

OPTIMIZATION OF MANAGEMENT SYSTEMS
USING SENSITIVITY ANALYSIS

by *1264*

RAVINDRA KUMAR

B. Tech. (Hons) - Mech., Indian Institute of Technology,
Bombay, India, 1966

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:

E. Stang Lee
Major Professor

LD
2668
T4
1970
K85

ACKNOWLEDGEMENT

The author wishes to express his deep sense of appreciation to his major professor, Dr. E. S. Lee for his guidance, constructive criticism and keen interest taken in preparing this master's thesis; to Mr. Syed Waziruddin for helpful suggestions and to Mr. P. K. Mehrotra for help in proof reading.

The author also wishes to express his thanks to Mrs. Roopa Shah for her drawings and Mrs. Marie Jirak for her excellent typing.

**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.**

TABLE OF CONTENTS

ACKNOWLEDGEMENT	ii
LIST OF TABLES	v
LIST OF FIGURES	vi
INTRODUCTION AND PURPOSE OF STUDY	1
NUMERICAL METHODS OF SOLUTION OF NON-LINEAR TWO POINT BOUNDARY VALUE PROBLEMS	3
Non-Linear Dynamic Systems	3
General Classification of Numerical Methods of Solution of Variational Problems	5
Direct Methods	5
Indirect Methods	6
BOUNDARY CONDITION ITERATION METHOD BASED ON DIFFERENTIAL SENSITIVITY ANALYSIS	10
The Problem	10
Discussion	20
APPLICATION TO AN INVENTORY MODEL	21
Development of the Model	21
Definition of the Problem	22
Formulation of the Problem	23
Boundary Condition Iteration	25
Numerical Aspects	28
Results	30
APPLICATION TO AN INVENTORY AND ADVERTISEMENT SCHEDULING MODEL	44
Development of the Model	44
Definition of the Problem	46
Formulation of the Problem	47
Boundary Condition Iteration Scheme	50
Numerical Aspects	53
Results	54
APPLICATION TO A PRODUCTION AND ADVERTISEMENT SCHEDULING MODEL	66
Definition of the Problem	70
Formulation of the Problem	71
Boundary Condition Iteration Scheme	73
Numerical Aspects	80
Results	82

CONCLUSION	94
REFERENCES	96
APPENDICES	97
1. Direct Second Variational Method	99
2. Computer Program for the Inventory Model	102
3. Computer Program for the Inventory and Advertisement Scheduling Model	110
4. Computer Program for the Production and Advertisement Scheduling Model	117

LIST OF TABLES

TABLE		Page
4.1	Initial Approximations Used for the Inventory Model	31
4.2	Convergence Rate of Final Inventory, $X_1(t_f)$	32
4.3	Convergence Rate of Total Cost, $X_2(t_f)$, in an Inventory Model	33
4.4	Convergence Rate of Residuals ψ_1 , ψ_2 and $ R $	34
4.5	Effect of Step Size γ_k and Weighting Factor q_k on Convergence Rate	36
5.1	Initial Approximations Used for Advertisement Model	55
5.2	Convergence Rate of Sales, $X_1(t_f)$	56
5.3	Convergence Rate of Final Inventory, $X_2(t_f)$	57
5.4	Convergence Rate of Total Profit, $X_3(t_f)$	58
5.5	Convergence Rate of Residuals ψ_1 , ψ_2 , ψ_3 and $ R $	63
5.6	Effect of Step Size γ_k and Weighting Factor q_k on Convergence Rate	64
6.1	Initial Approximations Used for the Production and Advertisement Scheduling Model	83
6.2	Effect of Weighting Factor q_k and Step Size γ_k on Convergence Rate for Set 1	84
6.3	Effect of Weighting Factor q_k and Step Size γ_k on Convergence Rate for Set 3	93

CHAPTER 1

INTRODUCTION AND PURPOSE OF STUDY

Science, since the earliest times, has been concerned with the mathematical expression of the laws governing the behavior of physical, biological, social and economic processes. Once the underlying laws of behavior have been determined — or, as is more often the case, approximated — for a given system, it is often observed that the behavior of the system can be influenced by the choice of certain parameters which man can control. For example, the demand for an economic good is affected by its price and the price can be set by the management. A question then arises as to how these controllable parameters can be chosen so as to attain some desired objective which may be either qualitative or quantitative. The rapid development of science and technology in recent years has brought competitive pressures on management and interest has focused on optimizing the behavior of the system. Some output of the system, such as the profit produced by an economic enterprise, is either to be maximized or minimized subject to the laws governing the behavior of the system. The search for the values of the parameters that maximize or minimize a well defined system performance criterion constitutes the fundamental problem of optimization theory.

The different rules of causality of various systems are described by mathematical relationships having a great variety of structures and in which the control variables appear in many ways. A large body of theory has been developed in recent years to deal with optimization

problems encountered in various fields. Several important optimization techniques like linear programming, dynamic programming, search techniques, maximum principle etc. have been developed to treat various classes of problems. Calculus of variations is a classical mathematical tool that treats certain problems in the area of optimization theory. Specifically, the mathematical models that calculus of variations treats associates values with functions and seeks a function which minimizes or maximizes the objective function. This particular model, though restricted, is capable of describing a very large number of important processes in many fields and has been the subject of continuous study for a long time.

The two point boundary value difficulty encountered in obtaining solutions to practical problems, however, limits the use of both calculus of variations and maximum principle. With the advent of fast digital computers, interest in this area has increased and a large amount of research has been devoted to devising techniques for overcoming this difficulty.

This work is a study of the way in which a recently proposed technique of boundary condition iteration based on differential sensitivity analysis [26, 28] can be used to solve variational problems arising in management decision making. More specifically, the object of this report is to investigate the computational features of this technique with respect to different problems. Chapter 4 deals with an inventory model, Chapter 5 deals with a more complex problem of inventory and advertisement scheduling and Chapter 6 deals with a production and advertisement scheduling model.

CHAPTER 2

NUMERICAL METHODS OF SOLUTION OF NONLINEAR TWO POINT
BOUNDARY VALUE PROBLEMS

NON-LINEAR DYNAMIC SYSTEMS

In general, the variational problem can be stated as follows: Find the m dimensional vector of control variables $\bar{T}(t)$ in the interval $t_0 \leq t \leq t_f$ such that a scalar performance index of the form

$$J = \phi(\bar{X}(t_f), t_f) + \int_{t_0}^{t_f} f_0(\bar{X}(t), \bar{T}(t), t) dt \quad (1)$$

is a minimum (or maximum), while the p dimensional vector of initial conditions $\bar{\psi}(\bar{X}(t_0), t_0) = 0$ and the q dimensional vector of final conditions $\bar{\eta}(\bar{X}(t_f), t_f) = 0$ are satisfied and the n first-order, nonlinear, differential equations

$$\dot{\bar{X}} = \bar{f}(\bar{X}(t), \bar{T}(t), t) \quad (2)$$

are also satisfied. The vector \bar{X} is an n dimensional vector of state variables, and t is the independent variable time. Note that, if the initial time is specified, $p \leq n$ and $q \leq n+1$. It is usually assumed that the problem is deterministic. Furthermore, in the following discussion it is assumed that $\bar{f}(\bar{X}, \bar{T}, t)$ and all of its derivatives are continuous in the interval of interest. It is assumed also that the control variable $\bar{T}(t)$ is unbounded and that there are no constraints on the state history except at the initial and terminal boundaries.

The conditions which must be satisfied if the extremal is to comply with the requirements of the problem statement are discussed by Bliss [1].

They may be summarized as follows. In the interval of interest,

$$\dot{\bar{X}} = \bar{H}^T_{\bar{\lambda}}, \quad \dot{\bar{\lambda}} = -\bar{H}^T_{\bar{X}}, \quad \bar{H}^T_{\bar{T}} = 0 \quad (3)$$

where superscript T denotes transpose and $\bar{\lambda}(t)$ are the Lagrange multipliers.

At the known initial time,

$$\bar{\psi}(\bar{X}(t_0), t_0) = 0, \quad [(P_{\bar{X}} + \bar{\lambda}^T)]_{t_0} d\bar{X}(t_0) = 0. \quad (4)$$

At the unknown final time,

$$\bar{\eta}(\bar{X}(t_f), t_f) = 0, \quad [(P_{\bar{X}} - \bar{\lambda}^T)]_{t_f} d\bar{X}(t_f) = 0, \quad [P_t + \bar{H}]_{t_f} dt_f = 0. \quad (5)$$

If the final time is known, the last condition of Equations (5) will not hold.

The scalar functions P and \bar{H} are defined as follows:

$$P = \phi(\bar{X}(t_f), t_f) + \bar{\mu}^T \bar{\psi}(\bar{X}(t_0), t_0) + \bar{v}^T \bar{\eta}(\bar{X}(t_f), t_f)$$

$$\bar{H} = f_0(\bar{X}, \bar{T}, t) + \bar{\lambda}^T \bar{f}(\bar{X}, \bar{T}, t) \quad (6)$$

where $\bar{\mu}$ and \bar{v} are constant multipliers and the scalar function \bar{H} is known as the generalized Hamiltonian. The conditions given above can be obtained by the standard calculus of variation methods and form a first-order necessary condition for the optimization problem. It is assumed that a well defined minimum of $\bar{H}(\bar{X}, \bar{T}, \bar{\lambda}, t)$ exists so that $\bar{H}_{\bar{T}} = 0$ and $\bar{H}_{\bar{T}, \bar{T}}$ is positive definite everywhere in the interval of interest. With these assumptions, the condition $\bar{H}_{\bar{T}} = 0$ theoretically yields m algebraic equations which can be used to eliminate the m control variables in Equation (3). The results can be expressed as

$$\dot{\bar{X}} = H_{\bar{\lambda}}^T, \quad \dot{\bar{\lambda}} = -H_{\bar{X}}^T \quad (7)$$

where $H = \bar{H}[\bar{X}, \bar{\lambda}, \bar{T}(\bar{X}, \bar{\lambda}, t), t]$. Equations (4), (5), and (7) lead to a conventional, two point boundary value problem in which the conditions on variables \bar{X} and $\bar{\lambda}$ are specified at both ends of the interval of interest.

GENERAL CLASSIFICATION OF NUMERICAL METHODS OF SOLUTION OF VARIATIONAL PROBLEMS

In general the methods used for numerical solution of variational problems can be classified as either direct or indirect. The direct methods use only the process equations and the desired terminal conditions as the starting point to minimize or maximize some desired performance index. The indirect methods, on the other hand, use the conditions required for mathematical optimality as a starting point and seek, by various iterative philosophies, to satisfy these conditions.

DIRECT METHODS

The direct numerical optimization procedure was first suggested by Kelley [9] and is referred to as the gradient method. Other direct optimization methods have been suggested by Bryson et. al. [2] and Kelley et. al. [10]. These methods have been applied with success to many practical problems in spacecraft guidance and control and aeronautics. Advantages usually associated with the steepest descent techniques, as they are frequently called, are that convergence does not depend upon the availability of a good initial estimate of the optimal trajectory as a starting point and that the techniques seek out relative minima rather than merely functionals which are stationary. The main disadvantage associated with the steepest descent techniques is that in many practical applications the convergence rate slows down as the optimum trajectory

is approached. The steepest descent methods have been discussed by several authors in detail, see [2], [9], [10] and [11].

INDIRECT METHODS

The indirect numerical optimization techniques were suggested as early as 1949 by Hestenes [7], who applied a calculus of variation formulation to the study of time-optimal solutions to the fixed end point problem. A number of methods have been proposed for the solution of two point boundary value problems arising in optimal control problems. These may be subdivided into three main classes:

- i. Boundary Condition Iteration Methods
- ii. Control Function Iteration Methods
- iii. Newton-type Iteration Methods

The choice of the method to be adopted depends on the problem and the nature of application. Each problem will have a certain structure and exhibit certain stability properties, although in a non-linear problem it might be very difficult to isolate either. Further the nature of the application may be the deciding criterion. For example, for on-line controls, rapidity of convergence is important. For some problems it may be necessary to obtain extremely accurate trajectories, while in others convergence of the performance functional to within a pre-assigned tolerance may be sufficient. These factors must be considered before selecting any one method.

i. Boundary Condition Iteration Methods

In these methods, typically the control function $T(t)$ is eliminated from the process and adjoint equations by solving $\bar{H}_T = 0$ and the resulting equations are solved by iteration on one of the unknown boundary values say, $\bar{\lambda}(t_0)$. Suitable scalar terminal error function

$V\{\bar{X}[t_f, \bar{\lambda}(t_0)], \bar{\lambda}[t_f, \bar{\lambda}(t_0)]\}$ is then constructed. The boundary value $\bar{\lambda}(t_0)$ is then adjusted till the error function V goes to zero. Several methods have been proposed for driving the error function to zero. Levine has proposed two methods. The first method [18] uses the gradient approach while the second one is concerned with the solution of two-point boundary value problems via a Newton-Raphson scheme using sensitivity information [19]. Levine [19] has reported success with these methods. However, for closed-loop problems, Padmanabhan et. al. [26] have drawn attention to the drawbacks of the method. The success of Newton-Raphson scheme depends upon availability of good estimates of $\bar{\lambda}(t_0)$ to achieve convergence where as in closed loop problems, it is generally impossible to get reasonably close estimates of the artificial variables, such as μ and ν . Padmanabhan et. al. [26] have proposed another method for boundary condition iteration using differential sensitivity measures. These local sensitivity variables are governed by certain differential equations derived from the process equations. They have reported success with recycle problems using the above method.

