_{ij})^2 = 1 \quad i = 1, 2, \dots, k \quad (3-4)$$

These constraints form a convex region R in E_m , which is assumed to be bounded. It follows, therefore, that there must be at least $m+1$ constraints, that is, $k \geq m+1$. Corresponding to each of the k constraints, the unit vector n_i is defined as

$$n_i = \{n_{i1}, n_{i2}, \dots, n_{im}\} \quad i = 1, 2, \dots, k \quad (3-5)$$

Equations (3-2) and (3-3) may now be written as

$$n_i^T x - b_i = 0 \quad i = 1, 2, \dots, e \quad (3-6)$$

$$n_i^T x - b_i \geq 0 \quad i = e+1, e+2, \dots, k \quad (3-7)$$

where

$$n_i^T n_i = 1 \quad i = 1, 2, \dots, k$$

In geometric terms, these equality and inequality constraints represent e hyperplanes H_i and $(k-e)$ closed half spaces, whose intersection in E_m is a convex polyhedron region R . The unit vector n_i is orthogonal to H_i and is directed so that if it originates at a point x in H_i , it points into the region R . A set of hyperplanes is linearly independent if the corresponding vectors n_i are linearly independent. If there are q linearly independent hyperplanes, then their intersection forms an $(m-q)$ dimensional subspace M_q in E_m . Clearly, there can at the most m linearly independent hyperplanes in E_m .

3.2 Constrained Maximum

Consider an objective function

$$f(x) = f(x_1, x_2, \dots, x_m) \quad (3-8)$$

If $f(x)$ is differentiable at all points x in E_m , each point x has associated with it an m -dimensional gradient vector

$$g(x) = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_m} \right] \quad (3-9)$$

The components of $g(x)$ can be considered as co-ordinates of another m -dimensional Euclidean space. If, however, $f(x)$ is twice differentiable, then in the neighborhood of x in E_m , the Hessian matrix of the second partial derivatives

$$G(x) = \left\{ \frac{\partial^2 f(x)}{\partial x_i \partial x_j} \right\} \quad (3-10)$$

defines a linear mapping of changes in position, dx , on to the change in the gradient, dg , as given by

$$dg = G(x)dx \quad (3-11)$$

The directions of the vectors dg and dx depend upon the ratios of the eigenvalues of $G(x)$. If, however, dx is an eigenvector of the Hessian matrix, both the vectors will point in the same direction.

Considering that $f(x)$ is a strictly concave function, it is desired to find a point x at which $f(x)$ has a global maximum. For this purpose, the important property of concavity requirement on $f(x)$ is that the symmetric matrix of second partial derivatives is negative semidefinite over the entire space E_m . Then $f(x)$ is given by the three-term Taylor expansion

$$f(x) = f_0 + a^T x + \frac{1}{2} x^T G x \quad (3-12)$$

We can write

$$g^{i+1} - g^i = G(x^{i+1} - x^i) \quad (3-13)$$

where

$$g^i = g(x^i)$$

For an unconstrained concave function, the region R is all of E_m . The necessary and sufficient condition that x be the maximum point is that $g(x) = 0$. The maximum for such a case can be found by Newton's method for solving m equations in m unknowns, namely, $g_j(x) = 0$, $j = 1, 2, \dots, m$, that result from the necessary and sufficient

conditions for a global maximum for a strictly concave function. If the region R is bounded and if there exists a point x in R such that $g(x) = 0$, the global maximum is called the interior global maximum of $f(x)$. If, however, $g(x)$ does not vanish in the interior of R , then the global maximum lies outside the boundary of R and is called a constrained global maximum.

In most of the actual problems $f(x)$ is not quadratic. In such cases the function can be approximated to be quadratic only in a small neighborhood of a point. Therefore, evaluation of the Hessian matrix and inversion of it at each step can be time consuming, since if the partial derivatives of $f(x)$ are approximated by finite differences, large number of functional evaluations are necessary to compute the Hessian matrix. Moreover, for non-concave functions, convergence cannot be guaranteed unless the starting point is close enough to the maximum. Davidon's variable metric method and Fletcher and Powell's method are, however, more attractive in that they require explicit calculation of only first partial derivatives, and yet develop quadratic convergence once a neighborhood of optimum is reached in which the second partial derivatives of the function remain sensibly constant.

Let us now consider a quadratic function $f(x)$ given by equation (3-12). It is required to locate the point at which the function is maximized in the linear manifold, M_q , formed by the intersection of q linearly independent hyperplanes H_i , $i = 1, 2, \dots, q$. Let $n_1, n_2, \dots, n_q$ be the unit normals to these hyperplanes. Define an $(m \times q)$ matrix

$$N_q = \{n_1, n_2, \dots, n_q\} \quad (3-14)$$

Now, the necessary and sufficient condition for $f(x^*)$ to be global maximum of $f(x)$ over M_q is that the gradient vector $g(x^*)$ be orthogonal to M_q . In other words

$$g^* = N_q \alpha \quad (3-15)$$

where α is a q dimensional vector. Substituting $g^{i+1} = N_q \alpha$ in equation (3-13) and premultiplying both sides by G^{-1} , we get

$$x^{i+1} - x^i = \Delta x^i = G^{-1}(N_q \alpha - g^i) \quad (3-16)$$

The vector Δx^i lies in the linear manifold M_q , and therefore

$$N_q^T \Delta x^i = 0 \quad (3-17)$$

or

$$N_q^T G^{-1} N_q \alpha - N_q^T G^{-1} g^i = 0 \quad (3-18)$$

The matrices N_q^T and G^{-1} are of the ranks q and m ($\geq q$), respectively.

The inverse of $N_q^T G^{-1} N_q$, therefore, exists [18]. Thus, from equation (3-18)

$$\alpha = (N_q^T G^{-1} N_q)^{-1} N_q^T G^{-1} g^i \quad (3-19)$$

Substituting the value of α in (3-16), we have

$$x^{i+1} = x^i + [G^{-1} N_q (N_q^T G^{-1} N_q)^{-1} N_q^T G^{-1} - G^{-1}] g^i \quad (3-20)$$

or

$$x^{i+1} = x^i - \hat{P}_q G^{-1} g^i \quad (3-21)$$

where

$$\hat{P}_q = I - G^{-1} N_q (N_q^T G^{-1} N_q)^{-1} N_q^T$$

$\hat{P}_q$ is the projection matrix and projects any point in E_m on to M_q .

In the conjugate gradient method the matrix $H_q = -\hat{P}_q G^{-1}$ acts as a variable metric. It is a positive semidefinite matrix. This matrix is updated at each step according to the approach followed by Fletcher and Powell, to incorporate the information about the local curvature of the function.

3.3 Algorithm

The basic algorithm used in the conjugate gradient method follows very closely the one used in Rosen's gradient projection (GP). But unlike that in GP, here the directions of search are determined by premultiplying the gradient vector by the matrix H_q , rather than by the orthogonal projection matrix P_q . The heart of the algorithm lies in the manner in which the matrix H_q is updated. The method requires the calculation of the matrix $(N_q^T N_q)^{-1}$ at each step. The cubic interpolation scheme used to locate the approximate maximum value of the function along the direction of search is given in Appendix B. The recursion relations used to compute the matrix $(N_q^T N_q)^{-1}$ are given in Appendix C.

The algorithm is given for an arbitrary point x^1 in the region R , having a function value $f(x^1)$ and a gradient $g^1 = g(x^1)$. It is assumed that x^1 lies in the subspace M_q formed by the intersection of q linearly independent hyperplanes. H_q^1 is, therefore, the matrix operator. For simplicity it is also assumed that there are no equality constraints. The algorithm for the conjugate gradient method can be stated as follows:

Step 1

Given x^1 , g^1 and H_q^1 , compute $H_q^1 g^1$ and $\alpha = (N_q^T N_q)^{-1} N_q^T g^1$

If $H_q^i g^i = 0$ and $\alpha_j \leq 0$, $j = 1, 2, \dots, q$, then x^i is the global maximum.

Step 2

If this is not the case test for the constraint drop. Either $||H_q^i g^i|| > \max \{0, \frac{1}{2} \alpha_q d_{qq}^{-1/2}\}$ or $||H_q^i g^i|| \leq \frac{1}{2} \alpha_q d_{qq}^{-1/2}$, where $\alpha_q d_{qq}^{-1/2} \geq \alpha_i d_{ii}^{-1/2}$, $i = 1, 2, \dots, q-1$ and where d_{ii} is the i^{th} diagonal element of the matrix $(N_q^T N_q)^{-1}$. If the former holds, go to step 3. If the latter holds, it means movement is restricted in the q th hyperplane. Therefore, drop H_q^i from the constraint basis and obtain H_{q-1}^i using the following equation

$$H_{q-1}^i = H_q^i + \frac{P_{q-1} n_q n_q^T P_{q-1}}{n_q^T P_{q-1} n_q} \quad (3-22)$$

where

$$P_{q-1} = I - N_{q-1} (N_{q-1}^T N_{q-1})^{-1} N_{q-1}^T.$$

Let $q = q-1$ and go to step 1.

Step 3

Let $s^i = H_q^i g^i$ and compute

$$\lambda_j = -(n_j^T x^i - b_j) / n_j^T s^i, \quad j = q+1, q+2, \dots, k$$

$$\text{and } \lambda^i = \min \{\lambda_j > 0\} \quad (3-23)$$

Now, using the cubic interpolation method obtain γ^i , $0 < \gamma^i \leq \lambda^i$, such that $f(x^i + \gamma^i s^i)$ is maximized.

Let $\sigma^i = \gamma^i s^i$ and $x^{i+1} = x^i + \sigma^i$ and compute $g^{i+1} = g(x^{i+1})$

Step 4

If $\gamma^i = \lambda^i$ and $(s^i)^T g^{i+1} \geq 0$, add to the constraint basis in step 3

the hyperplane corresponding to the $\min\{\lambda_j\}$. Compute

$$H_{q+1}^{i+1} = H_q^i - \frac{H_q^i n_j n_j^T H_q^i}{n_j^T H_q^i n_j} \quad (3-24)$$

Let $i = i+1$ and $q = q+1$ and return to step 1.

Step 5

Otherwise set $y^i = g^{i+1} - g^i$ and update the matrix operator H_q^i using the following relation

$$H_q^{i+1} = H_q^i - \frac{\sigma^i (\sigma^i)^T}{(\sigma^i)^T y^i} - \frac{H_q^i y^i (y^i)^T H_q^i}{(y^i)^T H_q^i y^i} \quad (3-25)$$

Let $i = i+1$ and return to step 1.

In actual practice an initial feasible point x^0 is obtained by Rosen's technique. If x^0 lies in the interior of the convex region R and also if there are no equality constraints, H_0^0 is chosen as an identity matrix I . If, however, x^0 lies on exactly q linearly independent hyperplanes, these hyperplanes (constraints) are added to the basis, the hyperplanes corresponding to the equality constraints, if any, being added first. H_q^0 is then computed from H_0^0 by using equation (3-24) q times.

CHAPTER 4

PROGRAM FLOW

The computer program for the conjugate gradient method is based on program known as GP 90 [16] developed for the gradient projection method of solving non-linear problems with linear constraints. The flow chart of the program is given in Appendix E. This includes logics for the matrix computations, matrix re-inversion and other subroutines used. A brief description of the limits and tolerances necessary for the successful running of the program, the main program itself and the subroutines, is given below.

4.1 Limits and Tolerances

There are three limits which need be set. Their setting depends upon the size and the type of the problem being solved.

LIMIT, refers to the maximum number of steps. It prevents the program from grinding away without accomplishing anything because of incorrect setting of tolerances or limits or because of wrong input data. It is difficult to predetermine the number of steps required to reach the maximum as it depends upon such factors as size of the problem, type of the function and number of constraints in basis. It keeps the run to a reasonable length without having to rely on a time limit.

MXRN, maximum number of re-inversions. The main purpose of this limit is to prevent the problem from re-inverting too often thus consuming extra time. The program is designed not to re-invert twice in a row to the same basis, but it may re-invert after only one basis

change or possibly repeat a series of re-inversions.

$\beta_{\max}$ is used to limit the number of interpolations required to find an x for which $f(x)$ has increased. It depends upon the non-linearity of the function. For a quadratic function it should be one.

The three tolerances required to be specified are gradient tolerance, ϵ_1 , constraint tolerance, ϵ_2 and linear dependence tolerance, ϵ_3 .

ϵ_1 is used to determine when the norm of the gradient is zero and the problem has reached the maximum value. Its value may vary from problem to problem as it depends on the relationship between the norm of the gradient and the corresponding change in the value of the objective function. A smaller value of ϵ_1 leads to better 'maximum' but at the cost of increased computational time. Value of ϵ_1 between .001 and .005 seems to be reasonable.

The constraint tolerance, ϵ_2 , is the acceptable error in satisfying a constraint. It determines when a constraint is active. ϵ_2 may be about $10^{-3} b'_i$, where b'_i is the largest value of the right hand side of constraints, b_i , divided by the corresponding scaling factor or 10^{-3} times the largest value of x .

The general value used for the linear dependence tolerance, ϵ_3 , is .005. This tolerance determines when a constraint is linearly dependent and, therefore, cannot be added to the basis.

Besides the limits and tolerances described above, it is necessary to specify the maximum step length $\tau_{\max}$. It is usually taken as one. The program has been designed to take an initial step of not less than

$\tau_{\min}$. If the initial step size is less than $\tau_{\min}$, it is taken to be $\tau_{\max}$ and then interpolation is done if necessary.

4.2 Main Program

For the purpose of description, the main program can be divided into three sections: input, starting procedure and step procedure. In the first section, the input data is read and the initial conditions are set for the problem. After the constraints have been read, they are normalized and stored. If equalities are indicated, these form the initial basis and the corresponding inverse is computed.

The second section of the program tests whether the initial guess of the vector x is feasible or not. If it is not, a feasible point is found by computing one x -correction so that all constraints in the current basis are satisfied. If, however, any constraint is still violated, it is considered as x -correction failure. A constraint in the basis is said to be violated if $|\lambda_1| > \epsilon_2$. If an x -correction has failed, a re-inversion is required. In case there is violation of constraints not in the basis ($\lambda_1 < -\epsilon_2$) the program tries to add to the basis the constraint corresponding to the most negative λ_1 , if that constraint is not linearly dependent and the procedure is repeated from x -correction. If the constraint is linearly dependent, the program tests against ϵ_3 for a constraint to drop. Failure of a constraint drop means that there is no feasible x . However, on steps other than the first step it only means an x -correction failure.

After finding a feasible x , subroutine FUNCT is called to compute $f(x)$ and $g(x)$. The program then enters the third phase, namely, the

step procedure. A step includes the projection of the gradient, testing for the maximum, changes in the basis, calculation of z and the updating of the matrix H_q . At the beginning of each step the program has a feasible x , $f(x)$ and $g(x)$. The non-basis constraints are classified for $v(|\lambda_i| \leq \epsilon_2)$ and $w(\lambda_i > \epsilon_2)$. The norm of the gradient is calculated and if $\|g\| \leq \epsilon_1$, it means 'maximum' has been reached.

If $\|g\| > \epsilon_1$ and there are no constraints in the basis, that is, $q = 0$, the program skips to compute z . If $q > 0$ and $\|Pg\| \leq \epsilon_1$ the program tests for a constraint to drop. If there is no constraint to drop, the maximum test is said to be satisfied and the current point is the maximum. If there is at least one constraint that can be dropped, the best one is dropped from the basis and the matrix H_{q-1} is computed using equation (3-22). The program returns to where the gradient is projected and tested. If $\|Pg\|$ is still zero, a re-inversion is required to continue.

If $\|Pg\| \geq \epsilon_1$ and $q < m$, the vector z is computed. If the current point is on constraints which are not in basis, that is, $v \neq 0$, the program computes and tests $z^T n_i$ for all i in v . If $\min(z^T n_i) < 0$, the corresponding constraint is added to the basis if it is linearly independent and H_{q+1} is computed. If the constraint is linearly dependent, test for negative $z^T n_i$, i in v , is repeated. When $v = 0$, test for the basis change is made. If q or $n = 0$ the program tests against $2\|H_q g\|$ for a constraint to drop. The program returns to the location where the gradient is projected and tested.

If the program finds no more changes to be made in the basis, the program uses the cubic interpolation scheme. First initial step length is determined and if $w = 0$, this is checked against constraints in w .

For all i in w for which $z^T n_i$ is negative, $\tau_i = -\lambda_i / z^T n_i$ is computed. If the minimum τ_i is less than the initial τ , the step is limited by the constraint H_i and τ is set equal to τ_i .

After interpolating, if $\gamma^i = \lambda^i$ and $(s^i)^T g^{i+1} \geq 0$ the constraint corresponding to the $\min\{\lambda_j\}$ (see step 3 of the algorithm) is added to the constrained basis and H_{q+1} is computed according to step 4. The step counter is tested against maximum step limit and if the limit is not reached, the program returns to the beginning of the step procedure. If, however, step 4 of the algorithm fails, H_q is updated according to the relation (3-25) and the program returns to the start of the step procedure.

4.3 Subroutines

Subroutine REINV is designed to compute the inverse matrix $(N_q^T N_q)^{-1}$ whereas subroutines COMP1 and COMP2 do the matrix computations required by the conjugate gradient algorithm. Subroutine AMDA calculates λ 's while CLASS classifies the constraints in to various categories, such as

u = linearly dependent constraint

v = constraints not in basis with $\lambda = 0$

w = constraints not in basis with $\lambda > 0$

All the above subroutines form a permanent part of the program. Another subroutine FUNCT is added to the program. It varies from problem to problem and is used to compute $f(x)$ and $g(x)$.

CHAPTER 5

APPLICATION OF THE CONJUGATE GRADIENT METHOD TO MANAGEMENT PROBLEMS

In this section three models relating to the optimization of management systems are considered. These models represent production, inventory and advertisement situations.

The first model is relatively a simpler production planning problem involving one state and one control variable. The second one is the well known diffusion model for advertisement involving two state variables and one control variable. At the end a highly non-linear production and advertisement scheduling problem, that involves six state variables and three control variable, is considered.

5.1 An Inventory Model

This model has been taken from [15]. Consider the case of a manufacturing concern where the sales rate $Q(t)$, at any time t , is known with certainty and that the rate of change of inventory level $I(t)$ is given by

$$\frac{dI(t)}{dt} = P(t) - Q(t) \quad (5-1)$$

where $P(t)$ is the production rate at time t . The problem, then, is to minimize the total cost, C_T , of inventory and production given by

$$C_T = \int_0^T [C_I(I_m - I(t))^2 + C_P \exp(P_m - P(T))^2] dt \quad (5-2)$$

Here C_P is the minimum production cost that occurs when the rate of production is equal to the production capacity, P_m , of the plant. An increase in production rate over the capacity may call for additional equipment and labor and can, therefore, be expensive. Similarly, a

decrease in production rate below the plant capacity can be equally expensive owing to the need for the maintenance of unused equipment and also because it would result in idle labor which cannot be reduced because of contract agreements. The quantity I_m may be considered as the storage capacity of the inventory and C_I , the cost of carrying inventory. Generally, the minimum storage cost occurs when the storage capacity is fully utilized.

The cost function given by (5-2) has a smoothing capability which is very much desirable for many manufacturing processes. In this case, I_m and P_m can be considered as desirable inventory and production levels. Now, let the sales rate, which is assumed to be known with certainty, be given by the linear equation

$$Q(t) = a + bt \quad (5-3)$$

and the initial inventory is

$$I(0) = c \quad (5-4)$$

where a , b and c are known constants. Substituting the value of $Q(t)$ in (5-1), we get

$$\frac{dI(t)}{dt} = P(t) - a - bt \quad (5-5)$$

To solve the problem by the conjugate gradient method, the above equations must be reduced to difference equations for digital computations. Thus, for small Δt , we obtain

$$I(t+\Delta t) = I(t) + [P(t) - a - bt] \Delta t \quad (5-6)$$

The integral in the cost equation (5-2) over the limits $k\Delta t$ and $(k+1)\Delta t$ can be approximated as follows

$$\begin{aligned}
& \int_{k\Delta t}^{(k+1)\Delta t} [C_I(I_m - I(t))^2 + C_P \exp(P_m - P(t))^2] dt \\
& = [C_I(I_m - I(t))^2 + C_P \exp(P_m - P(t))^2] \Delta t \quad (5-7)
\end{aligned}$$

The values of the various numerical quantities are assumed to be as follows

$$\begin{aligned}
a &= 2 & b &= 1 & c &= 5 \\
C_I &= 0.1 & C_P &= 0.001 & I_m &= 10 \\
P_m &= 5 & T &= 1
\end{aligned}$$

In addition, the production must not be more than 7 units.

5.1.1. Formulation of the Problem

Since the continuous process has to be approximated by a multistage process, the number of stages into which the process should be divided is influenced by the desired accuracy and computational cost. Obviously, the larger is the number of stages, the greater are the accuracy and computational cost. Let us assume that the process is divided into n stages. Since the conjugate gradient method maximizes the objective function, total cost C_T is minimized by maximizing $-C_T$. The problem may then be written as

maximize

$$F = -C_T = - \sum_{k=1}^n [C_I(I_m - \bar{I}(k\Delta t))^2 + C_P \exp(P_m - P(k\Delta t))^2] \Delta t \quad (5-8)$$

here $\bar{I}(k\Delta t)$ has been taken as the mean of the inventories at the k th stage. That is

$$\bar{I}(k\Delta t) = \frac{1}{2}[I(\overline{(k-1)\Delta t}) + I(k\Delta t)]$$

subject to

$$I(k\Delta t) = I(\overline{k-1}\Delta t) + [P(k\Delta t) - a - bk\Delta t]\Delta t \quad (5-9)$$

$$P(k\Delta t) \geq 0 \quad (5-10)$$

$$-P(k\Delta t) \geq -7 \quad (5-10a)$$

where $k = 1, 2, \dots, n$ and $\Delta t = T/n$

It is given that $I(0) = c$

Equation (5-9) represents n equality constraints on the state variable I , whereas (5-10) and (5-10a) represent 2 inequality constraints on each of the control variables P , the number of which equals the number of stages into which the process is divided, that is, n . These inequalities act as bounds on the control variables. Thus, in all there are $2n$ bounds on the control variables.

Since the conjugate gradient method does not recognize any state variables as such, the constraints given by equation (5-9) were eliminated, while calculating gradients, by evaluating the partial derivatives of the state variable with respect to the control variables at each stage, as given in Appendix D.

5.1.2. Numerical Results

For the purpose of this study, the problem was solved for 5-stage and 10-stage processes on an IBM 360/50 computer. Various limits and tolerances selected for the problem are as follows

$$\begin{aligned} \epsilon_1 &= 0.0002 & \text{LIMIT} &= 100 \\ \epsilon_2 &= 0.001 & \text{MXRN} &= 3 \\ \epsilon_3 &= 0.005 & \beta_{\max} &= 4 \\ \tau_{\max} &= 1.5 \end{aligned}$$

The subroutine FUNCT was so designed as to be independent of the number of stages into which the process is divided.

For the 5-stage process we have the number of variables, $m = 5$; the number of constraints, $k = 10$. Since all the constraints on the control variables are in the form of bounds, the number of bounds, $s = 10$ and the number of equality constraints, $e = 0$. Since, to start with, the program requires the initial guess of the control variables, three sets of initial approximations were used. They were: all $P_i = 1.00$, all $P_i = 4.00$ and all $P_i = 7.00$. Also, the estimated value of the minimum cost is required to be specified. It was taken as 1.0.

Using the three sets of initial approximations the problem was solved for both 5- and 10-stages. It was observed that for each process the optimal solution remained the same irrespective of the initial approximation used. In both the cases the initial guess of all $P_i = 7.00$ was found to be the best of the guesses used, and it required sufficiently smaller number of iterations as well as functional evaluations to reach the optimum, as compared to the other two approximations. A functional evaluation consists of an evaluation of the objective function and the corresponding gradients.

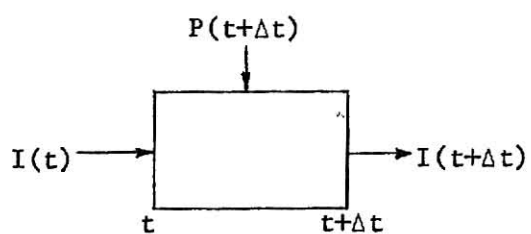
The optimal results for the two processes are summarized in Tables 1 and 2. The optimal values of production, inventory and cost are also shown in Figs. 1 and 2. The convergence rates of production and cost for the 5-stage and 10-stage processes are given in Tables 3 and 4, respectively. Table 5 gives the execution times for the different initial approximations. For the 5-stage process the minimum cost was found to be 0.940 and for the 10-stage process it was 0.932, showing a decrease of only 0.87%. The decrease is rather insignificant. This is due to the fact that the inventory and cost

functions are quite smooth, as can be seen from Figs. 1 and 2. The execution time for the 10-stage process, however, increased over the 5-stage process by about 278%. From Figs. 1 and 2 it can be observed that the cost function is more sensitive to the production rate in the initial stages than that in the later stages of the process.

From the manner in which the conjugate gradient method proceeded, it was observed that, to start with, there were no constraints in the basis. This was due to the absence of any equality constraint. Also, since all the initial approximations specified were feasible, no computer time was lost in finding a feasible point. It is, therefore, desirable to select a feasible point as the initial starting point. During the course of optimization, the constraints which became limiting were added to the basis. At the optimum, there were 2 active constraints for the 5-stage process and 4 active constraints for the 10-stage process. As can be seen from the optimum results, these correspond to the maximum allowable production rates at the initial stages of the two processes.

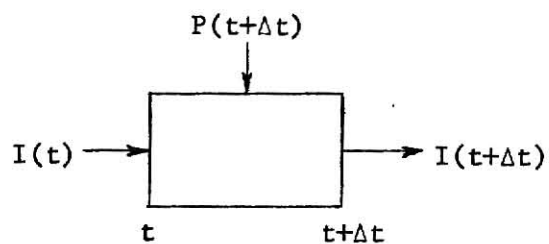
This model has also been solved by Nair [18] using Powell's conjugate gradient method which, incidently, is an unconstrained optimization technique. The minimum cost obtained by him for both 5-stage and 10-stage processes was 0.93. This agrees reasonably well with the present results. However, some difference was observed in the values of the control variables. This was due to the fact that Powell's method cannot handle any constraints.

Table 1. Optimal Results for the 5-Stage Process



t	$t+\Delta t$	$I(t)$	$P(t+\Delta t)$	$I(t+\Delta t)$	$Q(t+\Delta t)$	C_T
0.0	0.2	5.000	7.000	5.960	2.200	0.420
0.2	0.4	5.960	7.000	6.880	2.400	0.687
0.4	0.6	6.880	6.946	7.749	2.600	0.840
0.6	0.8	7.749	6.772	8.544	2.800	0.913
0.8	1.0	8.544	6.431	9.230	3.000	0.940

Table 2. Optimal Results for the 10-Stage Process



t	$t+\Delta t$	$I(t)$	$P(t+\Delta t)$	$I(t+\Delta t)$	$Q(t+\Delta t)$	C_T
0.0	0.1	5.000	7.000	5.490	2.100	0.232
0.1	0.2	5.490	7.000	5.970	2.200	0.419
0.2	0.3	5.970	7.000	6.440	2.300	0.569
0.3	0.4	6.440	7.000	6.900	2.400	0.685
0.4	0.5	6.900	6.972	7.347	2.500	0.773
0.5	0.6	7.347	6.899	7.777	2.600	0.836
0.6	0.7	7.777	6.809	8.188	2.700	0.879
0.7	0.8	8.188	6.691	8.577	2.800	0.907
0.8	0.9	8.577	6.517	8.939	2.900	0.924
0.9	1.0	8.939	6.165	9.255	3.000	0.932

Table 3 Convergence Rates of Production and Cost for the 5-Stage Process

Iteration No.	No. of Functional Evaluations	No. of Constraints in Basis	P ₁	P ₂	P ₃	P ₄	P ₅	C _T
0	1	0	1.000	1.000	1.000	1.000	1.000	8889.387
1	5	0	6.983	6.983	6.983	6.983	6.983	0.949
2	6	0	7.000	6.989	6.981	6.975	6.971	0.948
3	7	0	7.000	7.000	6.983	6.971	6.963	0.947
5	9	3	7.000	7.000	7.000	6.974	6.942	0.947
10	16	2	7.000	7.000	6.941	6.994	6.923	0.946
20	26	2	7.000	7.000	6.941	6.992	6.919	0.946
30	42	2	7.000	7.000	6.947	6.759	6.369	0.940
49	72	2	7.000	7.000	6.946	6.771	6.432	0.940
0	1	0	4.000	4.000	4.000	4.000	4.000	1.799
1	2	0	4.226	4.170	4.118	4.070	4.024	1.734
2	3	0	4.776	4.584	4.405	4.238	4.082	1.582
3	4	0	6.066	5.552	5.075	4.632	4.216	1.265
5	6	1	7.000	6.449	5.693	4.992	4.339	1.055
10	11	2	7.000	7.000	6.948	5.701	4.575	0.963
20	22	3	7.000	7.000	7.000	6.799	5.131	0.944
27	32	2	7.000	7.000	6.947	6.771	6.432	0.940
0	1	0	7.000	7.000	7.000	7.000	7.000	0.949
1	2	2	7.000	7.000	6.984	6.957	6.940	0.946
2	3	2	7.000	7.000	6.956	6.879	6.831	0.942
3	4	2	7.000	7.000	6.922	6.778	6.684	0.940
5	7	2	7.000	7.000	6.913	6.716	6.574	0.940
10	13	2	7.000	7.000	6.965	6.743	6.465	0.940
13	17	2	7.000	7.000	6.947	6.771	6.429	0.940

Table 4 Convergence Rates of Production and Cost for the 10-Stage Process

Iteration No.	No. of Functional Evaluations	No. of Constraints in Basis	P ₁	P ₄	P ₇	P ₁₀	C _T
0	1	0	1.000	1.000	1.000	1.000	8889.348
1	5	0	6.977	6.977	6.977	6.977	0.943
2	6	0	7.000	6.981	6.969	6.962	0.941
3	7	1	7.000	6.984	6.967	6.958	0.941
5	9	3	7.000	7.000	6.963	6.944	0.940
10	14	6	7.000	7.000	6.977	6.915	0.939
20	25	5	7.000	7.000	6.799	6.894	0.938
30	39	4	7.000	7.000	6.801	6.889	0.938
50	64	4	7.000	7.000	6.808	6.549	0.933
76	95	4	7.000	7.000	6.810	6.144	0.932
0	1	0	4.000	4.000	4.000	4.000	1.778
1	2	0	4.119	4.078	4.040	4.006	1.745
2	3	0	4.413	4.268	4.139	4.022	1.667
3	4	0	5.125	4.730	4.377	4.058	1.488
5	6	0	7.000	5.944	5.003	4.156	1.092
10	11	3	7.000	6.653	5.357	4.210	1.001
20	21	6	7.000	7.000	6.329	4.355	0.947
30	34	6	7.000	7.000	6.820	4.601	0.936
56	67	4	7.000	7.000	6.809	6.166	0.932
0	1	0	7.000	7.000	7.000	7.000	0.943
1	2	4	7.000	7.000	6.981	6.968	0.941
2	3	4	7.000	7.000	6.939	6.900	0.938
3	4	4	7.000	7.000	6.866	6.779	0.935
5	7	4	7.000	7.000	6.735	6.550	0.933
10	12	4	7.000	7.000	6.776	6.416	0.932
20	24	4	7.000	7.000	6.822	6.261	0.932
32	38	4	7.000	7.000	6.809	6.171	0.932

Table 5 - Execution Times for the 5-stage and 10-stage Processes

Initial Approximation	5-Stage Process		10-Stage Process	
	No. of Iterations	Execution Time in Secs	No. of Iterations	Execution Time in Secs
All $P_i = 1.00$	49	77.31	76	188.52
All $P_i = 4.00$	27	42.60	56	138.90
All $P_i = 7.00$	13	20.50	32	79.37

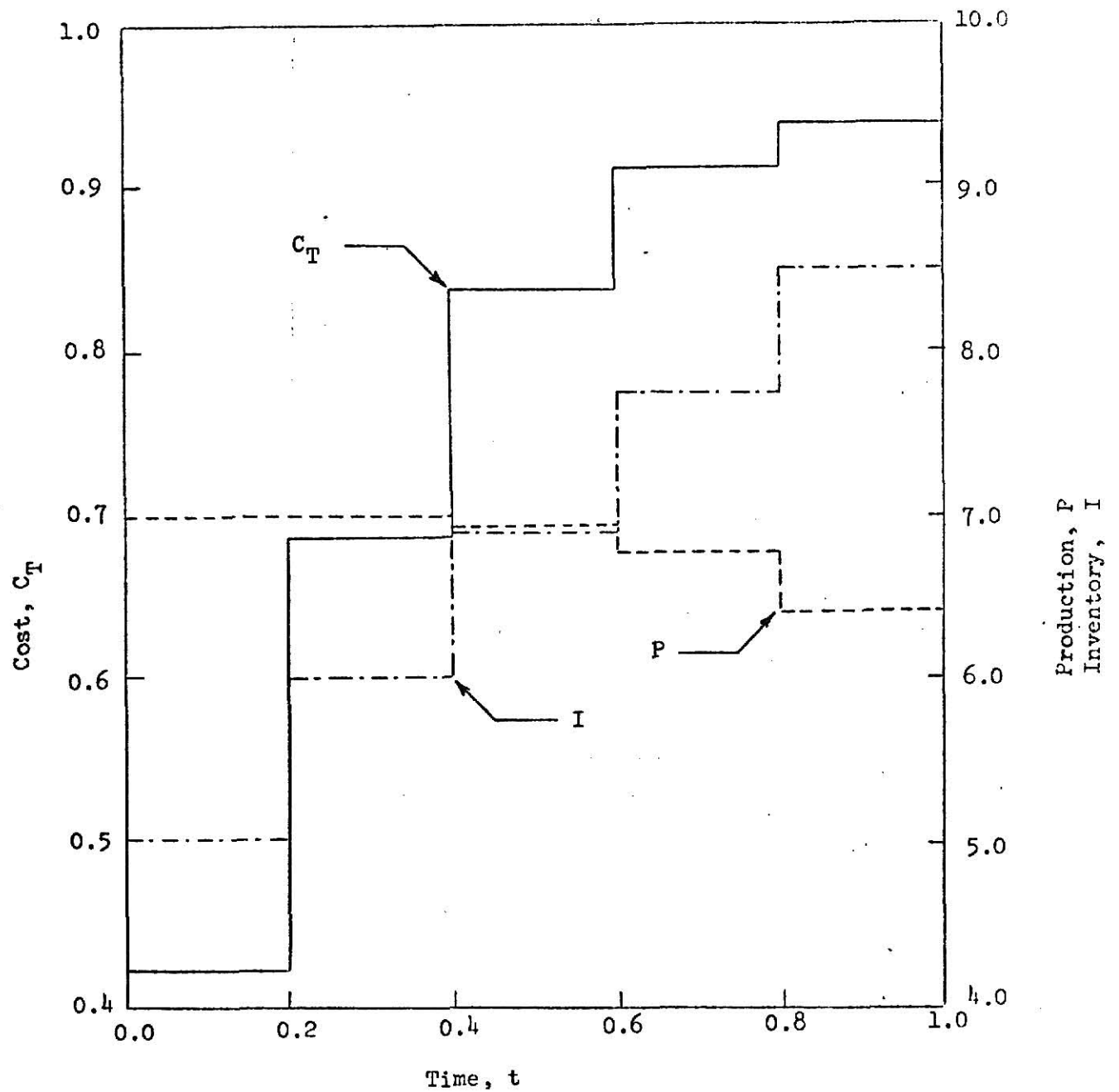


Fig. 1 Optimal Cost, Production and Inventory for the 5-Stage Process

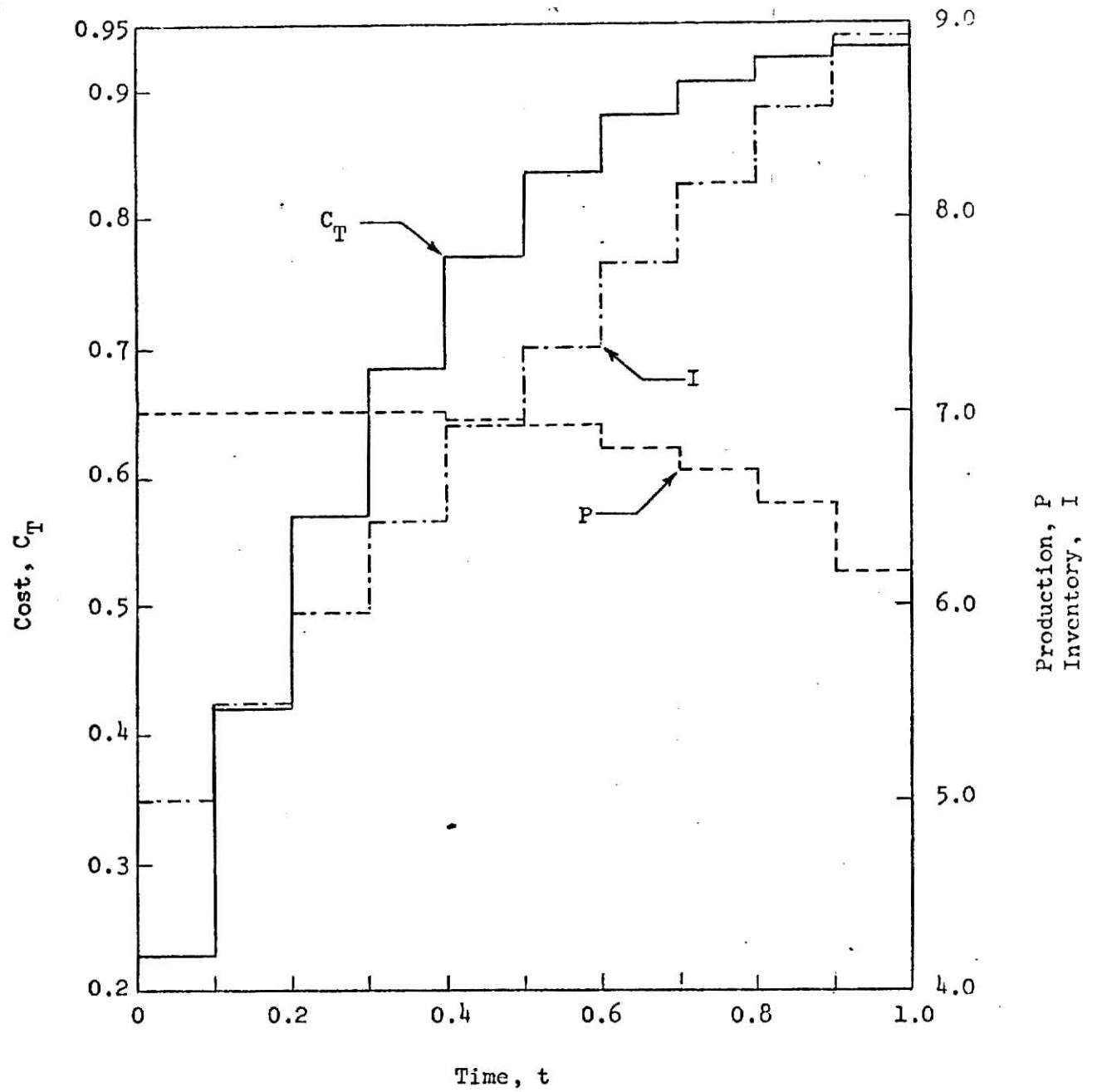


Fig. 2 Optimal Cost, Production and Inventory for the 10-Stage Process

5.2 An Inventory and Advertisement Model

Let us now use the conjugate gradient method to solve an advertising problem. The model has been taken from [18]. This model is more complex as compared to the one discussed earlier in that it involves two state variables. The block diagram of the model is shown in Fig. 3. Consider a group of potential customers of a company's product. Assume that the total number of persons in the group remains constant and that only a certain members of the group possess some information about the product. Assume further that diffusion of information within the group occurs through personal contact.

Let

$K(0)$ = number of informed persons at time t_0

$K(t)$ = number of informed persons at time t

N = total number of persons in the group

c = contact coefficient, the number of contacts made by one informed person per unit of time

Then $K(t)/N$ = proportion of informed persons at time t , and

$1-K(t)/N$ = proportion of uninformed persons at time t .

Since the number of informed persons can be increased only by making contacts with uninformed persons, the rate of increase in the number of informed persons at any time t is given by

$$\frac{dK(t)}{dt} = cK(t)[1 - K(t)/N] \quad (5-11)$$

Suppose that the firm can increase the number of contacts made by advertising by an additional number A per unit of time. Then equation (5-11) becomes

$$\frac{dK(t)}{dt} = (c+A(t)) K(t)[1 - K(t)/N] \quad (5-12)$$

If each successful contact increases the sale by one unit of the firm's product per unit time, then

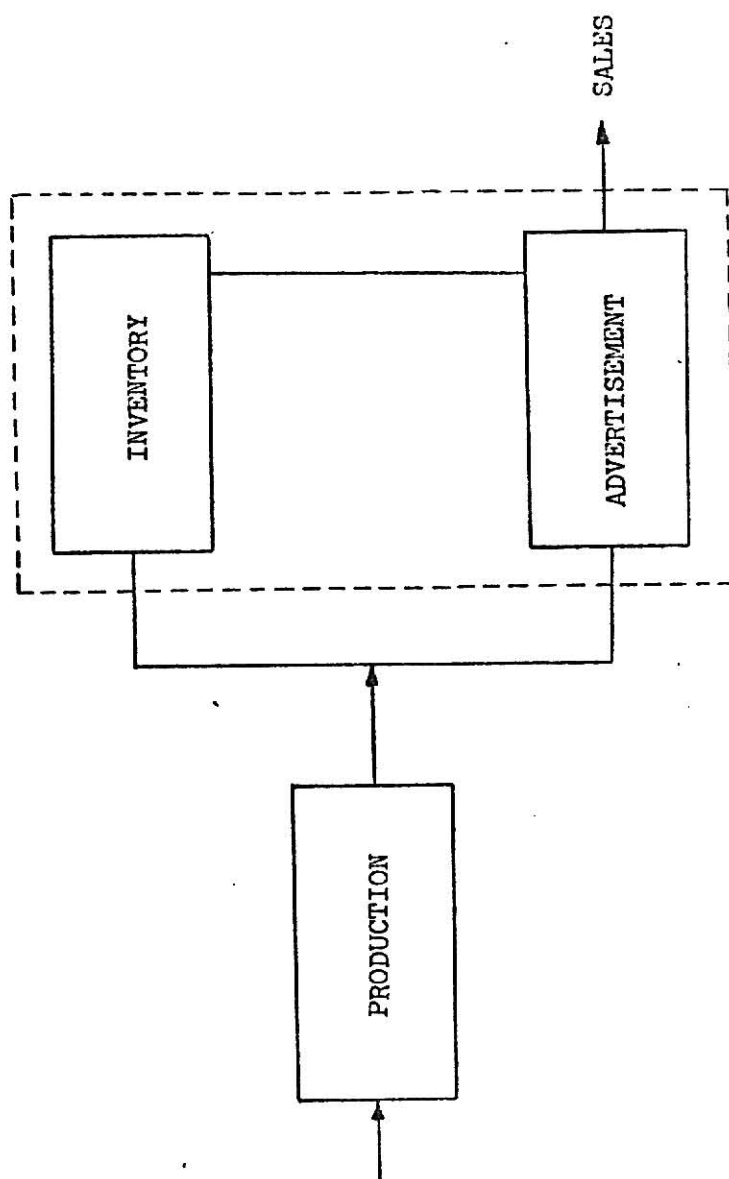


Fig. 3 Block Diagram of an Inventory and Advertisement Model

$$\frac{dQ(t)}{dt} = (c+A(t)) Q(t)[1 - Q(t)/N] \quad (5-13)$$

where $Q(t)$ represents the sales at time t .

The rate of change of company's inventory is given by

$$\frac{dI(t)}{dt} = P(t) - Q(t) \quad (5-14)$$

where $I(t)$ = inventory at time t , and

$P(t)$ = production rate at time t .

The production rate for the product under consideration is assumed to be given by the linear equation

$$P(t) = a + bt \quad (5-15)$$

where a and b are known constants. The company's management wishes to maximize the net profit, P_T , which can be obtained from the equation:

$$\text{Net Profit} = \text{Revenue} - \text{Cost of holding inventory} - \\ \text{Cost of advertising.}$$

In mathematical terms

$$P_T = \int_0^T [F Q(t) - C_I(I_m - I(t))^2 - C_A A(t) Q(t)] dt \quad (5-16)$$

where F is the revenue from the sale of one unit of the product, I_m is the optimal inventory level, C_I is the inventory carrying cost and C_A is the cost of advertising.

In this model, $I(t)$ and $Q(t)$ are the two state variables and $A(t)$ is the control variable.

In order to solve this problem by the conjugate gradient method, equations (5-13), (5-14) and (5-16) must be approximated by difference equations. Equation (5-13), then, can be written as

$$Q(t+\Delta t) = Q(t) + \dot{Q}(t)(c + A(t))[1 - \frac{Q(t)}{N}] \Delta t \quad (5-17)$$

Since it has been assumed that the number of persons in the group remains constant and that each successful contact increases the sale by one unit, $Q(t+\Delta t)$ should not exceed N for any value of t . In order to

avoid the possibility of $Q(t+\Delta t)$ exceeding N , the term $[1-Q(t)/N]$ should be replaced by $[1-Q(t+\Delta t)/N]$. Incorporating the change in equation (5-14) and rearranging the terms, we have

$$Q(t+\Delta t) = \frac{Q(t)[1+(c+A(t))\Delta t]}{1+(c+A(t))Q(t)\Delta t} \quad (5-18)$$

Equation (5-14) becomes

$$I(t+\Delta t) = I(t) + (P(t) - Q(t))\Delta t \quad (5-19)$$

The integral in the profit equation (5-16) over the limits $k\Delta t$ and $(k+1)\Delta t$ may be given by

$$\begin{aligned} & \int_{k\Delta t}^{(k+1)\Delta t} [FQ(t) - C_I(I_m - I(t))^2 - C_A A(t)Q(t)] dt \\ &= [FQ(t) - C_I(I_m - I(t))^2 - C_A A(t)Q(t)]\Delta t \end{aligned} \quad (5-20)$$

The values of the various constants are assumed to be as follows

$$\begin{aligned} a &= 70 & b &= 100 & c &= 2 \\ F &= 10 & C_I &= 0.15 & I_m &= 50 \\ C_A &= 1.5 & N &= 150 & T &= 1 \\ I(0) &= 20 & Q(0) &= 20 \end{aligned}$$

Because of the limitations of the funds, restriction has been imposed on the advertisement, which at any time should not be more than 6.

5.2.1 Formulation of the Problem

Let n be the number of stages into which the process is divided. The problem may, then, be written as

maximize

$$P_T = \sum_{k=1}^n [FQ(k\Delta t) - C_I(I_m - I(k\Delta t))^2 - C_A A(k\Delta t)Q(k\Delta t)]\Delta t \quad (5-21)$$

subject to

$$Q(k\Delta t) = \frac{Q(\overline{k-1}\Delta t)[1+(c+A(k\Delta t))\Delta t]}{1+(c+A(k\Delta t))Q(\overline{k-1}\Delta t)\Delta t} \quad (5-22)$$

$$I(k\Delta t) = I(\overline{k-1}\Delta t) + [P(\overline{k-1}\Delta t) - Q(\overline{k-1}\Delta t)]\Delta t \quad (5-23)$$

$$A(k\Delta t) > 0 \quad (5-24)$$

$$\text{and } -A(k\Delta t) \geq -6 \quad (5-24a)$$

where $k = 1, 2, \dots, n$ and $\Delta t = T/n$.

It is given that $I(0) = 20$ and $Q(0) = 20$.

On each of the two state variables Q and I , there are n equality constraints given by equations (5-22) and (5-23), respectively. Also, there are n control variables A , each being subject to two inequality constraints given by equation (5-24) and (5-24a), which act as bounds on the control variables. As in the first problem, here too, we have to eliminate the state variable constraints during gradient calculations by obtaining the partial derivatives of Q and I with respect to A at each stage, as given in Appendix D.

5.2.2 Numerical Results

The problem was solved for 5-stage and 10-stage processes. As in the inventory model, the subroutine FUNCT was designed to be independent of the number of stages involved.

Limits and tolerances used in the main program for this problem are given below

$$\begin{aligned} \epsilon_1 &= 0.005 & \text{LIMIT} &= 100 \\ \epsilon_2 &= 0.001 & \text{MXRN} &= 3 \\ \epsilon_3 &= 0.005 & \beta_{\max} &= 3 \\ \tau_{\max} &= 1.0 \end{aligned}$$

Again, considering the 5-stage process, we have the number of variables, $m = 5$; the number of constraints, $k = 10$. Since the constraints are in the form of bounds, the number of bounds $s = 10$ and there are no equality constraints, $e = 0$. The estimated value of the maximum profit was taken as 500.

Using the limits and tolerances given above, the original conjugate gradient program was tried for the 5-stage process. The program gave an execution error, after some iterations, while updating the matrix operator H_q . This was due to the fact that movement in the conjugate direction failed to improve further the values of the control variables, as a result of which $(\sigma^i)^T y^i$ in equation (3-25) became zero. A modification was therefore made in the computer program so as to change the direction of search to steepest ascent and then to switch over to the newly generated conjugate directions. It was also observed that during some of the iterations the initial steps were negative. Since it is always possible to take initially a positive step during each iteration [5], and, moreover, since the program checks the initial step against the constraints, another modification was made in selecting the initial step size. The manner in which the initial step size is selected is given in Appendix B.

Three different initial points were used for both 5-stage and 10-stage processes. They were: all $A_i = 1.00$, all $A_i = 4.00$ and all $A_i = 6.00$. In each case the problem converged to the same optimal values irrespective of the starting point. Values of inventory, production, advertisement and profit at optimal for the 5-stage and 10-stage processes are given in Tables 6 and 7 and Figs. 4 through 7.

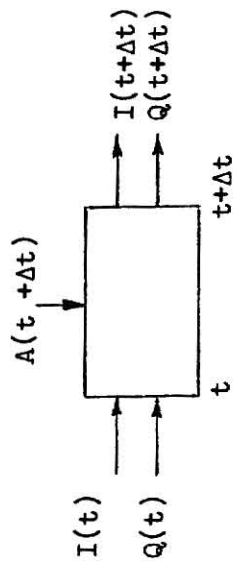
Convergence rates of advertisement and profit for the two processes, along with the number of functional evaluations required, are given in Tables 8 and 9, whereas the execution times for different initial approximations are given in Table 10. The maximum profit for the 5-stage process was found to be 609.088. The profit increased to 679.326 for the 10-stage process, an increase of about 11.5%. This was achieved for an increase of about 110% in the average execution time.

For both 5- and 10-stage processes, the value of A for the first stage is 6.000, the maximum allowable upper limit. The advertisement tapers off to zero as we reach the last stage. The trend of advertisement is as was expected. Maximum advertisement is needed in the initial stages to boost the sales. As the product picks up sales, the need for advertisement decreases.

As in the first problem, here too, since there were no equality constraints, the program started with no constraints in the basis, and the matrix operator H_0^0 was taken as the identity matrix. However, in the final solution, there were 4 active constraints for the 5-stage process and 8 active constraints for the 10-stage process as can be seen from Tables 8 and 9.

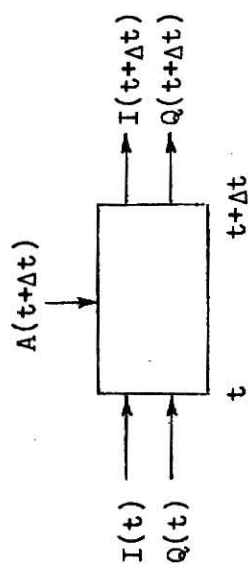
This problem has also been solved by Nair [18] using Powell's conjugate gradient method. The maximum profits obtained by him for the 5-stage and 10-stage processes were 607.62 and 683.50, respectively. However, since Powell's method cannot handle any constraints, some difference was observed between the optimal values of the control variables as obtained by the present method and those obtained by Powell's method.

Table 6 Optimal Results for the 5-Stage Process



t	$t+\Delta t$	$I(t)$	$Q(t)$	$P(t)$	$I(t+\Delta t)$	$Q(t+\Delta t)$	$A(t+\Delta t)$	P_T
0.0	0.2	20.000	20.000	70.000	30.000	42.857	6.000	-3.429
0.2	0.4	30.000	42.857	90.000	39.429	71.129	4.273	44.300
0.4	0.6	39.429	71.129	110.000	47.203	87.700	0.805	198.293
0.6	0.8	47.203	87.700	130.000	55.663	99.508	0.000	396.347
0.8	1.0	55.663	99.508	150.000	65.761	110.097	0.000	609.088

Table 7. Optimal Results for the 10-Stage Process



t	t+Δt	I(t)	Q(t)	P(t)	I(t+Δt)	Q(t+Δt)	A(t+Δt)	P _T
0.0	0.1	20.000	20.000	70.000	25.000	32.530	6.000	-6.122
0.1	0.2	25.000	32.530	80.000	29.747	49.897	6.000	-7.285
0.2	0.3	29.747	49.897	90.000	33.757	70.937	6.000	-4.149
0.3	0.4	33.757	70.937	100.000	36.664	86.092	3.014	40.351
0.4	0.5	36.664	86.092	110.000	39.054	94.023	0.469	125.966
0.5	0.6	39.054	94.023	120.000	41.652	100.259	0.000	225.180
0.6	0.7	41.652	100.259	130.000	44.626	106.124	0.000	330.871
0.7	0.8	44.626	106.124	140.000	48.014	111.563	0.000	442.374
0.8	0.9	48.014	111.563	150.000	51.858	116.540	0.000	558.862
0.9	1.0	51.858	116.540	160.000	56.203	121.040	0.000	679.325

Table 8 Convergence Rates of Advertisement and Profit for the 5-Stage Process

Iteration No.	No. of Functional Evaluations	No. of Constraints in Basis	A_1	A_2	A_3	A_4	A_5	P_T
0	1	0	1.000	1.000	1.000	1.000	1.000	442.991
1	2	0	2.683	2.169	1.504	0.755	0.032	540.194
2	3	0	2.719	2.192	1.511	0.742	0.000	542.179
3	4	1	4.782	3.511	1.859	0.000	0.000	595.451
4	5	2	6.000	3.952	1.477	0.000	0.000	607.666
5	7	3	6.000	3.985	0.871	0.000	0.000	608.911
6	9	3	6.000	4.273	0.798	0.000	0.000	609.087
7	11	3	6.000	4.273	0.805	0.000	0.000	609.088
0	1	0	4.000	4.000	4.000	4.000	4.000	334.882
1	2	0	4.118	3.068	1.811	0.717	0.000	577.126
2	3	1	4.882	3.426	1.675	0.000	0.000	586.673
3	4	2	6.000	3.893	1.442	0.000	0.000	607.777
4	6	3	6.000	3.986	0.853	0.000	0.000	608.908
5	8	3	6.000	4.274	0.800	0.000	0.000	609.087
6	10	3	6.000	4.274	0.802	0.000	0.000	609.087
7	12	3	6.000	4.273	0.804	0.000	0.000	609.088
0	1	0	6.000	6.000	6.000	6.000	6.000	84.199
1	2	0	4.436	2.988	1.508	0.505	0.000	583.768
2	3	1	5.026	3.312	1.484	0.000	0.000	597.985
3	4	2	6.000	3.788	1.387	0.000	0.000	607.869
4	6	3	6.000	4.013	0.808	0.000	0.000	608.928
5	8	3	6.000	4.273	0.805	0.000	0.000	609.087
6	10	3	6.000	4.273	0.805	0.000	0.000	609.087

Table 9 Convergence Rates of Advertisement and Profit for the 10-Stage Process

Iteration No.	No. of Functional Evaluations	No. of Constraints in Basis	A_1	A_4	A_7	A_{10}	P_T
0	1	0	1.000	1.000	1.000	1.000	466.694
1	2	0	1.876	1.510	0.987	0.476	529.163
2	3	0	2.521	1.864	0.919	0.000	571.683
3	4	1	2.961	2.090	0.840	0.000	594.074
5	6	3	5.375	3.111	0.000	0.000	661.935
10	12	7	6.000	3.093	0.000	0.000	679.318
14	20	8	6.000	3.015	0.000	0.000	679.326
0	1	0	4.000	4.000	4.000	4.000	352.134
1	2	0	4.198	2.837	1.343	0.624	598.199
2	3	0	4.495	2.866	1.024	0.000	629.478
3	4	1	4.647	2.888	0.881	0.000	638.843
5	6	3	5.629	3.042	0.000	0.000	665.011
10	12	7	6.000	3.098	0.000	0.000	679.316
14	20	8	6.000	3.015	0.000	0.000	679.326
0	1	0	6.000	6.000	6.000	6.000	74.240
1	2	0	4.392	2.180	0.493	0.000	631.315
2	3	1	4.480	2.221	0.458	0.000	635.449
3	4	2	4.727	2.333	0.358	0.000	643.809
5	6	4	6.000	2.808	0.000	0.000	669.810
10	13	8	6.000	3.014	0.000	0.000	679.325

Table 10 - Execution Times for the 5-stage and 10-stage Processes

Initial Approximation	5-Stage Process		10-Stage Process	
	No. of Iterations	Execution Time in Secs	No. of Iterations	Execution Time in Secs
All $A_i = 1.00$	7	26.46	14	69.63
All $A_i = 4.00$	7	26.46	14	69.63
All $A_i = 6.00$	6	22.68	10	49.73

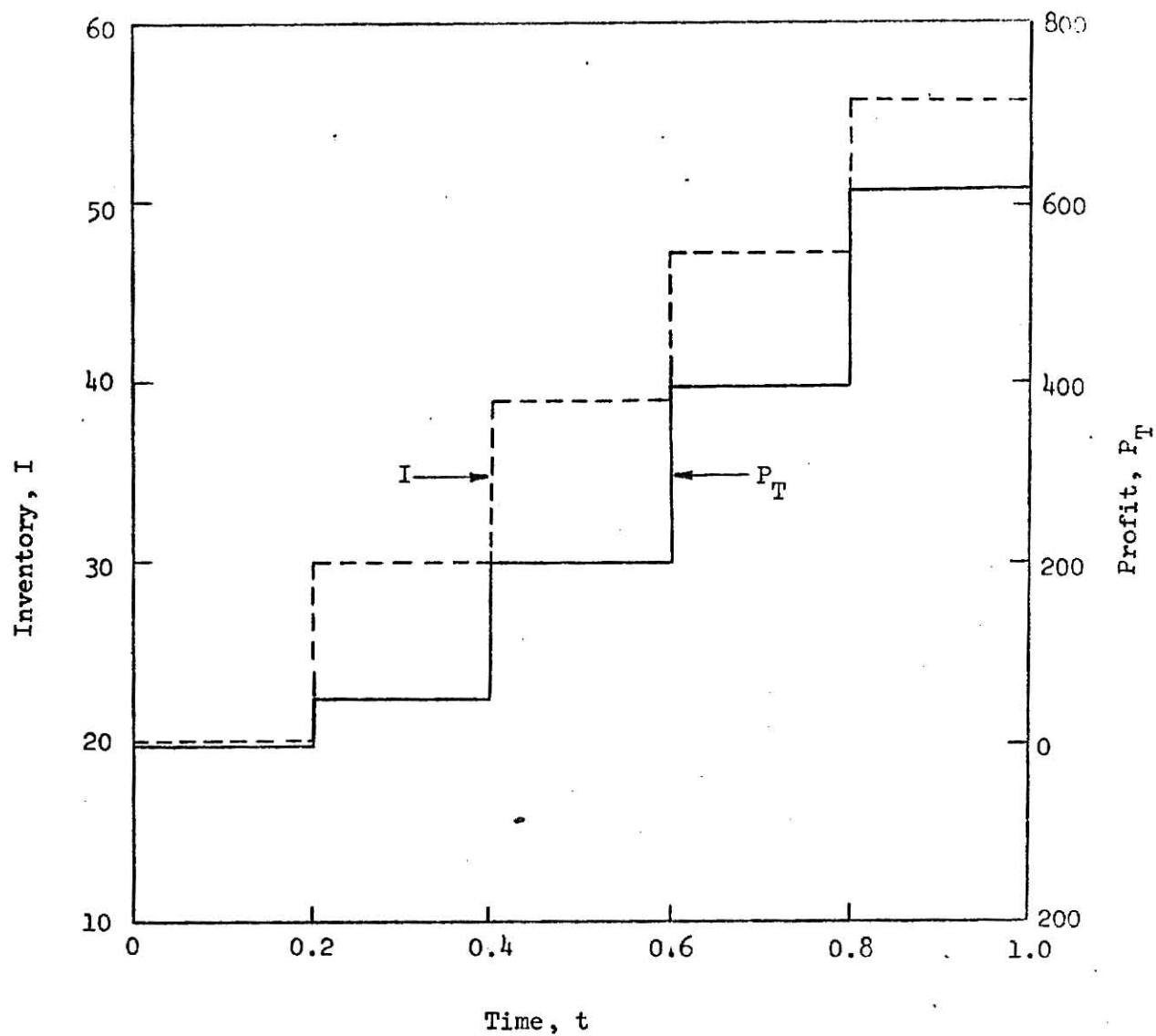


Fig. 4 Optimal Inventory and Profit for the 5-Stage Process

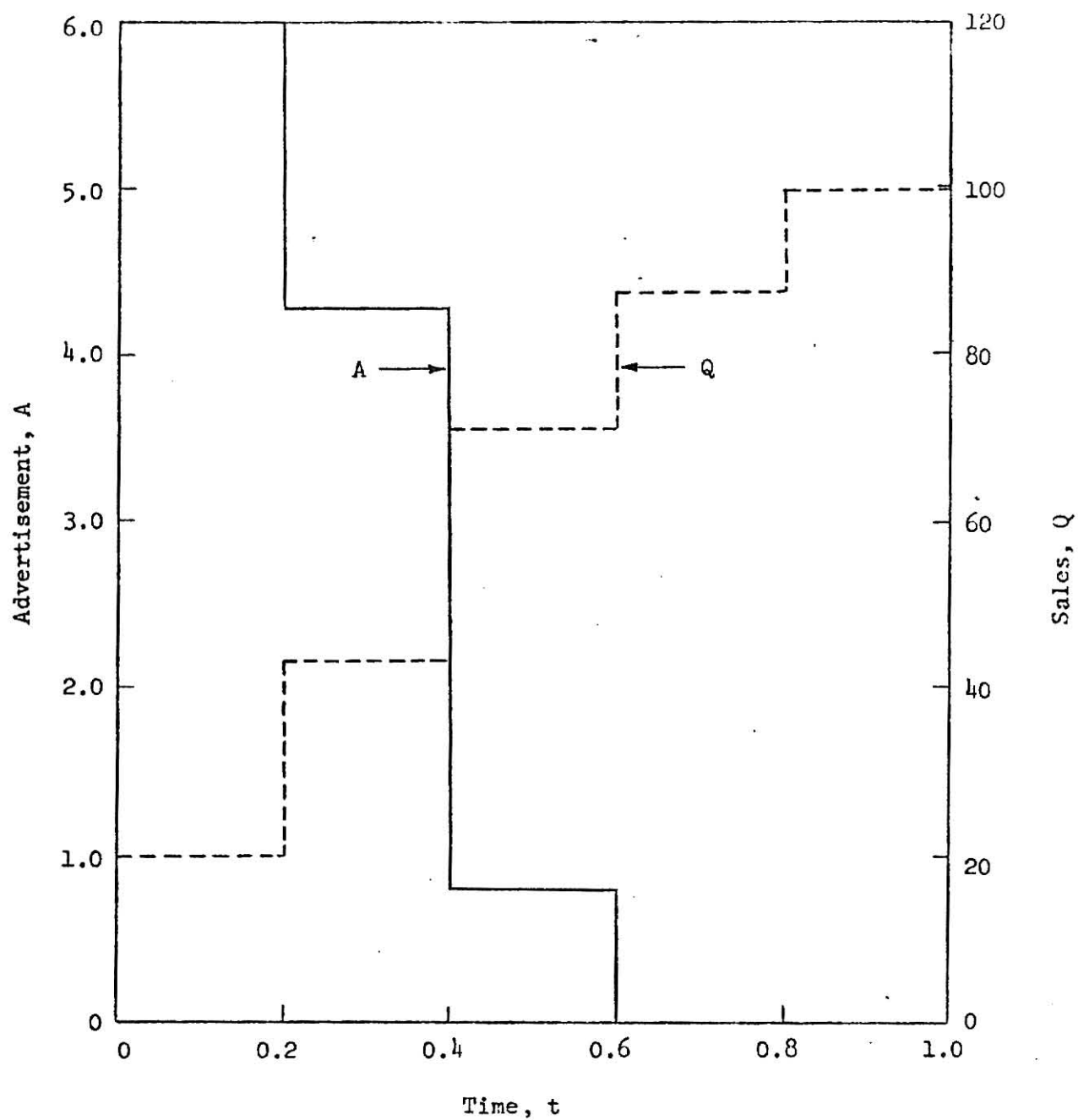


Fig. 5 Optimal Advertisement and Sales for the 5-Stage Process

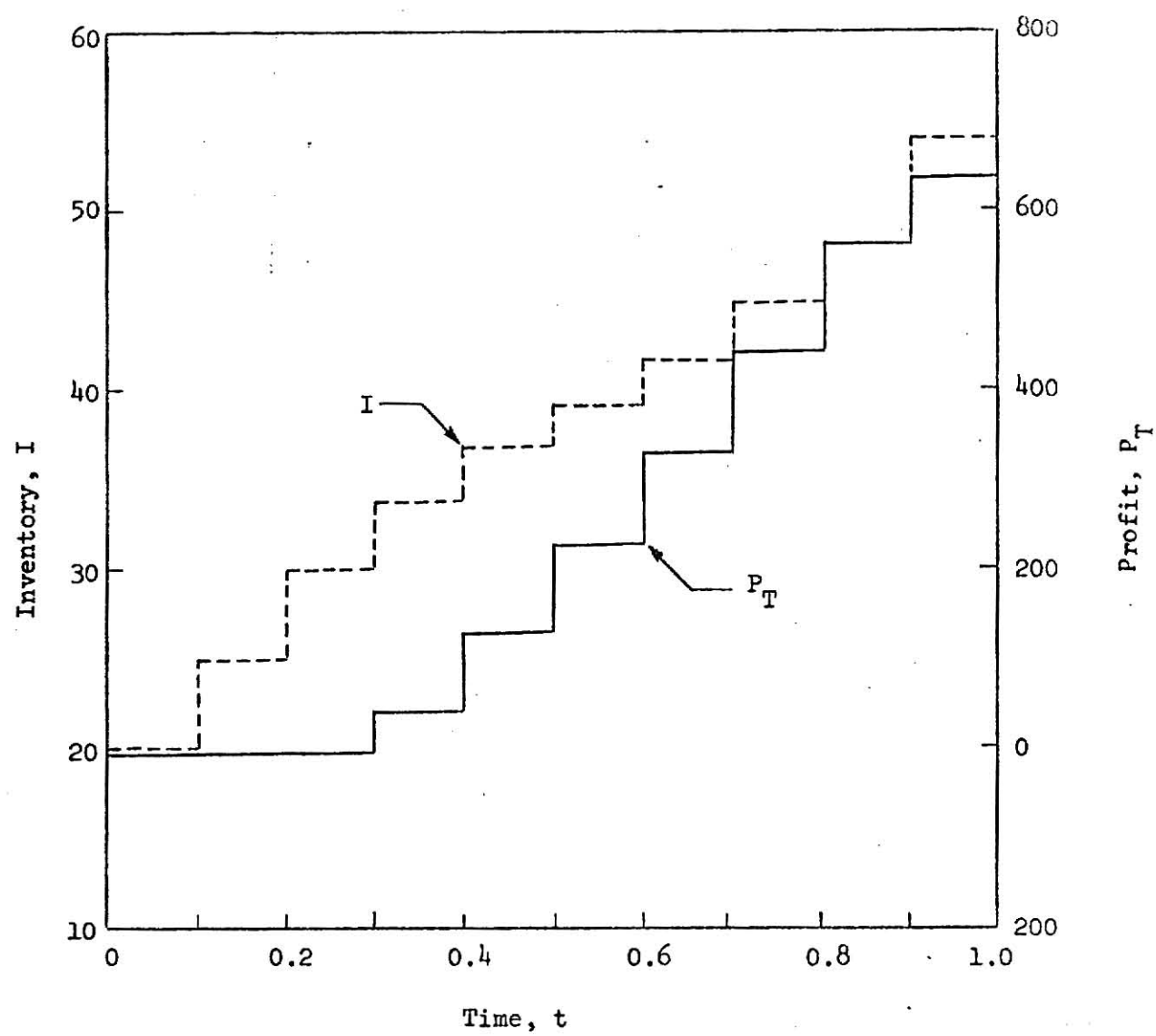


Fig. 6 Optimal Inventory and Profit for the 10-Stage Process

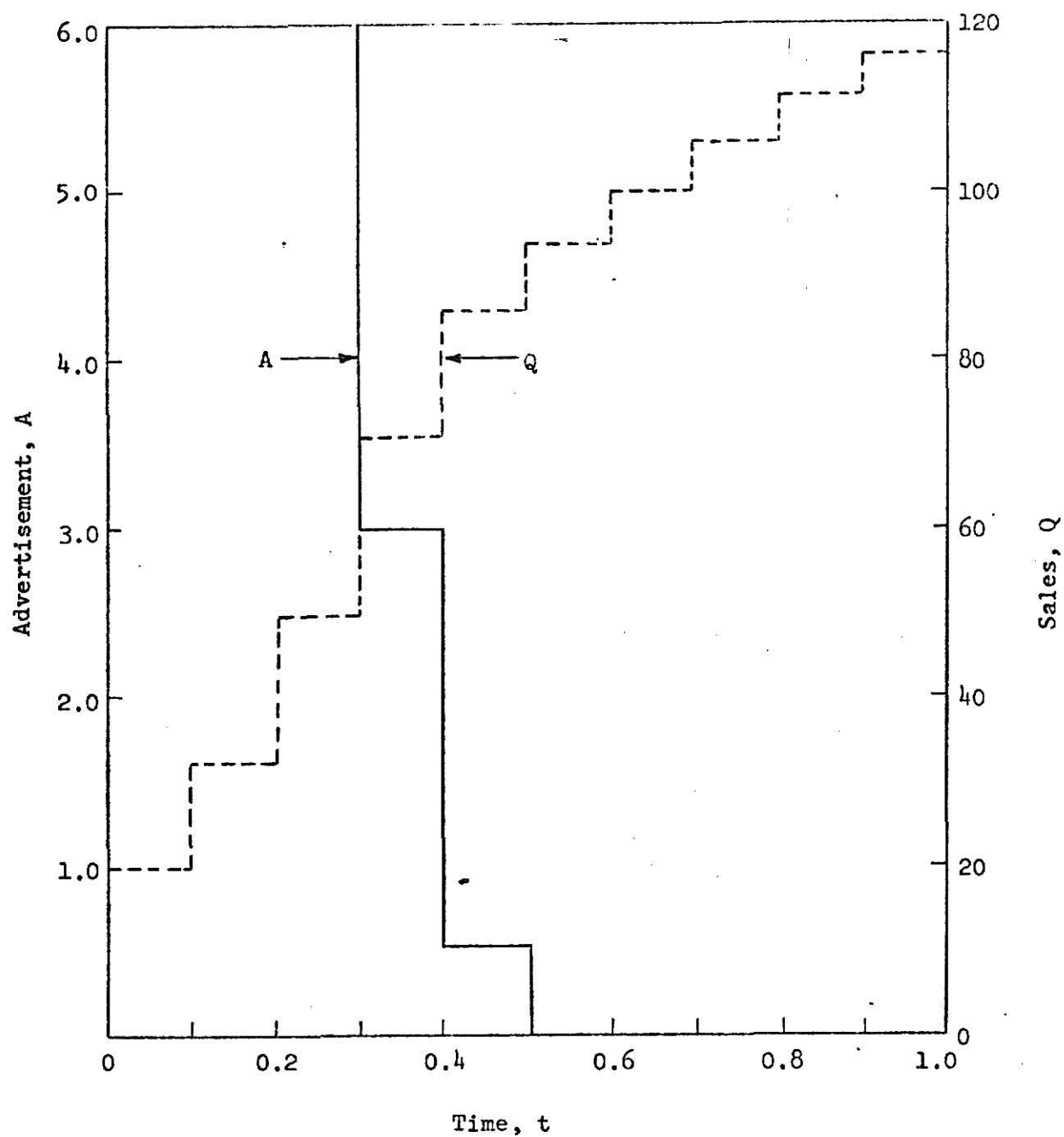


Fig. 7 Optimal Advertisement and Sales for the 10-Stage Process

5.3 A Production and Advertisement Model

We will now solve a more complex problem, namely, a production and advertisement model involving six state variables and three control variables. The model has been taken from [14]. Suppose that there are two chemical reactors processing a raw material containing three components A, B and C. Let the chemical reactions that take place in the reactors be represented as follows:



Component B is the most desirable product whereas C is the least valuable. Furthermore, B can be produced from A. However, certain amount of B also decomposes into C which is undesirable. Consider that B is a new product and therefore, needs advertisement to boost its sale. Also, to protect against fluctuations in demand, an inventory is assumed for B. A and C are assumed to have unlimited market at fixed price and they can be sold as soon as manufactured. Thus, no inventory for the raw material and products A and C is considered. The amount of B produced and that decomposed into C can be controlled by controlling the reaction temperatures. The block diagram of the process is shown in Fig. 8.