Another typical method used for boundary condition iteration is the method of Green's function due to Denn and Aris [5]. One of the drawbacks in these schemes is that they usually require manual (as opposed to automatic) step size adjustment resulting in program interruptions which make them unsuitable for high-speed machine computations (Lee [14]).

The boundary condition iteration methods have certain computer programming advantages. Computer logic is simple and fast storage requirements are small. In the problems where the method is successful, accurate trajectories are obtained. The main disadvantage is the inherent

instability of one of the Euler-Lagrange equations.

ii. Control Function Iteration

In recent years control function iteration methods have been extensively used in the solution of optimal control problems (Kelley [9], Fine and Bankoff [6], Luus and Lapidus [22], Merriam [23]). These methods can be broadly classified into first and second variation procedures. The first-variation technique uses a gradient approach to extremize the Hamiltonian \bar{H}_T , while the second-variation method takes into account the local curvature of the level surface of the Hamiltonian in the control space, and converges to the optimum with quadratic rate of convergence. Bryson [3] has developed a first-variation procedure which employs a control correction scheme for both fixed final time and free final time problems. Mitter's algorithm [24] is based on second-variations and exhibits improved rate of convergence. The algorithm is equivalent to Newton's Method in function space. However, for free-final time problems, it is too cumbersome to apply. Padmanabhan and Bankoff [27] have proposed a method which dispenses with Riccati equations used in Mitter's algorithm. Merriam [23] has proposed methods based on both first variations and second variations and has called them relaxation methods.

The primary advantage of these methods is that computations are always performed in the stable direction. However convergence tends to be slow in a certain neighborhood of the optimum.

iii. Newton's Method

Newton's method was first proposed by Hestenes [7] to solve fixed end point problems of the calculus of variations. Kalaba [8] has used this method for a special class of problems and called it quasi-linearization. A mathematical treatment of the quasilinearization method is given by

Lee [14] and has been applied to numerous problems in the fields of industrial and chemical engineering. Extensions of the generalized Newton-Raphson method for variable final time problems have been provided by Long [21], Conrad [4], and Lewallen [20].

Newton's Method essentially linearizes the set of non-linear differential equations around the previous iteration and uses the superposition principle in obtaining the solution of the two-point boundary value problems. The iteration scheme provides a sequence of trajectories which in general converges rather rapidly to the solution of the original non-linear equations.

Meaningful comparisons of these contemporary optimization methods are not found frequently in the literature. Recent studies by Kopp and Moyer [13] and Moyer & Pinkham [25] compare the generalized Newton-Raphson method [12], the second variation method [11], and the classical gradient method [9].

CHAPTER 3

BOUNDARY CONDITION ITERATION METHOD BASED ON
DIFFERENTIAL SENSITIVITY ANALYSIS

THE PROBLEM

The method which is introduced informally in Chapter 2, is developed here for the following variational problem: Find the function

$$T(t) \quad t_0 \leq t \leq t_f \quad (1)$$

Such that the set of functions

$$X_1(t), \dots, X_n(t) \quad t_0 \leq t \leq t_f \quad (2)$$

given by the differential equations

$$\dot{X}_i = f_i(X_1, \dots, X_n, T(t), t), \quad i = 1, \dots, n \quad (3)$$

and end conditions

$$\psi_j(t_0, X_1(t_0), \dots, X_n(t_0), t_f, X_1(t_f), \dots, X_n(t_f)) = 0$$

$$j = 1, \dots, p \leq 2n \quad (4)$$

minimize a function of the form

$$J = \phi(t_0, X_1(t_0), \dots, X_n(t_0), t_f, X_1(t_f), \dots, X_n(t_f)) \quad (5)$$

where the expression \dot{X} represents the first differential $\frac{dX}{dt}$. The variables $X_1(t), \dots, X_n(t)$ are the state variables and $T(t)$ is the control variable. The variable t is the independent variable and can be considered as time. The problem formulated above is essentially the problem of Mayer [1]. Following the classical treatment in the calculus of variations, let us introduce the set of Lagrange multipliers:

$$\lambda_i(t), \quad i = 1, \dots, n \quad (6)$$

and the set of constant multipliers

$$\mu_j, \quad j = 1, \dots, p. \quad (7)$$

Define the functions

$$F(t, \bar{X}, \dot{\bar{X}}, \bar{\lambda}, T) = \sum_{i=1}^n \lambda_i (\dot{X}_i - f_i(t, T, \bar{X})) \quad (8)$$

$$G(t_0, \bar{X}(t_0), t_f, \bar{X}(t_f), \bar{\mu}) = \phi(t_0, \bar{X}(t_0), t_f, \bar{X}(t_f)) \\ + \sum_{j=1}^p \mu_j \psi_j(t_0, \bar{X}(t_0), t_f, \bar{X}(t_f)) \quad (9)$$

where the vectors \bar{X} , $\dot{\bar{X}}$ and $\bar{\lambda}$ represent X_1, \dots, X_n ; $\dot{X}_1, \dots, \dot{X}_n$; and $\lambda_1, \dots, \lambda_n$, respectively. The Euler-Lagrange equations are

$$\frac{d}{dt} \frac{\partial F}{\partial \dot{X}_i} - \frac{\partial F}{\partial X_i} = 0, \quad i = 1, \dots, n \quad (10)$$

$$\frac{\partial F}{\partial T} = 0. \quad (11)$$

The transversality conditions are

$$\left. \frac{\partial G}{\partial X_i} \right|_{t=t_0} - \left. \frac{\partial F}{\partial \dot{X}_i} \right|_{t=t_0} = 0, \quad \left. \frac{\partial G}{\partial X_i} \right|_{t=t_f} + \left. \frac{\partial F}{\partial \dot{X}_i} \right|_{t=t_f} = 0 \\ i = 1, \dots, n. \quad (12)$$

Equations (10) to (12) form a necessary condition for the optimization problem and have been called the multiplier rule by Bliss [1]. Equation (10) can be reduced to

$$\dot{\lambda}_i = \frac{\partial F}{\partial X_i}, \quad i = 1, \dots, n. \quad (13)$$

The system of equations can be put in the canonical form by defining the Hamiltonian

$$H = \sum_{i=1}^n \lambda_i f_i(t, \bar{X}, T). \quad (14)$$

The original system of equations (3) may be written as

$$\dot{\bar{X}} = H_{\bar{X}}(\bar{X}, \bar{\lambda}, T). \quad (15)$$

The adjoint system of equations (10) or (13) may be written as

$$\dot{\bar{\lambda}} = -H_{\bar{\lambda}}(\bar{X}, \bar{\lambda}, T) \quad (16)$$

with equation (11) as

$$H_T(\bar{X}, \bar{\lambda}, T) = 0. \quad (17)$$

The system is now composed of $2n$ differential equations [Equations (15) and (16)], $2n$ transversality conditions [Equations (12)], one equation for the control variable [Equation (11) or (17)], and p end conditions [Equation (4)], to determine the values of the n state variables and n Lagrange multipliers, the $2n$ end values $X_i(t_0)$, $X_i(t_f)$, $i = 1, \dots, n$, the one control variable, and the p constant multipliers. Since the boundary conditions are not all given at the initial point t_0 , the above system forms a two-point boundary value problem. The differential equations which arise in great many practical situations are not linear and cannot be solved analytically. The numerical solution cannot be initiated at either $t = t_0$ or $t = t_f$ because neither $\bar{\lambda}(t_0)$ nor $\bar{X}(t_f)$ is known.

The following discussion on boundary condition iteration follows the work of Padmanabhan and Bankoff [26] closely. For more details, the reader is referred to the original work.

An obvious approach to resolving this computational dilemma is to select a set of boundary conditions on a trial-and-error basis at either boundary so as to reduce the problem to a final value problem or an initial value problem. Therefore a sequence of trajectories converging to the optimum may be obtained by iterating on the boundary values at either end of the process. However, the instability of the adjoint equations to numerical integration in the direction of the process, combined with the fact that the terminal values of the state variables (which are physical quantities) are more easily guessed than those of the adjoint variables, dictates that the iteration be performed on the exit values.

Thus having converted the problem into a final value problem by guessing the final values of the state variables not specified by end conditions given by equation (4) and the values of constant multipliers, μ_i , $i = 1, \dots, p$, the initial conditions given by equation (4) and (12) should match those obtained by backward integration. Presumably, the values at the initial end will not match and further iterations must be made. This can be stated in another way: The iteration process seeks to drive a residual vector \bar{R} to zero by iterating on the boundary conditions. Denoting the initial conditions on the adjoint variables by λ_i^0 , $i = 1, \dots, n$, the residual vector \bar{R} may be defined as

$$\bar{R} = \begin{pmatrix} \psi_1 \\ \vdots \\ \psi_m \\ \lambda_1(t_0) - \lambda_1^0 \\ \vdots \\ \lambda_n(t_0) - \lambda_n^0 \end{pmatrix} \quad m \leq p \quad (18)$$

It will be observed that only the initial conditions in Equation (4) are included.

It is clear that we will need some measure of the "miss distance" at every iteration. To this end, the residual norm Ω is defined as

$$\Omega = ||R||^2 = \sum_{i=1}^n \theta_i^2 + \sum_{i=1}^n \psi_i^2. \quad (19)$$

If $\Omega > 0$, then the vector $\bar{\pi} = \begin{pmatrix} \bar{X}(t_f) \\ \bar{\mu} \end{pmatrix}$ is perturbed such that Ω is

decreased. The most rapid decrease in Ω is accomplished by moving in the $\bar{\pi}$ space in the direction of the negative gradient of Ω . Obtaining Ω in terms of $\bar{\pi}$ through the expressions for \bar{R} , the implied partial derivative $\frac{\partial \Omega}{\partial \bar{\pi}}$ may be computed. However, the partial derivative does not give the true sensitivity of Ω with respect to $\bar{\pi}$, as it ignores the constraints imposed by the process. A valid measure is the constrained total derivative

$$\frac{d\Omega}{d\bar{\pi}} = \frac{\partial \Omega}{\partial \bar{\pi}} + \Omega \frac{\partial \bar{X}(t_0)}{\partial \bar{\pi}} + \Omega \frac{\partial \bar{\lambda}(t_0)}{\partial \bar{\pi}} \quad (20)$$

where the last two terms on the right side establish the connection to the canonical system (15), (16) and (17). From (19), it follows that

$$\frac{d\Omega}{d\bar{\pi}} = 2\Gamma(\bar{\pi})^T \bar{R}(\bar{\pi}) \quad (21)$$

where $\Gamma(\bar{\pi})$, the constrained Jacobian $\frac{d\bar{R}}{d\bar{\pi}}$, is given explicitly by:

$$\Gamma(\bar{\pi}) = \bar{R}_{\bar{\pi}} + \bar{R}_{\bar{X}(t_0)} \frac{\partial \bar{X}(t_0)}{\partial \bar{\pi}} + \bar{R}_{\bar{\lambda}(t_0)} \frac{\partial \bar{\lambda}(t_0)}{\partial \bar{\pi}}. \quad (22)$$

The boundary condition iteration can be constructed either as a gradient method for minimizing Ω , or as Newton's scheme for solving the system

$$\bar{R}(\bar{\pi}) = 0. \quad (23)$$

The advantages of the gradient method and variations of this method are in its simplicity. The convergence of the method is not contingent upon a good initial estimate as the starting condition. It is assured that the function to be minimized is decreased after each iteration cycle. The possible disadvantages are the slow convergence rate as the optimal trajectory is reached, and the unspecified step size. On the other hand, the Newton-Raphson has the advantage of possible improved rate of convergence in the terminal phase of the iteration technique.