Let

x_i = concentration of A in reactor i , $i = 1, 2$

y_i = concentration of B in reactor i

v_i = volume of reactor i

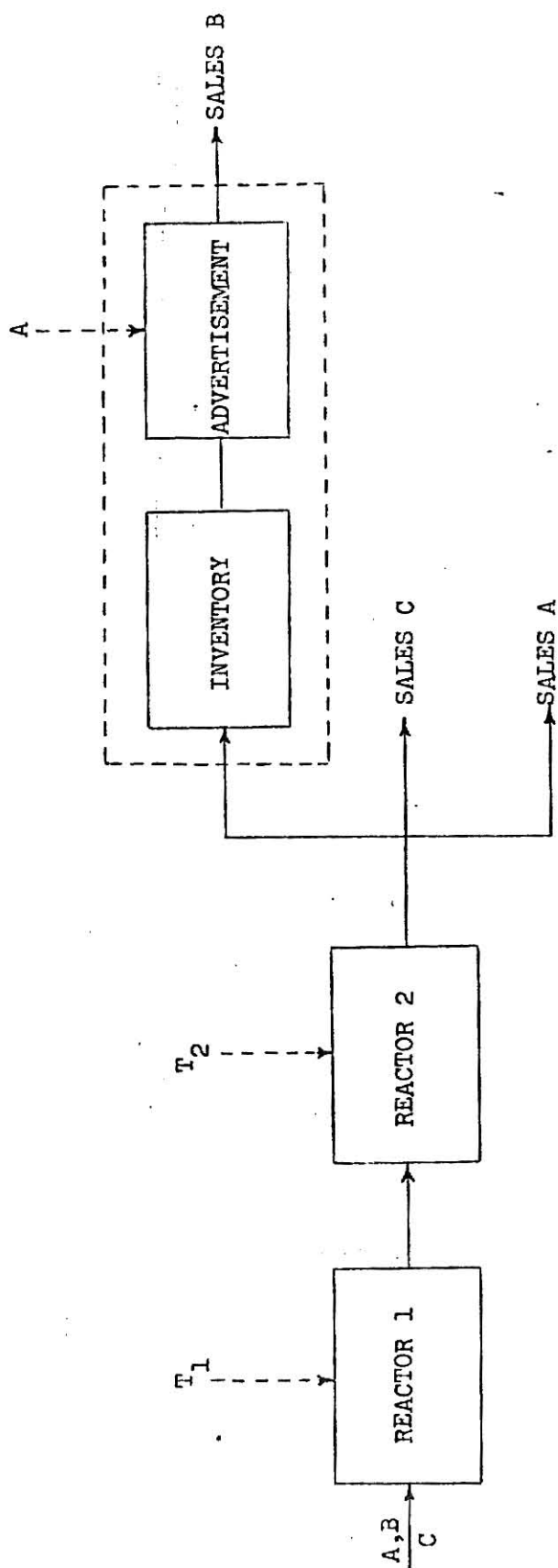


Fig. 8 Block Diagram of a Production and Advertisement Model

q = flow rate

k_{a1} = reaction rate constant of A in reactor 1

k_{b1} = reaction rate constant of B in reactor 1

G_a, G_b = frequency factor constants for A and B, respectively

E_a, E_b = activation energies of reactions

R = gas constant, and

T_i = temperature in reactor i

Under the steady state conditions, the production rates in the two reactors are given by

$$v_1 \frac{dx_1}{dt} = q(x_0 - x_1) - v_1 k_{a1} x_1 \quad (5-25)$$

$$v_1 \frac{dy_1}{dt} = q(y_0 - y_1) - v_1 k_{b1} y_1 + v_1 k_{a1} x_1 \quad (5-26)$$

$$v_2 \frac{dx_2}{dt} = q(x_1 - x_2) - v_2 k_{a2} x_2 \quad (5-27)$$

$$v_2 \frac{dy_2}{dt} = q(y_1 - y_2) - v_2 k_{b2} y_2 + v_2 k_{a2} x_2 \quad (5-28)$$

where x_0 and y_0 are the raw material concentrations of A and B, respectively.

The reaction rate constants are defined as

$$k_{a1} = G_a \exp(-E_a/RT_1) \quad (5-29)$$

$$k_{b1} = G_b \exp(-E_b/RT_1) \quad (5-30)$$

$$k_{a2} = G_a \exp(-E_a/RT_2) \quad (5-31)$$

$$k_{b2} = G_b \exp(-E_b/RT_2) \quad (5-32)$$

As indicated earlier, B is the desired product for which inventory and advertisement are assumed. The performance equation for inventory, $I(t)$, at any time t is given by

$$\frac{dI(t)}{dt} = qy_2 - Q(t) \quad (5-33)$$

where the term qy_2 represents the production rate of B and $Q(t)$ represents the sales rate. The performance equation for the sales rate is obtained as discussed in problem 2. It is given by

$$\frac{dQ(t)}{dt} = Q(t)(c + A(t))(1 - \frac{Q(t)}{N}) \quad (5-34)$$

where c is the contact coefficient, $A(t)$ is the advertisement and N is the total number of persons in the group.

Equations (5-25) through (5-34) describe the system. We have six state variables, x_1, x_2, y_1, y_2, I and Q and three control variables, A, T_1 and T_2 .

The management of the plant wishes to obtain the values of the control variables so as to maximize the following net profit, P_T .

$$P_T = \begin{array}{l} \text{Revenue} + \text{Revenue} + \text{Revenue} - \text{Inventory} \\ \text{from B} \quad \text{from A} \quad \text{from C} \quad \text{cost of B} \\ - \text{Advertisement} - \text{Manufacturing} \\ \text{cost of B} \quad \text{cost} \end{array}$$

In mathematical terms

$$P_T = \int_0^{t_f} [c_1 Q + c_2 q x_2 + c_3 q (1 - x_2 - y_2) - C_I (I_m - I)^2 - C_A A^2 Q^2 - C_T (T_m - T_1)^2 - C_T (T_1 - T_2)^2] dt \quad (5-35)$$

where c_1 , c_2 and c_3 are the revenues from the sale per unit of B, A and C, respectively. Last two terms represent the manufacturing cost. T_m may be considered as the temperature of the raw material before processing. Like the first two models, here too, we need to approximate the various differential equations representing the system, and the profit equation by difference equations. For a small interval Δt equations (5-25) through (5-28), (5-33) and (5-34) become

$$x_1(t+\Delta t) = x_1(t) + [\frac{q}{v_1}(x_0 - x_1(t)) - k_{a1}x_1(t)]\Delta t \quad (5-36)$$

$$y_1(t+\Delta t) = y_1(t) + [\frac{q}{v_1}(y_0 - y_1(t)) - k_{b1}y_1(t) + k_{a1}x_1(t)]\Delta t \quad (5-37)$$

$$x_2(t+\Delta t) = x_2(t) + [\frac{q}{v_2}(x_1(t) - x_2(t)) - k_{a2}x_2(t)]\Delta t \quad (5-38)$$

$$y_2(t+\Delta t) = y_2(t) + [\frac{q}{v_2}(y_1(t) - y_2(t)) - k_{b2}y_2(t) + k_{a2}x_2(t)]\Delta t \quad (5-39)$$

$$I(t+\Delta t) = I(t) + [qy_2(t) - Q(t)] \quad (5-40)$$

$$Q(t+\Delta t) = Q(t) + [A(t)(c + A(t))(1 - \frac{Q(t)}{N})]\Delta t \quad (5-41)$$

The profit over the small interval Δt is given by

$$\begin{aligned} P_T &= [c_1 Q + c_2 q x_2 + c_3 q (1 - x_2 - y_2) - C_I (I_m - I)^2 \\ &= C_A A^2 Q^2 - C_T \{(T_m - T_1)^2 + (T_1 - T_2)^2\}] \Delta t \end{aligned} \quad (5-42)$$

The values of the various numerical constants are assumed as follows:

$G_a = 0.535 \times 10^{11}$ per unit time	$N = 100$
$G_b = 0.461 \times 10^{18}$ per unit time	$c = 1$
$E_a = 18000$ cal/mole	$c_1 = 5.0$
$E_b = 30000$ cal/mole	$c_2 = c_3 = 0$
$R = 2$ cal/mole $^{\circ}\text{K}$	$C_I = 1.0$
$q = 60$ gal/unit	$C_A = 0.01$
$v_1 = v_2 = 12$ gal	$C_T = 0.0005$
$I_m = 20$	$t_f = 1$
$T_m = 340^{\circ}\text{K}$	$y_0 = 0.43$
$x_0 = 0.53$	$y_1(0) = 0.43$
$x_1(0) = 0.53$	$y_2(0) = 0.43$
$x_2(0) = 0.53$	$Q(0) = 1.0$
$I(0) = 8.0$	

Because of the limitations of funds, the advertisement must not be more than 20 and also the temperatures in the two reactors must not be more than 450°K .

5.3.1 Formulation of the Problem

The process was divided into 5 stages. The problem can then be defined as

maximize

$$P_T = \sum_{k=1}^5 c_1 Q(k\Delta t) + c_2 q x_2(k\Delta t) + c_3 q (1 - x_2(k\Delta t) - y_2(k\Delta t) - C_I (I_m - I(k\Delta t)))^2 - C_A (A(k\Delta t) Q(k\Delta t))^2 - C_T \{ (T_m - T_1(k\Delta t))^2 + (T_1(k\Delta t) - T_2(k\Delta t))^2 \} \Delta t \quad (5-43)$$

subject to the constraints

$$x_1(k\Delta t) = x_1(\overline{k-1}\Delta t) + [\frac{q}{v_1}(x_0 - x_1(\overline{k-1}\Delta t)) - k_{a1}x_1(\overline{k-1}\Delta t)]\Delta t \quad (5-44)$$

$$y_1(k\Delta t) = y_1(\overline{k-1}\Delta t) + [\frac{q}{v_1}(y_0 - y_1(\overline{k-1}\Delta t)) - k_{b1}y_1(\overline{k-1}\Delta t) - k_{a1}x_1(\overline{k-1}\Delta t)]\Delta t \quad (5-45)$$

$$x_2(k\Delta t) = x_2(\overline{k-1}\Delta t) + [\frac{q}{v_2}(x_1(\overline{k-1}\Delta t) - x_2(\overline{k-1}\Delta t)) - k_{a2}x_2(\overline{k-1}\Delta t)]\Delta t \quad (5-46)$$

$$y_2(k\Delta t) = y_2(\overline{k-1}\Delta t) + [\frac{q}{v_2}(y_1(\overline{k-1}\Delta t) - y_2(\overline{k-1}\Delta t)) - k_{b2}y_2(\overline{k-1}\Delta t) - k_{a2}x_2(\overline{k-1}\Delta t)]\Delta t \quad (5-47)$$

$$I(k\Delta t) = I(\overline{k-1}\Delta t) + [qy_2(\overline{k-1}\Delta t) - Q(\overline{k-1}\Delta t)]\Delta t \quad (5-48)$$

$$Q(k\Delta t) = Q(\overline{k-1}\Delta t) + [Q(\overline{k-1}\Delta t)(c + A(k\Delta t))(1 - \frac{Q(\overline{k-1}\Delta t)}{N})]\Delta t \quad (5-49)$$

$$A(k\Delta t) \geq 0 \quad (5-50)$$

$$-A(k\Delta t) \geq -20 \quad (5-50a)$$

$$-T_1(k\Delta t) \geq -450 \quad (5-51)$$

$$-T_2(k\Delta t) \geq -450 \quad (5-52)$$

where $k = 1, 2, \dots, 5$ and $\Delta t = t_f/5$

It is given that

$$\begin{aligned} x_0 &= 0.53 & y_0 &= 0.43 \\ x_1(0) &= 0.53 & y_1(0) &= 0.43 \\ x_2(0) &= 0.53 & y_2(0) &= 0.43 \\ I(0) &= 8.0 & Q(0) &= 1.0 \end{aligned}$$

In this problem, equations (5-44) through (5-49) represent the constraints on the six state variables, x_1 , y_1 , x_2 , y_2 , I and Q , respectively. On

dividing the process into five stages, the number of control variables becomes 15. They are $A(k\Delta t)$, $T_1(k\Delta t)$ and $T_2(k\Delta t)$ where $k = 1, 2, \dots, 5$. The constraints on the control variables are given by the inequalities (5-50), (5-50a), (5-51), and (5-52). These serve as bounds. Obviously, we have five equality constraints on each of the state variables, ten inequality constraints on advertisement and five inequality constraints on each of the temperatures, T_1 and T_2 . Like the first two problems, the equality constraints on the state variables were eliminated while computing gradients by obtaining partial derivatives of the state variables with respect to the control variables at each stage. The calculations for the partial derivatives are given in Appendix D.

5.3.2 Numerical Results

The problem was first solved for 3 stages using six sets of initial approximations. The results showed the existence of two optimal solutions. It was, however, observed that at higher optimum, the concentration x_2 , at the first stage, became negative. This seemed to be unreasonable and was due to the fact that the division of the process into three stages involved a considerable amount of approximation.

The problem was then solved for 5 stages. Since there are six state variables, higher number of stages involves extremely complicated computations for gradients. Therefore, more than 5 stages were not considered. The subroutine FUNCT designed to evaluate the objective function and the gradients for the 5-stage process is shown in Appendix F. Limits and tolerances selected for the 5-stage process are as follows

$$\begin{aligned}
\epsilon_1 &= 0.002 & \text{LIMIT} &= 150 \\
\epsilon_2 &= 0.001 & \text{MXRN} &= 5 \\
\epsilon_3 &= 0.005 & \beta_{\max} &= 2 \\
\tau_{\max} &= 1.5
\end{aligned}$$

For the 5-stage problem, the number of variables, $m = 15$; the number of constraints, $k = 20$. Again, since all the constraints act as bounds, $s = 20$ and $c = 0$. The estimated value of the maximum profit was taken as 60.

The problem was run using six sets of initial approximation to the control variables. They were

$$\begin{aligned}
A_j &= 1.00, & T_{ij} &= 345 \\
A_j &= 1.00, & T_{ij} &= 350 \\
A_j &= 1.00, & T_{ij} &= 355 \\
A_j &= 1.00, & T_{ij} &= 365 \\
A_j &= 5.00, & T_{ij} &= 370 \\
\text{and } A_j &= 10.00, & T_{ij} &= 375
\end{aligned}$$

where A_j and T_{ij} represent advertisement and temperatures respectively at stage j ; $i = 1, 2$, and $j = 1, 2, \dots, 5$. Again two distinct optimal solutions were obtained. This seems quite likely since the model is highly nonlinear and, moreover, the conjugate gradient method finds only the local extremum. It was observed that during the course of optimization, none of the constraints were active. Therefore, the constraint tolerance, ϵ_2 and the linear dependence tolerance ϵ_3 had no effect on the solution of the problem. With the gradient tolerance, $\epsilon_2 = 0.002$ and for the same maximum profit, the results showed some difference in the values of the control variables, as obtained with

different initial approximations. The gradient tolerance was, therefore, reduced to 0.0001 and the limit on the maximum number of iterations was set equal to 60. The problem was run further starting from the solutions obtained with $\epsilon_1 = 0.002$. In each case the problem ran for 60 iterations, since the gradient tolerance of 0.0001 was not achieved. However, the results obtained after 60 iterations did not show any significant difference from those obtained with $\epsilon_1 = 0.002$.

The two values of the maximum profit obtained were 49.704 and 51.436. The higher of the two values, that is, 51.436 is probably the actual maximum profit. The results corresponding to the two optima are summarized in Tables 11 and 12. Values of the profit, advertisement, temperatures, sales and inventory at the actual optimum are also shown in Figs. 9 through 11. The convergence rates of advertisement, temperatures and profit are given in Tables 13 and 14. The execution times for the various sets of initial approximations are given in Table 15. These include the execution times for the additional 60 iterations. Table 15 also gives the total number of iterations as well as the number of iterations required to obtain solutions with $\epsilon_1 = 0.002$.

This problem has also been solved by Shah [24], as problem D, by quasilinearization. The maximum profit obtained by him was 66.26. The difference between the present results and those obtained by Shah is probably due to the approximation involved in treating the continuous process as a 5-stage process.

Table 11 - Results Corresponding to the Relative Optimum

t	$t+\Delta t$	$A(t+\Delta t)$	$T_1(t+\Delta t)$	$T_2(t+\Delta t)$	$x_1(t+\Delta t)$	$y_1(t+\Delta t)$	$x_2(t+\Delta t)$	$y_2(t+\Delta t)$	$I(t+\Delta t)$	$Q(t+\Delta t)$	P_T
0.0	0.2	15.86	318.47	315.91	0.527	0.433	0.528	0.432	12.960	4.399	-15.096
0.2	0.4	6.82	316.36	308.07	0.528	0.432	0.526	0.434	17.280	10.833	-16.729
0.4	0.6	3.46	321.03	311.41	0.526	0.434	0.526	0.434	20.322	19.452	-6.413
0.6	0.8	1.78	331.89	322.57	0.521	0.438	0.522	0.438	21.638	28.168	16.168
0.8	1.0	0.75	341.60	342.59	0.510	0.447	0.499	0.456	21.256	35.248	49.704

Table 12 - Results Corresponding to the Actual Optimum

t	t+Δt	A(t+Δt)	T ₁ (t+Δt)	T ₂ (t+Δt)	x ₁ (t+Δt)	y ₁ (t+Δt)	x ₂ (t+Δt)	y ₂ (t+Δt)	I(t+Δt)	Q(t+Δt)	P _T
0.0	0.2	15.07	354.93	368.03	0.475	0.468	0.394	0.487	12.960	4.182	-13.717
0.2	0.4	6.59	362.27	380.99	0.447	0.468	0.243	0.342	17.967	10.265	-13.513
0.4	0.6	3.40	360.89	384.09	0.459	0.462	0.274	0.295	20.019	18.378	- 3.059
0.6	0.8	1.81	338.59	337.37	0.516	0.442	0.452	0.469	19.887	26.799	19.047
0.8	1.0	0.79	339.62	339.40	0.513	0.444	0.501	0.454	20.151	33.808	51.436

Table 13 - Convergence Rates of Advertisement, Temperatures and Profit for the Relative Optimum

Iteration No.	No. of Functional Evaluations	No. of Constraints in Basis	A_1	A_3	A_5	T_{11}	T_{13}	T_{15}	T_{21}	T_{23}	T_{25}	P_T
0	1	0	1.00	1.00	1.00	345.00	345.00	345.00	345.00	345.00	345.00	-47.611
5	11	0	13.85	3.22	0.96	344.56	344.76	345.00	344.60	344.67	345.00	45.238
10	18	0	16.46	3.50	0.72	344.29	344.64	344.99	344.43	344.47	345.00	48.120
25	35	0	15.90	3.47	0.75	310.93	329.10	343.65	326.18	318.94	344.98	49.608
47	66	0	15.87	3.46	0.75	318.37	321.44	342.06	315.62	310.76	343.37	49.702
107	170	0	15.86	3.46	0.75	318.47	321.03	341.60	315.91	311.41	342.59	49.704
0	1	0	1.00	1.00	1.00	350.00	350.00	350.00	350.00	350.00	350.00	-51.818
5	11	0	14.03	3.21	0.96	349.49	349.72	349.99	349.51	349.67	350.00	44.307
10	18	0	16.34	3.51	0.72	349.14	349.55	349.97	349.28	349.44	350.00	47.264
25	34	0	15.90	3.47	0.75	309.98	328.49	347.45	322.50	320.89	349.84	49.597
56	75	0	15.86	3.46	0.75	318.45	320.87	342.95	315.97	311.01	344.65	49.703
116	188	0	15.86	3.46	0.75	318.53	321.12	342.08	316.03	311.37	343.48	49.704
0	1	0	1.00	1.00	1.00	355.00	355.00	355.00	355.00	355.00	355.00	-55.992
5	11	0	14.79	3.03	1.17	354.50	354.70	354.99	354.47	354.81	355.00	43.008
10	18	0	16.83	3.52	0.71	354.17	354.53	354.96	354.23	354.67	355.00	46.257
25	35	0	15.91	3.47	0.75	302.69	327.61	350.93	317.73	330.51	354.85	49.464
51	67	0	15.86	3.46	0.75	318.25	319.61	345.62	313.14	311.19	349.12	49.697
111	167	0	15.87	3.46	0.75	317.95	321.94	342.70	316.22	311.13	344.55	49.703

Note - T_{ij} denotes temperatures T_i at stage j ; $i = 1, 2$ and $j = 1, 2, \dots, 5$

Table 14 - Convergence Rates of Advertisement, Temperatures and Profit for the Actual Optimum

Iter- ation No.	No. of Functional Evaluations	No. of Constraints in Basis	A ₁	A ₃	A ₅	T ₁₁	T ₁₃	T ₁₅	T ₂₁	T ₂₃	T ₂₅	P _T
0	1	0	1.00	1.00	1.00	365.00	365.00	365.00	365.00	365.00	365.00	-56.275
5	11	0	14.27	3.18	0.98	365.37	365.07	364.98	365.11	366.14	365.00	43.33
10	18	0	16.82	3.53	0.73	365.72	365.09	364.93	365.31	367.19	365.00	46.491
25	39	0	15.09	3.41	0.78	368.89	365.20	363.95	369.92	378.24	364.99	51.243
50	69	0	15.05	3.40	0.79	353.01	361.47	353.74	367.95	383.03	363.67	51.349
75	103	0	15.07	3.40	0.79	351.77	359.24	343.12	367.91	384.58	340.48	51.430
98	135	0	15.07	3.40	0.79	354.94	360.88	339.62	368.02	384.09	339.40	51.436
158	251	0	15.07	3.40	0.79	354.93	360.89	339.62	368.03	384.09	339.40	51.436
0	1	0	5.00	5.00	5.00	370.00	370.00	370.00	370.00	370.00	370.00	-65.028
5	9	0	13.08	3.56	0.16	370.63	370.21	369.98	370.50	371.17	370.00	45.076
10	16	0	16.20	3.45	0.74	371.55	370.43	369.92	371.19	372.81	370.00	49.578
25	35	0	15.10	3.41	0.79	371.50	370.45	368.82	371.35	375.37	369.99	51.137
50	65	0	15.05	3.40	0.79	355.58	365.68	358.58	367.27	381.68	368.81	51.309
84	107	0	15.07	3.40	0.79	354.93	360.53	340.88	368.01	384.07	341.60	51.436
144	216	0	15.07	3.40	0.78	354.84	361.12	340.79	368.04	384.08	341.23	51.436
0	1	0	10.00	10.00	10.00	375.00	375.00	375.00	375.00	375.00	375.00	-3840.679
5	9	0	11.15	2.98	0.61	375.24	375.09	374.99	375.24	375.38	375.00	46.333
10	17	0	15.20	3.41	0.78	375.52	375.20	374.94	375.43	375.87	375.00	50.837
25	36	0	15.08	3.41	0.79	372.40	374.46	372.86	367.86	374.55	374.98	51.097
50	64	0	15.09	3.40	0.79	358.45	353.83	355.33	368.07	383.66	371.89	51.321
89	116	0	15.07	3.40	0.79	354.42	359.08	338.17	367.91	384.14	337.82	51.435
149	224	0	15.08	3.40	0.79	354.71	360.18	338.61	367.97	384.15	337.95	51.435

Note - T_{ij} denotes temperatures T_i at stage j ; $i = 1, 2$ and $j = 1, 2, \dots, 5$

Table 15 - Execution Times for Different Initial Approximations

Initial Approximation	Total No. of Iterations	No. of Iterations with $\epsilon_1 = 0.002$	Execution Time in Minutes
$A_j = 1.00$ $T_{ij} = 345$	107	47	6.01
$A_j = 1.00$ $T_{ij} = 350$	116	56	6.25
$A_j = 1.00$ $T_{ij} = 355$	111	51	6.21
$A_j = 1.00$ $T_{ij} = 365$	158	92	7.98
$A_j = 5.00$ $T_{ij} = 370$	144	84	7.32
$A_j = 10.00$ $T_{ij} = 375$	149	89	7.93

Note - A_j and T_{ij} denote advertisement and temperatures, respectively,
at stage j ; $i = 1, 2$ and $j = 1, 2, \dots, 5$.

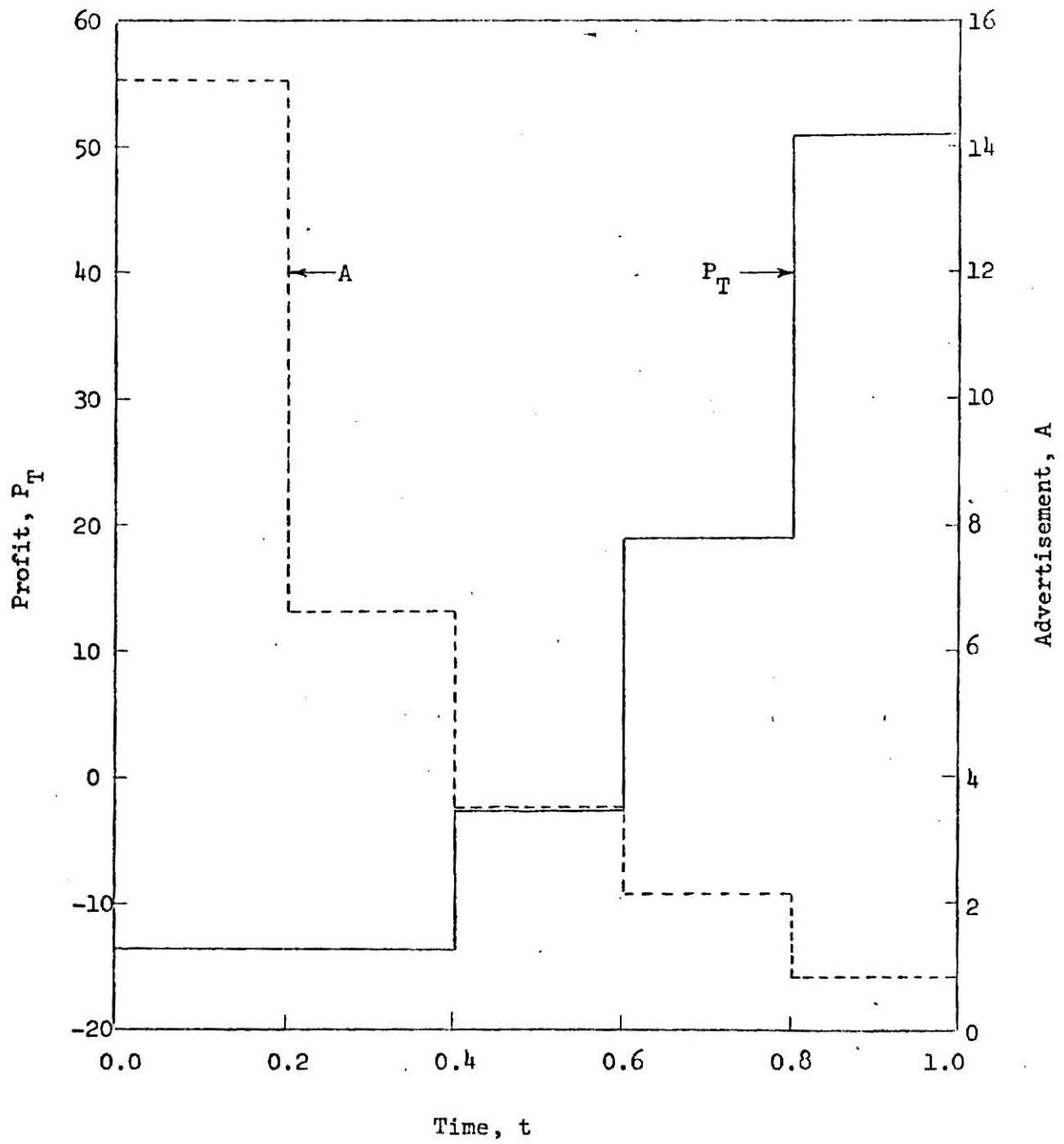


Fig. 9 Profit and Advertisement at Actual Optimum

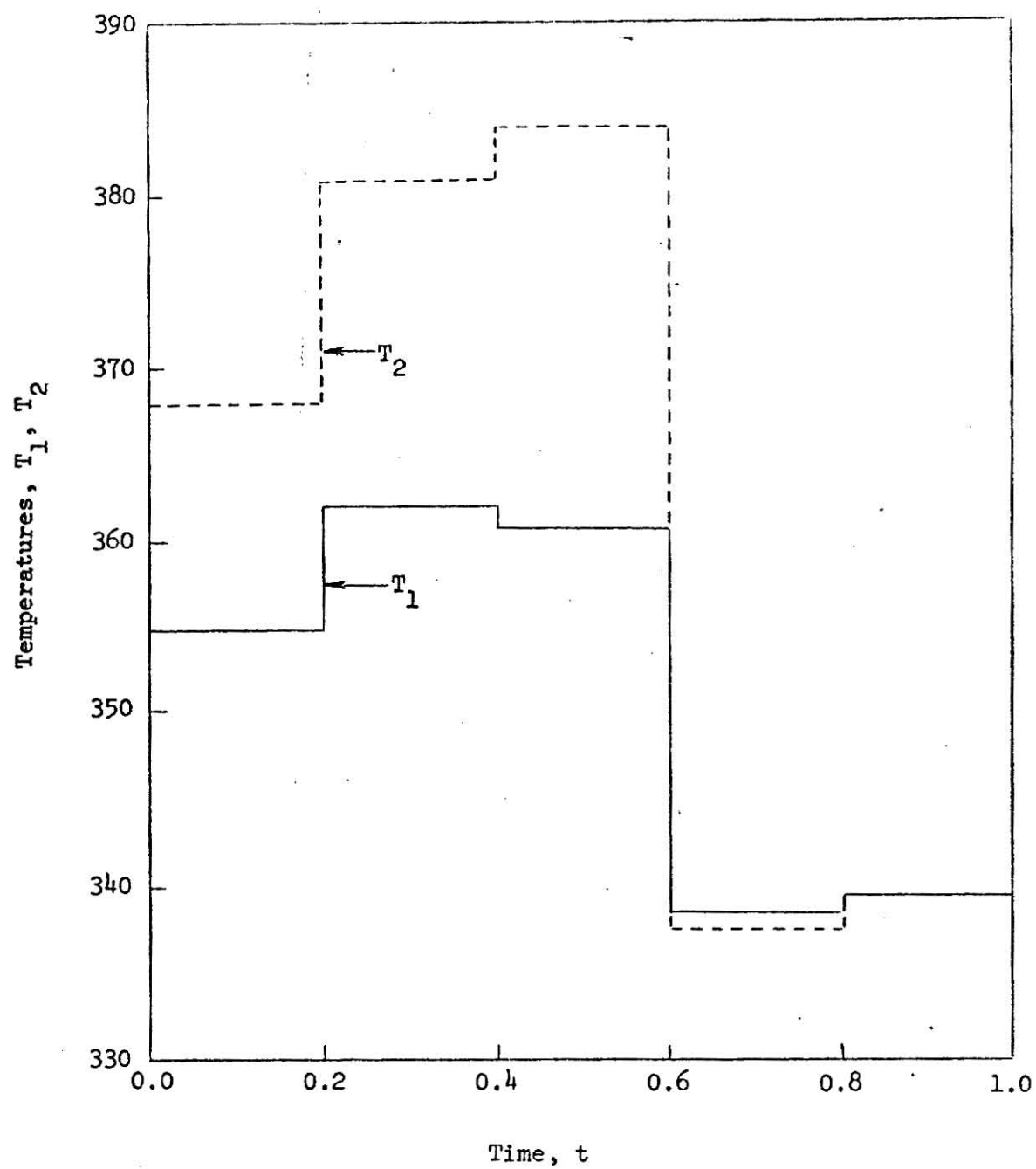


Fig. 10 Temperatures at Actual Optimum

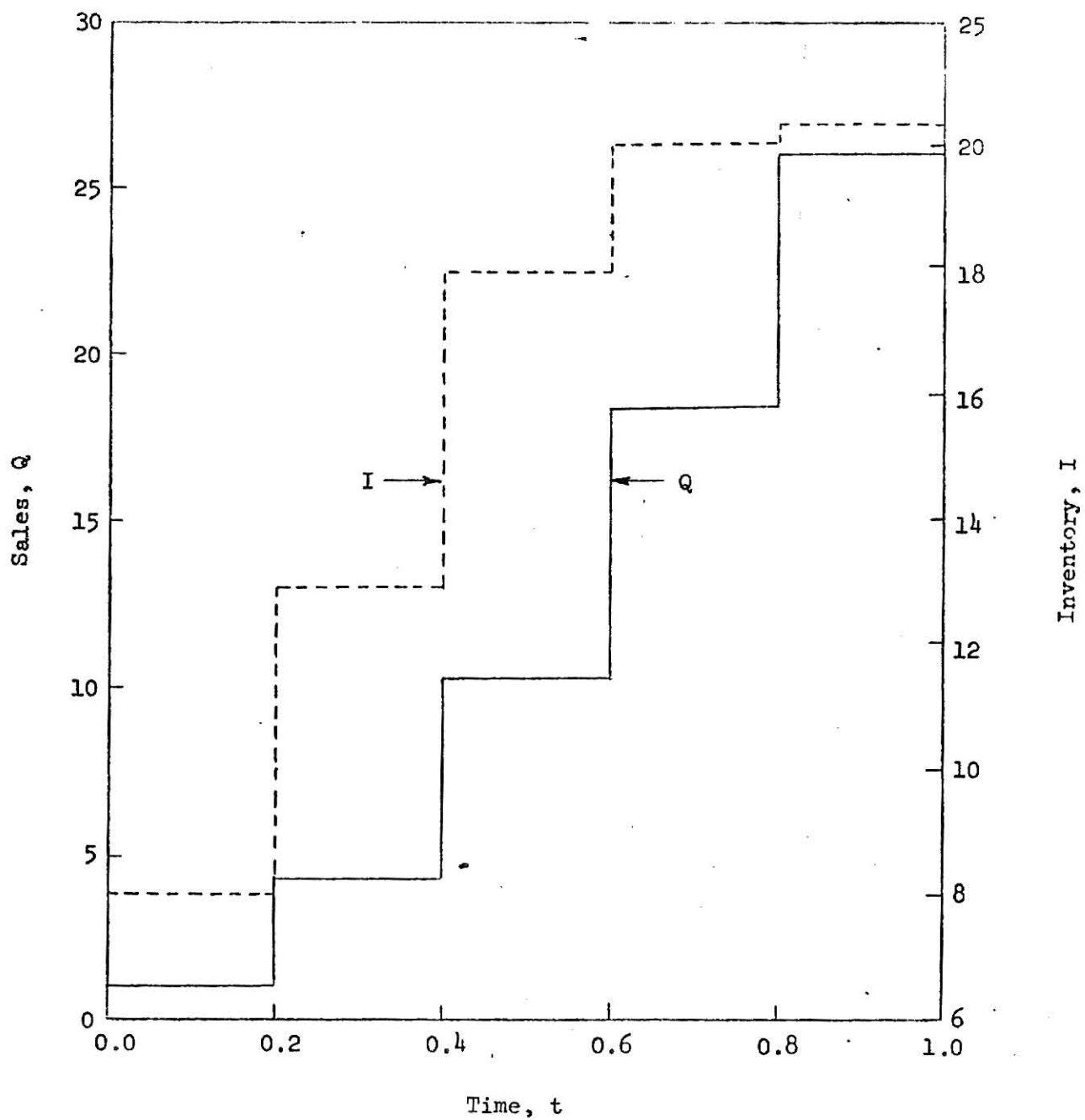


Fig. 11 Sales and Inventory at Actual Optimum

CHAPTER 6

DISCUSSION

Solutions of the different test problems presented in this study demonstrate the usefulness of mathematical programming approach to industrial management systems. In particular the conjugate gradient method developed by Goldfarb [7] has proved quite efficient in solving a class of non-linear problems. The efficiency of the technique is affected considerably by the manner in which the problem is formulated. For example, presence of bounds and equality constraints definitely make the method more efficient. It is therefore, essential to have the basic knowledge about the system being solved and formulate the model in the proper form.