As is usually the case, it is difficult to state dogmatically the superiority of any one method over another. A combination of two of the methods in practice might well be used to advantage. It is obvious that one can easily combine both these schemes as follows. The $(k+1)^{st}$ iterate is obtained from:

$$\bar{\pi}^{k+1} = \bar{\pi}^k - [(1-q_k) \gamma_k' \Gamma(\bar{\pi}^k)^T + q_k \Gamma(\bar{\pi}^k)^{-1}] \bar{R}(\bar{\pi}^k). \quad (24)$$

The introduction of q_k in the above formula gives control over the iteration process and its rate of convergence. If $q_k = 1$, Equation (24) reduces to Newton's Method, while $q_k = 0$ corresponds to the gradient approach. This allows one to use the gradient scheme in the beginning with a crude estimate of $\bar{\pi}$ and progress towards the solution. Once the iterates are brought to the proximity of the solution, the Newton-Raphson scheme can be used to achieve rapid convergence.

In order to use the iteration formula (24), the sensitivities of the residuals $\bar{\theta}$ and $\bar{\psi}$ to small perturbations in $\bar{\pi}$ are required. These in turn, are expressible in terms of the sensitivities of $\bar{X}(t)$ and $\bar{\lambda}(t)$, which are defined as the local gradients $\frac{\partial \bar{X}(t)}{\partial \bar{X}(t_f)}$, $\frac{\partial \bar{\lambda}(t)}{\partial \bar{X}(t_f)}$, $\frac{\partial \bar{X}(t)}{\partial \bar{\mu}}$, $\frac{\partial \bar{\lambda}(t)}{\partial \bar{\mu}}$.

These are tied to the process via the sensitivity differential equations, whose derivation is typified by the following treatment for $\frac{\partial \bar{X}(t)}{\partial \bar{X}(t_f)}$.

Representing equation (15) as the nonlinear integral equation

$$\bar{X}(t) = \bar{X}(t_f) + \int_{t_f}^t H_{\bar{\lambda}}(\bar{X}(\xi), T(\xi), \xi) d\xi \quad (25)$$

we get by direct differentiation

$$\frac{\partial \bar{X}(t)}{\partial \bar{X}(t_f)} = I + \int_{t_f}^t \left\{ H_{\bar{\lambda}, \bar{X}} \frac{\partial \bar{X}(\xi)}{\partial \bar{X}(t_f)} + H_{\bar{\lambda}, T} \frac{\partial T(\xi)}{\partial \bar{X}(t_f)} \right\} d\xi. \quad (26)$$

By forcing $T(t)$ to satisfy equation (17), we restrict the trajectory displacements $\delta \bar{X}(t)$, $\delta \bar{\lambda}(t)$ to be tangential to the surface $H_T = 0$. This results in the variational equation:

$$H_{T,T} \frac{\partial T(t)}{\partial \bar{X}(t_f)} + H_{T,\bar{X}} \frac{\partial \bar{X}(t)}{\partial \bar{X}(t_f)} + H_{T,\bar{\lambda}} \frac{\partial \bar{\lambda}(t)}{\partial \bar{X}(t_f)} = 0. \quad (27)$$

Combining (26) and (27) we arrive at

$$\frac{\partial \bar{X}(t)}{\partial \bar{X}(t_f)} = I + \int_{t_f}^t \left\{ A(\xi) \frac{\partial \bar{X}(\xi)}{\partial \bar{X}(t_f)} + B(\xi) \frac{\partial \bar{\lambda}(\xi)}{\partial \bar{X}(t_f)} \right\} d\xi \quad (28)$$

$$\text{where } A(t) = H_{\bar{\lambda},\bar{X}} - H_{\bar{\lambda},T} H_{T,T}^{-1} H_{T,\bar{X}} \quad (29)$$

$$B(t) = - H_{\bar{\lambda},T} H_{T,T}^{-1} H_{T,\bar{\lambda}}. \quad (30)$$

Equivalently, (28) can be represented as the differential system:

$$\frac{d}{dt} \left(\frac{\partial \bar{X}(t)}{\partial \bar{X}(t_f)} \right) = A(t) \frac{\partial \bar{X}(t)}{\partial \bar{X}(t_f)} + B(t) \frac{\partial \bar{\lambda}(t)}{\partial \bar{X}(t_f)} \quad (31)$$

$$\left. \frac{\partial \bar{X}(t)}{\partial \bar{X}(t_f)} \right|_{t=t_f} = I. \quad (32)$$

A similar treatment for the other sensitivity co-efficients leads to their respective equations,

$$\frac{d}{dt} \left(\frac{\partial \bar{X}(t)}{\partial \bar{\mu}} \right) = A(t) \frac{\partial \bar{X}(t)}{\partial \bar{\mu}} + B(t) \frac{\partial \bar{\lambda}(t)}{\partial \bar{\mu}} \quad (33)$$

$$\left. \frac{\partial \bar{X}(t)}{\partial \bar{\mu}} \right|_{t=t_f} = 0 \quad (34)$$

$$\frac{d}{dt} \left(\frac{\partial \bar{\lambda}(t)}{\partial \bar{X}(t_f)} \right) = C(t) \frac{\partial \bar{X}(t)}{\partial \bar{X}(t_f)} + D(t) \frac{\partial \bar{\lambda}(t)}{\partial \bar{X}(t_f)} \quad (35)$$

$$\left. \frac{\partial \bar{\lambda}(t)}{\partial \bar{X}(t_f)} \right|_{t=t_f} = 0 \quad (36)$$

$$\frac{d}{dt} \left(\frac{\partial \bar{\lambda}(t)}{\partial \bar{\mu}} \right) = C(t) \frac{\partial \bar{X}(t)}{\partial \bar{\mu}} + D(t) \frac{\partial \bar{\lambda}(t)}{\partial \bar{\mu}} \quad (37)$$

$$\left. \frac{\partial \bar{\lambda}(t)}{\partial \bar{\mu}} \right|_{t=t_f} = \bar{\psi}_{\bar{X}(t_f)} \cdot \quad (38)$$

The integration of the above system of differential equations backwards yields the sensitivity co-efficients at the initial point and hence the constrained Jacobian matrix $\Gamma(\bar{\pi})$ is completely defined. Note that the system of differential sensitivity equations forms a final value problem.

It will be noted that in the iteration scheme given by Equation (24), a factor γ_k' has been introduced. The factor γ_k' is given by

$$\gamma_k' = \gamma_k \frac{||R^k||^2}{||\Gamma_k^T R^k||^2} \cdot \quad (39)$$

The above representation explicitly shows the normalization of the gradient $\frac{d\Omega}{d\bar{\pi}} = 2\Gamma^T(\bar{\pi})\bar{R}$. Our immediate concern is how to fix the bounds on the weighting factor q_k and the step size factor γ_k , for which the iteration process reaches the solution of the residual equation $\bar{R}(\bar{\pi}) = 0$.

It has been proved by Padmanabhan and Bankoff [28] that for $\gamma_k, q_k \in [0, 1]$

1. $\Omega^k \rightarrow 0$ monotonically as $k \rightarrow \infty$, where k is the iteration number.
2. $\bar{\pi}^k \rightarrow \bar{\pi}_0$ at a superlinear rate, where $\bar{\pi}_0$ is the solution at the optimum.

3. For any arbitrary initial policy, the algorithm converges to the same optimum. It has been assumed here that the equation $\bar{R}(\pi) = 0$ has a unique solution at $\bar{\pi} = \bar{\pi}_0$.

The computational scheme can now be summarized as follows:

1. Estimate the vector $\bar{\pi}$ from physical considerations or otherwise.
2. If Equation (17) admits of explicit solution for the control variable $T(t)$, eliminate $T(t)$ from Equations (15) and (16). Otherwise assume a reasonable value of the control variable $T(t)$.
3. Calculate exit values of the adjoint variables from Equation (12).
4. Integrate the system of Equations (15) and (16) backwards.
5. If the control variable $T(t)$ was assumed in step 2, check if Equation (17) is satisfied. If not, find a solution of the Equation (17) by control iteration using direct second variation method (DSV) outlined in Appendix (1). If the control variable was not assumed, generate the control program.
6. Compute the residual vector \bar{R} , Equation (18).
7. If $||\bar{R}||$ is sufficiently close to zero, the problem is solved and the computation terminated. If not, improve the estimate of $\bar{\pi}$ by iterating on boundary conditions. The following additional steps are required.
8. Compute the exit values of sensitivity co-efficients from Equations (32), (34), (36) and (38).
9. Integrate all the sensitivity differential equations (31), (33), (35) and (37) from $t = t_f$ to $t = t_0$. Compute the constrained Jacobian matrix $\frac{d\bar{R}}{d\bar{\pi}}$.
10. Improve the estimate of $\bar{\pi}$ by using Equation (24) and return to step 3.

DISCUSSION

The main advantage of this technique lies in its simplicity. The programming is comparatively straightforward. This method has the additional advantage of being able to converge starting from a crude estimate of the final values of the state variables. It combines the advantages of both the gradient approach and the quadratic convergence characteristic of the Newton-Raphson approach. Quadratic convergence means that the error in the $(k+1)^{\text{st}}$ iteration tends to be proportional to the square of the error in the k^{th} iteration. This method has also been found to be effective in closed-loop control problems where quasilinearization failed to converge [28].

In spite of all the advantages, this technique also has its difficulties. There are two main difficulties.

The first difficulty arises from the fact that the number of sensitivity differential equations in case of open loop problems, are $2.n^2$ where n is the number of state variables. Hence, as the number of state variables increase, the computer memory required is tremendously large.

The second difficulty arises because of the unspecified step size γ_k and the weighting factor q_k . Trial and error is required to find a suitable combination of the factors for each problem.

For a detailed mathematical treatment of this technique and related convergence theorems, the reader is referred to Padmanabhan and Bankoff [26,28]. This technique has been used to solve three typical management problems in Chapters 4, 5 and 6. The numerical and computational aspects of this method have also been discussed.

CHAPTER 4

APPLICATION TO AN INVENTORY MODEL

In this chapter, the computational aspects of this technique will be discussed with respect to its application to an inventory model having one state variable and one control variable. As will be seen later, this being a smaller dimensioned problem, it is easier to solve.

DEVELOPMENT OF THE MODEL

This model is taken from a paper by Lee and Shaikh [16]. Consider the case of a manufacturing organization whose sales rate is known with certainty. The rate of change of inventory level $X_1(t)$ is given by

$$\frac{dX_1(t)}{dt} = P(t) - Q(t) \quad t_0 \leq t \leq t_f \quad (1)$$

where $P(t)$ = production rate at time t
and $Q(t)$ = sales rate at time t .

The management wishes to minimize the cost functional given by

$$C_T = \int_{t_0}^{t_f} [C_I(I_m - X_1(t))^2 + C_p \exp(P_m - P(t))^2] dt \quad (2)$$

where C_T is the total cost of production and inventory. C_p is the minimum production cost which occurs when the production rate is P_m . The quantity P_m can be considered as the production capacity of the manufacturing plant. Since the plant is designed for a capacity P_m , an increase in capacity may require additional equipment and manpower and this can be very expensive. On the other hand, a decrease in production rate below P_m will be equally expensive due to maintenance of unused equipment and idle labor which cannot be decreased due to contract agreements. C_I is the cost of carrying inventory and the quantity I_m

can be considered as the capacity for storage of inventory. In actual practice, the minimum storage cost is obtained when the storage capacity is completely utilized. In addition, the cost functional has the smoothing capability which is frequently desirable for many manufacturing processes. In this case I_m and P_m can be considered as the desirable inventory and production levels. It is further assumed that the sales forecast is known and is given by the linear relation

$$Q(t) = a + bt \quad t_0 \leq t \leq t_f \quad (3)$$

and the initial inventory is given as

$$X_1(t_0) = X_1^0 \quad (4)$$

where a , b and X_1^0 are known constants.

The role of management in this particular case is to select the optimal policy from among all feasible solutions which gives the minimum cost.

DEFINITION OF THE PROBLEM

Find the control function $P(t)$ in the interval $t_0 \leq t \leq t_f$ such that the scalar performance index

$$C_T = \int_{t_0}^{t_f} [C_I (I_m - X_1(t))^2 + C_p \exp(P_m - P(t))^2] dt \quad (5)$$

is a minimum, while the initial condition

$$X_1(t_0) = X_1^0 \quad (6)$$

and the first order, nonlinear, differential equation

$$\frac{dX_1(t)}{dt} = P(t) - (a + bt) \quad (7)$$

is satisfied. Note that, it has been assumed that the problem is deterministic, that the control variable $P(t)$ and its derivatives are continuous in the interval of interest. It is also assumed that the control variable $P(t)$ is unbounded and that there are no constraints on the state history except at the initial boundary. In a practical problem, the management may wish to specify the final inventory also.