The conjugate gradient method has an advantage over Rosen's gradient projection method in that it requires less number of functional evaluations [7]. Since in complicated problems, each functional evaluation, together with the gradient calculations, may consume a considerable amount of execution time, the computational cost, in such cases, for the conjugate gradient method would obviously be less. Less number of functional evaluations is probably due to the use of the cubic interpolation scheme. Moreover, since the conjugate gradient method is quadratically convergent, it would normally require less number of iterations to reach the optimum, than does the gradient projection method.

The applications of the method also demonstrate that in many cases a continuous process can easily be solved by approximating it to a multistage process. The number of stages into which the process may

be divided is generally determined by the accuracy requirements and the computational cost. Like other gradient methods, the conjugate gradient method too requires the calculation of first partial derivatives. In complicated problems involving a large number of state variables, the computations for the partial derivatives almost limit the number of stages into which the process can be divided. In such cases, one may have to sacrifice some accuracy in order to avoid laborious or sometimes practically impossible calculations.

ACKNOWLEDGEMENT

The author wishes to express his deep sense of gratitude to his major professor, Dr. E. S. Lee for his continuous guidance, valuable advice and the personal interest he has taken in the preparation of this thesis.

REFERENCES

1. Beckman, F. S., "The Solution of Linear Equations by the Conjugate Gradient Method", Mathematical Methods for Digital Computers, A. Ralston and H. S. Wilf (Eds.), John Wiley and Sons, New York (1960).
2. Box, M. J., "A New Method of Constrained Optimization and a Comparison with Other Methods", Computer Journal, Vol. 8, p. 42 (1965).
3. Box, M. J., "A Comparison of Several Current Optimization Techniques and the Use of Transformations in Constrained Problems", Computer Journal, Vol. 9, p. 67 (1966).
4. Davidon, W. C., "Variable Metric Method for Minimization", Research and Development Report ANL-5990 (Rev.), Atomic Energy Commission (1959).
5. Fletcher, R., and Powell, M. J. D., "A Rapidly Convergent Descent Method for Minimization", Computer Journal, Vol. 6, p. 163 (1963).
6. Fletcher, R., and Reeves, C. M., "Function Minimization by Conjugate Gradients", Computer Journal, Vol. 7, p. 149 (1964).
7. Goldfarb, D., and Lapidus, L., "Conjugate Gradient Method for Nonlinear Programming with Linear Constraints", I & EC Fundamentals, Vol. 7, p. 142 (1968).
8. Hadley, G., Nonlinear and Dynamic Programming, Addison Wesley Publishing Co., Inc. (1964).
9. Kelley, H. J., "Method of Gradients", Optimization Techniques, G. Leitman (Ed.), Academic Press, New York, New York (1962).

10. Kelley, H. J., Denham, W. F., Johnson, I. L., and Wheatley, P. O.,
"An Accelerated Gradient Method for Parameter Optimization
with Nonlinear Constraints", Journal of the Astronautical
Sciences, Vol. 9, p. 166 (1966).
11. Kelley, H. J., and Myers, G. E., "Conjugate Direction Methods
for Parameter Optimization", paper presented at the 18th
International Astronautical Federation, Belgrade, Yugoslavia
(1967).
12. Kelley, H. J., and Speyer, J. L., "Accelerated Gradient Projection",
paper presented at the Colloquium on Optimization, Nice, France
(1969).
13. Kelley, J. E. Jr., "The Cutting Plane Method for Solving Convex
Programs", SIAM Journal, Vol. 8, p 703 (1960).
14. Lee, E. S., and Shah, P. D., "Optimization of Production Planning
by the Generalized Newton-Raphson Method", paper presented at
the AIIE National Meeting, Houston, Texas (1969).
15. Lee, E. S., and Shaikh, M. A., "Optimal Production Planning
by a Gradient Technique I. First Variations", to appear in
Management Science.
16. Merrill, R. P., "Gradient Projection 7090", SHARE Program
Library Report No. 3430 (1966).
17. Myers, G. E., "Properties of the Conjugate Gradient Methods",
Journal of Optimization Theory and Applications, Vol. 2, p. 209
(1968).
18. Nair, B. R., "Optimization of Management Systems by the Conjugate
Gradient Methods", Master's Report, Kansas State University (1969).

19. Pearson, J. D., "Variable Metric Methods of Minimization",
Computer Journal, Vol. 12, p. 171.
20. Rosen, J. B., "The Gradient Projection Method for Nonlinear
Programming, Part-I, Linear Constraints", SIAM Journal, Vol. 8,
p. 181 (1960).
21. Rosen, J. B., "The Gradient Projection Method for Non-linear
Programming, Part-II, Nonlinear Constraints", SIAM Journal,
Vol 9, p. 514 (1961).
22. Rosenbrock, H. H., "An Automatic Method for Finding the Greatest
or Least Value of a Function", Computer Journal, Vol. 3, p. 175
(1960).
23. Spang, H. A., "A Review of Minimization Techniques for Nonlinear
Functions", SIAM Review, Vol. 4, p. 343 (1962).
24. Shah, P. D., "Application of Quasilinearization to Industrial
Management Systems", Master's Thesis, Kansas State University (1969).

APPENDIX A

NOMENCLATURE

The nomenclature used in the conjugate gradient method is as follows:

b_i	right hand side of a constraint
d_{ij}	element of the inverse matrix
e	number of equalities
f	value of the objective function
g	gradient in the direction of increasing f
H_i	a constraint (hyperplane)
H_q^i	matrix operator
Hg	conjugate gradient
k	number of constraints
LIMIT	maximum number of iterations
m	number of variables
MXRN	maximum number of re-inversions
n_i	normalized constraint vector
N_k	constraint matrix
N_q	basis
$(N_q^T N_q)^{-1}$	inverse matrix
Pg	projected gradient
Pn	projected constraint vector
q	constraints in the basis
q^*	constraints added to the initial basis
s	number of constraints that are bounds

u	linearly dependent constraints
v	constraints not in the basis with $\lambda = 0$
w	constraints not in the basis with $\lambda > 0$
x	variable vector
z	vector in the direction of step
$\beta_{\max}$	maximum interpolations
ϵ_1	gradient tolerance
ϵ_2	constraint tolerance
ϵ_3	linear dependence tolerance
λ_1	normal distance to a constraint

APPENDIX B

CUBIC INTERPOLATION SCHEME

The interpolation scheme used in the conjugate gradient method, to locate the maximum along a line, is based on the one suggested by Davidon [4]. The choice of the scheme is, however, arbitrary. In fact, any method suitable for finding a value of γ^i , such that $0 \leq \gamma^i \leq \lambda^i$, which maximizes $f(x^i + \gamma^i s^i)$ along s^i , can be used. The linear search technique used is as follows.

For interpolation purposes some point must be selected between x^i and $x^i + \lambda^i s^i$. Since H_q^i is an approximation to $-\hat{P}_q G^{-1}(x^i)$, a reasonable priori estimate could be a point given by

$$\bar{x}^i = x^i + \tau^i s^i \quad (B-1)$$

where

$$\tau^i = \min \left\{ \tau_{\max}, \frac{2[f(x^i) - f^*]}{(g^i)^T s^i} \right\} \quad (B-2)$$

f^* being the estimated value of the maximum. If τ^i is negative, it has been found useful to take $\tau^i = \tau_{\max}$. Further, τ^i may exceed λ^i and in that case, the point given by (B-1) will not lie between x^i and $x^i + \lambda^i s^i$. In order to avoid that, a better choice of the point is given by

$$\hat{x}^i = x^i + \tau s^i \quad (B-3)$$

where $\tau = \min \{\tau^i, \lambda^i\}$

The approximate value of γ^i , that maximizes $f(x^i + \gamma^i s^i)$ on the line segment between x^i and $\hat{x}^i$, given by (B-3), is then computed using the well known

cubic interpolation formulas given below.

$$\gamma^i = \tau(1-D) \quad (B-4)$$

where

$$D = \frac{W - (s^i)^T g^i - Z}{(s^i)^T g^i - (s^i)^T \hat{g}^i + 2W}$$

$$Z = \frac{3}{\tau} [f(x^i) - f(\hat{x}^i)] - (s^i)^T g^i - (s^i)^T \hat{g}^i$$

$$\text{and } W = [Z^2 - (s^i)^T g^i (s^i)^T \hat{g}^i]^{1/2}$$

Before interpolating, however, it is necessary to ensure that both $\{f(\hat{x}^i) - f(x^i)\}$ and $(s^i)^T g^i$ are not positive. Otherwise, $f(x)$ may not attain a maximum between x^i and $\hat{x}^i$. If one of them is positive, the method described by Davidon [4] is used to handle the situation. Many a time it may happen that the interpolation formulas may yield a point $x^i + \gamma^i s^i$ such that $f(x^i + \gamma^i s^i) \leq \max \{f(x^i), f(\hat{x}^i)\}$. In such cases, interpolation should be repeated over the portion of the divided range having the greatest function value at one of its end points.

APPENDIX C

RECURSION RELATIONS

As indicated in the algorithm of the conjugate gradient method, the matrix $(N_q^T N_q)^{-1}$ need to be computed for each iteration. For a particular set of q vectors n_1 , the matrix can be obtained by first forming matrix $N_q^T N_q$ and then inverting it. This would, however, require prohibitive amount of calculations for a large problem. To overcome this difficulty, two recursion relations suggested by Rosen [20] are used. These relations permit the calculation of $(N_{q-1}^T N_{q-1})^{-1}$ from $(N_q^T N_q)^{-1}$, when a hyperplane is dropped, with approximately q^2 multiplications and divisions. Also, when a hyperplane is added, $(N_q^T N_q)^{-1}$ can be calculated from $(N_{q-1}^T N_{q-1})^{-1}$ with approximately $2q^2 + mq$ multiplications and divisions.

Suppose that the matrix $(N_q^T N_q)^{-1}$ is partitioned as

$$(N_q^T N_q)^{-1} = \begin{pmatrix} B_1 & B_2 \\ B_3 & B_4 \end{pmatrix} \quad (C-1)$$

where B_1 , B_2 , B_3 and B_4 are, respectively $(q-1 \times q-1)$, $(q-1 \times 1)$, $(1, q-1)$ and (1×1) matrices. When the q th hyperplane, H_q , is dropped from the constrained basis, $(N_{q-1}^T N_{q-1})^{-1}$ can be obtained from $(N_q^T N_q)^{-1}$ using the following relation

$$(N_{q-1}^T N_{q-1})^{-1} = B_1 - B_2 B_4^{-1} B_3 \quad (C-2)$$

Since $(N_q^T N_q)^{-1}$ is known, the partitions B_1 , B_2 , B_3 and B_4 are also known.

When a hyperplane is added to the constrained basis, $(N_q^T N_q)^{-1}$ is given by equation (C-1) where the partitions B_1 , B_2 , B_3 and B_4 are computed as follows:

$$B_1 = (N_{q-1}^T N_{q-1})^{-1} + A_0^{-1} r_{q-1} r_{q-1}^T \quad (C-3)$$

$$B_2 = B_3^T = -A_0^{-1} r_{q-1} \quad (C-4)$$

$$B_4 = A_0^{-1} \quad (C-5)$$

where

$$r_{q-1} = (N_{q-1}^T N_{q-1})^{-1} N_{q-1}^T n_q$$

$$A_0 = ||P_{q-1} n_q||^2$$

and $P_{q-1} n_q = n_q - N_{q-1} x_{q-1}$

For certain applications a very useful recursion relation is one which gives P_q using P_{q-1} and n_q .

$$\begin{aligned} P_q &= I - N_q (N_q^T N_q)^{-1} N_q^T \\ &= I - [N_{q-1} n_q] \begin{pmatrix} B_1 & B_2 \\ B_3 & B_4 \end{pmatrix} \begin{pmatrix} N_{q-1}^T \\ N_q^T \end{pmatrix} \end{aligned} \quad (C-6)$$

Carrying out the matrix multiplications and substituting the values of B_1 , B_2 , B_3 and B_4 , we can write

$$P_q = P_{q-1} - \frac{P_{q-1} n_q n_q^T P_{q-1}}{n_q^T P_{q-1} n_q} \quad (C-7)$$

The use of the recursion relations given by equations (C-1) through (C-7) facilitate the computation of matrices $(N_i^T N_i)^{-1}$ and P_i with minimum computations.

APPENDIX D

GRADIENT CALCULATIONS

D.1 Inventory Model

Since it is a minimization problem, the total cost is minimized by maximizing the negative of the cost C_T . For that let $F = -C_T$. Rewriting the equations for F and the state variable I , we have

$$F = - \sum_{k=1}^n [C_I \{I_m - \frac{1}{2}(I(k-1)\Delta t) + I(k\Delta t)\}^2 + C_P \exp (P_m - P(k\Delta t))^2] \Delta t \quad (D-1)$$

$$I(k\Delta t) = I(k-1)\Delta t + (P(k\Delta t) - a - bk\Delta t)\Delta t \quad (D-2)$$

where $k = 1, 2, \dots, n$ and $\Delta t = T/n$.

For simplicity, let us denote $I(k\Delta t)$ by I_k and $P(k\Delta t)$ by P_k , $k = 1, 2, \dots, n$. Equations (D-1) and (D-2) can then be written as

$$F = - \sum_{k=1}^n [C_I \{I_m - \frac{1}{2}(I_{k-1} + I_k)\}^2 + C_P \exp (P_m - P_k)^2] \Delta t \quad (D-3)$$

$$\text{and } I_k = I_{k-1} + (P_k - a - bk\Delta t)\Delta t \quad (D-4)$$

Since P 's are the control variables, the gradients are given by the first partial derivatives of F with respect to P 's as follows

$$\begin{aligned} \frac{\partial F}{\partial P_i} = & \left[\sum_{k=1}^n \left\{ C_I (I_m - \frac{1}{2}(I_{k-1} + I_k)) \left(\frac{\partial I_{k-1}}{\partial P_i} + \frac{\partial I_k}{\partial P_i} \right) \right. \right. \\ & \left. \left. + 2C_P \exp (P_m - P_i)^2 (P_m - P_i) \right\} \Delta t; \quad i = 1, 2, \dots, n \right] \quad (D-5) \end{aligned}$$

where

$$\frac{\partial I_k}{\partial P_i} = 0 \quad i > k$$

From equation (D-4), we have

$$\frac{\partial I_k}{\partial P_i} = \Delta t \quad i = k$$

and

$$\begin{aligned} \frac{\partial I_k}{\partial P_i} &= \frac{\partial I_k}{\partial I_{k-1}} \cdot \frac{\partial I_{k-1}}{\partial I_{k-2}} \cdots \frac{\partial I_i}{\partial P_i} \quad i < k \\ &= \Delta t \end{aligned}$$

Let us consider the 5-stage process. Substituting the values of the derivatives on the right hand side of equation (D-5), we get

$$\begin{aligned} \frac{\partial F}{\partial P_1} &= C_I \{I_m - \frac{1}{2}(I_0 + I_1)\}(\Delta t)^2 + 2C_I \{I_m - \frac{1}{2}(I_1 + I_2)\}(\Delta t)^2 \\ &\quad + 2C_I \{I_m - \frac{1}{2}(I_2 + I_3)\}(\Delta t)^2 + 2C_I \{I_m - \frac{1}{2}(I_3 + I_4)\}(\Delta t)^2 \\ &\quad + 2C_I \{I_m - \frac{1}{2}(I_4 + I_5)\}(\Delta t)^2 + 2C_P \exp (P_m - P_1)^2 (P_m - P_1)\Delta t \\ &= 2C_I (\Delta t)^2 \left[\frac{1}{2} \{I_m - \frac{1}{2}(I_0 + I_1)\} + \sum_{k=2}^5 (I_m - \frac{1}{2}(I_{k-1} + I_k)) \right] \\ &\quad + 2C_P \exp (P_m - P_1)^2 (P_m - P_1)\Delta t \end{aligned} \quad (D-6)$$

Similarly

$$\begin{aligned} \frac{\partial F}{\partial P_2} &= 2C_I (\Delta t)^2 \left[\frac{1}{2} \{I_m - \frac{1}{2}(I_1 + I_2)\} + \sum_{k=3}^5 (I_m - \frac{1}{2}(I_{k-1} + I_k)) \right] \\ &\quad + 2C_P \exp (P_m - P_2)^2 (P_m - P_2)\Delta t \end{aligned} \quad (D-7)$$

$$\begin{aligned} \frac{\partial F}{\partial P_3} &= 2C_I (\Delta t)^2 \left[\frac{1}{2} \{I_m - \frac{1}{2}(I_2 + I_3)\} + \sum_{k=4}^5 (I_m - \frac{1}{2}(I_{k-1} + I_k)) \right] \\ &\quad + 2C_P \exp (P_m - P_3)^2 (P_m - P_3)\Delta t \end{aligned} \quad (D-8)$$

$$\begin{aligned} \frac{\partial F}{\partial P_4} = & 2C_I (\Delta t)^2 \left[\frac{1}{2} \{I_m - \frac{1}{2} (I_3 + I_4)\} + \{I_m - \frac{1}{2} (I_4 + I_5)\} \right] \\ & + 2C_P \exp (P_m - P_4)^2 (P_m - P_4) \Delta t \end{aligned} \quad (D-9)$$

$$\frac{\partial F}{\partial P_5} = C_I (\Delta t)^2 \{I_m - \frac{1}{2} (I_3 + I_4)\} + 2C_P \exp (P_m - P_5)^2 (P_m - P_5) \Delta t$$

Thus, equations (D-6) through (D-10) represent the gradient equations for the 5-stage process. (D-10)

D.2 An Inventory and Advertisement Model

Rewriting the profit equation and the state variable equations for the model, we have

$$P_T = \sum_{k=1}^n [FQ(k\Delta t) - C_I (I_m - I(k\Delta t))^2 - C_A A(k\Delta t)Q(k\Delta t)]\Delta t \quad (D-11)$$

$$Q(k\Delta t) = \frac{Q(k-1\Delta t)[1 + (c + A(k\Delta t))\Delta t]}{1 + (c + A(k\Delta t))Q(k-1\Delta t)\Delta t} \quad (D-12)$$

$$I(k\Delta t) = I(k-1\Delta t) + [P(k-1\Delta t) - Q(k-1\Delta t)]\Delta t \quad (D-13)$$

where $k = 1, 2, \dots, n$ and $\Delta t = T/n$

In the above equations, let

$$I_k = I(k\Delta t)$$

$$Q_k = Q(k\Delta t)$$

$$A_k = A(k\Delta t)$$

$$\text{and } P_k = P(k\Delta t)$$

Equations (D-11), (D-12) and (D-13) become

$$P_T = \sum_{k=1}^n [FQ_k - C_I (I_m - I_k)^2 - C_A A_k Q_k]\Delta t \quad (D-14)$$

$$Q_k = \frac{Q_{k-1}[1 + (c + A_k)\Delta t]}{1 + (c + A_k)Q_{k-1}\Delta t} \quad (D-15)$$

and

$$I_k = I_{k-1} + (P_{k-1} - Q_{k-1})\Delta t \quad (D-16)$$

Here, since A 's are the control variables, we need to compute the first derivatives of P_T with respect to A 's. Let us consider the 5-stage process. The gradients may be computed as follows

$$\begin{aligned}
\frac{\partial P_T}{\partial A_1} &= [(F - C_{A_1}) \frac{\partial Q_1}{\partial A_1} + (F - C_{A_2}) \frac{\partial Q_2}{\partial A_1} + \dots + (F - C_{A_5}) \frac{\partial Q_5}{\partial A_1} \\
&\quad + 2C_I(I_m - I_1) \frac{\partial I_1}{\partial A_1} + 2C_I(I_m - I_2) \frac{\partial I_2}{\partial A_1} + \dots + 2C_I(I_m - I_5) \frac{\partial I_5}{\partial A_1} - C_A Q_1] \Delta t \\
&= \left[\sum_{k=1}^5 (F - C_{A_k}) \frac{\partial Q_k}{\partial A_1} + \sum_{k=1}^5 2C_I(I_m - I_k) \frac{\partial I_k}{\partial A_1} - C_A Q_1 \right] \Delta t \quad (D-17)
\end{aligned}$$

From equation (D-15), we have

$$\frac{\partial Q_i}{\partial A_1} = \frac{Q_{i-1} [1 + (c + A_i) Q_{i-1} \Delta t] \Delta t - Q_{i-1} [1 + (c + A_i) \Delta t] Q_{i-1} \Delta t}{(1 + (c + A_i) Q_{i-1} \Delta t)^2} \quad (D-18)$$

$$\frac{\partial Q_k}{\partial A_i} = 0 \quad i > k \quad (D-19)$$

$$\text{and } \frac{\partial Q_k}{\partial A_i} = \frac{\partial Q_k}{\partial Q_{k-1}} \cdot \frac{\partial Q_{k-1}}{\partial Q_{k-2}} \dots \frac{\partial Q_i}{\partial A_i} \quad i < k \quad (D-20)$$

Again, from equation (D-15), we have

$$\frac{\partial Q_k}{\partial Q_{k-1}} = \frac{(1 + (c + A_k) \Delta t)(1 + (c + A_k) Q_{k-1} \Delta t) - Q_{k-1}(1 + (c + A_k) \Delta t) A_k \Delta t}{(1 + (c + A_k) Q_{k-1} \Delta t)^2} \quad k > 2 \quad (D-21)$$

From equation (D-15) and (D-16), we get

$$\frac{\partial I_k}{\partial A_i} = 0 \quad i \geq k \quad (D-22)$$

$$\frac{\partial I_k}{\partial A_i} = \frac{\partial I_k}{\partial Q_{k-1}} \cdot \frac{\partial Q_{k-1}}{\partial A_k} \quad i = k - 1 \quad (D-23)$$

$$\text{and } \frac{\partial I_k}{\partial A_i} = \frac{\partial I_k}{\partial I_{k-1}} \cdot \frac{\partial I_{k-1}}{\partial I_{k-2}} \cdots \frac{\partial I_{i+1}}{\partial Q_i} \cdot \frac{\partial Q_i}{\partial A_i} + \frac{\partial I_k}{\partial Q_{k-1}} \cdot \frac{\partial Q_{k-1}}{\partial Q_{k-2}} \cdots \frac{\partial Q_i}{\partial A_i}$$

$$i < k - 1 \quad (D-24)$$

where

$$\frac{\partial I_k}{\partial I_{k-1}} = 1 \quad k > 1 \quad (D-25)$$

and

$$\frac{\partial I_k}{\partial Q_{k-1}} = -\Delta t \quad k > 1 \quad (D-26)$$

Substituting the partial derivatives, obtained from equations (D-25)

and (D-26), in equations (D-23) and (D-24), we get

$$\frac{\partial I_k}{\partial A_i} = -\Delta t \frac{\partial Q_i}{\partial A_i} \quad i = k - 1 \quad (D-27)$$

$$\frac{\partial I_k}{\partial A_i} = -\Delta t \frac{\partial Q_i}{\partial A_i} - \Delta t \frac{\partial Q_{k-1}}{\partial Q_{k-2}} \cdot \frac{\partial Q_{k-2}}{\partial Q_{k-3}} \cdots \frac{\partial Q_i}{\partial A_i} \quad i < k - 1 \quad (D-28)$$

Representing the expressions given by equations (D-18) and (D-21)

by x_i and y_k , and substituting the values of partial derivatives on

the right hand side of equation (D-1), we have

$$\begin{aligned} \frac{\partial P_T}{\partial A_1} = & (F - C_A A_1) \Delta t x_1 + (F - C_A A_2) \Delta t x_1 y_2 \\ & + (F - C_A A_3) \Delta t x_1 y_2 y_3 + (F - C_A A_4) \Delta t x_1 y_2 y_3 y_4 \\ & + (F - C_A A_5) \Delta t x_1 y_2 y_3 y_4 y_5 - 2C_I (I_m - I_2) (\Delta t)^2 x_1 \\ & - 2C_I (I_m - I_3) (\Delta t)^2 (x_1 + x_1 y_2) - \end{aligned}$$

$$\begin{aligned}
& 2C_I (I_m - I_4) (\Delta t)^2 (x_1 + x_1 y_2 + x_1 y_2 y_3) \\
& - 2C_I (I_m - I_5) (\Delta t)^2 (x_1 + x_1 y_2 + x_1 y_2 y_3 + x_1 y_2 y_3 y_4) \\
& - C_A Q_1 \Delta t
\end{aligned} \tag{D-29}$$

Similarly

$$\begin{aligned}
\frac{\partial P_T}{\partial A_2} &= (F - C_A A_2) \Delta t x_2 + (F - C_A A_3) \Delta t x_2 y_3 \\
&+ (F - C_A A_4) \Delta t x_2 y_3 y_4 + (F - C_A A_5) \Delta t x_2 y_3 y_4 y_5 \\
&- 2C_I (I_m - I_3) (\Delta t)^2 x_2 - 2C_I (I_m - I_4) (\Delta t)^2 (x_2 + x_2 y_3) \\
&- 2C_I (I_m - I_5) (\Delta t)^2 (x_2 + x_2 y_3 + x_2 y_3 y_4) - C_A Q_2 \Delta t
\end{aligned} \tag{D-30}$$

$$\begin{aligned}
\frac{\partial P_T}{\partial A_3} &= (F - C_A A_3) x_3 \Delta t + (F - C_A A_4) \Delta t x_3 y_4 + (F - C_A A_5) \Delta t x_3 y_4 y_5 \\
&- 2C_I (I_m - I_4) (\Delta t)^2 x_3 - 2C_I (I_m - I_5) (x_3 + x_3 y_4) (\Delta t)^2 \\
&- C_A Q_3 \Delta t
\end{aligned} \tag{D-31}$$

$$\begin{aligned}
\frac{\partial P_T}{\partial A_4} &= (F - C_A A_4) x_4 \Delta t + (F - C_A A_5) \Delta t x_4 y_5 \\
&- 2C_I (I_m - I_5) (\Delta t)^2 x_4 - C_A Q_4 \Delta t
\end{aligned} \tag{D-32}$$

and

$$\frac{\partial P_T}{\partial A_5} = (F - C_A A_5) x_5 \Delta t - C_A Q_5 \Delta t \tag{D-33}$$

Thus, equations (D-30) through (D-33) give the gradients for the 5-stage process.

D.3 A Production and Advertisement Model.

Here the total profit, P_T , is given by

$$\begin{aligned}
 P_T = & \sum_{k=1}^5 [c_1 Q(k\Delta t) + c_2 q x_2(k\Delta t) + c_3 q (1 - x_2(k\Delta t) - y_2(k\Delta t)) \\
 & - C_I (I_m - I(k\Delta t))^2 - C_A (A(k\Delta t) Q(k\Delta t))^2 \\
 & - C_T (T_m - T_1(k\Delta t))^2 - C_T (T_1(k\Delta t) - T_2(k\Delta t))^2] \Delta t
 \end{aligned} \tag{D-34}$$

The performance equations are

$$x_1(k\Delta t) = x_1(\overline{k-1}\Delta t) + \left[\frac{q}{v_1} (x_0 - x_1(\overline{k-1}\Delta t)) - k_{a1} x_1(\overline{k-1}\Delta t) \right] \Delta t \tag{D-35}$$

$$\begin{aligned}
 y_1(k\Delta t) = & y_1(\overline{k-1}\Delta t) + \left[\frac{q}{v_1} (y_0 - y_1(\overline{k-1}\Delta t)) - k_{b1} y_1(\overline{k-1}\Delta t) - \right. \\
 & \left. k_{a1} x_1(\overline{k-1}\Delta t) \right] \Delta t
 \end{aligned} \tag{D-36}$$

$$x_2(k\Delta t) = x_2(\overline{k-1}\Delta t) + \left[\frac{q}{v_2} (x_1(\overline{k-1}\Delta t) - x_2(\overline{k-1}\Delta t)) - k_{a2} x_2(\overline{k-1}\Delta t) \right] \Delta t \tag{D-37}$$

$$\begin{aligned}
 y_2(k\Delta t) = & y_2(\overline{k-1}\Delta t) + \left[\frac{q}{v_2} (y_1(\overline{k-1}\Delta t) - y_2(\overline{k-1}\Delta t)) - k_{b2} y_2(\overline{k-1}\Delta t) \right. \\
 & \left. - k_{a2} x_2(\overline{k-1}\Delta t) \right] \Delta t
 \end{aligned} \tag{D-38}$$

$$I(k\Delta t) = I(\overline{k-1}\Delta t) + [q y_2(\overline{k-1}\Delta t) - Q(\overline{k-1}\Delta t)] \Delta t \tag{D-39}$$

$$Q(k\Delta t) = Q(\overline{k-1}\Delta t) + [Q(\overline{k-1}\Delta t) (c + A(k\Delta t)) (1 - \frac{Q(\overline{k-1}\Delta t)}{N})] \Delta t \tag{D-40}$$

where $k = 1, 2, \dots, 5$ and $\Delta t = t_f/5$

Also we have

$$k_{a1} = G_a \exp(-E_a/RT_1) \quad (D-41)$$

$$k_{b1} = G_b \exp(-E_b/RT_1) \quad (D-42)$$

$$k_{a2} = G_a \exp(-E_a/RT_2) \quad (D-43)$$

$$k_{b2} = G_b \exp(-E_b/RT_2) \quad (D-44)$$

Since $c_2 = c_3 = 0$, the second and third terms may be dropped from equation (D-17). Thus

$$P_T = \sum_{k=1}^5 [c_1 Q(k\Delta t) - C_I (I_m - I(k\Delta t))^2 - C_A (A(k\Delta t) Q(k\Delta t))^2 - C_T (T_m - T_1(k\Delta t))^2 - C_T (T_1(k\Delta t) - T_2(k\Delta t))^2] \Delta t \quad (D-45)$$

Now we have A , T_1 and T_2 as the control variables and x_1, y_1, x_2, y_2 , I and Q as the state variables. For sake of simplicity

let

$$\begin{aligned} A_k &= A(k\Delta t) & x_{1k} &= x_1(k\Delta t) \\ T_{1k} &= T_1(k\Delta t) & x_{2k} &= x_2(k\Delta t) \\ T_{2k} &= T_2(k\Delta t) & y_{1k} &= y_1(k\Delta t) \\ Q_k &= Q(k\Delta t) & \text{and } y_{2k} &= y_2(k\Delta t) \\ I_k &= I(k\Delta t) \end{aligned}$$

Then the profit equation and the performance equations can be written as

$$P_T = \sum_{k=1}^5 [c_1 Q_k - C_I (I_m - I_k)^2 - C_A (A_k Q_k)^2 - C_T (T_m - T_{1k})^2 - C_T (T_{1k} - T_{2k})^2] \Delta t \quad (D-46)$$

$$x_{1k} = x_{1k-1} + \left[\frac{q}{v_1} (x_0 - x_{1k-1}) - k_{a1} x_{1k-1} \right] \Delta t \quad (D-47)$$

$$y_{1k} = y_{1k-1} + \left[\frac{q}{v_1} (y_0 - y_{1k-1}) - k_{b1} y_{1k-1} - k_{a1} x_{1k-1} \right] \Delta t \quad (D-48)$$

$$x_{2k} = x_{2k-1} + \left[\frac{q}{v_2} (x_{1k-1} - x_{2k-1}) - k_{a2} x_{2k-1} \right] \Delta t \quad (D-49)$$

$$y_{2k} = y_{2k-1} + \left[\frac{q}{v_2} (y_{1k-1} - y_{2k-1}) - k_{b2} y_{2k-1} - k_{a2} x_{2k-1} \right] \Delta t \quad (D-50)$$

$$I_k = I_{k-1} + [q y_{2k-1} - Q_{k-1}] \Delta t \quad (D-51)$$

$$\text{and } Q_k = Q_{k-1} + [Q_{k-1} (c + A_k) (1 - \frac{Q_{k-1}}{N})] \Delta t \quad (D-52)$$

Since A , T_1 and T_2 are the control variables, we need to evaluate the first derivatives of P_T with respect to A , T_1 and T_2 .