FORMULATION OF THE PROBLEM

This problem was formulated by following the classical treatment of Calculus of Variations and the numerical results are obtained by using the technique outlined in Chapter 3.

Introduce another state variable

$$X_2(t) = \int_{t_0}^t [C_I(I_m - X_1(t))^2 + C_P \exp(P_m - P(t))^2] dt \quad (8)$$

differentiating with respect to the independent variable t ,

$$\frac{dX_2(t)}{dt} = C_I(I_m - X_1(t))^2 + C_P \exp(P_m - P(t))^2 \quad (9)$$

$$\text{with } X_2(t_f) = C_T \quad (10)$$

$$\text{and } X_2(t_0) = 0.$$

Therefore the problem may be reformulated as the Mayer's Problem as

Minimize $X_2(t_f)$ subject to

$$\dot{X}_1 = P(t) - (a + bt) \quad (12)$$

$$\dot{X}_2 = C_I(I_m - X_1(t))^2 + C_P \exp(P_m - P(t))^2 \quad (13)$$

and boundary conditions

$$X_1(t_0) = X_1^0 \quad (14)$$

$$x_2(t_0) = 0 \quad (15)$$

where the notation $(\dot{})$ represents $\frac{d}{dt}$.

Introduce the Lagrange multipliers $\lambda_1(t)$, $\lambda_2(t)$ and the constant multipliers μ_1 and μ_2 and define the functions

$$F = \lambda_1(t) [\dot{x}_1 + a + bt - P(t)] + \lambda_2(t) [\dot{x}_2 - C_I(I_m - x_1(t))^2 - C_P \exp(P_m - P(t))^2] \quad (16)$$

$$\text{and } G = \mu_1(x_1(t_0) - x_1^0) + \mu_2 x_2(t_0). \quad (17)$$

The Euler Lagrange equations [15]

$$\frac{d}{dt} \frac{\partial F}{\partial \dot{x}_i} - \frac{\partial F}{\partial x_i} = 0 \quad i = 1, 2 \quad (18)$$

and

$$\frac{\partial F}{\partial P} = 0 \quad (19)$$

can now be applied to equation (16) to obtain the following relationships

$$\frac{d\lambda_1(t)}{dt} = 2\lambda_2 C_I (I_m - x_1)^2 \quad (20)$$

$$\frac{d\lambda_2(t)}{dt} = 0 \quad (21)$$

$$\text{and } (P_m - P(t)) \exp(P_m - P(t))^2 - \frac{\lambda_1}{2\lambda_2 C_P} = 0. \quad (22)$$

We need two boundary conditions for the two Lagrange multipliers λ_1 and λ_2 . They are obtained by applying the transversality conditions [15]

$$\left. \frac{\partial G}{\partial \dot{X}_1} \right|_{t=t_f} + \left. \frac{\partial F}{\partial \dot{X}_1} \right|_{t=t_f} = 0. \quad (23)$$

Applying this condition to equations (16) and (17), we get

$$\lambda_1(t_f) = 0 \quad (24)$$

$$\lambda_2(t_f) = -1. \quad (25)$$

Now we have four differential conditions with two initial and two final conditions, which make the problem, a two point boundary value type.

In addition we have an implicit equation (22) for determining the control variable $P(t)$.

BOUNDARY CONDITION ITERATION

As discussed in Chapter 3, we need the differential sensitivity measures $\frac{\partial \bar{X}(t)}{\partial \bar{X}(t_f)}$ and $\frac{\partial \bar{\lambda}(t)}{\partial \bar{X}(t_f)}$ to iterate on the boundary conditions.

The differential sensitivity equations are given by

$$\frac{d}{dt} \left(\frac{\partial \bar{X}(t)}{\partial \bar{X}(t_f)} \right) = A(t) \frac{\partial \bar{X}(t)}{\partial \bar{X}(t_f)} + B(t) \frac{\partial \bar{\lambda}(t)}{\partial \bar{X}(t_f)} \quad (26)$$

$$\frac{d}{dt} \left(\frac{\partial \bar{\lambda}(t)}{\partial \bar{X}(t_f)} \right) = C(t) \frac{\partial \bar{X}(t)}{\partial \bar{X}(t_f)} + D(t) \frac{\partial \bar{\lambda}(t)}{\partial \bar{X}(t_f)} \quad (27)$$

$$\text{and } \left. \frac{\partial \bar{X}(t)}{\partial \bar{X}(t_f)} \right|_{t=t_f} = I \quad (28)$$

$$\left. \frac{\partial \bar{\lambda}(t)}{\partial \bar{X}(t_f)} \right|_{t=t_f} = 0 \quad (29)$$

where I is the identity matrix. For the particular problem, the matrices $A(t)$, $B(t)$, $C(t)$ and $D(t)$ are given by

$$A(t) = \begin{pmatrix} 0 & 0 \\ -2C_I(I_m - X_1) & 0 \end{pmatrix} \quad (30)$$

$$B(t) = \begin{pmatrix} \frac{-1}{H_{P,P}} & \frac{2C_P(P_m - P(t)) \exp(P_m - P(t))^2}{H_{P,P}} \\ \frac{2C_P(P_m - P(t)) \exp(P_m - P(t))^2}{H_{P,P}} & \frac{-4C_P^2(P_m - P(t))^2 (\exp(P_m - P(t))^2)^2}{H_{P,P}} \end{pmatrix} \quad (31)$$

$$C(t) = \begin{pmatrix} -2\lambda_2 C_I & 0 \\ 0 & 0 \end{pmatrix} \quad (32)$$

$$\text{and } D(t) = \begin{pmatrix} 0 & 2C_I(I_m - X_1) \\ 0 & 0 \end{pmatrix} \quad (33)$$

$$\text{where } H_{P,P} = 2\lambda_2 C_P \exp(P_m - P(t))^2 [1 + 2(P_m - P(t))^2]. \quad (34)$$

Thus for the differential sensitivity measures we have four differential equations given by equation (26), four differential equations given by equation (27) and eight boundary conditions given by equations (28) and (29).

In the terminology of Chapter 3, the residual norm vector is

$$\bar{R} = \begin{pmatrix} \psi_1 \\ \psi_2 \end{pmatrix} \quad (35)$$

where

$$\begin{aligned} \psi_1 &= X_1(0) - X_1^0 = 0 \\ \psi_2 &= X_2(0) = 0 \end{aligned} \quad (36)$$

$$\text{and } \bar{\pi} = \begin{pmatrix} X_1(t_f) \\ X_2 \quad t_f \end{pmatrix}. \quad (37)$$

The elements of the constrained Jacobian, $\frac{d\bar{R}}{d\bar{\pi}}$ are given by

$$\Gamma(\bar{\pi}) = \begin{pmatrix} \frac{\partial X_1(t_0)}{\partial X_1(t_f)} & \frac{\partial X_1(t_0)}{\partial X_2(t_f)} \\ \frac{\partial X_2(t_0)}{\partial X_1(t_f)} & \frac{\partial X_2(t_0)}{\partial X_2(t_f)} \end{pmatrix}. \quad (38)$$

The elements of the above matrix are obtained by integrating the differential sensitivity equations backwards. All the necessary information for the iteration scheme

$$\bar{\pi}^{k+1} = \bar{\pi}^k - [(1-q_k) \gamma_k' \Gamma(\bar{\pi}^k)^T + q_k \Gamma(\bar{\pi}^k)^{-1}] \bar{R}(\bar{\pi}^k) \quad (39)$$

is available.

It would be observed that equation (22) does not admit explicit analytic solution for $P(t)$. In general, however, this control can be found by

iterative procedures for solving the algebraic equation (22), in conjunction with differential system (7), (9), (20) and (21). The Direct Second Variational Method, a procedure proposed by Padmanabhan and Bankoff [26] was used. The details of the method are given in Appendix (1).

NUMERICAL ASPECTS

In order to solve this problem numerically, the constants were assumed to have the following values:

$$\begin{aligned} a &= 2.0 & I_m &= 10.0 \\ b &= 1.0 & C_p &= 0.001 \\ X_1^0 &= 5.0 & P_m &= 5.0 \\ C_I &= 0.1 & t_0 &= 0 \\ t_f &= 1.0 \end{aligned}$$

This problem was solved on an IBM 360-50 computer. The Runge-Kutta integration formulae were used to integrate equations (12), (13), (20), (21), (26) and (27). The step size used for the numerical integration was 0.01.

As discussed previously, we need initial approximations of the final values of the state variables to start the solution of this problem. Since there are only two state variables, we needed the initial approximations of $X_1(t_f)$ and $X_2(t_f)$. These values were obtained from intuition and knowledge of the system. After obtaining optimum solution from one initial approximation, other initial values were tried to study the convergence properties of the technique. The set of initial approximations used are given in Table 4.1. It would be observed that the initial approximations used were on both sides of the optimum.

The convergence rates of the control variable — production rate — are shown in Figs. (4.1) and (4.2). The convergence rates of the inventory level and the total cost are shown in Figs. (4.3), (4.4), (4.5) and (4.6) respectively.

Its observed from Figs. (4.1) and (4.2) that production rate curve is very close to the optimum after the fifth iteration. This near optimum trajectory is reached irrespective of the initial guess on control variable in five iterations. However, the trajectories for cost and inventory level are far from optimum at the fifth iteration.

The rates of convergence of the final inventory $X_1(t_f)$, total cost $X_2(t_f)$, and the residuals are given in Tables (4.2), (4.3) and (4.4) respectively. The rate of convergence during the first five or six iterations is fast while the rate of convergence is slow thereafter. This is illustrated in Fig. (4.7). Similar convergence rates hold for other variables.

It can be seen from Tables (4.2), (4.3) and (4.4) that with initial guesses far from optimum, the correct solution is obtained without significant increase in the number of iterations. It is an important advantage of the scheme that correct solutions are obtained starting with crude estimates of the final values of the state variables. The convergence rate of the final inventory is shown in Figure (4.7).

Since the weighting factor q_k provides control over the iteration process, the effect of q_k on convergence rate was studied. The results are shown in Table (4.5). The number of iterations required were minimum with $q_k = 1$. This is because of the fact that the initial approximation was close to the optimum and Newton-Raphson techniques better near the optimum than the Gradient technique.

RESULTS

The optimal cost in the problem was 0.9187 and the optimal initial and final values of the state variables were:

$$x_1(0) = 5.0 \quad x_1(1) = 9.3088$$

$$x_2(0) = 0.0 \quad x_2(1) = 0.9187$$

Table 4.1. Initial Approximations Used for the Inventory Model

Set No.	$x_1^0(t_f)$	$x_2^0(t_f)$
1	9.0	0.95
2	7.0	0.5
3	4.0	1.5
4	20.0	2.0
5	50.0	10.0
Optimum	9.3088	0.9186

Table 4.2. Convergence Rate of Final Inventory, $X_1(t_f)$ Initial Approximation of Production $P(t) = 6.0$

$$q_k = \left(\frac{k}{100+k}\right)^3, \gamma_k = 0.5$$

$x_1^0(t_f)$	9.0	7.0	4.0	20.0	50.0
$x_2^0(t_f)$	0.95	0.5	1.5	2.0	10.0
Iteration					
1	9.0	7.0	4.0	20.0	
2	9.1346	7.9525	5.8178	15.7915	
3	9.2144	8.5157	7.0792	10.9164	
4	9.2683	8.8583	8.1279	8.7561	
5	9.2884	9.0921	8.8487	8.8187	
6	9.2987	9.2282	9.0135	9.1387	
7	9.3036	9.2483	9.2038	9.1644	
8	9.3062	9.2866	9.2246	9.2574	
9	9.3075	9.2914	9.2784	9.2672	
10	9.3082	9.3024	9.2846	9.2936	
11	9.3084	9.3038	9.2999	9.2968	
12	9.3086	9.3069	9.3018	9.3044	
13	9.3087	9.3073	9.3062	9.3053	
14	9.3087	9.3082	9.3067	9.3075	
15	—	9.3084	9.3080	9.3078	
16	—	9.3086	9.3082	9.3084	
17	—	9.3087	9.3086	9.3085	
18	—	9.3087	9.3086	9.3087	
19	—	—	9.3087	9.3087	
20	—	—	9.3087	9.3088	

Did not converge in 1 hour of computing time.