Let us first consider the derivatives with respect to advertisement at various stages. We have

$$\begin{aligned} \frac{\partial P_T}{\partial A_1} = & [(c_1 - 2C_A A_1^2 Q_1) \frac{\partial Q_1}{\partial A_1} + (c_1 - 2C_A A_2^2 Q_2) \frac{\partial Q_2}{\partial A_1} + \dots \\ & \dots + (c_1 - 2C_A A_5^2 Q_5) \frac{\partial Q_5}{\partial A_1} + 2C_I (I_m - I_1) \frac{\partial I_1}{\partial A_1} + \\ & 2C_I (I_m - I_2) \frac{\partial I_2}{\partial A_1} + \dots + 2C_I (I_m - I_5) \frac{\partial I_5}{\partial A_1} \\ & - 2C_A A_1 Q_1^2] \Delta t \end{aligned} \quad (D-53)$$

where, from equation (D-52)

$$\frac{\partial Q_1}{\partial A_1} = Q_0 (1 - Q_0/N) \Delta t$$

$$\text{and } \frac{\partial Q_k}{\partial A_1} = \frac{\partial Q_k}{\partial Q_{k-1}} \cdot \frac{\partial Q_{k-1}}{\partial Q_{k-2}} \dots \frac{\partial Q_1}{\partial A_1} \quad k \geq 2$$

Again, from (D-52), we have

$$\begin{aligned}\frac{\partial Q_k}{\partial Q_{k-1}} &= 1 + (c + A_k) \left(1 - 2 \frac{Q_{k-1}}{N}\right) \Delta t \\ &= a_k \text{ (say)}\end{aligned}$$

Equation (D-51) gives

$$\frac{\partial I_1}{\partial A_1} = 0$$

$$\frac{\partial I_2}{\partial A_1} = \frac{\partial I_2}{\partial Q_1} \cdot \frac{\partial Q_1}{\partial A_1}$$

$$= - \Delta t Q_0 \left(1 - \frac{Q_0}{N}\right) \Delta t$$

$$\frac{\partial I_3}{\partial A_1} = \frac{\partial I_3}{\partial I_2} \cdot \frac{\partial I_2}{\partial A_1} + \frac{\partial I_3}{\partial Q_2} \cdot \frac{\partial Q_2}{\partial A_1}$$

$$= - Q_0 \left(1 - \frac{Q_0}{N}\right) (\Delta t)^2 - \Delta t a_2 Q_0 \left(1 - \frac{Q_0}{N}\right) \Delta t$$

$$= - Q_0 \left(1 - \frac{Q_0}{N}\right) (\Delta t)^2 (1 + a_2)$$

$$\frac{\partial I_4}{\partial A_1} = \frac{\partial I_4}{\partial I_3} \cdot \frac{\partial I_3}{\partial A_1} + \frac{\partial I_4}{\partial Q_3} \cdot \frac{\partial Q_3}{\partial A_1}$$

$$= - Q_0 \left(1 - \frac{Q_0}{N}\right) (\Delta t)^2 [1 + a_2 + a_2 a_3]$$

Similarly

$$\frac{\partial I_5}{\partial A_1} = - Q_0 \left(1 - \frac{Q_0}{N}\right) (\Delta t)^2 [1 + a_2 + a_2 a_3 + a_2 a_3 a_4]$$

Substituting the values in equation (D-53), we have

$$\begin{aligned}
\frac{\partial P_T}{\partial A_1} = & Q_0 \left(1 - \frac{Q_0}{N}\right) (\Delta t)^3 [(c_1 - 2C_A A_1^2 Q_1) + (c_1 - 2C_A A_1^2 Q_2) a_2 \\
& + \dots + (c_1 - 2C_A A_5^2 Q_5) a_2 a_3 a_4 a_5 - \{2C_I (I_m - I_2) \\
& + 2C_I (I_m - I_3) (1 + a_2) + \dots + 2C_I (I_m - I_5) (1 + a + a_2 a_3 \\
& + a_2 a_3 a_4)\} - 2C_A A_1^2 Q_1^2 \Delta t]
\end{aligned} \tag{D-54}$$

Similarly we obtain

$$\begin{aligned}
\frac{\partial P_T}{\partial A_2} = & Q_1 \left(1 - \frac{Q_1}{N}\right) (\Delta t)^3 [(c_1 - 2C_A Q_2 A_2^2) + (c_1 - 2C_A A_3^2 Q_3) a_3 \\
& + (c_1 - 2C_A A_4^2 Q_4) a_3 a_4 + (c_1 - 2C_A A_5^2 Q_5) a_3 a_4 a_5 \\
& - \{2C_I (I_m - I_3) + 2C_I (I_m - I_4) (1 + a_3) + \\
& 2C_I (I_m - I_5) (1 + a_3 + a_3 a_4)\}] - 2C_A A_2^2 Q_2^2 \Delta t
\end{aligned} \tag{D-55}$$

$$\begin{aligned}
\frac{\partial P_T}{\partial A_3} = & Q_2 \left(1 - \frac{Q_2}{N}\right) (\Delta t)^3 [(c_1 - 2C_A Q_3 A_3^2) + (c_1 - 2C_A A_4^2 Q_4) a_4 \\
& + (c_1 - 2C_A A_5^2 Q_5) a_4 a_5 - \{2C_I (I_m - I_4) + \\
& 2C_I (I_m - I_5) (1 + a_4)\}] - 2C_A A_3^2 Q_3^2 \Delta t
\end{aligned} \tag{D-56}$$

$$\begin{aligned}
\frac{\partial P_T}{\partial A_4} = & Q_3 \left(1 - \frac{Q_3}{N}\right) (\Delta t)^3 [(c_1 - 2C_A Q_4 A_4^2) + (c_1 - 2C_A Q_5 - A_5^2) a_5 \\
& - 2C_I (I_m - I_5)] - 2C_A A_4^2 Q_4^2 \Delta t
\end{aligned} \tag{D-57}$$

and

$$\frac{\partial P_T}{\partial A_5} = Q_4 \left(1 - \frac{Q_4}{N}\right) (\Delta t)^3 (c_1 - 2C_A Q_5 A_5^2) - 2C_A A_5 Q_5^2 \Delta t \quad (D-58)$$

Let us now compute the first derivatives of the profit function with respect to the temperatures in the two reactors at various stages.

In order to simplify the presentation of the calculations, the following quantities are used to denote the values mentioned against each.

$$p_j = \frac{\partial x_{1j+1}}{\partial x_{1j}} = 1 - \frac{q}{v} (1 + G_a \exp(-E_a/RT_{1j}) \Delta t)$$

$$r_j = \frac{\partial y_{1j+1}}{\partial y_{1j}} = 1 - \frac{q}{v} (1 + G_b \exp(-E_b/RT_{1j}) \Delta t)$$

$$s_j = \frac{\partial y_{1j+1}}{\partial x_{1j}} = G_a \exp(-E_a/RT_{1j}) \Delta t$$

$$u_j = \frac{\partial x_{2j+1}}{\partial x_{2j}} = 1 - \frac{q}{v} (1 + G_a \exp(-E_a/RT_{2j}) \Delta t)$$

$$w_j = \frac{\partial y_{2j+1}}{\partial y_{2j}} = 1 - \frac{q}{v} (1 + G_b \exp(-E_b/RT_{2j}) \Delta t)$$

$$z_j = \frac{\partial y_{2j+1}}{\partial x_{2j}} = G_a \exp(-E_a/RT_{2j}) \Delta t$$

$$b_{ij} = \frac{\partial k_{ai}}{\partial T_{ij}} = G_a \exp(-E_a/RT_{ij}) \left(\frac{E_a}{RT_{ij}^2}\right) \quad i = 1, 2$$

$$d_{ij} = \frac{\partial k_{bi}}{\partial T_{ij}} = G_b \exp(-E_b/RT_{ij}) \left(\frac{E_b}{RT_{ij}^2}\right) \quad i = 1, 2$$

$$\alpha_{ij} = \frac{\partial x_{1j}}{\partial T_{ij}} = -\Delta t x_{ij-1} b_{ij} \quad i = 1, 2$$

$$\beta_{ij} = \frac{\partial y_{1j}}{\partial T_{ij}} = -\Delta t [y_{ij} d_{ij} + x_{ij} b_{ij}] \quad i = 1, 2$$

Now, from equation (D-46), we have

$$\begin{aligned} \frac{\partial P_T}{\partial T_{11}} = & [2C_I(I_m - I_1) \frac{\partial I_1}{\partial T_{11}} + 2C_I(I_m - I_2) \frac{\partial I_2}{\partial T_{11}} + \dots + 2C_I(I_m - I_5) \frac{\partial I_5}{\partial T_{11}} \\ & + 2C_T(T_m - T_{11}) - 2C_T(T_{11} - T_{21})] \Delta t \end{aligned} \quad (D-59)$$

where $\frac{\partial I_k}{\partial T_{11}}$; $i = 1, 2, \dots, 5$, are given as follows

$$\frac{\partial I_1}{\partial T_{11}} = 0$$

$$\frac{\partial I_2}{\partial T_{11}} = \frac{\partial I_2}{\partial y_{21}} \cdot \frac{\partial y_{21}}{\partial T_{11}}$$

$$= q \Delta t (0) = 0$$

$$\frac{\partial I_3}{\partial T_{11}} = \frac{\partial I_3}{\partial I_2} \cdot \frac{\partial I_2}{\partial T_{11}} + \frac{\partial I_3}{\partial y_{22}} \cdot \frac{\partial y_{22}}{\partial T_{11}}$$

$$= 0 + q \Delta t \left[\frac{\partial y_{22}}{\partial y_{11}} \cdot \frac{\partial y_{11}}{\partial T_{11}} \right]$$

$$= q \Delta t \left[\frac{q}{v} \Delta t \beta_{11} \right]$$

$$= \frac{q^2}{v} (\Delta t)^2 \beta_{11}$$

$$\frac{\partial I_4}{\partial T_{11}} = \frac{\partial I_4}{\partial I_3} \cdot \frac{\partial I_3}{\partial T_{11}} + \frac{\partial I_4}{\partial y_{23}} \cdot \frac{\partial y_{23}}{\partial T_{11}}$$

$$= \frac{\partial I_3}{\partial T_{11}} + q \Delta t \left[\frac{\partial y_{23}}{\partial y_{22}} \cdot \frac{\partial y_{22}}{\partial T_{11}} + \frac{\partial y_{23}}{\partial y_{12}} \cdot \frac{\partial y_{12}}{\partial T_{11}} + \frac{\partial y_{23}}{\partial x_{22}} \cdot \frac{\partial x_{22}}{\partial T_{11}} \right]$$

$$= \frac{\partial I_3}{\partial T_{11}} + q \Delta t \left[w_2 \frac{q}{v} \Delta t \beta_{11} + \frac{q}{v} \Delta t \left(\frac{\partial y_{12}}{\partial y_{11}} \cdot \frac{\partial y_{11}}{\partial T_{11}} + \frac{\partial y_{12}}{\partial x_{11}} \cdot \frac{\partial x_{11}}{\partial T_{11}} \right) \right]$$

$$\begin{aligned}
& + z_2 \left(\frac{\partial x_{22}}{\partial x_{21}} \cdot \frac{\partial x_{21}}{\partial T_{11}} + \frac{\partial x_{22}}{\partial x_{11}} \cdot \frac{\partial x_{11}}{\partial T_{11}} \right)] \\
& = \frac{\partial I_3}{\partial T_{11}} + q \Delta t \left\{ w_2 \frac{q}{v} \Delta t \beta_{11} + \frac{q}{v} \Delta t (r_1 \beta_{11} + s_1 \alpha_{11}) \right. \\
& \quad \left. + z_2 (u_1 \cdot 0 + \frac{q}{v} \Delta t \cdot \alpha_{11}) \right\} \\
& = \frac{\partial I_3}{\partial T_{11}} + \frac{q^2}{v} (\Delta t)^2 \{ w_2 \beta_{11} + r_1 \beta_{11} + s_1 \alpha_{11} + z_2 \alpha_{11} \} \\
\frac{\partial I_5}{\partial T_{11}} & = \frac{\partial I_5}{\partial I_4} \cdot \frac{\partial I_4}{\partial T_{11}} + \frac{\partial I_5}{\partial y_{24}} \cdot \frac{\partial y_{24}}{\partial T_{11}} \\
& = \frac{\partial I_4}{\partial T_{11}} + q \Delta t \left\{ \frac{\partial y_{24}}{\partial y_{23}} \cdot \frac{\partial y_{23}}{\partial T_{11}} + \frac{\partial y_{24}}{\partial y_{13}} \cdot \frac{\partial y_{13}}{\partial T_{11}} + \frac{\partial y_{24}}{\partial x_{23}} \cdot \frac{\partial x_{23}}{\partial T_{11}} \right\} \\
& = \frac{\partial I_4}{\partial T_{11}} + q \Delta t \left\{ w_3 \cdot \frac{\partial y_{23}}{\partial T_{11}} + \frac{q}{v} t \frac{\partial y_{13}}{\partial T_{11}} + z_3 \frac{\partial x_{23}}{\partial T_{11}} \right\}
\end{aligned}$$

In the above equation

$$\frac{\partial y_{23}}{\partial T_{11}} = \frac{q}{v} \Delta t (w_2 \beta_{11} + r_1 \beta_{11} + s_1 \alpha_{11} + z_2 \alpha_{11})$$

$$\frac{\partial y_{13}}{\partial T_{11}} = \frac{\partial y_{13}}{\partial y_{12}} \left(\frac{\partial y_{12}}{\partial y_{11}} \cdot \frac{\partial y_{11}}{\partial T_{11}} + \frac{\partial y_{12}}{\partial x_{11}} \cdot \frac{\partial x_{11}}{\partial T_{11}} \right)$$

$$+ \frac{\partial y_{13}}{\partial x_{12}} \cdot \frac{\partial x_{12}}{\partial x_{11}} \cdot \frac{\partial x_{11}}{\partial T_{11}}$$

$$= r_2 (r_1 \beta_{11} + s_1 \alpha_{11}) + s_2 p_1 \alpha_{11}$$

and
$$\frac{\partial x_{23}}{\partial T_{11}} = \frac{\partial x_{23}}{\partial x_{22}} \cdot \frac{\partial x_{22}}{\partial T_{11}} + \frac{\partial x_{23}}{\partial x_{12}} \cdot \frac{\partial x_{12}}{\partial T_{11}}$$

$$\begin{aligned}
&= \frac{\partial x_{23}}{\partial x_{22}} \cdot \frac{\partial x_{22}}{\partial x_{11}} \cdot \frac{\partial x_{11}}{\partial T_{11}} + \frac{\partial x_{23}}{\partial x_{12}} \cdot \frac{\partial x_{12}}{\partial x_{11}} \cdot \frac{\partial x_{11}}{\partial T_{11}} \\
&= u_2 \frac{q}{v} \Delta t \alpha_{11} + \frac{q}{v} \Delta t u_1 \alpha_{11} \\
&= \frac{q}{v} \Delta t \alpha_{11} (u_2 + u_1)
\end{aligned}$$

The first derivative of P_T with respect to the temperature T_1 at the second stage is given by

$$\frac{\partial P_T}{\partial T_{12}} = \left[\sum_{k=1}^5 \{ 2C_I (I_m - I_k) \frac{\partial I_k}{\partial T_{12}} \} + 2C_T (T_m - T_{12}) - 2C_T (T_{12} - T_{22}) \right] \Delta t$$

Since $\frac{\partial I_1}{\partial T_{12}} = 0$, $\frac{\partial I_2}{\partial T_{12}} = 0$ and $\frac{\partial I_3}{\partial T_{12}} = 0$, the

above equation can be written as

$$\begin{aligned}
\frac{\partial P_T}{\partial T_{12}} &= [2C_I (I_m - I_4) \frac{\partial I_4}{\partial T_{12}} + 2C_I (I_m - I_5) \frac{\partial I_5}{\partial T_{12}} \\
&\quad + 2 (T_m - T_{12}) - 2 (T_{12} - T_{22})] \Delta t
\end{aligned} \tag{D-60}$$

Here, the only unknowns are $\frac{\partial I_4}{\partial T_{12}}$ and $\frac{\partial I_5}{\partial T_{12}}$.

These may be computed as follows

$$\begin{aligned}
\frac{\partial I_4}{\partial T_{12}} &= \frac{\partial I_4}{\partial I_3} \cdot \frac{\partial I_3}{\partial T_{12}} + \frac{\partial I_4}{\partial y_{23}} \cdot \frac{\partial y_{23}}{\partial T_{12}} \\
&= \frac{\partial I_4}{\partial y_{23}} \cdot \frac{\partial y_{23}}{\partial T_{12}} \quad \text{since } \frac{\partial I_3}{\partial T_{12}} = 0 \\
&= q \Delta t \left(\frac{\partial y_{23}}{\partial y_{12}} \cdot \frac{\partial y_{12}}{\partial T_{12}} \right)
\end{aligned}$$

$$= q \Delta t \left(\frac{q}{v} \Delta t \beta_{12} \right)$$

$$= \frac{q^2}{v} (\Delta t)^2 \beta_{12}$$

$$\frac{\partial I_5}{\partial T_{12}} = \frac{\partial I_5}{\partial I_4} \cdot \frac{\partial I_4}{\partial T_{12}} + \frac{\partial I_5}{\partial y_{24}} \cdot \frac{\partial y_{24}}{\partial T_{12}}$$

$$= 1 \cdot \frac{\partial I_4}{\partial T_{12}} + q \Delta t \left\{ \frac{\partial y_{24}}{\partial y_{23}} \cdot \frac{\partial y_{23}}{\partial T_{12}} + \frac{\partial y_{24}}{\partial y_{13}} \cdot \frac{\partial y_{13}}{\partial T_{12}} + \frac{\partial y_{24}}{\partial x_{23}} \cdot \frac{\partial x_{23}}{\partial T_{12}} \right\}$$

$$= \frac{\partial I_4}{\partial T_{12}} + q \Delta t \left\{ w_3 \cdot \left(\frac{\partial y_{23}}{\partial y_{12}} \cdot \frac{\partial y_{12}}{\partial T_{12}} \right) + \frac{q}{v} \Delta t \left(\frac{\partial y_{13}}{\partial y_{12}} \cdot \frac{\partial y_{12}}{\partial T_{12}} + \right. \right.$$

$$\left. \frac{\partial y_{13}}{\partial x_{12}} \cdot \frac{\partial x_{12}}{\partial T_{12}} \right) + z_3 \left(\frac{\partial x_{23}}{\partial x_{12}} \cdot \frac{\partial x_{12}}{\partial T_{12}} \right) \}$$

$$= \frac{q^2}{v} (\Delta t)^2 \beta_{12} + q \Delta t \left\{ w_3 \cdot \frac{q}{v} \cdot \Delta t \beta_{12} + \frac{q}{v} \Delta t (r_2 \beta_{12} \right.$$

$$+ s_2 \alpha_{12}) + z_3 \cdot \frac{q}{v} \Delta t \alpha_{12} \}$$

$$= \frac{q^2}{v} (\Delta t)^2 [\beta_{12} + w_3 \beta_{12} + r_2 \beta_{12} + s_2 \alpha_{12} + z_3 \alpha_{12}]$$

The derivative of P_T with respect to T_{13} is given by

$$\frac{\partial P_T}{\partial T_{13}} = 2C_I (I_m - I_5) \frac{\partial I_5}{\partial T_{13}} + 2C_T (T_m - T_{13}) - 2C_T (T_{13} - T_{23}) \quad (D-61)$$

where

$$\frac{\partial I_5}{\partial T_{13}} = \frac{\partial I_5}{\partial I_4} \cdot \frac{\partial I_4}{\partial T_{13}} + \frac{\partial I_5}{\partial y_{24}} \cdot \frac{\partial y_{24}}{\partial T_{13}}$$

$$= 0 + q \Delta t \left[\frac{\partial y_{24}}{\partial y_{13}} \cdot \frac{\partial y_{13}}{\partial T_{13}} \right]$$

$$= q \Delta t \frac{q}{v} \Delta t \cdot \beta_{13}$$

$$= \frac{q}{v} (\Delta t)^2 \beta_{13}$$

$$\frac{\partial P_T}{\partial T_{14}} = 2C_T (T_m - T_{14}) - 2C_T (T_{14} - T_{24}) \quad (D-62)$$

$$\text{and } \frac{\partial P_T}{\partial T_{15}} = 2C_T (T_m - T_{15}) - 2C_T (T_{15} - T_{25}) \quad (D-63)$$

The first derivatives of P_T with respect to T_2 at various stages are given as follows

$$\frac{\partial P_T}{\partial T_{21}} = \left[\sum_{k=1}^5 \{2C_I (I_m - I_k) \cdot \frac{\partial I_k}{\partial T_{2k}}\} + 2C_T (T_{11} - T_{21}) \right] \Delta t \quad (D-64)$$

where

$$\frac{\partial I_1}{\partial T_{21}} = 0$$

$$\frac{\partial I_2}{\partial T_{21}} = \frac{\partial I_2}{\partial y_{21}} \cdot \frac{\partial y_{21}}{\partial T_{21}}$$

$$= q \Delta t \beta_{21}$$

$$\frac{\partial I_3}{\partial T_{21}} = \frac{\partial I_3}{\partial I_2} \cdot \frac{\partial I_2}{\partial T_{21}} + \frac{\partial I_3}{\partial y_{22}} \cdot \frac{\partial y_{22}}{\partial T_{21}}$$

$$= q \Delta t \cdot \beta_{21} + q \Delta t \left\{ \frac{\partial y_{22}}{\partial y_{21}} \cdot \frac{\partial y_{21}}{\partial T_{21}} + \frac{\partial y_{22}}{\partial x_{21}} \cdot \frac{\partial x_{21}}{\partial T_{21}} \right\}$$

$$= q \Delta t \beta_{21} + q \Delta t \{w_1 \beta_{21} + z_1 \alpha_{21}\}$$

$$\begin{aligned}
\frac{\partial I_4}{\partial T_{21}} &= \frac{\partial I_4}{\partial I_3} \cdot \frac{\partial I_3}{\partial T_{21}} + \frac{\partial I_4}{\partial y_{23}} \cdot \frac{\partial y_{23}}{\partial T_{21}} \\
&= \frac{\partial I_3}{\partial T_{21}} + q \cdot \Delta t \left[\frac{\partial y_{23}}{\partial y_{22}} \cdot \frac{\partial y_{22}}{\partial T_{21}} + \frac{\partial y_{23}}{\partial x_{22}} \cdot \frac{\partial x_{22}}{\partial T_{21}} \right] \\
&= \frac{\partial I_3}{\partial T_{21}} + q \cdot \Delta t [w_2(w_1 \beta_{21} + z_1 \alpha_{21}) + z_2 u_1 \alpha_{21}]
\end{aligned}$$

$$\begin{aligned}
\frac{\partial I_5}{\partial T_{21}} &= \frac{\partial I_5}{\partial I_4} \cdot \frac{\partial I_4}{\partial T_{21}} + \frac{\partial I_5}{\partial y_{24}} \cdot \frac{\partial y_{24}}{\partial T_{21}} \\
&= \frac{\partial I_4}{\partial T_{21}} + q \cdot \Delta t \left[\frac{\partial y_{24}}{\partial y_{23}} \cdot \frac{\partial y_{23}}{\partial T_{21}} + \frac{\partial y_{24}}{\partial x_{23}} \cdot \frac{\partial x_{23}}{\partial T_{21}} \right] \\
&= \frac{\partial I_4}{\partial T_{21}} + q \cdot \Delta t [w_3 \{w_2(w_1 \beta_{21} + z_1 \alpha_{21}) + z_2 u_1 \alpha_{21}\} + z_3 u_2 u_1 \alpha_{21}]
\end{aligned}$$

The partial derivative of P_T with respect to T_2 at second stage is given by

$$\frac{\partial P_T}{\partial T_{22}} = \left[\sum_{k=3}^5 \{2C_I(I_m - I_k) \frac{\partial I_k}{\partial T_{22}}\} + 2C_T(T_{12} - T_{22}) \right] \Delta t \quad (D-65)$$

where

$$\begin{aligned}
\frac{\partial I_3}{\partial T_{22}} &= \frac{\partial I_3}{\partial y_{22}} \cdot \frac{\partial y_{22}}{\partial T_{22}} \\
&= q \Delta t \beta_{22}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial I_4}{\partial T_{22}} &= \frac{\partial I_4}{\partial I_3} \cdot \frac{\partial I_3}{\partial T_{22}} + \frac{\partial I_4}{\partial y_{23}} \cdot \frac{\partial y_{23}}{\partial T_{22}} \\
&= \frac{\partial I_3}{\partial T_{22}} + \frac{\partial I_4}{\partial y_{23}} \left\{ \frac{\partial y_{23}}{\partial y_{22}} \cdot \frac{\partial y_{22}}{\partial T_{22}} + \frac{\partial y_{23}}{\partial x_{22}} \cdot \frac{\partial x_{22}}{\partial T_{22}} \right\} \\
&= q \Delta t \beta_{22} + q \cdot \Delta t \{ w_2 \beta_{22} + z_2 \alpha_{22} \}
\end{aligned}$$

and

$$\begin{aligned}
\frac{\partial I_5}{\partial T_{22}} &= \frac{\partial I_5}{\partial I_4} \cdot \frac{\partial I_4}{\partial T_{22}} + \frac{\partial I_5}{\partial y_{24}} \cdot \frac{\partial y_{24}}{\partial T_{22}} \\
&= \frac{\partial I_4}{T_{22}} + q \cdot \Delta t \left\{ \frac{\partial y_{24}}{y_{23}} \cdot \frac{\partial y_{23}}{T_{22}} + \frac{\partial y_{24}}{x_{23}} \cdot \frac{\partial x_{23}}{x_{22}} \cdot \frac{\partial x_{22}}{T_{22}} \right\} \\
&= \frac{\partial I_4}{\partial T_{22}} + q \Delta t [w_3 \{w_2 \beta_{22} + z_2 \alpha_{22}\} + z_3 u_2 \alpha_{22}] \\
&= q \Delta t \beta_{22} + q \Delta t (w_2 \beta_{22} + z_2 \alpha_{22}) + q \Delta t [w_3 (w_2 \beta_{22} + z_2 \alpha_{22}) + z_3 u_2 \alpha_{22}]
\end{aligned}$$

The partial derivative of P_T with respect to T_2 at third stage is given by

$$\begin{aligned}
\frac{\partial P_T}{\partial T_{23}} &= [2C_I (I_m - I_4) \frac{\partial I_4}{\partial T_{23}} + 2C_I (I_m - I_5) \frac{\partial I_5}{\partial T_{23}} \\
&\quad + 2C_T (T_{13} - T_{23})] \Delta t
\end{aligned} \tag{D-66}$$

where

$$\begin{aligned}\frac{\partial I_4}{\partial T_{23}} &= \frac{\partial I_4}{\partial y_{23}} \cdot \frac{\partial y_{23}}{\partial T_{23}} \\ &= q \Delta t \beta_{23}\end{aligned}$$

and

$$\begin{aligned}\frac{\partial I_5}{\partial T_{23}} &= \frac{\partial I_5}{\partial I_4} \cdot \frac{\partial I_4}{\partial T_{23}} + \frac{\partial I_5}{\partial y_{24}} \cdot \frac{\partial y_{24}}{\partial T_{23}} \\ &= \frac{\partial I_4}{\partial T_{23}} + \frac{\partial I_5}{\partial y_{24}} \left[\frac{\partial y_{24}}{\partial y_{23}} \cdot \frac{\partial y_{23}}{\partial T_{23}} + \frac{\partial y_{24}}{\partial x_{23}} \cdot \frac{\partial x_{23}}{\partial T_{23}} \right] \\ &= q \Delta t \beta_{23} + q \Delta t [w_3 \beta_{23} + z_3 \alpha_{23}]\end{aligned}$$

The derivatives of P_T with respect to T_2 at 4th and 5th stages are given by

$$\begin{aligned}\frac{\partial P_T}{\partial T_{24}} &= [2C_I(I_m - I_5) \frac{\partial I_5}{\partial T_{24}} + 2C_T(T_{14} - T_{24})] \Delta t \\ &= [2C_I(I_m - I_5) q \Delta t \beta_{24} + 2C_T(T_{14} - T_{24})] \Delta t\end{aligned}\tag{D-67}$$

and

$$\frac{\partial P_T}{\partial T_{25}} = 2C_T(T_{15} - T_{25}) \Delta t\tag{D-68}$$

Equations (D-54) through (D-68) represent the gradients for the 5-stage production and advertisement model.