8 iterations, $\|R\| = 9.85$

Table 4.3. Convergence Rate of Total Cost, $X_2(t_f)$, in an Inventory Model
Initial Approximation of Production $P(t)$, = 6.0

$$q_k = \left(\frac{k}{100+k}\right)^3, \gamma_k = 0.5$$

$x_1^0(t_f)$	9.0	7.0	4.0	20.0	50.0
$x_2^0(t_f)$	0.95	0.5	1.5	2.0	10.0

Iteration

1	0.95	0.5	1.5	2.0
2	0.9934	0.9301	2.1611	3.0733
3	1.0049	1.1538	2.5591	3.1671
4	0.9929	1.2419	2.6475	2.1629
5	0.9558	1.2339	1.7985	1.5022
6	0.9373	1.1022	1.3226	1.3275
7	0.9280	1.0060	1.1664	1.1144
8	0.9233	0.9714	1.0370	1.0426
9	0.9210	0.9439	0.9918	0.9782
10	0.9198	0.9339	0.9537	0.9550
11	0.9192	0.9259	0.9399	0.9361
12	0.9189	0.9230	0.9288	0.9292
13	0.9188	0.9207	0.9248	0.9237
14	0.9187	0.9199	0.9216	0.9217
15	—	0.9192	0.9204	0.9201
16	—	0.9190	0.9195	0.9195
17	—	0.9188	0.9191	0.9190
18	—	0.9187	0.9188	0.9189
19	—	—	0.9187	0.9187
20	—	—	0.9187	0.9187

Did not converge in 1 hour of computing time.

After 8 iterations $||R|| = 9.85$

Table 4.4. Convergence Rate of Residuals ψ_1 , ψ_2 and $||R||$

Initial Approximation of Production $P(t) = 6.0$

$$a_k = \left(\frac{k}{100+k}\right)^3, \quad \gamma_k = 0.5$$

Iterations	$x_1^0(t_f)$	$x_2^0(t_f)$								
			ψ_1	ψ_2	$ R $	ψ_1	ψ_2	$ R $	ψ_1	ψ_2
1	-5.658			-5.534	7.914	1.472 x 10 ⁻¹	-7.987	1.676 x 10 ⁻¹		
2	-3.775			-2.143	4.341	1.041 x 10 ⁻¹	-2.138 x 10 ⁻¹	1.041 x 10 ⁻¹		
3	-2.451			-2.424	2.463	5.051	3.117	5.936		
4	-1.328			8.526 x 10 ⁻¹	1.578	-6.359 x 10 ⁻¹	8.669 x 10 ⁻¹	1.075		
5	-5.318 x 10 ⁻¹			5.696 x 10 ⁻¹	7.792 x 10 ⁻¹	-5.655 x 10 ⁻¹	2.517 x 10 ⁻¹	6.190 x 10 ⁻¹		
6	-3.443 x 10 ⁻¹			2.092 x 10 ⁻¹	4.028 x 10 ⁻¹	-1.998 x 10 ⁻¹	2.986 x 10 ⁻¹	3.593 x 10 ⁻¹		
7	-1.239 x 10 ⁻¹			1.802 x 10 ⁻¹	2.188 x 10 ⁻¹	-1.699 x 10 ⁻¹	1.025 x 10 ⁻¹	1.985 x 10 ⁻¹		
8	-9.959 x 10 ⁻²			6.433 x 10 ⁻²	1.186 x 10 ⁻²	-6.089 x 10 ⁻²	9.119 x 10 ⁻²	1.096 x 10 ⁻²		
9	-3.609 x 10 ⁻²			5.379 x 10 ⁻²	6.477 x 10 ⁻²	-4.931 x 10 ⁻²	3.304 x 10 ⁻²	5.936 x 10 ⁻²		
10	-2.878 x 10 ⁻²			1.971 x 10 ⁻²	3.488 x 10 ⁻²	-1.799 x 10 ⁻²	2.681 x 10 ⁻²	3.228 x 10 ⁻²		
11	-1.054 x 10 ⁻²			1.569 x 10 ⁻²	1.891 x 10 ⁻²	-1.424 x 10 ⁻²	9.898 x 10 ⁻³	1.734 x 10 ⁻³		
12	-8.320 x 10 ⁻³			5.808 x 10 ⁻³	1.015 x 10 ⁻³	-5.236 x 10 ⁻³	7.777 x 10 ⁻³	9.376 x 10 ⁻³		
13	-3.055 x 10 ⁻³			4.550 x 10 ⁻³	5.481 x 10 ⁻³	-4.115 x 10 ⁻³	2.883 x 10 ⁻³	5.025 x 10 ⁻³		
14	-2.416 x 10 ⁻³			1.682 x 10 ⁻³	2.944 x 10 ⁻³	-1.513 x 10 ⁻³	2.252 x 10 ⁻³	2.713 x 10 ⁻³		
15	-8.965 x 10 ⁻⁴			1.317 x 10 ⁻³	1.593 x 10 ⁻³	-1.202 x 10 ⁻³	8.297 x 10 ⁻⁴	1.460 x 10 ⁻³		
16	-6.914 x 10 ⁻⁴			4.912 x 10 ⁻⁴	8.481 x 10 ⁻⁴	-4.436 x 10 ⁻⁴	6.543 x 10 ⁻⁴	7.905 x 10 ⁻⁴		
17	-2.594 x 10 ⁻⁴			3.767 x 10 ⁻⁴	4.574 x 10 ⁻⁴	-3.490 x 10 ⁻⁴	2.421 x 10 ⁻⁴	4.247 x 10 ⁻⁴		
18	-2.050 x 10 ⁻⁴			1.376 x 10 ⁻⁴	2.469 x 10 ⁻⁴	-1.306 x 10 ⁻⁴	1.901 x 10 ⁻⁴	2.307 x 10 ⁻⁴		
19	-8.106 x 10 ⁻⁵			1.089 x 10 ⁻⁴	1.358 x 10 ⁻⁴	-1.049 x 10 ⁻⁴	6.893 x 10 ⁻⁵	1.255 x 10 ⁻⁴		
20	-			-	-	-3.814 x 10 ⁻⁵	5.697 x 10 ⁻⁵	6.856 x 10 ⁻⁵		

Table 4.4. Convergence Rate of Residuals ψ_1 , ψ_2 and $||R||$ (contd.)

Initial Approximation of Production, $P(t) = 6.0$

$$q_k = \left(\frac{k}{100+k} \right)^3, \quad \gamma_k = 0.5$$

$x_1^0(t_f)$	9.0	7.0				
$x_2^0(t_f)$	0.95	0.5				
Iteration	ψ_1	ψ_2	$ R $	ψ_1	ψ_2	$ R $
1	-3.597×10^{-1}	-1.726×10^{-1}	3.990×10^{-1}	-2.535	-2.386	3.481
2	-2.046×10^{-1}	-3.822×10^{-2}	2.081×10^{-1}	-1.518	-1.018	1.828
3	-1.115×10^{-1}	2.567×10^{-2}	1.144×10^{-1}	-9.036×10^{-1}	-3.240×10^{-2}	9.598×10^{-1}
4	-4.799×10^{-2}	4.851×10^{-2}	6.824×10^{-2}	-5.208×10^{-1}	1.987×10^{-2}	5.212×10^{-1}
5	-2.418×10^{-2}	2.423×10^{-2}	3.423×10^{-2}	-2.539×10^{-2}	1.737×10^{-1}	3.077×10^{-1}
6	-1.196×10^{-2}	1.235×10^{-2}	1.719×10^{-2}	-9.529×10^{-2}	1.318×10^{-1}	1.627×10^{-1}
7	-6.090×10^{-3}	6.121×10^{-3}	8.634×10^{-3}	-7.163×10^{-2}	4.870×10^{-2}	8.662×10^{-2}
8	-2.997×10^{-3}	3.136×10^{-3}	4.338×10^{-3}	-2.632×10^{-2}	3.872×10^{-2}	4.682×10^{-2}
9	-1.569×10^{-3}	1.524×10^{-3}	2.187×10^{-3}	-2.061×10^{-2}	1.432×10^{-2}	2.510×10^{-2}
10	-7.210×10^{-4}	8.246×10^{-4}	1.095×10^{-3}	-7.586×10^{-3}	1.125×10^{-2}	1.356×10^{-2}
11	-4.368×10^{-4}	3.562×10^{-4}	5.636×10^{-4}	-5.961×10^{-3}	4.170×10^{-3}	7.275×10^{-3}
12	-1.698×10^{-4}	2.388×10^{-5}	2.930×10^{-4}	-2.185×10^{-3}	3.266×10^{-3}	3.929×10^{-3}
13	-1.326×10^{-4}	8.710×10^{-5}	1.586×10^{-4}	-1.733×10^{-3}	1.206×10^{-3}	2.111×10^{-3}
14	-	-	-	-6.370×10^{-4}	9.492×10^{-4}	1.143×10^{-3}
15	-	-	-	-5.045×10^{-4}	3.488×10^{-4}	6.133×10^{-4}
16	-	-	-	-1.917×10^{-4}	2.746×10^{-4}	3.349×10^{-4}
17	-	-	-	-1.478×10^{-4}	1.106×10^{-4}	1.794×10^{-4}
18	-	-	-	-5.627×10^{-5}	7.942×10^{-5}	9.733×10^{-5}
19	-	-	-	-	-	-
20	-	-	-	-	-	-

Table 4.5. Effect of Step Size γ_k and Weighting Factor q_k on
Convergence Rate

$$x_1^0(t_f) = 9.0, \quad x_2^0(t_f) = 0.95, \quad P(t) = 6.0.$$

Step Size γ_k	Weighting Factor q_k	No. of Iterations Required	Time Min.
0.5	$(\frac{k}{100+k})^3$ *	14	11.2
0.5	$(\frac{k}{100+k})^2$	14	11.2
0.5	$(\frac{k}{100+k})$	13	10.4
0.5	$(\frac{k}{50+k})$	13	10.4
0.5	$(\frac{k}{30+k})$	12	9.6
0.5	$(\frac{k}{10+k})$	10	8.0
0.5	$(\frac{k}{1+k})$	7	5.6
0.5	$(\frac{k}{.1+k})$	5	4.0
0.5	$\frac{k}{0+k}$	4	3.2

*_k = Iteration Number

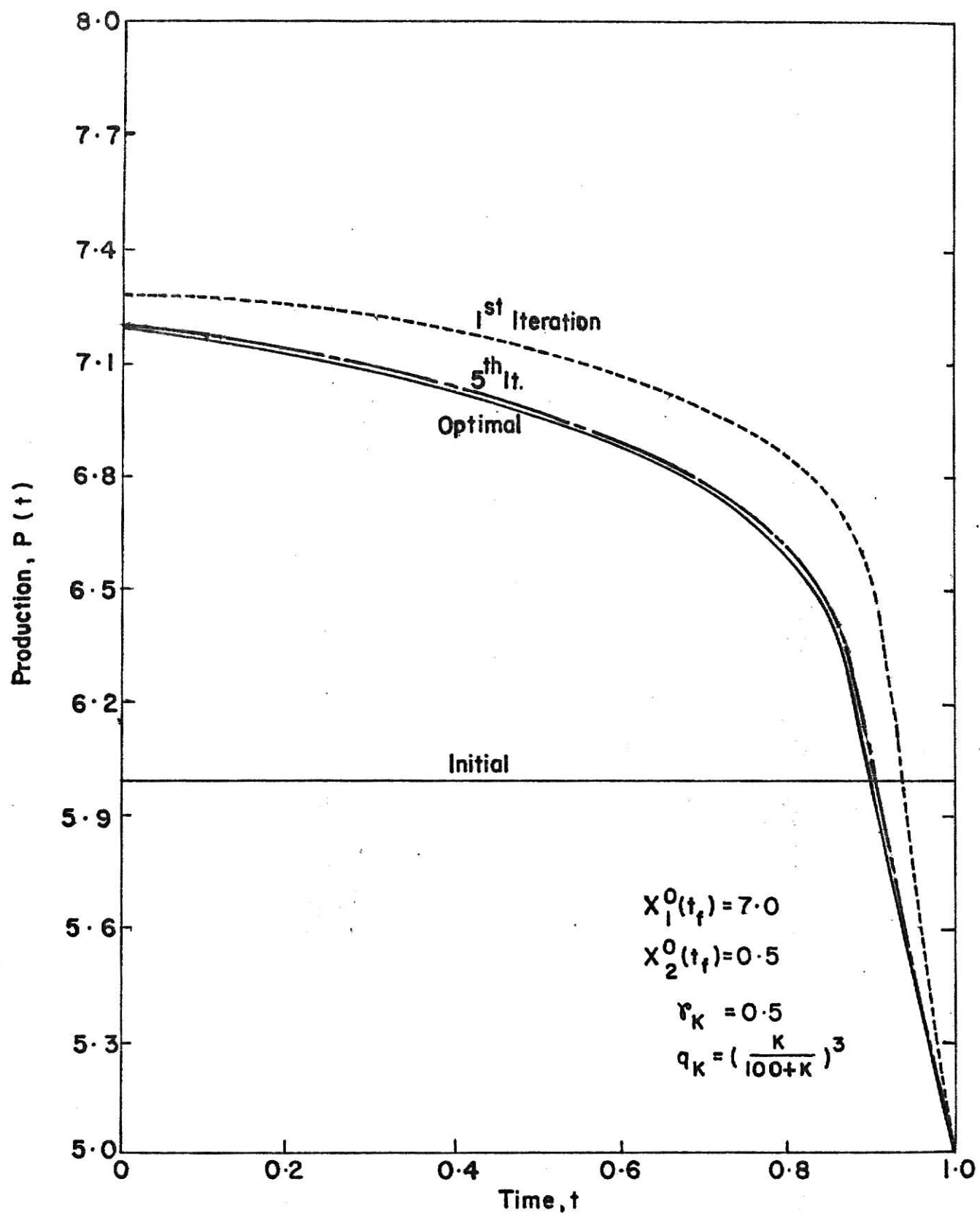


Fig.4.1. Convergence rate of production, $P(t)$, in an inventory model.

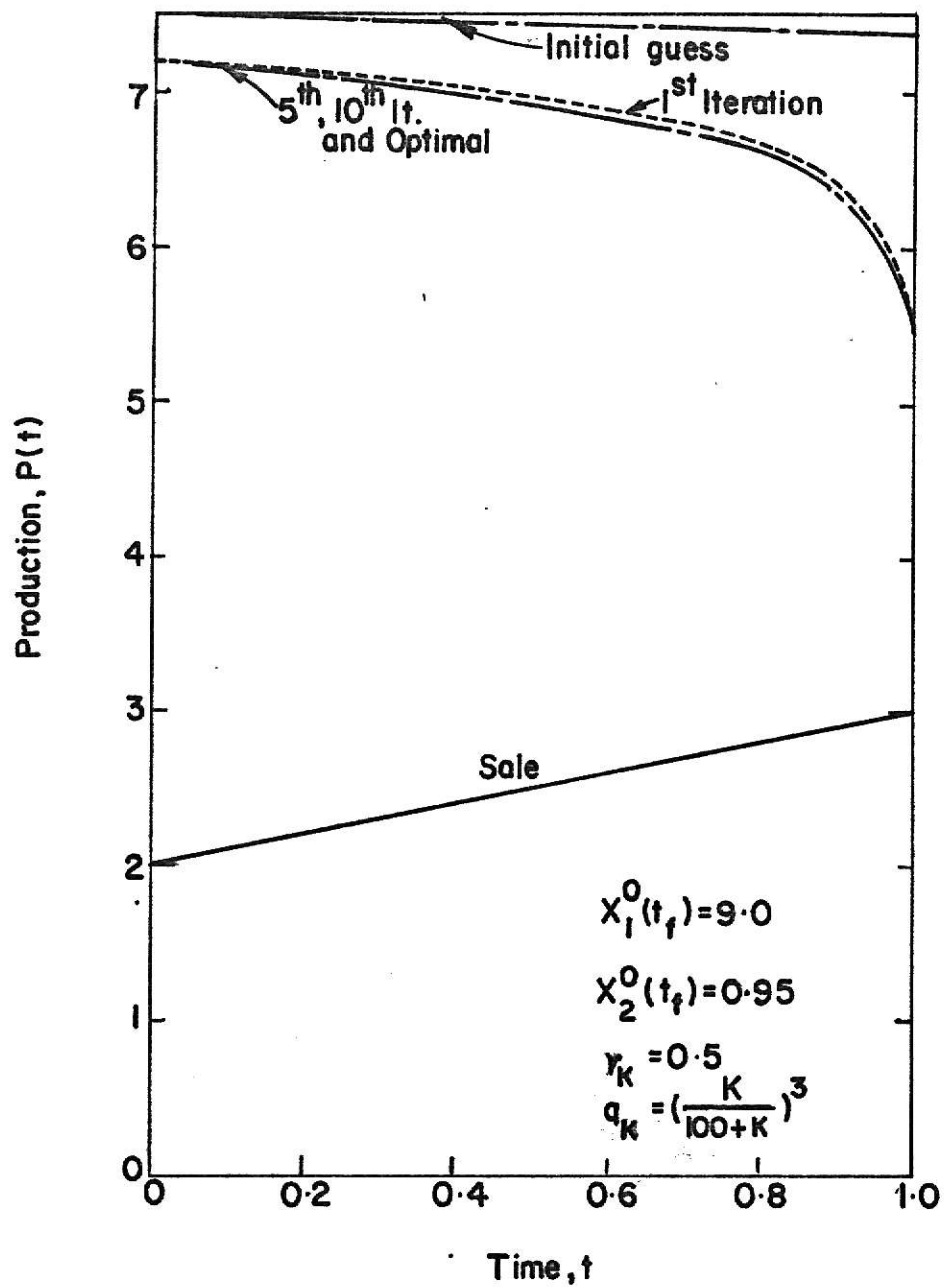


Fig.4.2. Convergence rate of production, $P(t)$ in an inventory model.

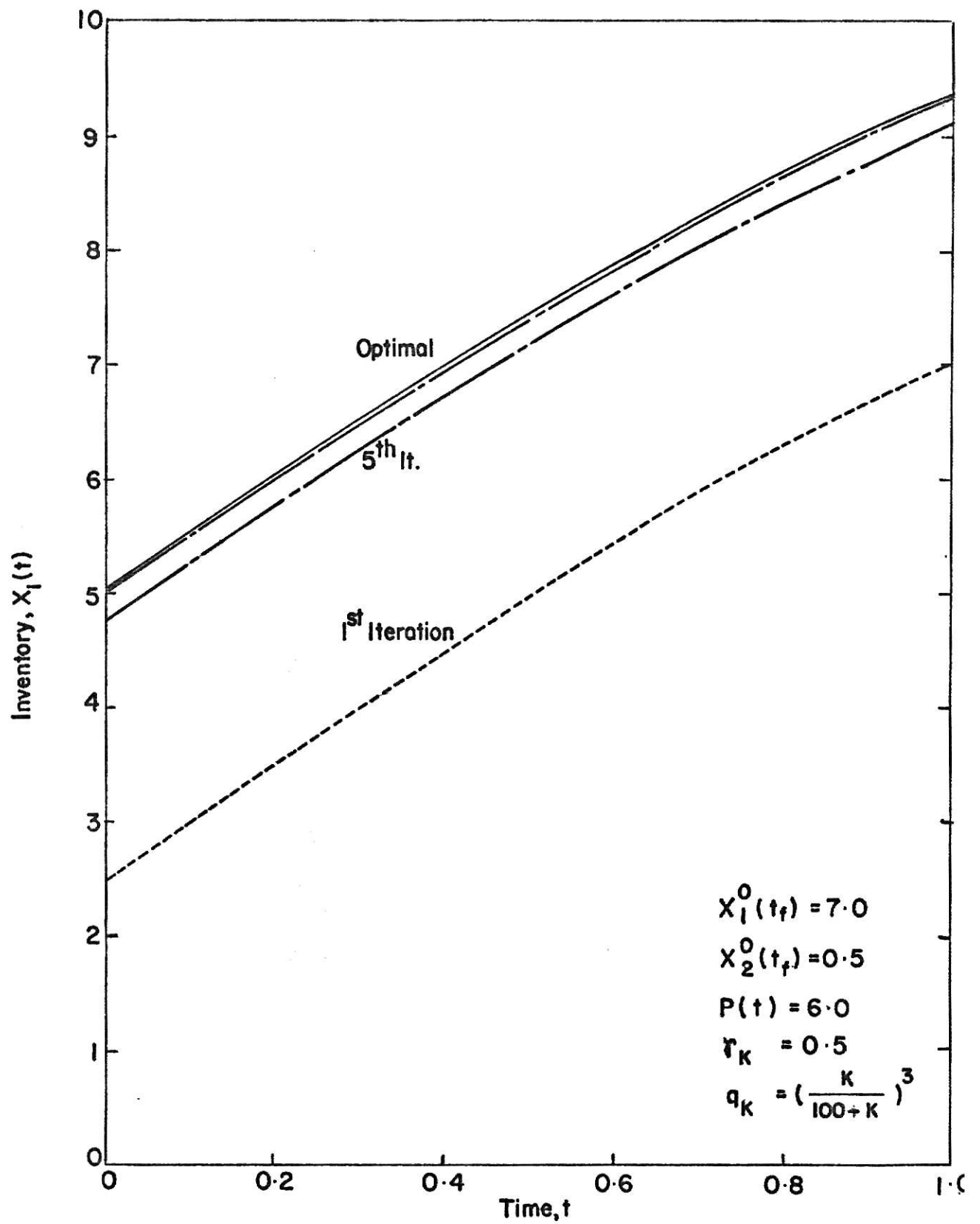


Fig.4.3. Convergence rate of inventory level, $X_1(t)$ in an inventory model.

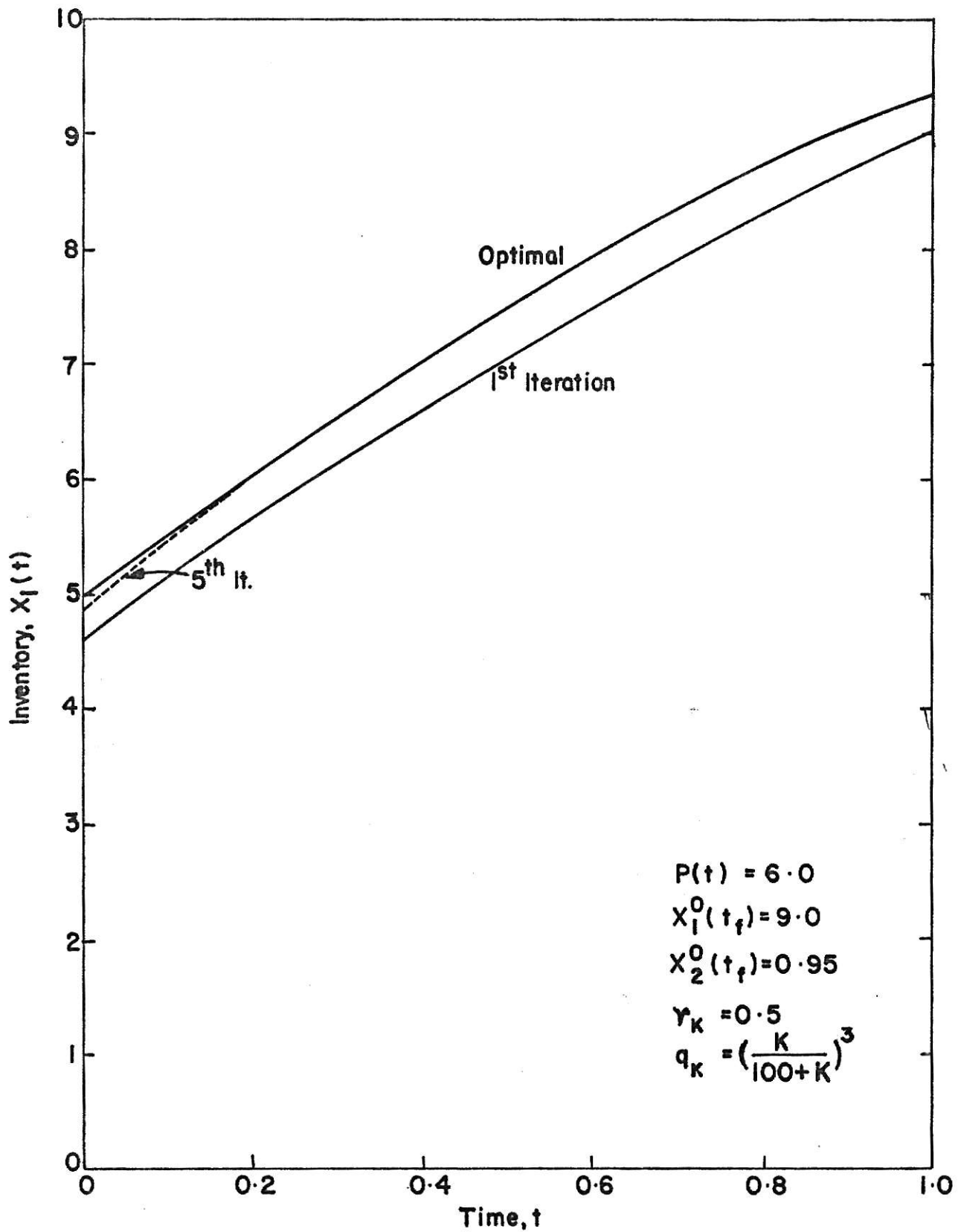


Fig.4.4. Convergence rate of inventory, $X_I(t)$ in an inventory model.