APPENDIX E

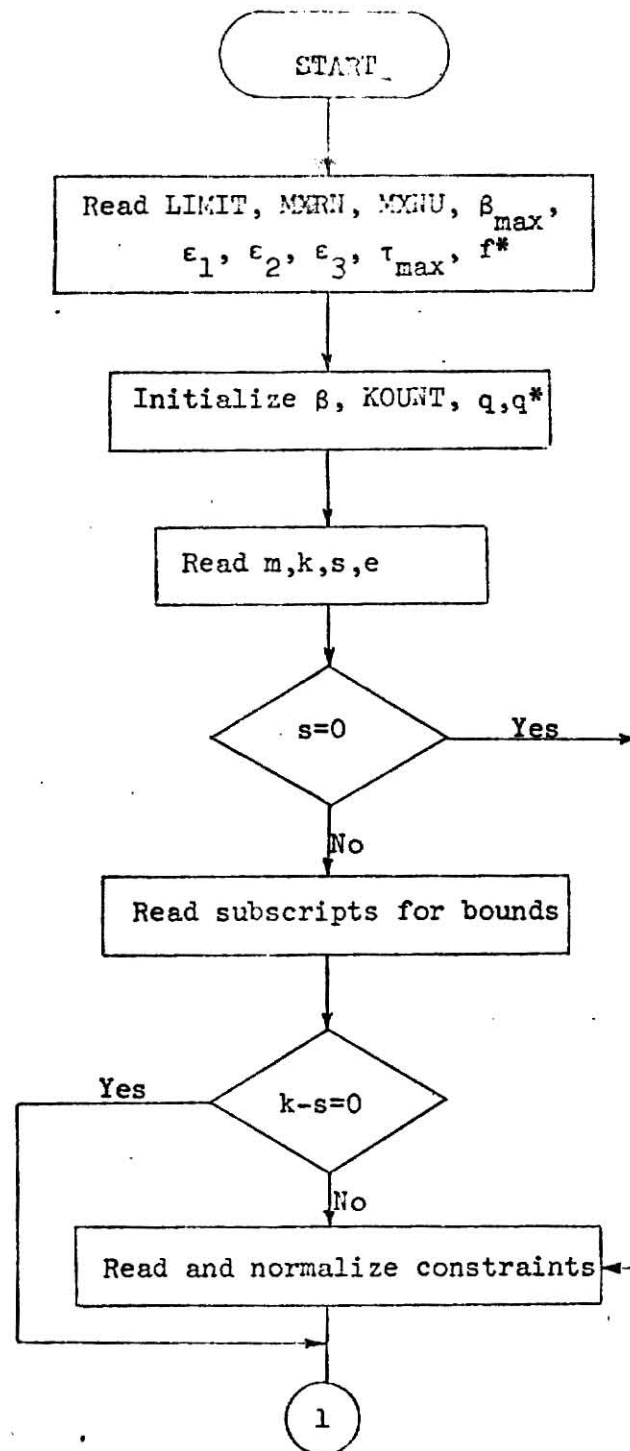
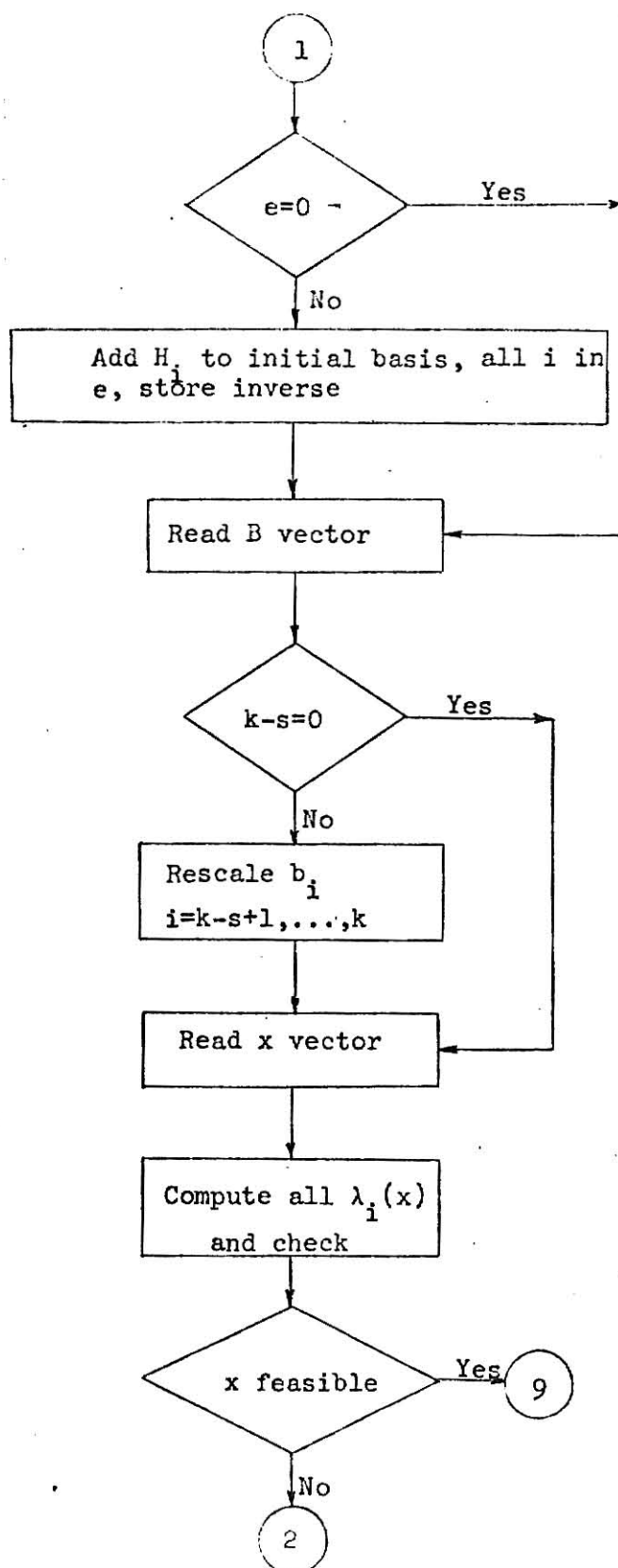
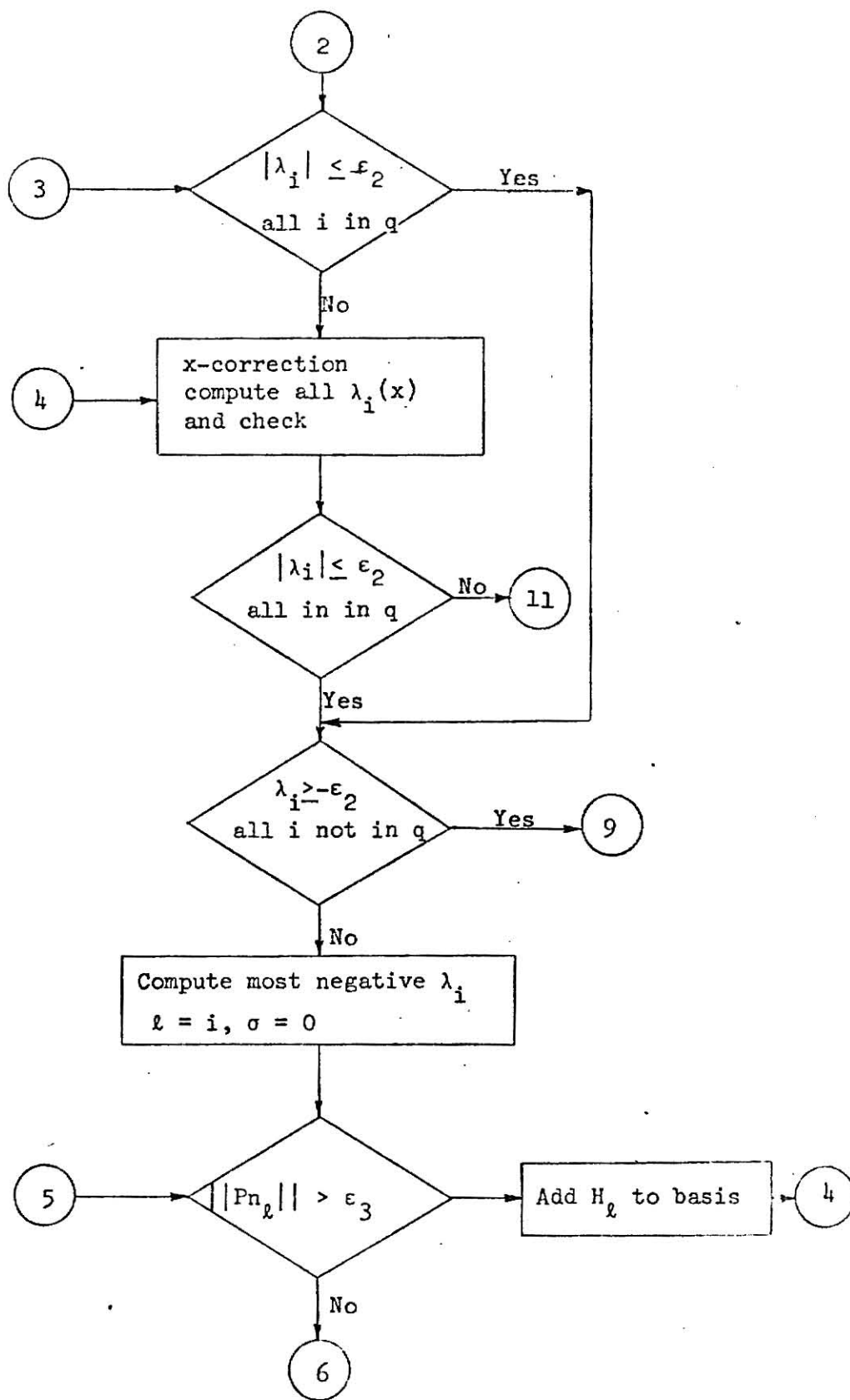
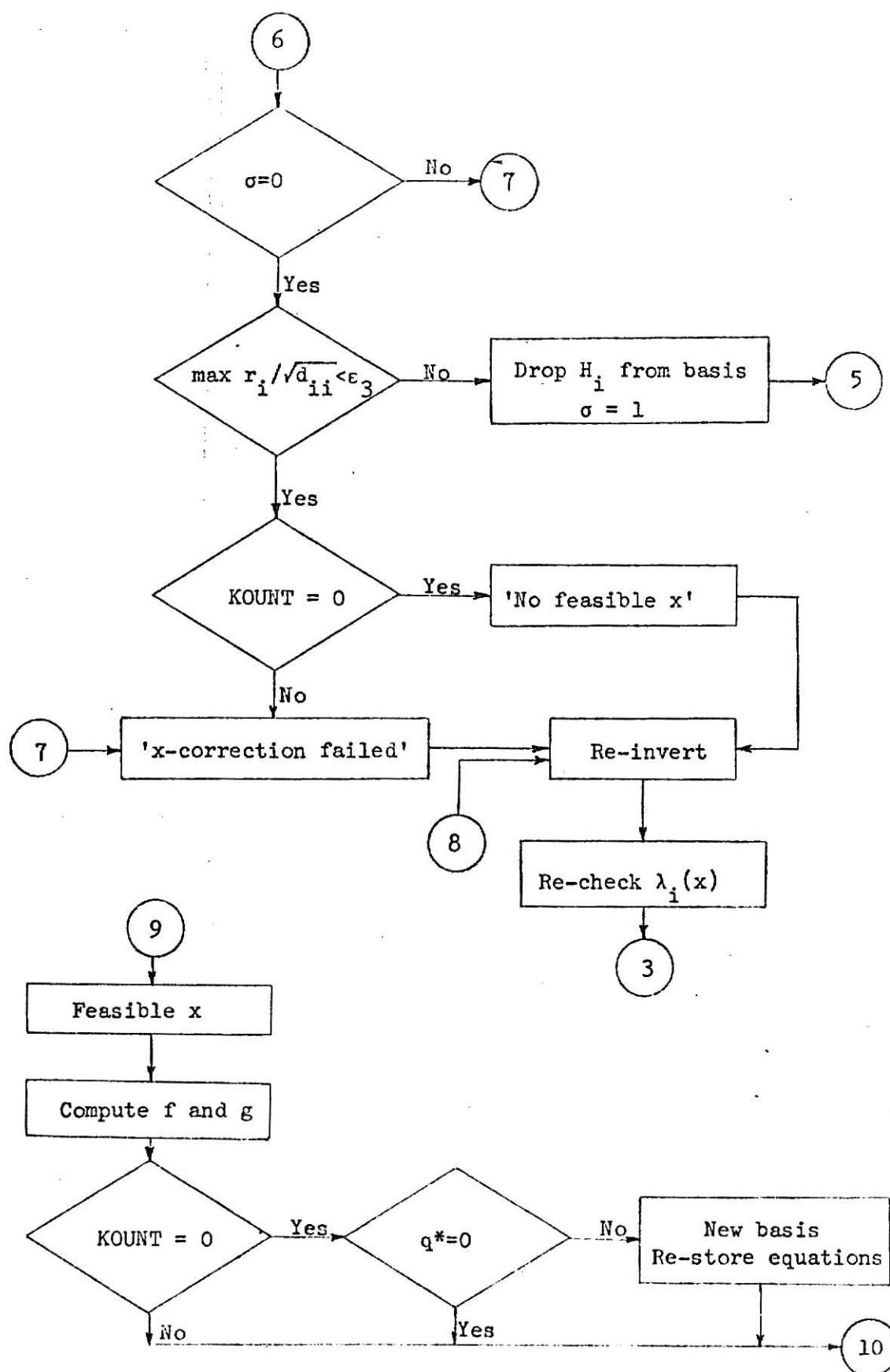
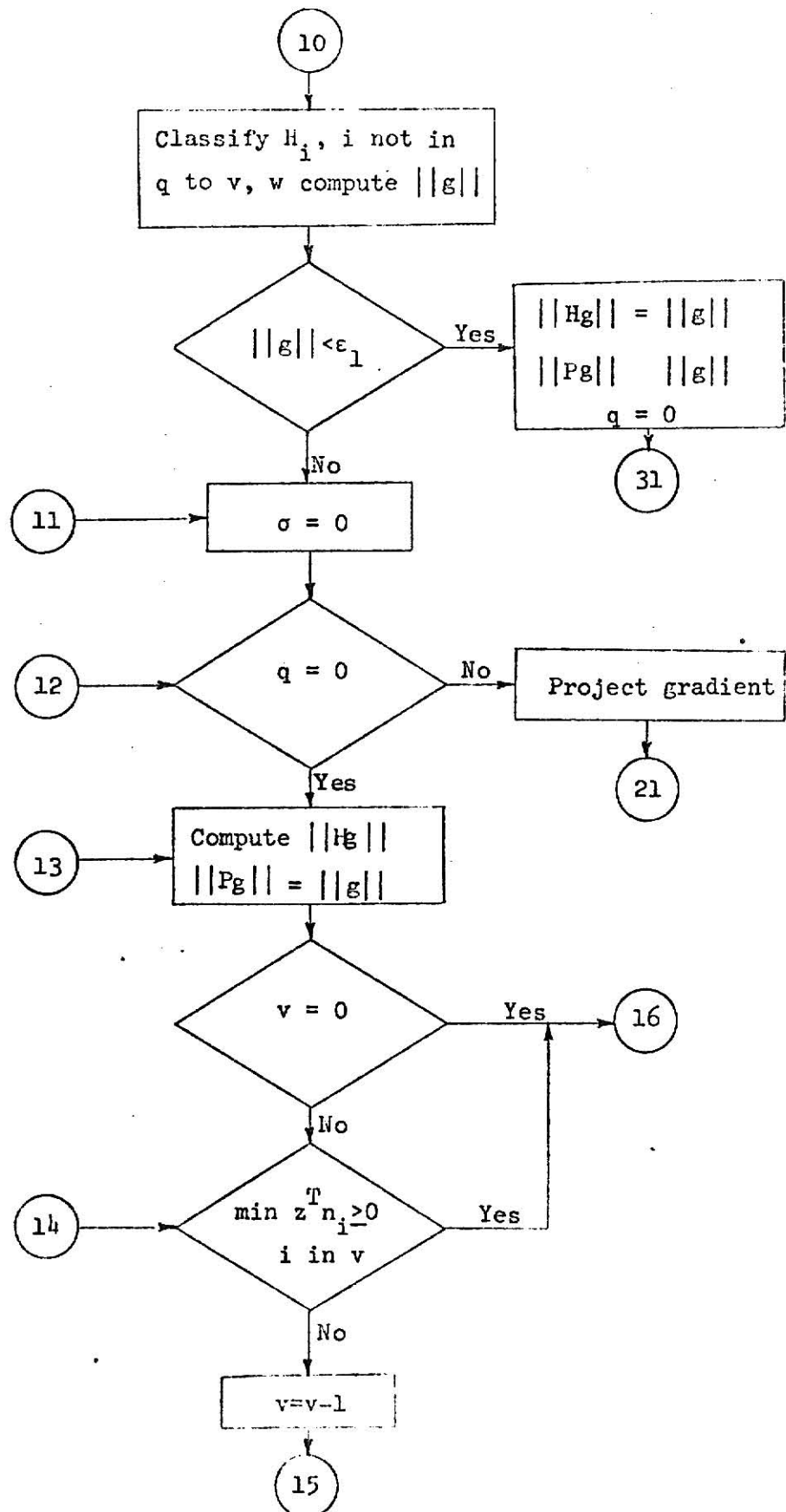


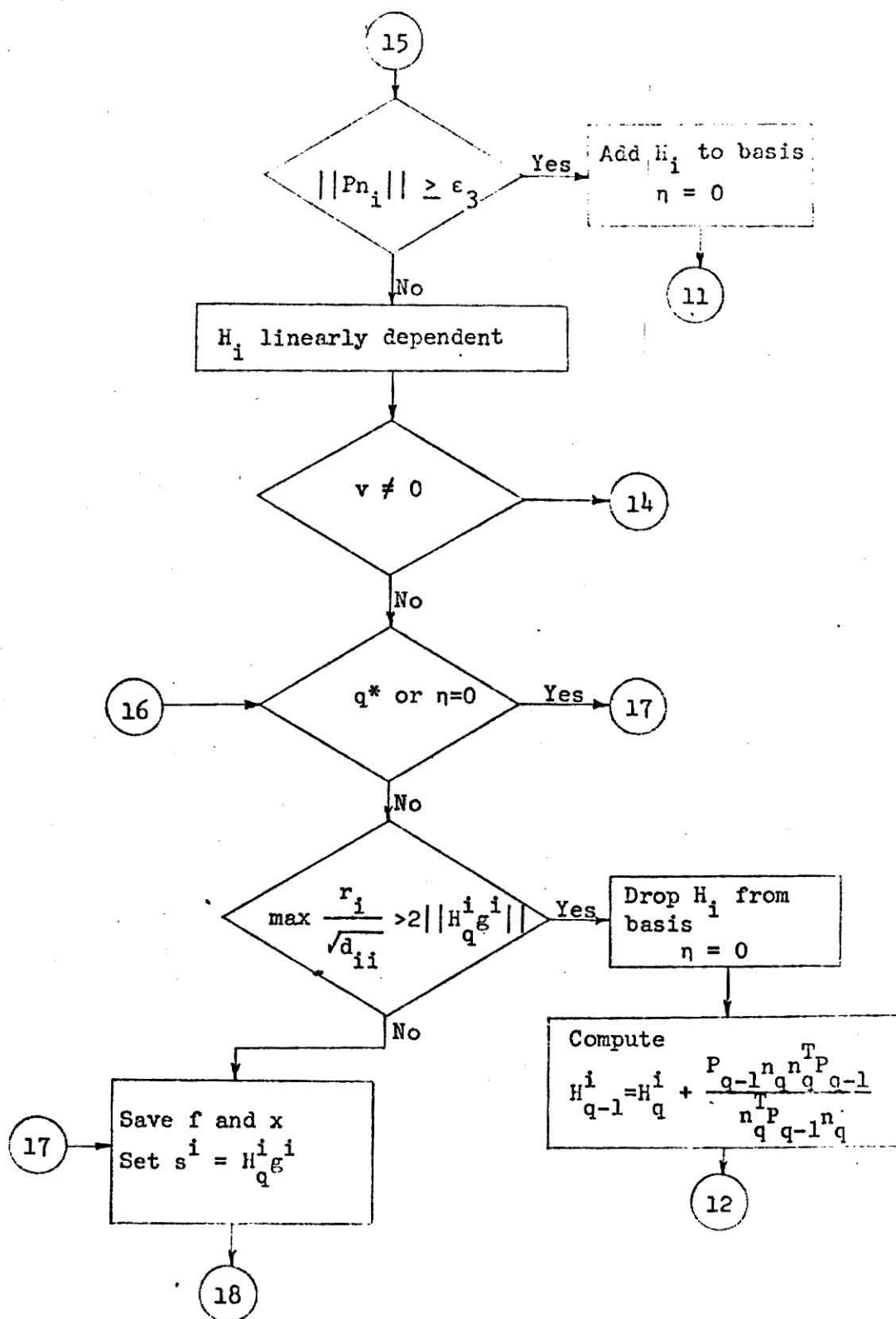
Fig.12 Flow Chart for the Main Program of the Conjugate Gradient Method

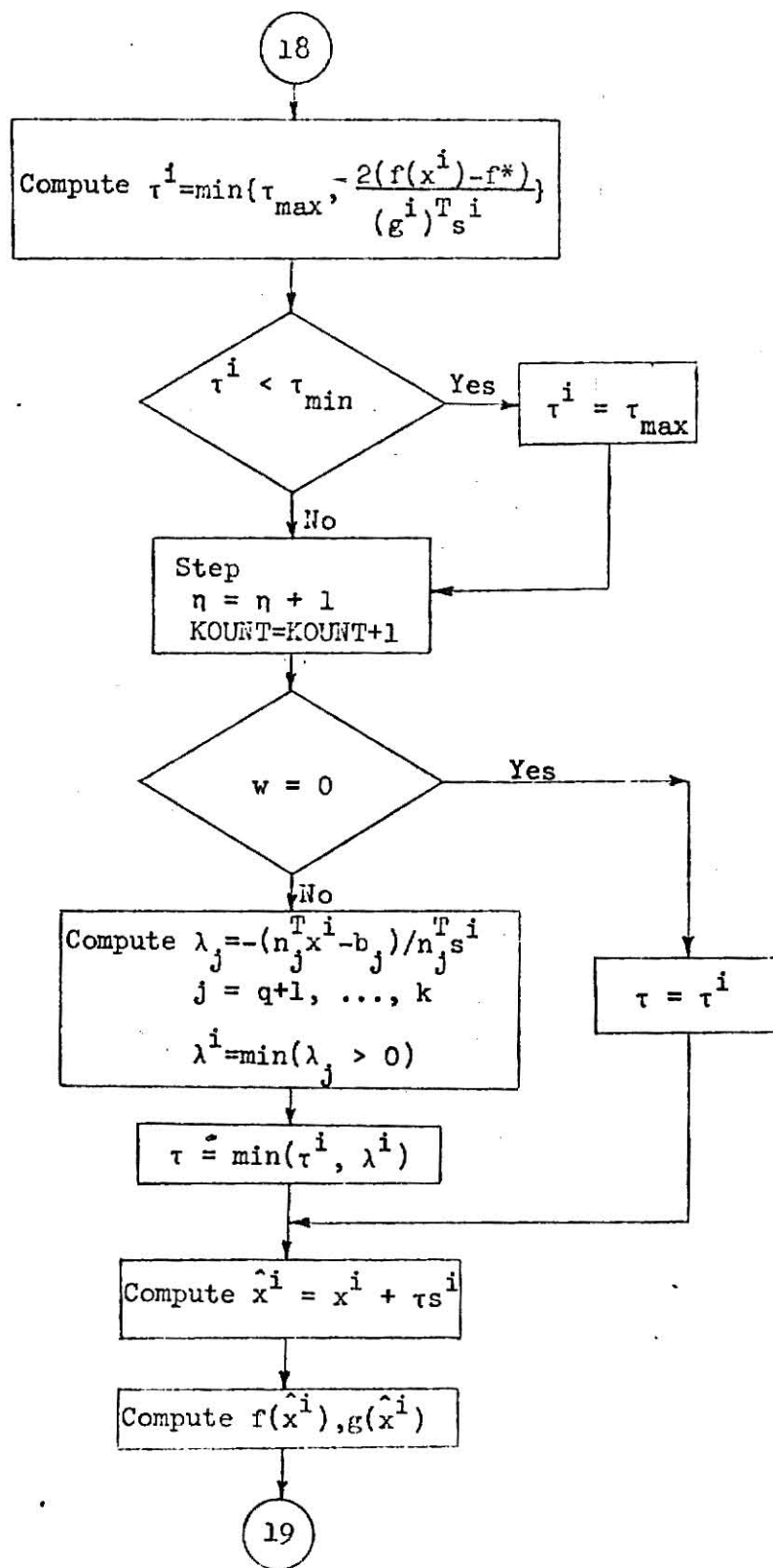


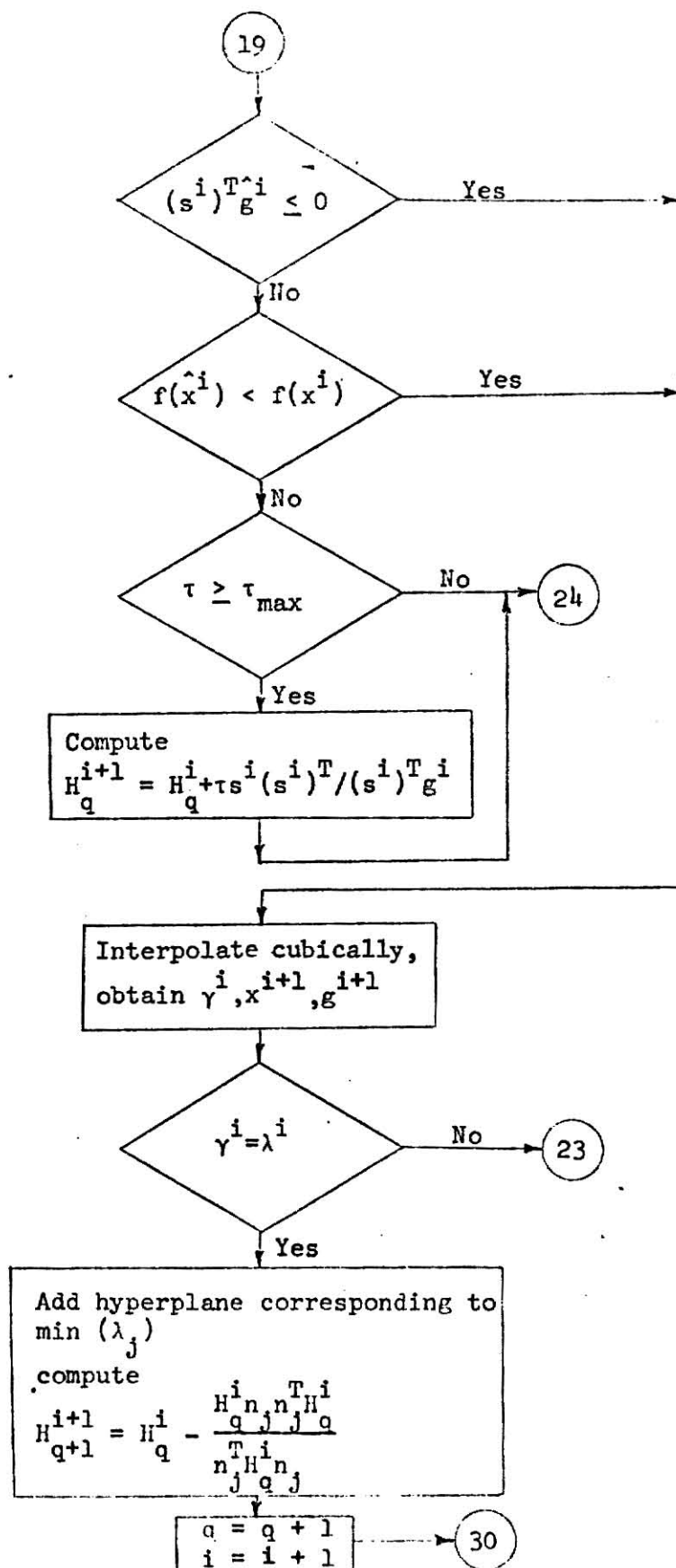


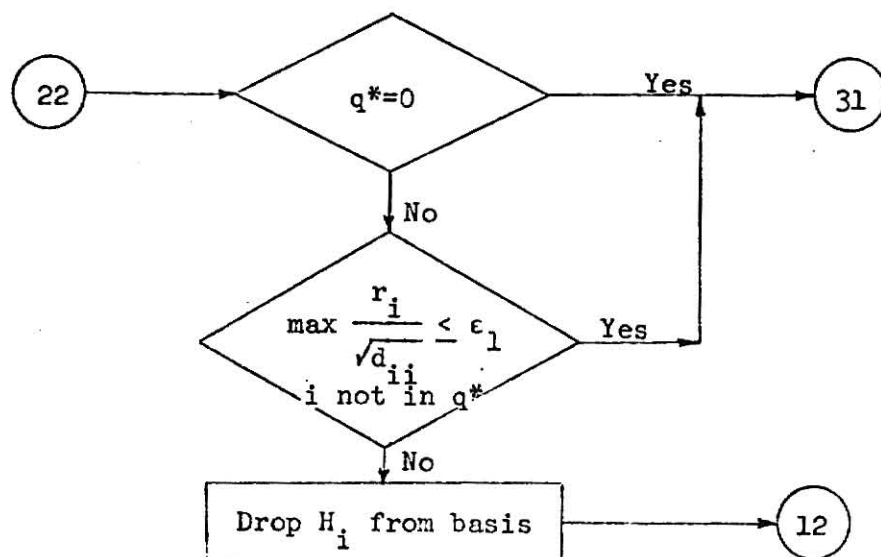
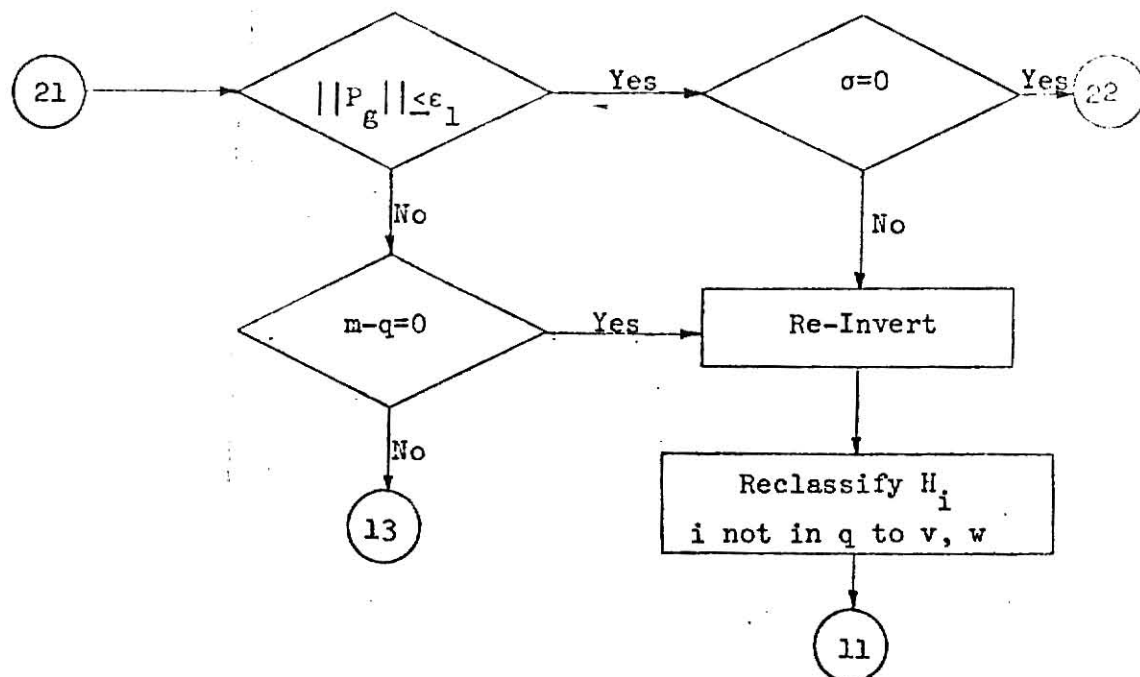