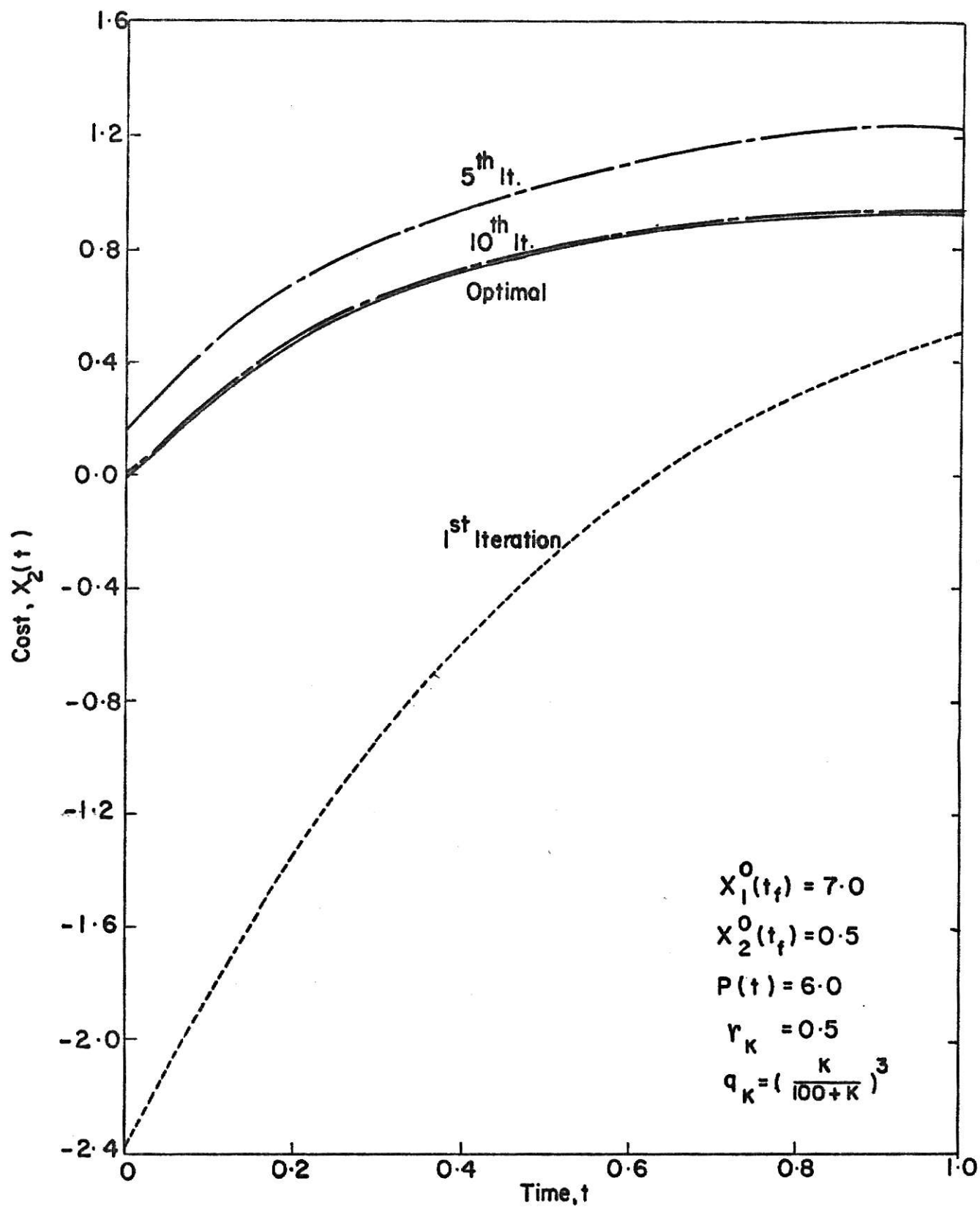


Fig.4.5. Convergence rate of cost, $X_2(t)$.

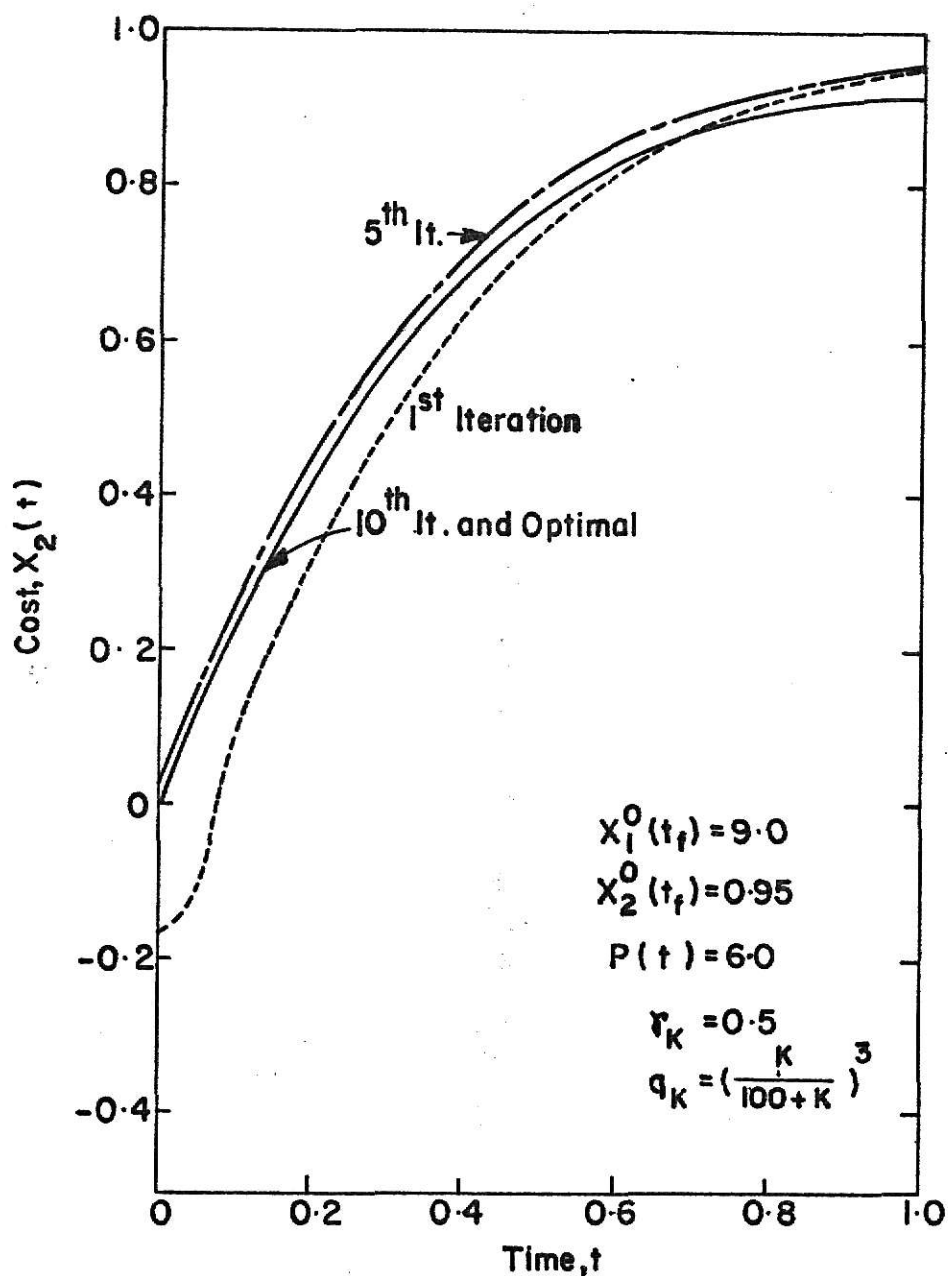


Fig.4.6. Convergence rate of cost in an inventory problem.

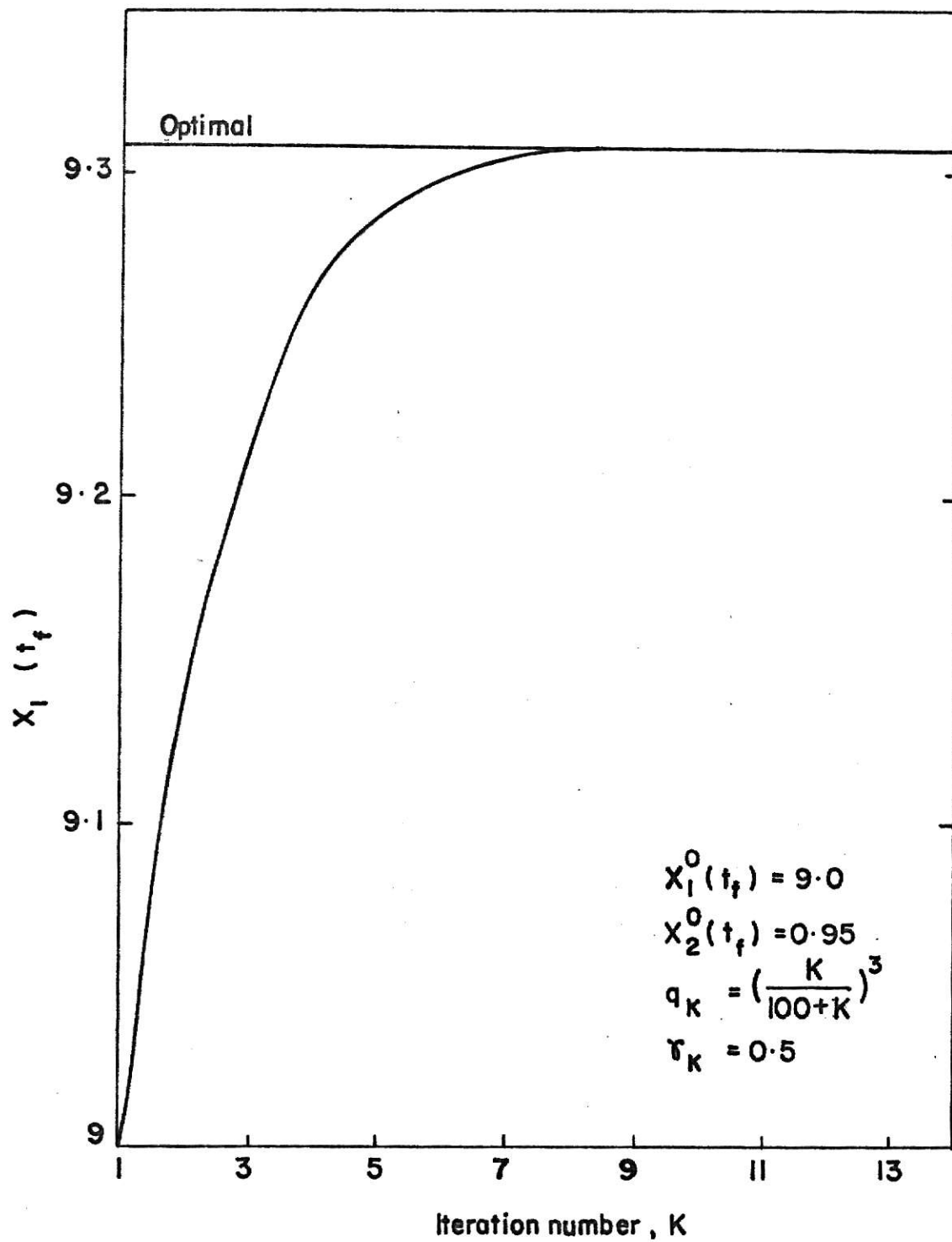


Fig.4.7. Convergence rate of final inventory, $X_1(t_f)$.

CHAPTER 5

APPLICATION TO AN INVENTORY AND ADVERTISEMENT SCHEDULING MODEL

We now wish to apply the technique to a more complex problem, namely an inventory and advertisement model. This model has two state variables and one control variable. As will be seen, the number of differential equations increase rapidly as the number of state variables increase.

DEVELOPMENT OF THE MODEL

This model was originally developed by Teichroew [29]. Consider a group of people in which only certain members possess a particular piece of information, say, about a company's product. Suppose that the total number of such persons remains constant and that the diffusion of information occurs only through personal contact. The number of contacts made by an average informed person in a unit of time is known. In a contact, the contactee receives information if he does not already have it, if he already has it, the contact is wasted in so far as increasing the number of informed persons is concerned.