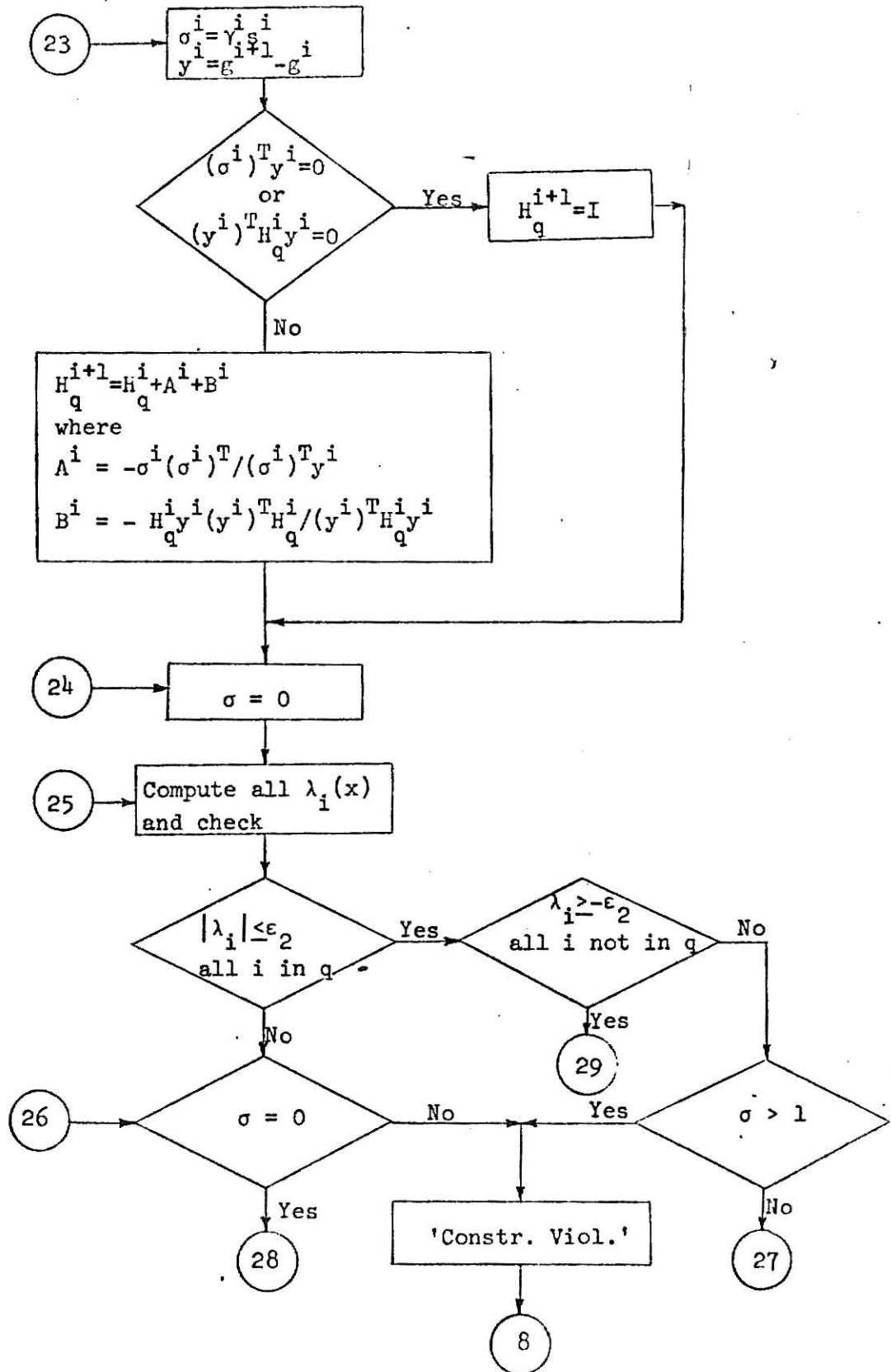


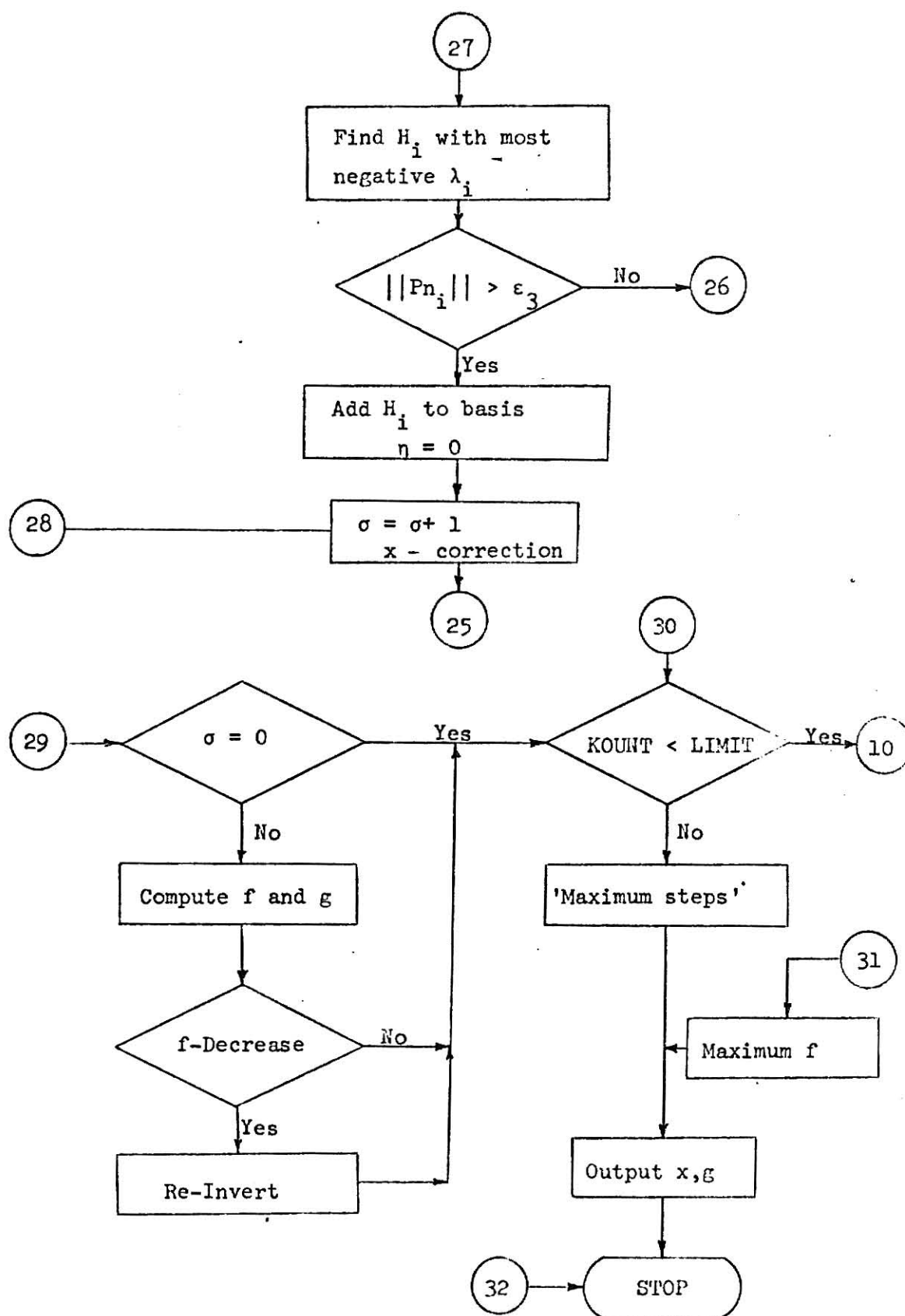












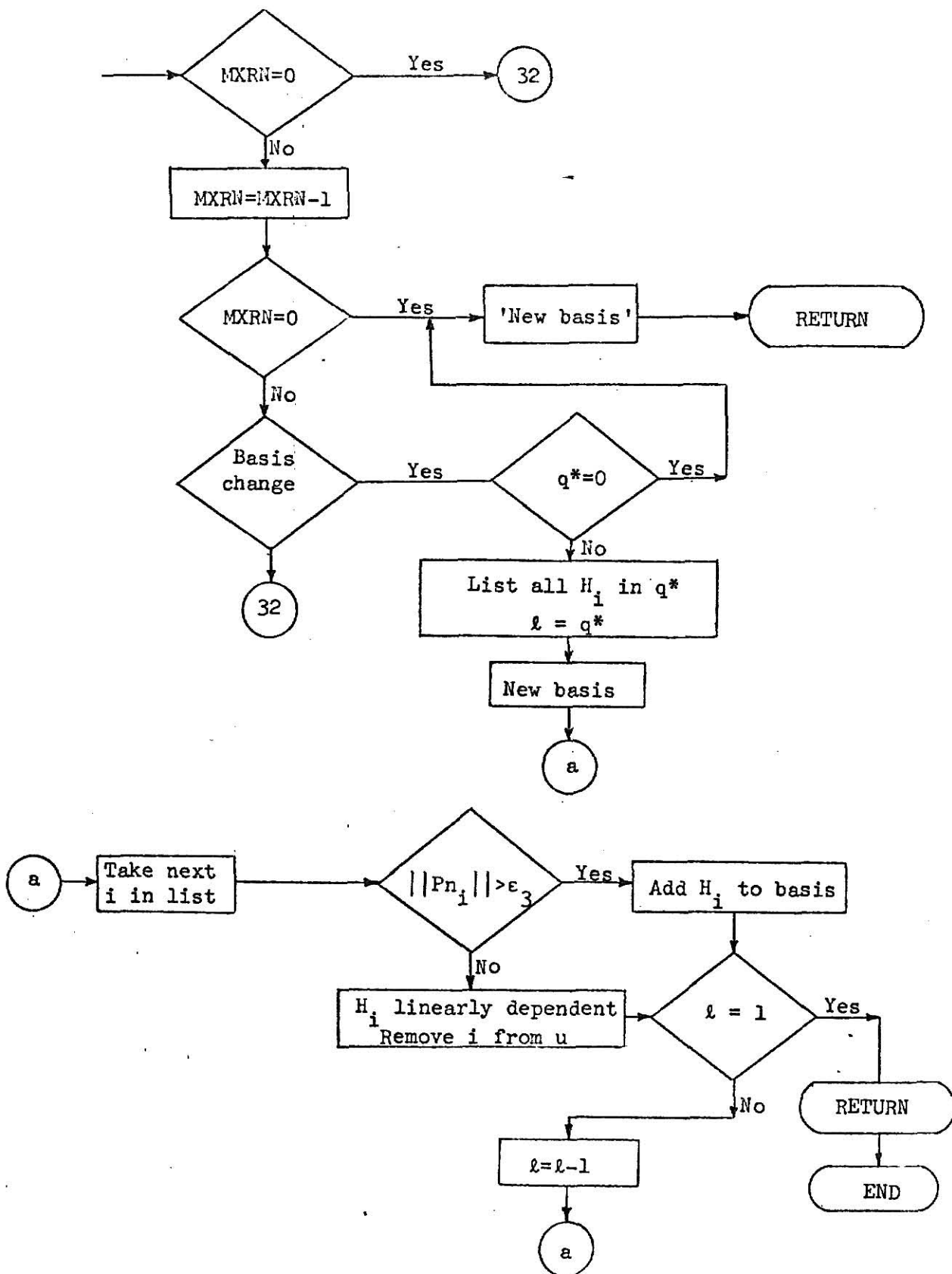


Fig. 13 Flow Chart for Subroutine REINV

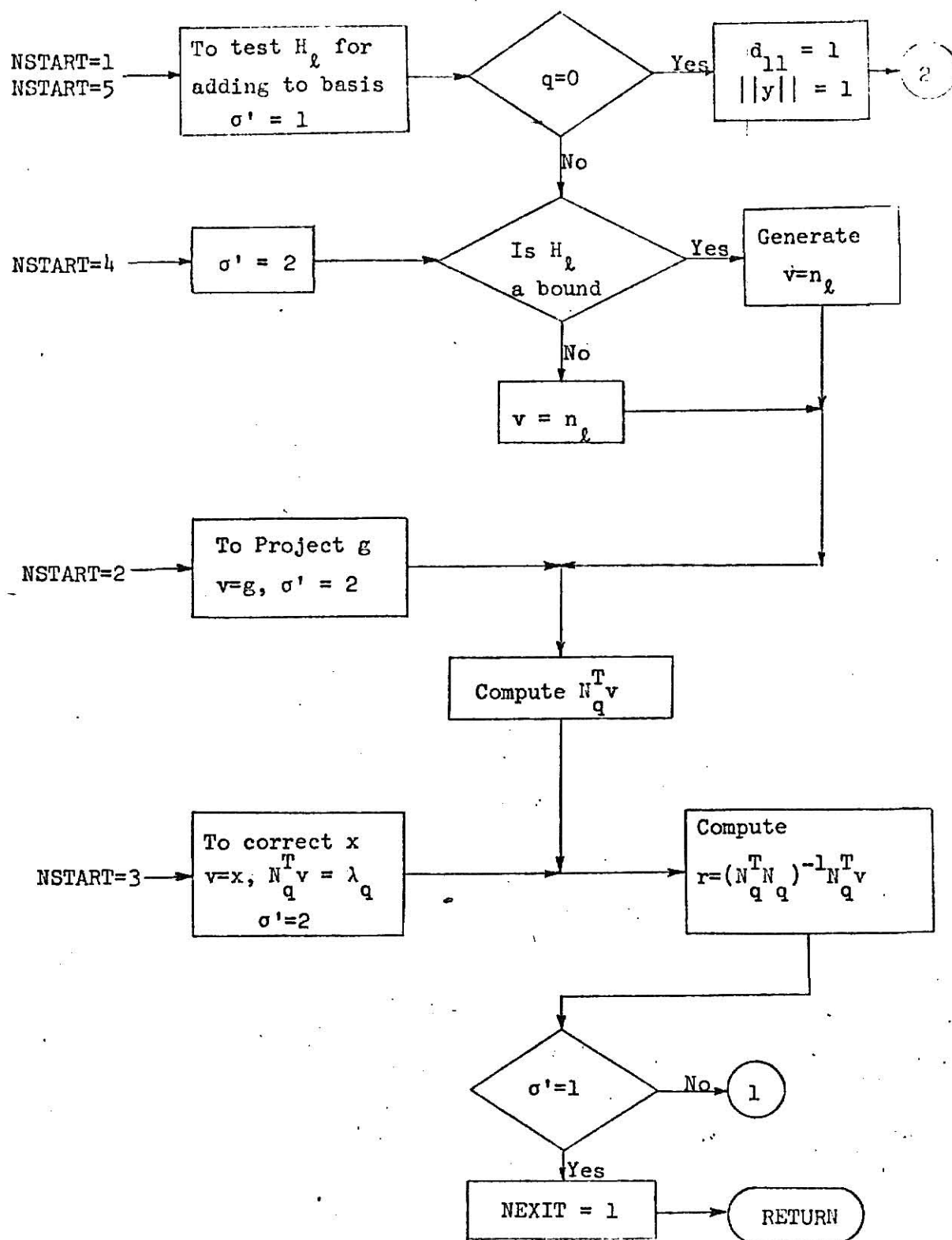
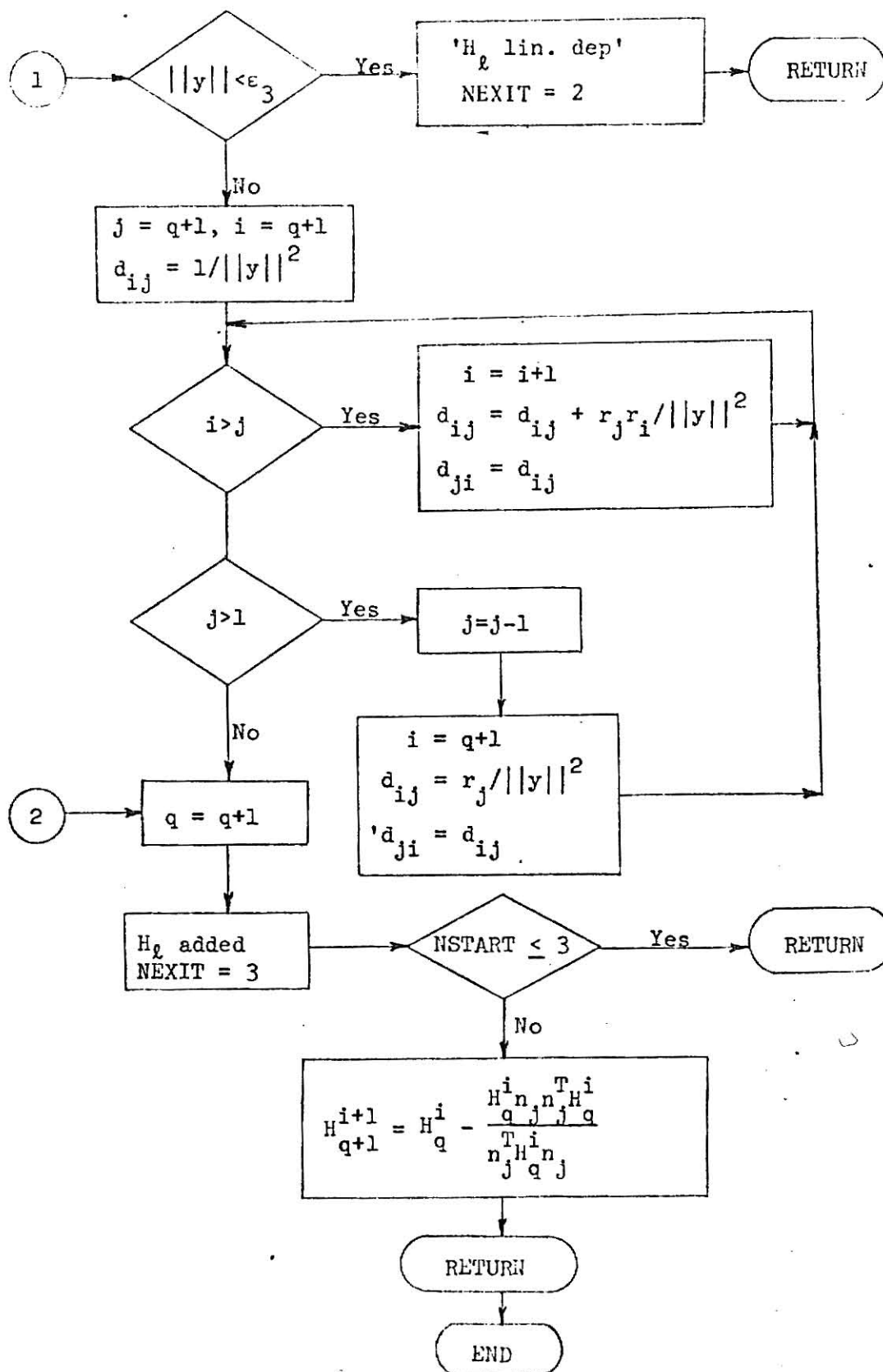


Fig. 14 Flow Chart for Subroutine COMPl



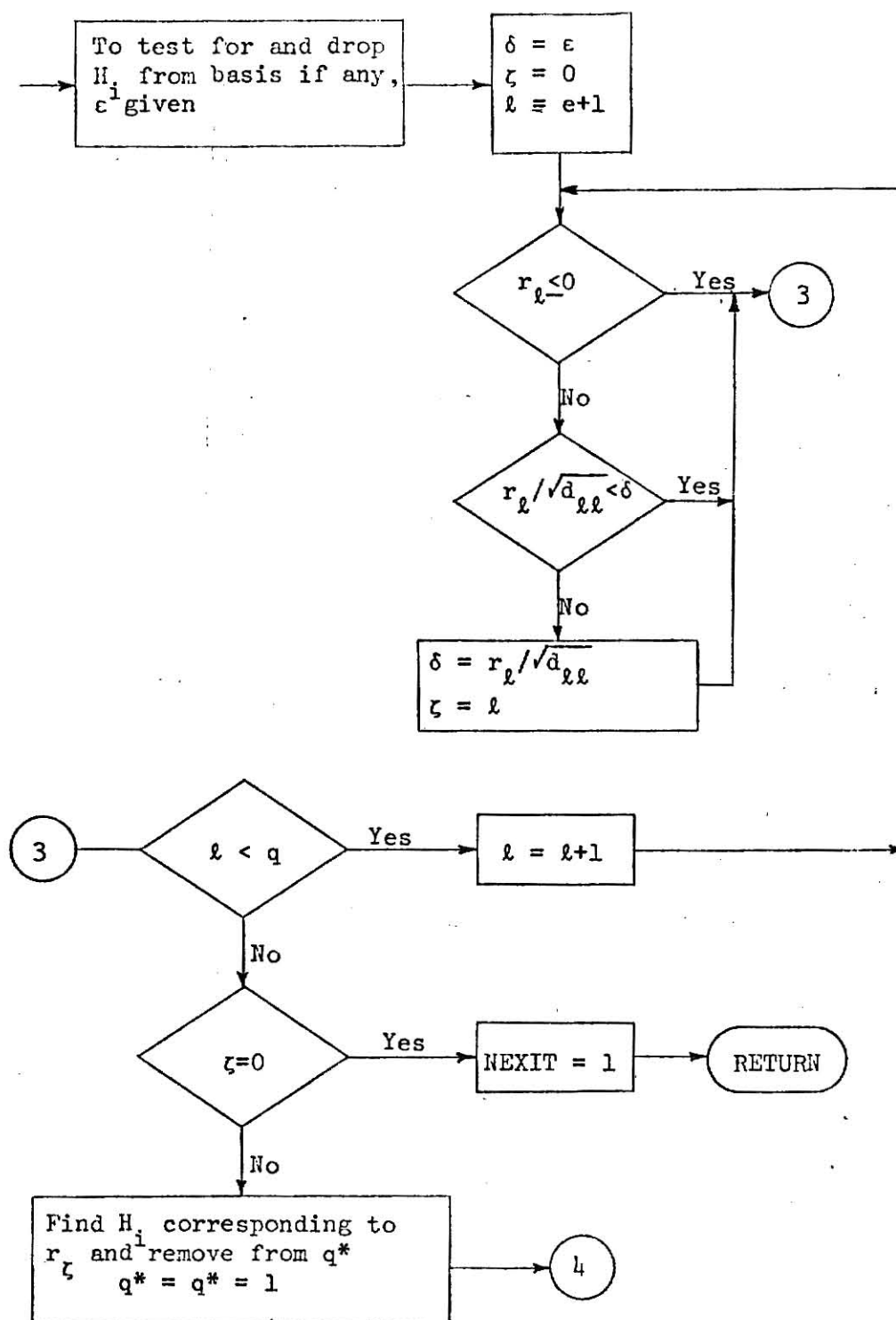
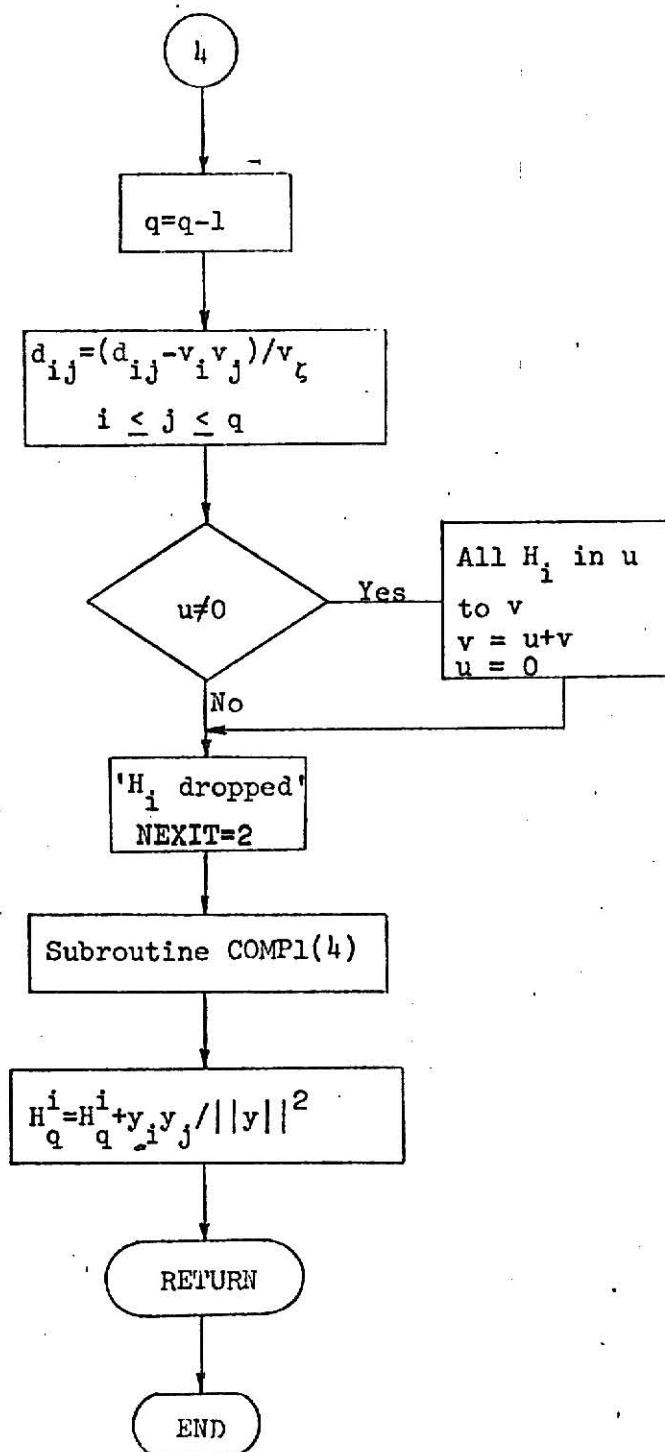


Fig. 15 Flow Chart for Subroutine COMP2



APPENDIX F

```

C      COMPUTER PROGRAM FOR THE CONJUGATE GRADIENT METHOD
C
      DIMENSION G(20),X(20),P(20),V(20),JXB(40),IHI(20),R(20),
1SD(40),RR(20),Z(20),XO(20),GO(20),X1(20),DELX(20),X2(20),Y(20),
1AMBDA(40),FG(20),IU(40),IV(40),IW(40),B(40),PG(20),AX(20)
      DIMENSION D(20,20),DN(20,20),A(40,20),H(20,20),YY(20),HG(20)
      COMMON Y,PN,AMBDA,M,G,X,F,P,EPSI1,NEXIT,KQ,D,IH,NB,V,A,JXB,IHI,
1EPSI3,EPSI2,LD,IU,R,KEQ,MEXIT,IV,KV,B,KMNB,IW,KW,K,MXRN,INV,DN,
1MU,NETA,INT,LDC,H
      COMMON/ARO/NFUNC,KOUNT,NOFE
      EQUIVALENCE (Y,PG),(PN,PGNORM),(SD,AMBDA),(Z,HG)
909  FORMAT(F10.4)
910  FORMAT(4I4,4F10.5)
1000 FORMAT(4(I5))
1010 FORMAT(/' INITIAL X')
1020 FORMAT(20I4)
1040 FORMAT(6F12.6)
1050 FORMAT(6F12.6)
1060 FORMAT(I3)
C
C      NOMENCLATURE
C
C      NSP = NUMBER OF SETS OF STARTING POINTS
C      LIMIT = MAXIMUM NUMBER OF ITERATIONS
C      MXRN = MAXIMUM NUMBER OF RE-INVERSIONS
C      MUMAX = MAXIMUM NUMBER OF INTERPOLATIONS
C      EPSI1 = GRADIENT TOLERANCE
C      EPSI2 = CONSTRAINT TOLERANCE
C      EPSI3 = LINEAR DEPENDENCE TOLERANCE
C      TMAX = MAXIMUM STEP SIZE
C      M = NUMBER OF VARIABLES
C      K = NUMBER OF CONSTRAINTS INCLUDING BOUNDS
C      NB = NUMBER OF BOUNDS
C      NE = NUMBER OF EQUALITY CONSTRAINTS
C      KMNB = NUMBER OF CONSTRAINTS EXCLUDING BOUNDS
C      KQ = NUMBER OF CONSTRAINTS IN THE PRESENT BASIS
C      KEQ = NUMBER OF CONSTRAINTS IN THE INITIAL BASIS
C      LDC = NUMBER OF LINEARLY DEPENDENT CONSTRAINS
C      INV = REINVERSION FLAG USED TO DETERMINE IF ANY CHANGE IN BASIS
C      IS MADE BETWEEN TWO REINVERSIONS
C
      READ 1060,NSP
      READ 910, LIMIT,MXRN,MUMAX,INTMAX,EPSI1,EPSI2,EPSI3,TMAX
      READ 909, ESTF
C
C      INITIALIZE COUNTERS
C
      IN = 0
      NOFE=0
10  INT=0

```

```

      NFUNC=0
      KQ=0
      KEQ=0
      LDC=0
      INV=1
11  MU=0
      KOUNT=0
      IN = IN + 1
      IF(IN-1) 12,12,27
12  TMIN = 0.0001*TMAX
      READ 1000,M,K,NB,NE
      NBP1=NB+1
      KMNB=K-NB
      IF(NB)24,24,22
C
C      READ AND PRINT SUBSCRIPTS FOR BOUNDS
C
22  READ 1020,((JXB(I),I=1,NB)
      PRINT 1020,((JXB(I),I=1,NB)
      IF(KMNB)27,27,24
C
C      READ AND PRINT CONSTRAINTS
C
24  READ 1040,((A(I,J),J=1,M),I=1,KMNB)
      PRINT 1040,((A(I,J),J=1,M),I=1,KMNB)
      DO 26 I=1,KMNB
C
C      NORMALIZE CONSTRAINTS
C
      SD(I)=0.
      DO 25 J=1,M
25  SD(I)=SD(I)+A(I,J)**2
      SD(I)=SQRT(SD(I))
      DO 26 J=1,M
26  A(I,J)=A(I,J)/SD(I)
      PRINT 1040, ((A(I,J),J=1,M),I=1,KMNB)
27  IF(NE)29,29,28
29  DO 606 I=1,M
      DO 605 J=1,M
605  H(I,J)=0.
606  H(I,I)=1.
      GO TO 30
C
C      SET SUBSCRIPTS FOR EQUALITY CONSTRAINTS, ADD H TO INITIAL
C      BASIS FOR ALL I IN KE, STORE INVERSE
C
28  KQ=0
      IH=NB+1
      DO 610 I=1,NE
610  IW(I)=0

```

```

      CALL COMP1(1)
      IHI(1)=IH
615  IF(NE-KQ) 675,675,620
620  N=1
      INDEX=0
      DELTA=EPSI3
625  N=N+1
      IF(IW(N)) 630,630,650
630  DO 635 J=1,M
635  V(J)=A(N,J)
      CALL COMP1(2)
      PN=0.0
      DO 640 J=1,M
640  PN=PN+V(J)**2
      PN=SQRT(PN)
      IF(PN-DELTA) 650,650,645
645  DELTA=PN
      INDEX=N
650  IF(N-NE) 625,655,655
655  IF(INDEX) 675,675,660
660  IH=INDEX+NB
      CALL COMP1(1)
      IW(INDEX)=777
      IHI(KQ)=IH
      GO TO 615
675  KEQ=KQ
      DO 676 I=1,M
      DO 676 J=1,KEQ
      DN(I,J)=0.
      DO 676 N=1,KEQ
      KK=IHI(N)-NB
676  DN(I,J)=DN(I,J)+A(KK,I)*D(N,J)
      DO 678 I=1,M
      DO 677 J=1,M
      H(I,J)=0.
      DO 677 N=1,KEQ
      KK=IHI(N)-NB
677  H(I,J)=H(I,J)-A(KK,I)*DN(J,N)
678  H(I,I)=H(I,I)+1.
      DO 680 I=1,KQ
      DO 680 J=1,KQ
680  DN(I,J)=D(I,J)
30  IF(IN-1) 33,33,45

```

```

C
C      READ AND PRINT RIGHT HAND SIDE OF CONSTRAINTS
C
33  READ 1050,(B(I),I=1,K)
    PRINT 1050,(B(I),I=1,K)
    IF(KMNB) 40,40,31
31  DO 32 I=1,KMNB

```



```

      J=I+NB
32  B(J)=B(J)/SD(I)
      PRINT 1040,(B(I),I=1,K)
40  CONTINUE
C
C      READ AND PRINT INITIAL X-VECTOR
C
45  READ 1040,(X(I),I=1,M)
      PRINT 1010
      PRINT 1040,(X(I),I=1,M)
C
C      COMPUTE ALL LAMBDA'S AND CHECK
C
52  CALL AMDA
      IF(NB) 54,54,64
64  DO 53 I=1,NB
      IF(AMBDA(I)+EPSI2) 59,53,53
53  CONTINUE
54  IF(NBP1-K)65,65,130
65  DO 57 J=NBP1,K
      IF(J-NBP1-NE) 55,56,56
55  IF(ABS(AMPDA(J))-EPSI2) 57,57,59
56  IF(AMBDA(J)+EPSI2) 59,57,57
57  CONTINUE
      GO TO 130
59  PRINT 1040,(X(I),I=1,M)
      PRINT 1050,(AMBDA(J),J=1,K)
      IF(KQ) 80,80,60
60  DO 61 I=1,KQ
      J=IHI(I)
      IF(ABS(AMBDA(J))-EPSI2) 61,61,70
61  CONTINUE
      GO TO 80
C
C      X-CORRECTION  COMPUTE ALL LAMBDA'S AND CHECK
C
70  DO 71 J=1,M
71  V(J)=X(J)
      CALL COMP1(3)
      DO 72 J=1,M
72  X(J)=Y(J)
      CALL AMDA
      IF(KQ) 80,80,66
66  DO 62 I=1,KQ
      J=IHI(I)
      IF(ABS(AMBDA(J))-EPSI2) 62,62,110
62  CONTINUE
80  DO 83 I=1,K
      IF(AMBDA(I)+EPSI2) 85,83,83
83  CONTINUE

```

```

      GO TO 130
C
C      FIND MOST NEGATIVE LAMBDA
C
      85 N=1
        SIGMA=AMBDA(1)
        IF(K-2) 89,86,86
      86 DO 88 I=2,K
        IF(AMBDA(I)-SIGMA) 87,88,88
      87 N=I
        SIGMA=AMBDA(I)
      88 CONTINUE
      89 SIGMA=0.0
        IH=N
      90 CALL COMPI(5)
        GO TO (100,100,95),NEXIT
      95 KQM1=KQ-1
        IF (KQM1) 97,97,94
      94 DO 96 I=1,KQM1
      96 AMBDA(I)=0.0
      97 INV=1
        GO TO 70
    100 IF(SIGMA) 110,101,110
    101 CALL COMP2(EPSI3)
        GO TO (105,103),MEXIT
C
C      DROP CONSTRAINT FROM BASIS
C
    103 SIGMA=1.0
        INV=1
        IH=N
        GO TO 90
    105 IF(KOUNT) 106,106,110
    106 PRINT 5001
5001 FORMAT(/'  NOT FEASIBLE X')
        GO TO 120
C
C      X-CORRECTION FAILED. RE-INVERT AND RE-CHECK LAMBDA'S TO FIND
C      FEASIBLE X
C
    110 PRINT 5002
5002 FORMAT(/'  X-CORRECTION FAILED')
    120 CALL REINV
        IF(NEXIT) 60,60,460
    130 PRINT 5003
5003 FORMAT(/'  FEASIBLE X')
        PRINT 1040,(X(J),J=1,M)
        PRINT 5004
5004 FORMAT(/'  LAMBDA'S')
        PRINT 1050,(AMBDA(J),J=1,K)

```

```

        CALL FUNCT(M,X,F,G,KQ)
        IF(KOUNT) 133,133,140
133 IF(KQ-KEQ) 9999,138,134
134 CONTINUE
138 CONTINUE
    NETA=0
C
C    CLASSIFY CONSTRAINTS
C
140 CALL CLASS
    INV=1
    GNORM=0
    DO 141 J=1,M
141 GNORM=GNORM+G(J)**2
    GNORM=SQRT(GNORM)
    PRINT 5006,GNORM,(G(J),J=1,M)
5006 FORMAT(/'    GRADIENT',F12.4/(10F12.4))
C
C    CHECK GRADIENT AGAINST TOLERANCE
C
    IF(GNORM-EPSI1) 142,142,150
142 HG NORM=GNORM
    PG NORM=GNORM
    KQ=0
    GO TO 420
150 SIGMA=0.0
160 IF(KQ) 161,161,170
161 HG NORM=0.0
    PG NORM=GNORM
    DO 163 J=1,M
    PG(J)=G(J)
    HG(J)=0.0
    DO 162 I=1,M
162 HG(J)=HG(J)+H(J,I)*G(I)
163 HG NORM=HG NORM+HG(J)**2
    HG NORM=SQRT(HG NORM)
    PRINT 7000,HG NORM,(HG(J),J=1,M)
7000 FORMAT(/'    CONJ GRAD',F12.4/(10F12.4))
    GO TO 180
C
C    PROJECT GRADIENT AND CHECK AGAINST TOLERANCE
C
170 DO 169 J=1,M
169 V(J)=G(J)
    CALL COMP1(2)
    PG NORM=0.0
    HG NORM=0.0
    DO 171 J=1,M
    HG(J)=0.0
    DO 1715 I=1,M

```

```

1715 HG(J)=HG(J)+H(J,I)*G(I)
      HGNORM=HGNORM+HG(J)**2
171 PGNORM=PGNORM+PG(J)**2
      PGNORM=SQRT(PGNORM)
      HGNORM=SQRT(HGNORM)
      PRINT 5007,PGNORM,(PG(J),J=1,M)
5007 FORMAT(/' PROJ GRAD',F12.4/(10F12.4))
      PRINT 7000,HGNORM,(HG(J),J=1,M)
      IF(PGNORM-EPSI1) 175,175,172
172 IF(M-KQ) 9999,173,180
173 PRINT 5008
5008 FORMAT(' PROJ NOT ZERO AT VERTEX')
C
C RE-INVERT AND RE-CLASSIFY CONSTRAINTS IF PROJECTION NOT ZERO
C AT VERTEX
C
174 CALL REINV
      IF(NEXIT) 1745,1745,460
1745 CALL CLASS
      GO TO 150
175 IF(SIGMA) 176,177,176
176 PRINT 5009
5009 FORMAT (' PROJ.ZERO AFTER DROP')
      GO TO 174
177 IF(KQ-KEQ) 9999,420,178
178 CALL COMP2(EPSI1)
      GO TO (420,179),MEXIT
C
C DROP CONSTRAINT FROM BASIS
C
179 NETA=0
      INV=1
      SIGMA=5.0
      GO TO 160
180 LDC=0
      IF(KV)9999,200,190
190 KEQP1=KEQ+1
      KEQP=KEQP1
      IF(KEQP1-KQ)1903,1903,1902
1903 DO 1901 J=KEQP,KQ
1901 RR(J)=R(J)
1902 L=0
      INDEX=0
      DELTA=0.0
191 L=L+1
      IF(IV(L)-NB) 1912,1912,1915
1912 KK=IV(L)
      J=JXB(KK)
      IF(J) 1913,9999,1914
1913 J=-J

```

```

      ZN=-Z(J)
      GO TO 1925
1914  ZN=Z(J)
      GO TO 1925
1915  KK=IV(L)-NB
      ZN=0.0
      DO 192 J=1,M
192   ZN=ZN+Z(J)*A(KK,J)
1925  IF(ZN-DELTA) 193,1935,1935
193   INDEX=IV(L)
      LL=L
      DELTA=ZN
1935  IF(KV-L) 194,194,191
194   IF(INDEX) 9999,200,195
195   KV=KV-1
      IF(KV-LL) 1995,198,198
198   DO 199 I=LL,KV
199   IV(I)=IV(I+1)
1995  IH=INDEX
      CALL COMP1(5)
      GO TO (197,197,196),NEXIT
196   NETA=0
      INV=1
      GO TO 150
197   IF(KV) 9999,1997,190
1997  IF(KEQP-KQ) 1998,1998,200
1998  DO 1999 J=KEQP,KQ
1999  R(J)=RR(J)
200   IF(NETA) 201,204,201
201   IF(KQ-KEQ)202,204,202
202   CALL COMP2(2.*HGNORM)
      GO TO (204,203),MEXIT
203   NETA=0
      INV=1
      GO TO 160
204   FY=F
      MU=0
      INT=0

C
C   PRINT, SUMMARY
C
C   Q - CONSTRAINTS IN BASIS
C   E - EQUALITY CONSTRAINTS
C   Q* - CONSTRAINTS ADDED TO THE INITIAL BASIS
C   U - LINEARLY DEPENDENT CONSTRAINTS
C   V - CONSTRAINTS NOT IN BASIS WITH LAMBDA = 0
C   W - CONSTRAINTS NOT IN BASIS WITH LAMBDA > 0
C
      KQS=KQ-KEQ
      PRINT 5010,KQ,NE,KQS,LDC,KV,KW

```

```

5010 FORMAT(4X2H Q, I2, 3H E , I2, 4H Q* , I2, 3H U , I2, 3H V , I2, 3H W , I2)
      IF(KQ)212,212,211
211 PRINT 9211, (IH(I), J=1, KQ)
212 IF(LDC)214,214,213
213 PRINT 9213, (IU(J), J=1, LDC)
214 IF(KV)216,216,215
215 PRINT 9215, (IV(J), J=1, KV)
216 IF(KW)218,218,217
217 PRINT 9217, (IW(J), J=1, KW)
9211 FORMAT('      Q' /10I3)
9213 FORMAT('      U' /10I3)
9215 FORMAT('      V' /10I3)
9217 FORMAT('      W' /10I3)
218 ZGY=0.0
      DO 205 I=1, M
      ZGY=ZGY+Z(I)*G(I)
      GO(I)=G(I)
205 Y(I)=X(I)
C
C      COMPUTE INITIAL STEP SIZE
C
208 T=AMIN1(TMAX, 2.0*(ESTF-F)/ZGY)
      IF(T.GE.TMIN) GO TO 230
      T=TMAX
230 NETA=NETA+1
C
C      STEP ITERATION COUNTER
C
      KOUNT=KOUNT+1
      L=0
      IF(KW)9999,240,231
231 DO 239 I=1, KW
      J=IW(I)
      IF(J-NB)232,232,235
232 KK=JXB(J)
      IF(KK) 233,9999,234
233 KK=-KK
      ZN=-Z(KK)
      GO TO 2365
234 ZN=Z(KK)
      GO TO 2365
235 KK=J-NB
      ZN=0.0
      DO 236 N=1, M
236 ZN=ZN+Z(N)*A(KK, N)
2365 IF(ZN) 237,239,239
237 TI=-AMBDA(J)/ZN
      IF(TI-T)238,239,239
238 T=TI
      L=J

```

```

239 CONTINUE
PRINT 5011,KOUNT
5011 FORMAT(/' STEP',I4)
C
C STEP ALONG THE CONJUGATE DIRECTION
C
240 DO 241 J=1,M
241 X(J)=Y(J)+T*Z(J)
XMUD=PGNORM
250 CALL FUNCT(M,X,F,G,KQ)
ZG=0.0
DO 251 J=1,M
251 ZG=ZG+Z(J)*G(J)
PRINT 5012,MU,INT,NETA,L,T,F,ZG
5012 FORMAT(4(3X I3),3F12.6)
ZGX=ZG
IF(ZGX)260,260,253
253 IF(FY-F)254,260,260
254 IF(T-TMAX) 257,255,255
255 DO 256 I=1,M
DO 256 J=1,M
H(I,J)=H(I,J)+Z(I)*Z(J)*T/ZGY
256 CONTINUE
257 GO TO 280
C
C INTERPOLATE CUBICALLY
C
260 ZZ=(3./T)*(F-FY)-ZGX-ZGY
W=SQRT(ZZ**2-ZGX*ZGY)
FRAC=(-ZGX+W-ZZ)/(-ZGX+ZGY+2.*W)
GAMMA=T*(1.-FRAC)
DO 262 J=1,M
YY(J)=X(J)
R(J)=GAMMA*Z(J)
262 X(J)=Y(J)+R(J)
FYY=F
CALL FUNCT(M,X,F,G,KQ)
IF(ABS(F-FYY).LT.EPSI1) GO TO 269
ZGZ=0.
DO 265 J=1,M
265 ZGZ=ZGZ+Z(J)*G(J)
IF(AMAX1(FY,FYY)-F)269,269,264
264 IF(MU-MUMAX)2695,2696,2696
2695 IF(FY-FYY)267,266,266
2696 PRINT 7001
7001 FORMAT(' MAXIMUM INTERPOLATION')
CALL REINV
IF(NEXIT)280,280,460
266 T=(1.-FRAC)*T
ZGX=ZGZ

```

```

        GO TO 260
267  T=FRAC*T
      FY=F
      F=FPY
      ZGY=ZGZ
      DO 268 J=1,M
      Y(J)=X(J)
268  X(J)=YY(J)
      GO TO 260
269  CONTINUE
270  DO 271 J=1,M
271  Y(J)=G(J)-GO(J)
      SIGY=0.
      YHY=0.
      DO 273 I=1,M
      RR(I)=0.
      DO 272 J=1,M
      RR(I)=RR(I)+Y(J)*H(J,I)
272  CONTINUE
      SIGY=SIGY+R(I)*Y(I)
273  YHY=YHY+RR(I)*Y(I)
      IF(SIGY*YHY.NE.0.) GO TO 274
C
C      SEARCH IN THE DIRECTION OF STEEPEST ASCENT IF RESULTS ARE
C      NOT SATISFACTORY
C
278  DO 276 I=1,M
      DO 276 J=1,M
      IF(I.EQ.J) GO TO 277
      H(I,J)=0.0
      GO TO 276
277  H(I,J)=1.0
276  CONTINUE
      GO TO 280
274  CONTINUE
C
C      UPDATE MATRIX OPERATOR AS PER STEP 5 OF THE ALGORITHM
C
      DO 275 I=1,M
      DO 275 J=1,M
      H(I,J)=H(I,J)-(R(I)*R(J))/SIGY-(RR(I)*RR(J))/YHY
275  CONTINUE
280  SIGMA=0.
C
C      COMPUTE LAMBDA'S FOR NEW X AND CHECK
C
290  CALL AMDA
      IF(KQ)284,284,283
283  DO 281 I=1,KQ
      J=IHI(I)

```



```

        PRINT 6002,AMBDA(J)
6002 FORMAT(/'  AMDA(J)',10F10.4)
        IF(ABS(AMBDA(J))-EPSI2)281,281,300
281 CONTINUE
284 DO 282 I=1,K
        IF(AMBDA(I)+EPSI2)292,282,282
282 CONTINUE
        GO TO 320
292 IF(SIGMA-1.0)293,293,301
293 L=1
C
C      FIND MOST NEGATIVE LAMBDA AND ADD CONSTRAINT TO THE BASIS
C
        RHO=AMBDA(1)
        IF(K-2)298,299,299
299 DO 295 I=2,K
        IF(AMBDA(I)-RHO)294,295,295
294 L=I
        RHO=AMBDA(I)
295 CONTINUE
298 IH=L
        CALL COMP1(5)
        GO TO(300,300,296),NEXIT
296 NETA=0
        INV=1
        GO TO 310
300 IF(SIGMA)301,310,301
C
C      CONSTRAINT VIOLATION
C
301 PRINT 5021
5021 FORMAT('  CONSTRAINT VIOLATION')
        GO TO 120
310 SIGMA=SIGMA+1.0
C
C      X-CORRECTION
C
        PRINT 5032
5032 FORMAT(/'  X-CORRECTION')
        DO 311 J=1,M
311 V(J)=X(J)
        CALL COMP1(3)
        DO 312 J=1,M
312 X(J)=Y(J)
        INV=1
        GO TO 290
320 IF(SIGMA)321,325,321
321 CALL FUNCT(M,X,F,G,KQ)
        IF(F-FY)322,325,325
C

```

```

C      F DECREASE
C
  322 PRINT 5013
5013 FORMAT('      F DECREASE')
      CALL REINV
      IF(NEXIT)325,325,460
  325 IF(KOUNT-LIMIT)327,326,326
  326 PRINT 5014
5014 FORMAT('      MAXIMUM STEPS TAKEN')
      GO TO 460
  327 PRINT 5015,(X(J),J=1,M)
5015 FORMAT(6F12.6)
      GO TO 140
C
C      MAXIMUM F
C
  420 PRINT 5018 ,F
5018 FORMAT(/'      MAXIMUM F =',F12.6)
  460 CALL CLASS
  470 PRINT 5019
5019 FORMAT(/'      J      X VECTOR      GRADIENT')
      DO 5678 J=1,M
5678 PRINT 5020,J,X(J),G(J)
5020 FORMAT(I3,2F12.6)
      GO TO 480
9999 PRINT 5031
5031 FORMAT ('      ERROR')
  480 IF(IN-NSP) 485,481,481
  485 PRINT 486
  486 FORMAT(1H1)
      GO TO 10
  481 STOP
      END

```

SUBROUTINE REINV

THIS SUBROUTINE REFORMS THE INVERSE MATRIX

```

      DIMENSION G(20),X(20),P(20),V(20),JXB(40),IHI(20),R(20),
      1SD(40),RR(20),Z(20),XD(20),GO(20),X1(20),DELX(20),X2(20),Y(20),
      1AMBDA(40),FG(20),IU(40),IV(40),IW(40),B(40),PG(20)
      DIMENSION D(20,20),DN(20,20),A(40,20),H(20,20),YY(20),HG(20)
      COMMON Y,PN,AMBDA,M,G,X,F,P,EPSI1,NEXIT,KQ,D,IH,NB,V,A,JXB,IHI,
      1 EPSI3,EPSI2,LD,IU,R,KEQ,MEXIT,IV,KV,B,KMNB,IW,KW,K,MXRN,INV,DN,
      1MU,NETA,INT,LDC,H
      EQUIVALENCE (Y,PG),(PN,PGNORM),(SD,AMBDA),(Z,HG)
      IF(MXRN .GT. 0)GO TO 10

```

RETURN IF MAXIMUM NUMBER OF RE-INVERSIONS HAVE BEEN DONE

```

      9 NEXIT=1
      RETURN
     10 NEXIT=0
      MXRN=MXRN-1
      IF(MXRN .LE. 0)GO TO 20
      GO TO 40
     20 IF(KEQ .EQ. 0)GO TO 35

```

RESTORE INITIAL BASIS AND RESET COUNTERS

```

     15 DO 30 I=1,KEQ
      DO 30 J=1,KEQ
     30 D(I,J)=DN(I,J)
     35 LDC=0
      NETA=0
      KQ=KEQ
      INV=0
      MU=0
      INT=0
    935 FORMAT('      NEW BASIS')
      PRINT 935
      RETURN
     40 IF(INV)9,9,50

```

RETURN IF THERE IS NO CHANGE IN BASIS

```

     50 IF(KQ-KEQ)15,15,51

```

LIST ALL CONSTRAINTS ADDED TO THE INITIAL BASIS

```

     51 L=KQ-KEQ
      IF(KEQ .GT. 0)GO TO 54
      DO 53 I=1,M
      DO 52 J=1,M

```

```

52 H(I,J)=0.
53 H(I,I)=1.
   GO TO 59
54 DO 46 I=1,M
   DO 45 J=1,KEQ
   D(I,J)=0.
   DO 55 N=1,KEQ
   KK=IHI(N)-NB
55 D(I,J)=D(I,J)+A(KK,I)+DN(N,J)
45 CONTINUE
46 CONTINUE
   DO 57 I=1,M
   DO 56 J=1,M
   H(I,J)=0.
   DO 56 N=1,KEQ
   KK=IHI(N)-NB
56 H(I,J)=H(I,J)-A(KK,I)*D(J,N)
57 H(I,I)=H(I,I)+1.
   DO 58 I=1,KEQ
   DO 58 J=1,KEQ
58 D(I,J)=DN(I,J)
59 LDC=0
   NETA=0
   KQ=KEQ
   INV=0
   MU=0
   INT=0
60 IH=IHI(KQ+1)
C
C   ADD CONSTRAINTS IN OEDER OF CONSTRAINT NUMBERS
C
   CALL COMP1(5)
65 IF(L .LE. 1)GO TO 70
   L=L-1
   GO TO 60
70 NEXIT=0
   RETURN
   END

```

SUBROUTINE COMPI(NSTART)

THIS SUBROUTINE CARRIES OUT THE MATRIX COMPUTATIONS REQUIRED BY
THE CONJUGATE GRADIENT METHOD

```

C
C
C
C
      DIMENSION G(20),X(20),P(20),V(20),JXB(40),IHI(20),R(20),
      1SD(40),RR(20),Z(20),XO(20),GO(20),X1(20),DELX(20),X2(20),Y(20),
      1LAMBDA(40),FG(20),IU(40),IV(40),IW(40),B(40),PG(20),TT(20)
      DIMENSION D(20,20),DN(20,20),A(40,20),H(20,20),YY(20),HG(20)
      COMMON Y,PN,AMBDA,M,G,X,F,P,EPSI1,NEXIT,KQ,D,IH,NB,V,A,JXB,IHI,
      1 EPSI3,EPSI2,LD,IU,R,KEQ,MEXIT,IV,KV,B,KMNB,IW,KW,K,MXRN,INV,DN,
      1MU,NETA,INT,LDC,H
      EQUIVALENCE (Y,PG),(PN,PGNORM),(SD,AMBDA),(Z,HG)
      GO TO (10, 80, 110, 25, 10 ), NSTART
10  IFLAG = 1
      IF (KQ) 20,20,30
20  D(1,1) = 1.0
      PN = 1.0
      GO TO 260
25  IFLAG = 2

C
C
C
      CHECK WHETHER THE CONSTRAINT IS A BOUND
30  IF(IH-NB) 40,40,35
35  I = IH-NB
      DO 36 J = 1,M
36  V(J) = A(I,J)
      GO TO 90

C
C
C
      GENERATE VECTOR V IF THE CONSTRAINT IS A BOUND
40  DO 50 I = 1,M
50  V(I) = 0.0
      JK = JXB(IH)
      IF (JK) 60,60,70
60  JKM = -JK
      V(JKM) = -1.0
      GO TO 90
70  V(JK) = 1.0
      GO TO 90
80  IFLAG = 2
90  IF (KQ) 85,85,88
85  DO 86 J = 1,M
      Y(J) = V(J)
86  CONTINUE
      RETURN
88  DO 100 I = 1,KQ
      KK = IHI(I)-NB
      IF (KK) 91,91,95
91  JBD = IHI(I)

```

```

      JK = JXB(JBD)
      IF (JK) 92,92,93
92   JKM = -JK
      P(I) = -V(JKM)
      GO TO 100
93   P(I) = V(JK)
      GO TO 100
95   P(I) = 0.0
      DO 99 J = 1,M
99   P(I) = P(I)+A(KK,J)*V(J)
100  CONTINUE
      GO TO 130
110  IFLAG = 2
      DO 120 I = 1,KQ
      KK = IHI(I)
120  P(I) = AMBDA(KK)
130  DO 140 I = 1,KQ
      R(I) = 0.0
      DO 140 J = 1,KQ
      R(I) = R(I)+D(I,J)*P(J)
140  CONTINUE

```

C
C
C

PROJECT THE VECTOR

```

      DO 160 J = 1,M
      Y(J) = 0.0
      DO 150 I = 1,KQ
      KK = IHI(I)-NB
      IF (KK) 151,151,155
151  JBD = IHI(I)
      JK = JXB(JBD)
      IF (JK) 152,152,153
152  IF (J+JK) 150,156,150
156  Y(J) = Y(J)-R(I)
      GO TO 150
153  IF (J-JK) 150,154,150
154  Y(J) = Y(J)+R(I)
      GO TO 150
155  Y(J) = Y(J)+A(KK,J)*R(I)
150  CONTINUE
160  Y(J) = V(J)-Y(J)
      GO TO (180,170), IFLAG
170  NEXIT = 1
      RETURN

```

C
C
C

COMPUTE PROJECTED NORM

```

180  PN = 0.0
      DO 190 J = 1,M
190  PN = PN + Y(J)**2

```

```

        YB2 = PN
        PN = SORT(PN)
C
C      CHECK FOR LINEAR DEPENDENCE
C
        IF (PN-EP3) 200,210,210
200  NEXIT = 2
        LDC = LDC+1
        IU(LDC) = IH
        PRINT 9200, IH, PN
9200  FORMAT('      H ',I2,' LINEARLY DEPENDENT      YB=',F12.6)
        RETURN
C
C      ADD LINEARLY INDEPENDENT CONSTRAINTS
C
210  J = KQ + 1
        I = J
        D(I,J) = 1.0/YB2
220  IF (I-J) 240,240,230
C
C      COMPUTE NEW INVERSE
C
230  I = I-1
        D(I,J) = D(I,J)+R(J)*R(I)/YB2
        D(J,I) = D(I,J)
        GO TO 220
240  IF (J-1) 260,260,250
250  J = J - 1
        I = KQ + 1
        D(I,J) = -R(J)/YB2
        D(J,I) = D(I,J)
        GO TO 220
260  KQ = KQ + 1
        IHI(KQ) = IH
        PRINT 9260, IH, PN
9260  FORMAT('      H ',2X I2,'      ADDED      YB',F12.6)
        NEXIT = 3
        IF (NSTART-3) 270,270,290
270  RETURN
290  IF (IH-NB) 300,300,330
300  JK = JXB(IH)
        IF (JK) 310,310,320
310  JK = -JK
C
C      COMPUTE H(Q+1) AS PER STEP 4 OF THE ALGORITHM
C
        DO 311 J = 1,M
311  TT(J) = -H(JK,J)
        YB2 = H(JK,JK)
        GO TO 335

```

```
320 DO 321 J = 1,M
321 TT(J) = H(JK,J)
    YB2 = H(JK,JK)
    GO TO 335
330 KK = IH-NB
    YB2 = 0.0
    DO 333 I = 1,M
    TT(I) = 0.0
    DO 332 J = 1,M
332 TT(I) = TT(I)+A(KK,J)*H(J,I)
333 YB2 = YB2+TT(I)*A(KK,I)
335 DO 340 I = 1,M
    DO 340 J = 1,M
340 H(I,J) = H(I,J)-TT(I)*TT(J)/YB2
    RETURN
    END
```


SUBROUTINE COMP2(DEL)

THIS SUBROUTINE CARRIES OUT THE MATRIX COMPUTATIONS REQUIRED BY
THE CONJUGATE GRADIENT METHOD

DIMENSION G(20),X(20),P(20),V(20),JXB(40),IHI(20),R(20),
1SD(40),RR(20),Z(20),XO(20),GO(20),X1(20),DETX(20),X2(20),Y(20),
1AMBDA(40),FG(20),IU(40),IV(40),IW(40),B(40),PG(20)
DIMENSION D(20,20),DN(20,20),A(40,20),H(20,20),YY(20),HG(20)
COMMON Y,PN,AMBDA,M,G,X,F,P,EPSI1,NEXIT,KQ,D,IH,NB,V,A,JXB,IHI,
1 EPSI3,EPSI2,LD,IU,R,KEQ,MEXIT,IV,KV,B,KMNB,IW,KW,K,MXRN,INV,DN,
1MU,NETA,INT,LDC,H
EQUIVALENCE (Y,PG),(PN,PGNORM),(SD,AMBDA),(Z,HG)
DELTA = DEL
NZ = 0
L = KEQ + 1

CHECK FOR CONSTRAINT DROP AND DROP CONSTRAINT CORRESPONDING TO
MAXIMUM RD > DELTA

40 IF (R(L)) 43,43,41
41 RD = R(L)/SQRT(D(L,L))
IF (RC-DELTA) 43,42,42
42 DELTA = RD
NZ = L
43 IF (L-KQ) 44,45,45
44 L = L+1
GO TO 40
45 IF (NZ) 46,46,50
46 MEXIT = 1
RETURN

COMPUTE NEW INVERSE

50 DO 60 I = 1,KQ
60 V(I) = D(I,NZ)
IH = IHI(NZ)
NZM = NZ - 1
KQ = KQ - 1
IF (NZ-KQ) 63,63,62
62 VNZ = V(NZ)
GO TO 105
63 DO 80 I = NZ, KQ
IHI(I) = IHI(I+1)
IF (NZM) 75,75,65
65 DO 70 J = 1,NZM
D(I,J) = D(I+1,J)
70 D(J,I) = D(I,J)
75 DO 80 J = NZ, KQ
80 D(I,J) = D(I+1,J+1)

```

      VNZ = V(NZ)
      DO 100 I = NZ,KQ
100   V(I) = V(I+1)
105   DO 130 I = 1,KQ
      DO 130 J = 1,KQ
      IF (I-J) 110,110,120
110   D(I,J) = D(I,J)-V(I)*V(J)/VNZ
      GO TO 130
120   D(I,J) = D(J,I)
130   CONTINUE
C
C   RESET LINEARLY DEPENDENT CONSTRAINTS = 0
C
      IF (LDC) 150,150,140
140   DO 142 I = 1,LDC
      J = KV + I
142   IV(J) = IU(I)
      KV = KV + LDC
      LDC = 0
150   MEXIT = 2
      PRINT 9150, IH, DELTA
9150  FORMAT('      H  'I2,' DROPPED   DELTA=',F12.6)
C
C   COMPUTE H(Q-1) AS PER STEP 2 OF THE ALGORITHM
C
      CALL COMP1(4)
      PN = 0.0
      DO 160 J = 1,M
160   PN = PN + Y(J)**2
      DO 170 I = 1,M
      DO 170 J = 1,M
170   H(I,J) = H(I,J) + Y(I)*Y(J)/PN
      RETURN
      END

```

SUBROUTINE AMDA

C
C
C

THIS SUBROUTINE COMPUTES LAMBDA

```

      DIMENSION G(20),X(20),P(20),V(20),JXB(40),IHI(20),R(20),
1SD(40),RR(20),Z(20),XO(20),GO(20),X1(20),DELX(20),X2(20),Y(20),
1AMBDA(40),FG(20),IU(40),IV(40),IW(40),B(40),PG(20)
      DIMENSION D(20,20),DN(20,20),A(40,20),H(20,20),YY(20),HG(20)
      COMMON Y,PN,AMBDA,M,G,X,F,P,EPSI1,NEXIT,KQ,D,IH,NB,V,A,JXB,IHI,
1 EPSI3,EPSI2,LD,IU,R,KEQ,MEXIT,IV,KV,B,KMNB,IW,KW,K,MXRN,INV,DN,
1MU,NETA,INT,LDC,H
      EQUIVALENCE (Y,PG),(PN,PGNORM),(SD,AMBDA),(Z,HG)
      IF (NB) 2,2,4
4 DO 7 I = 1,NB
      J = JXB(I)
      IF (J) 5,5,6
5 KK = -J
      AMBDA(I) = -X(KK) - B(I)
      GO TO 7
6 AMBDA(I) = X(J) - B(I)
7 CONTINUE
2 IF (KMNB)10,10,3
3 DO 9 I = 1,KMNB
      SUM = 0.0
      KK = NB + I
      DO 8 J = 1,M
8 SUM = SUM + A(I,J)*X(J)
9 AMBDA(KK) = SUM - B(KK)
10 RETURN
      END

```

SUBROUTINE CLASS

THIS SUBROUTINE CLASSIFIES THE CONSTRAINTS

```

C
C
C
  DIMENSION G(20),X(20),P(20),V(20),JXB(40),IHI(20),R(20),
1SD(40),RR(20),Z(20),XD(20),GO(20),X1(20),DELX(20),X2(20),Y(20),
1AMBDA(40),FG(20),IU(40),IV(40),IW(40),B(40),PG(20)
  DIMENSION D(20,20),DN(20,20),A(40,20),H(20,20),YY(20),HG(20)
  COMMON Y,PN,AMBDA,M,G,X,F,P,EPSI1,NEXIT,KQ,D,IH,NB,V,A,JXB,IHI,
1 EPSI3,EPSI2,LD,IU,R,KEQ,MEXIT,IV,KV,B,KMNB,IW,KW,K,MXRN,INV,DN,
1MU,NETA,INT,LDC,H
  EQUIVALENCE (Y,PG),(PN,PGNORM),(SD,AMBDA),(Z,HG)
  KW = 0
  KV = 0
  DO 50 I = 1,K
  IF (KQ) 41,41,30
30 DO 40 J = 1,KQ
  IF (I-IHI(J)) 40,50,40
40 CONTINUE
41 IF (ABS(AMBDA(I))-EPSI2) 42,42,43
42 KV = KV + 1
  IV(KV) = I
  GO TO 50
43 KW = KW + 1
  IW(KW) = I
50 CONTINUE
  RETURN
  END

```

```

SUBROUTINE FUNCT(M,X,F,G,KQ)
C
C THIS SUBROUTINE COMPUTES THE VALUES OF THE OBJECTIVE FUNCTION
C AND THE GRADIENTS FOR THE INVENTORY MODEL
C
  DIMENSION X(10),G(10),VI(15),Q(15),FT(10),ST(10),F1(15),BR(10),
  ICR(15),GR(15)
  COMMON/ARO/NFUNC,KOUNT,NOFE
  95 FORMAT(8F8.4)
  200 FORMAT(/'      ITERATION # = ',I3,'      # OF FUNCTIONAL EVALUATIONS
  1=' ,I3,'      # OF CONSTRAINTS IN BASIS = ',I3,/)
  440 FORMAT(21X,' STAGE 1      STAGE 2      STAGE 3      STAGE 4      STAGE
  1 5',/)
  451 FORMAT('      PRODUCTION',5F12.3,/'      SALE',5F12.3,
  1/'      INVENTORY',5F12.3,/'      COST',5F12.3,/)
  NFUNC = NFUNC + 1
  NOFE=NOFE+1
  IF(NOFE.GT.1) GO TO 90
  READ 95,A,B,C,CI,CP,VIM,PM
  N=M
  EN=N
  DELT=1./EN
  N1=N+1
  AS=-2.*CI*(DELT**2)
  VI(1)=C
  F1(1)=0.0
  Q(1)=A
  90 CONTINUE
  DO 100 J=2,N1
  I=J-1
  ED=I
  Q(J)=A+B*DELT*ED
  VI(J)=VI(I)+(X(I)-Q(J))*DELT
  FT(I)=CI*DELT*(VIM-VI(I)/2.-VI(J)/2.）**2
  ST(I)=CP*DELT*(EXP((PM-X(I))**2))
  BR(I)=VIM-VI(I)/2.-VI(J)/2.
  CR(I)= -2.*DELT*CP*(PM-X(I))*EXP((PM-X(I))**2)
  100 F1(J)=F1(I)+FT(I)+ST(I)
  F=-F1(N1)
  DO 102 I=1,N
  GR(I)=0.0
  BR(I)=BR(I)/2.
  DO 101 J=I,N
  101 GR(I)=GR(I)+BR(J)
  102 G(I)=-GR(I)*AS-CR(I)
  PRINT 200,KOUNT,NFUNC,KQ
  PRINT 440
  PRINT 451,(X(I),I=1,N),(Q(I),I=2,N1),(VI(I),I=2,N1),(F1(I),I=2,N1)
  RETURN
  END

```

```

SUBROUTINE FUNCT(M,X,F,G,KQ)
C
C THIS SUBROUTINE COMPUTES THE VALUES OF THE OBJECTIVE FUNCTION
C AND THE GRADIENTS FOR THE INVENTORY AND ADVERTISEMENT MODEL
C
  DIMENSION X(10),G(10),P(15),Q(15),VI(15),U(15),V(15),AW(15),W(15),
  ID(15),E(15),F1(15),GR(15),QX(15),XI(15),R(15)
  COMMON/ARO/NFUNC,KOUNT,NOFE
200 FORMAT(/'      ITERATION # = ',I3,'      # OF FUNCTIONAL EVALUATIONS
1=' ,I3,'      # OF CONSTRAINTS IN BASIS = ',I3,/)
440 FORMAT(23X,' STAGE 1      STAGE 2      STAGE 3      STAGE 4      STAGE
1 5',/)
451 FORMAT('      INC. CONTACTS',5F12.3,/'      PRODUCTION',5F12.3,/
1'      SALE',5F12.3,/'      INVENTORY',5F12.3,/
2'      PROFIT',5F12.3,/)
920 FORMAT(10F7.2)
  NOFE=NOFE+1
  NFUNC=NFUNC+1
  IF(NOFE.GT.1) GO TO 90
  READ 920,A,B,C,EF,AN,PI,CA,CI,VI(1),Q(1)
  N=M
  N1=N+1
  EN=N
  DELT=1./EN
  P(1)=A
  F1(1)=0.0
90 DO 100 J=2,N1
  I=J-1
  ED=I
  P(J)=A+B*DELT*ED
  U(I)=Q(I)*(1.+(C+X(I))*DELT)
  V(I)=1.+(C+X(I))*Q(I)*DELT/AN
  Q(J)=U(I)/V(I)
  VI(J)=VI(I)+(P(I)-Q(I))*DELT
100 F1(J)=F1(I)+(EF*Q(J)-CI*((PI-VI(J))**2)-CA*X(I)*Q(J))*DELT
  F= F1(N1)
  DO 101 I=1,N
  AW(I)=(Q(I)*V(I)*DELT-Q(I)*U(I)*DELT/AN)/(V(I)**2)
  D(I)=EF*DELT-CA*X(I)*DELT
  E(I)=2.*CI*(PI-VI(I+1))*DELT
  R(I)=-CA*Q(I+1)*DELT
101 CONTINUE
  DO 102 J=2,N
102 W(J)=((1.+(C+X(J))*DELT)*V(J)-U(J)*(C+X(J))*DELT/AN)/(V(J)**2)
  E(N1)=0.0
  W(N1)=0.0
  DO 150 I=1,N
  L=I+1
  GR(I)=0.0
  QX(I)=AW(I)

```

```
XI(I)=-DELT*QX(I)
DO 140 J=L,N1
QX(J)=W(J)*QX(J-1)
XI(J)=XI(J-1)-QX(J)*DELT
140 GR(I)=GR(I)+D(J-1)*QX(J-1)+E(J)*XI(J-1)
150 G(I)=GR(I)+R(I)
PRINT 200,KOUNT,NFUNC,KQ
PRINT 440
PRINT 451,(X(I),I=1,N),(P(I),I=2,N1),(Q(I),I=2,N1),(VI(I),I=2,N1),
1(F1(I),I=2,N1)
RETURN
END
```

```

SUBROUTINE FUNCT(M,X,F,G,KQ)
C
C THIS SUBROUTINE COMPUTES THE OBJECTIVE FUNCTION AND THE
C GRADIENTS FOR THE PRODUCTION AND ADVERTISEMENT MODEL
C
  DIMENSION X(15),G(15),U(15),W(15),Y(15),Z(15),Q(15),VI(15),GR(15),
  1AK1(15),BK1(15),AK2(15),BK2(15),DAK1(15),DAK2(15),DBK1(15),AR(15),
  2DBK2(15),D(15),E(15),F1(10),QQ(15),QX(15),XI(15),QI(15),CA(15)
  DIMENSION UU(10),WW(10),WU(10),YY(10),YU(10),ZZ(10),VIZ(10),
  1WX(10),UX(10),YX(10),ZX(10),XK(10),XL(10),ZY(10),ZW(10)
  COMMON/ARO/NFUNC,KOUNT,NOFE
100 FORMAT(2F7.0,9F5.0,F7.4,F6.3)
450 FORMAT(15F8.3)
451 FORMAT(/'          SALE',5F12.3,/'          INVENTORY',5F12.3,/'
1'          PROFIT',5F12.3,/'
930 FORMAT(/'          ITERATION #=',I3,/' # OF FUNCTIONAL EVALUATIONS=',
1I3,/' # OF CONSTRAINTS IN BASIS=',I3,/)
1000 FORMAT(/5F6.3,3X,5F6.3,3X,5F6.3,3X,5F6.3)
1001 FORMAT(' EA=',F8.0,/' EB=',F8.0,/' R=',F4.0,/' AQ=',F4.0,/' V=',
1F4.0,/' VIM=',F4.0,/' TM=',F5.0,/' AN=',F5.0,/' C=',F3.0,/' C1=',
2F3.0,/' CI=',F3.0,/' CT=',F6.4,/' CA=',F5.3,/)
  NFUNC=NFUNC+1
  NOFE=NOFE+1
  IF(NOFE.GT.1) GO TO 90
  READ 100,EA,EB,R,AQ,V,VIM,TM,AN,C,C1,CI,CT,CA
  PRINT 1001,EA,EB,R,AQ,V,VIM,TM,AN,C,C1,CI,CT,CA
  N=M/3
  EN=N
  DT=1./EN
  N1=N+1
  U(1)=0.53
  Y(1)=0.53
  W(1)=0.43
  Z(1)=0.43
  Q(1)=1.0
  VI(1)=8.0
  XO=0.53
  YO=0.43
  GA=C.535E11
  GB=C.461E18
90 CONTINUE
  F1(1)=C.0
  DO 101 I=1,N
  K=I+N
  L=K+N
  AK1(I)=GA*EXP(-EA/(R*X(K)))
  BK1(I)=GB*EXP(-EB/(R*X(K)))
  AK2(I)=GA*EXP(-EA/(R*X(L)))
  BK2(I)=GB*EXP(-EB/(R*X(L)))
  DAK1(I)=AK1(I)*(EA/(R*(X(K)**2)))

```



```

DAK2(I)=AK2(I)*(EA/(R*(X(L)**2)))
DBK1(I)=BK1(I)*(EB/(R*(X(K)**2)))
DBK2(I)=BK2(I)*(EB/(R*(X(L)**2)))
101 CONTINUE
DO 120 J=2,N1
  I=J-1
  K=I+N
  L=K+N
  Q(J)=Q(I)+(C+X(I))*(Q(I)-(Q(I)**2)/AN)*DT
  VI(J)=VI(I)+(AQ*Z(I)-Q(I))*DT
  U(J)=U(I)+(AC*(XC-U(I))/V-AK1(I)*U(I))*DT
  W(J)=W(I)+(AC*(YD-W(I))/V-BK1(I)*W(I)+AK1(I)*U(I))*DT
  Y(J)=Y(I)+(AC*(U(I)-Y(I))/V-AK2(I)*Y(I))*DT
  Z(J)=Z(I)+(AC*(W(I)-Z(I))/V-BK2(I)*Z(I)+AK2(I)*Y(I))*DT
  F1(J)=F1(I)+(C1*Q(J)-C1*(VIM-VI(J))**2-CA*(X(I)**2)*(Q(J)**2)-
1CT*((TM-X(K))**2+(X(K)-X(L))**2))*DT
120 CONTINUE
  F=F1(N1)
  DO 130 I=1,N
    J=I+1
    D(I)=C1*DT-2.*CA*(X(I)**2)*Q(J)*DT
    E(I)=2.*C1*(VIM-VI(J))*DT
    AR(I)=-2.*CA*X(I)*(Q(J)**2)*DT
    QA(J)=(Q(I)-(Q(I)**2)/AN)*DT
130 CONTINUE
  DO 140 J=2,N
    QQ(J)=1.+(C+X(J))*(1.-2.*Q(J)/AN)*DT
140 CONTINUE
  E(N1)=0.0
  QQ(N1)=0.0
  DO 160 I=1,N
    L=I+1
    GR(I)=0.0
    QX(I)=QA(L)
    XI(I)=-DT*QX(I)
    DO 150 J=L,N1
      QX(J)=QQ(J)*QX(J-1)
      XI(J)=XI(J-1)-DT*QX(J)
150 GR(I)=GR(I)+D(J-1)*QX(J-1)+E(J)*XI(J-1)
160 G(I)=GR(I)+AR(I)
  DO 250 I=1,N
    J=I+1
    K=I+N
    L=K+N
    XK(I)=2.*CT*DT*(TM-X(K))-2.*CT*DT*(X(K)-X(L))
    XL(I)=2.*CT*DT*(X(K)-X(L))
    UX(I)=-U(I)*CT*CAK1(I)
    WX(I)=-DT*(W(I)*DBK1(I)-U(I)*DAK1(I))
    YX(I)=-DT*(Y(I)*DBK2(I)-Y(I)*DAK2(I))
    ZX(I)=-DT*(Z(I)*DBK2(I)-Y(I)*DAK2(I))

```

250 CONTINUE

DO 260 J=2,N
 UU(J)=1.-DT*(AQ/V+AK1(J))
 WW(J)=1.-DT*(AQ/V+BK1(J))
 WU(J)=DT*AK1(J)
 YY(J)=1.-DT*(AQ/V+AK2(J))
 YU(J)=AQ*DT/V
 ZW(J)=AQ*DT/V
 ZZ(J)=1.-DT*(AQ/V+BK2(J))
 ZY(J)=DT*AK2(J)
 VIZ(J)=AQ*DT

260 CONTINUE

DZ3X6=ZW(2)*WX(1)
 DI4X6=VIZ(3)*DZ3X6
 DW3X6=WW(2)*WX(1)+WU(2)*UX(1)
 DY3X6=YU(2)*UX(1)
 DU3X6=UU(2)*UX(1)
 DI5X6=DI4X6+VIZ(4)*(ZZ(3)*DZ3X6+ZW(3)*DW3X6+ZY(3)*DY3X6)
 DZ4X6=ZZ(3)*DZ3X6+ZW(3)*DW3X6+ZY(3)*DY3X6
 DW4X6=WW(3)*DW3X6+WU(3)*DU3X6
 DY4X6=YY(3)*DY3X6+YU(3)*DU3X6
 DI6X6=DI5X6+VIZ(5)*(ZZ(4)*DZ4X6+ZW(4)*DW4X6+ZY(4)*DY4X6)
 DW4X7=WW(3)*WX(2)+WU(3)*UX(2)
 DZ4X7=ZW(3)*WX(2)
 DY4X7=YU(3)*UX(2)
 DI5X7=VIZ(4)*DZ4X7
 DI6X7=DI5X7+VIZ(5)*(ZZ(4)*DZ4X7+ZW(4)*DW4X7+ZY(4)*DY4X7)
 DI6X8=VIZ(5)*ZW(4)*WX(3)
 DI3X1=VIZ(2)*ZX(1)
 DZ3X1=ZZ(2)*ZX(1)+ZY(2)*YX(1)
 DY3X1=YY(2)*YX(1)
 DY4X1=YY(3)*DY3X1
 DY5X1=YY(4)*DY4X1
 DI4X1=DI3X1+VIZ(3)*DZ3X1
 DZ4X1=ZZ(3)*DZ3X1+ZY(3)*DY3X1
 DI5X1=DI4X1+VIZ(4)*DZ4X1
 DZ5X1=ZZ(4)*DZ4X1+ZY(4)*DY4X1
 DI6X1=DI5X1+VIZ(5)*DZ5X1
 DI4X2=VIZ(3)*ZX(2)
 DZ4X2=ZZ(3)*ZX(2)+ZY(3)*YX(2)
 DZ5X2=ZZ(4)*DZ4X2+ZY(4)*YY(3)*YX(2)
 DI5X2=DI4X2+VIZ(4)*DZ4X2
 DI6X2=DI5X2+VIZ(5)*DZ5X2
 DI5X3=VIZ(4)*ZX(3)
 DI6X3=DI5X3+VIZ(5)*(ZZ(4)*ZX(3)+ZY(4)*YX(3))
 G(6)=E(3)*DI4X6+E(4)*DI5X6+E(5)*DI6X6+XK(1)
 G(7)=E(4)*DI5X7+E(5)*DI6X7+XK(2)
 G(8)=E(5)*DI6X8+XK(3)
 G(9)=XK(4)
 G(10)=XK(5)

```
G(11)=E(2)*DI3X1+E(3)*DI4X1+E(4)*DI5X1+E(5)*DI6X1+XL(1)
G(12)=E(3)*DI4X2+E(4)*DI5X2+E(5)*DI6X2+XL(2)
G(13)=E(4)*DI5X3+E(5)*DI6X3+XL(3)
G(14)=E(5)*VI7(5)*ZX(4)+XL(4)
G(15)=XL(5)
PRINT 930,KOUNT,NFUNC,KC
PRINT 450,(X(I),I=1,M)
PRINT 1000,(U(J),J=2,N1),(W(J),J=2,N1),(Y(J),J=2,N1),(Z(J),J=2,N1)
PRINT 451,(O(I),I=2,N1),(VI(I),I=2,N1),(F1(I),I=2,N1)
RETURN
END
```

OPTIMIZATION OF INDUSTRIAL MANAGEMENT SYSTEMS

BY

THE CONJUGATE GRADIENT METHOD

by

ASHOK KUMAR ARORA

B. Tech. (Hons), Indian Institute of Technology

Kharagpur, India, 1965

AN ABSTRACT OF A MASTER'S THESIS

submitted in partial fulfillment of the
requirement for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1970

ABSTRACT

Motivated by the successful applications of the gradient methods to unconstrained nonlinear problems, a number of methods have been proposed to solve constrained nonlinear problems. In 1966, Goldfarb developed a quadratically convergent method, called the conjugate gradient method, to solve a class of nonlinear problems characterized by linear constraints. This method is the extension of Davidon's variable metric method and has the inherited ability to use the information about the local curvature of the nonlinear function without requiring calculation of second partial derivatives.

The purpose of this thesis is to apply the conjugate gradient method to some industrial management systems and to analyze the results.

The method is initially described and then three problems concerning production, inventory control and advertisement are solved with a view to making a critical study of the technique. The first problem is a simple inventory model involving the minimization of production and inventory costs. The second model is the well known diffusion model requiring to maximize profit by controlling advertisement. The last model is relatively more complicated and involves six state variables and three control variables. The results obtained have been found satisfactory.