Let $k(0) = k_0$ = number of informed persons at time t_0

N = total number of persons

C = contact co-efficient; the number of contacts made
by one informed person per unit time

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

then $k(t)/N$ = proportion of informed persons at time t

$1 - \frac{k(t)}{N}$ = proportion of uninformed persons at time t

$C k(t) dt$ = contacts made during a time interval dt .

The increase in the total number of informed people during a short interval of time dt is obtained by multiplying the number of contacts by the proportion of uninformed persons, because an increase in informed members is caused only by contacts with uninformed group. Hence

$$dk(t) = C \cdot k(t) \cdot dt \cdot \left(1 - \frac{k(t)}{N}\right) \quad (1)$$

or

$$\frac{dk(t)}{dt} = C \cdot k(t) \left(1 - \frac{k(t)}{N}\right) \quad (2)$$

Suppose next that the firm can influence the number of contacts by spending money on advertising. In particular, it can increase the number of contacts made by the informed persons by an additional number T per unit of time. Thus,

$$\frac{dk(t)}{dt} = k(t) [C + T(t)] \left[1 - \frac{k(t)}{N}\right] \quad (3)$$

If each informed person buys n units of the company's products and if $X_1(t)$ represents the sales at time t , then

$$X_1(t) = n \cdot k(t) \quad (4)$$

Let $n = 1$, and substituting for $k(t)$ in equation (3) we get

$$\frac{dX_1(t)}{dt} = X_1(t) \cdot (C + T(t)) \cdot \left[1 - \frac{X_1(t)}{N}\right] \quad (5)$$

Next, the rate of change of the firm's inventory $X_2(t)$, is given by

$$\frac{dX_2(t)}{dt} = P(t) - X_1(t) \quad (6)$$

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

The production rate is assumed to be a linear function given by

$$P(t) = a+bt \quad (7)$$

where a, b are known constants and t is time.

This is a typical industrial management problem where the management wishes to maximize the profit given by

$$J(T) = \int_{t_0}^{t_f} [C_s \cdot X_1(t) - C_I(P_I - X_2(t))^2 - C_A \cdot T^2(t)X_1(t)]dt \quad (8)$$

where C_s is the revenue from sale of one unit of the product, C_I is the inventory carrying cost, and P_I can be considered as the capacity for the storage of inventory. In many practical situations, the minimum storage cost is obtained when the storage capacity is completely filled. Furthermore, the cost function, equation (8), has the smoothing capability, which is frequently desirable for many processes. In this case, P_I can be considered as the desirable inventory level. The last term on the right hand side of equation (6) represents the total cost of advertising, where C_A is the cost of advertising.

The initial conditions for equations (5) and (6) are

$$X_1(t_0) = X_1^0 \quad (9)$$

$$X_2(t_0) = X_2^0. \quad (10)$$

The role of management in this particular case is to select the optimal policy from among all feasible solutions which gives the maximum profit.

DEFINITION OF THE PROBLEM

Find the control function $T(t)$ in the interval $t_0 \leq t \leq t_f$ such that the scalar performance index

$$J(T) = \int_{t_0}^{t_f} [C_s X_1(t) - C_I(P_I - X_2(t))^2 - C_A \cdot T^2(t)X_1(t)] dt \quad (11)$$

is a maximum, while the first order, non-linear, differential equations

$$\frac{dX_1(t)}{dt} = X_1(t) \cdot (C + T(t)) \left[1 - \frac{X_1(t)}{N}\right] \quad (12)$$

$$\text{and } \frac{dX_2(t)}{dt} = P(t) - X_1(t) \quad (13)$$

and the initial conditions

$$X_1(t_0) = X_1^0 \quad (14)$$

$$X_2(t_0) = X_2^0 \quad (15)$$

are satisfied. The above problem has two state variables, $X_1(t)$ and $X_2(t)$, and one control variable $T(t)$. Note that, since advertisement rate cannot be negative we have a lower bound on the control function

$$T(t) \geq 0 \quad (16)$$

It has been assumed that the problem is deterministic, that the control variable and its derivatives are continuous in the interval $[t_0, t_f]$.

FORMULATION OF THE PROBLEM

As discussed in Chapter 3, it is convenient to reformulate the problem as a Mayer's problem. To this end define

$$X_3(t) = \int_{t_0}^t [C_s X_1(t) - C_I(P_I - X_2(t))^2 - C_A T^2(t) \cdot X_1(t)] dt. \quad (17)$$

Direct differentiation yields

$$\frac{dX_3(t)}{dt} = C_s X_1(t) - C_I(P_I - X_2(t))^2 - C_A T^2(t) \cdot X_1(t) \quad (18)$$

$$\text{with } X_3(t_f) = J(T) \quad (19)$$

$$\text{and } X_3(t_0) = 0. \quad (20).$$

Thus, with the introduction of the state variable $X_3(t)$, the original problem may be reformulated as:

Maximize $X_3(t_f)$ subject to the constraints of equations (12), (13) and (18) and satisfying the initial conditions given by equations (14), (15) and (20).

To solve this problem, following the classical technique of the Calculus of Variations, introduce the Lagrange multipliers $\lambda_1(t)$, $\lambda_2(t)$, $\lambda_3(t)$, $\mu_1(t)$ and the constant multipliers θ_1 , θ_2 and θ_3 . Define the functions

$$F = \lambda_1 [\dot{X}_1 - X_1(C + T)(1 - \frac{X_1}{N})] + \lambda_2 [\dot{X}_2 - a - bt + X_1] \\ + \lambda_3 [\dot{X}_3 - C_s X_1 + C_I(P_I - X_2)^2 + C_A T^2 X_1] + \mu_1(t) \cdot T(t)$$

$$\text{and } G = X_3(t_f) + \theta_1[X_1(t_0) - X_1^0] + \theta_2[X_2(t_0) - X_2^0] + \theta_3[X_3(t_0)] \quad (21)$$

$$(22)$$

where the notation $(\dot{})$ represents $\frac{d}{dt}$.

In equation (21) $\mu_1 = 0$, when the constraint equation (16) is not violated. However when the constraint equation (16) is violated, the control variable $T(t)$ is determined by the equality in equation (16). and the multiplier $\mu_1(t)$ is determined from the Euler-Lagrange equations.

Applying the Euler-Lagrange equations [15] to equation (21) we get the adjoint system,

$$\frac{d\lambda_1(t)}{dt} = -\lambda_1(C + T)(1 - \frac{2X_1}{N}) + \lambda_2 + \lambda_3 [-C_s + C_A T^2] \quad (23)$$

$$\frac{d\lambda_2(t)}{dt} = -2\lambda_3 C_I (P_I - X_2) \quad (24)$$

$$\frac{d\lambda_3(t)}{dt} = 0 \quad (25)$$

$$\text{and } -\lambda_1 X_1 \left(1 - \frac{X_1}{N}\right) + 2\lambda_3 C_A X_1 T + \mu_1 = 0. \quad (26)$$

However, since the multiplier μ_1 does not enter into the equations (23), (24) and (25), the numerical solution procedure will be the same as for the unconstrained case, except that whenever the constraint is violated, the control variable will be obtained from the equality in equation (16). Therefore, the multiplier μ_1 can be dropped from equation (26), which then reduces to

$$T(t) = \frac{\lambda_1}{2\lambda_3 C_A} \left(1 - \frac{X_1}{N}\right). \quad (27)$$

The required boundary conditions for the Lagrange multipliers are obtained by applying the transversality conditions [15]

$$\left. \frac{\partial G}{\partial X_1} \right|_{t=t_f} + \left. \frac{\partial F}{\partial \dot{X}_1} \right|_{t=t_f} = 0 \quad (28)$$

to equations (21) and (22). The boundary conditions are

$$\lambda_1(t_f) = 0 \quad (29)$$

$$\lambda_2(t_f) = 0 \quad (30)$$

$$\lambda_3(t_f) = -1. \quad (31)$$

The system is now composed of 6 differential conditions with three initial and three final conditions which make the problem a two point

boundary value type. In addition we have equation (27) for determining the control variable $T(t)$. Notice that the control variable can be eliminated from equations (12), (18) and (23) by using equation (27).

BOUNDARY CONDITION ITERATION SCHEME

In order to use the iteration scheme given by equation (24) of Chapter 3, we need the sensitivity co-efficients $\frac{\partial \bar{X}(t)}{\partial \bar{X}(t_f)}$ and $\frac{\partial \bar{\lambda}(t)}{\partial \bar{X}(t_f)}$ where $\bar{X}(t)$ and $\bar{\lambda}(t)$ are (3×1) vectors. To obtain these sensitivity co-efficients, the following eighteen differential equations are required

$$\frac{d}{dt} \left(\frac{\partial \bar{X}(t)}{\partial \bar{X}(t_f)} \right) = A(t) \frac{\partial \bar{X}(t)}{\partial \bar{X}(t_f)} + B(t) \frac{\partial \bar{\lambda}(t)}{\partial \bar{X}(t_f)} \quad (32)$$

$$\frac{d}{dt} \left(\frac{\partial \bar{\lambda}(t)}{\partial \bar{X}(t_f)} \right) = C(t) \frac{\partial \bar{X}(t)}{\partial \bar{X}(t_f)} + D(t) \frac{\partial \bar{\lambda}(t)}{\partial \bar{X}(t_f)} \quad (33)$$

with the boundary conditions

$$\left. \frac{\partial \bar{X}(t)}{\partial \bar{X}(t_f)} \right|_{t=t_f} = I, \text{ where } I \text{ is a } 3 \times 3 \text{ Identity matrix} \quad (34)$$

$$\text{and } \left. \frac{\partial \bar{\lambda}(t)}{\partial \bar{X}(t_f)} \right|_{t=t_f} = 0. \quad (35)$$

For the particular problem, the matrices $A(t)$, $B(t)$, $C(t)$ and $D(t)$ can be easily derived to be

$$A(t) = \begin{pmatrix} (C+T) \left(1 - \frac{2X_1}{N}\right) - \frac{X_1}{H_{T,T}} \left(1 - \frac{X_1}{N}\right) [\lambda_1 \left(1 - \frac{2X_1}{N}\right) - 2\lambda_3 C_A T] & 0 & 0 \\ -1 & 0 & 0 \\ C_s - C_A T^2 + \frac{2C_A T X_1}{H_{T,T}} [\lambda_1 \left(1 - \frac{2X_1}{N}\right) - 2\lambda_3 C_A T] & 2C_I (P_I - X_2) & 0 \end{pmatrix} \quad (36)$$

$$B(t) = \frac{-1}{H_{T,T}} \begin{pmatrix} X_1^2 \left(1 - \frac{X_1}{N}\right)^2 & 0 & -2C_A T X_1^2 \left(1 - \frac{X_1}{N}\right) \\ 0 & 0 & 0 \\ -2C_A T X_1^2 \left(1 - \frac{X_1}{N}\right) & 0 & 4C_A^2 T^2 X_1^2 \end{pmatrix} \quad (37)$$

$$C(t) = \begin{pmatrix} \frac{2\lambda_1}{N} (C+T) + \frac{1}{H_{T,T}} [\lambda_1 \left(1 - \frac{2X_1}{N}\right) - 2\lambda_3 C_A T]^2 & 0 & 0 \\ 0 & 2\lambda_3 C_I & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (38)$$

$$\text{and } D(t) = - [A(t)]^T \quad (39)$$

$$\text{where } H_{T,T} = -2X_1 \lambda_3 C_A; \quad (40)$$

Thus, for the differential sensitivity measures we have nine differential equations given by equation (32), nine differential equations given by equation (33) and eighteen boundary conditions given by equations (34) and (35). Since the boundary conditions are given at the final time only, the system forms a final value type problem and can be easily solved.

Since iterations are carried out on final values of the state variables, the vector $\bar{\Pi}$ is defined as

$$\bar{\Pi} = \begin{pmatrix} X_1(t_f) \\ X_2(t_f) \\ X_3(t_f) \end{pmatrix} \quad (41)$$

and in the terminology of Chapter 3, the residual norm vector to be driven to zero is

$$\bar{R} = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \end{pmatrix} \quad (42)$$

$$= \begin{pmatrix} X_1(t_0) - X_1^0 \\ X_2(t_0) - X_2^0 \\ X_3(t_0) \end{pmatrix} \quad (43)$$

The elements of the constrained Jacobian $\frac{d\bar{R}}{d\bar{\Pi}}$ are given by

$$\Gamma(\bar{\pi}) = \begin{pmatrix} \frac{\partial X_1(t_0)}{\partial X_1(t_f)} & \frac{\partial X_1(t_0)}{\partial X_2(t_f)} & \frac{\partial X_1(t_0)}{\partial X_3(t_f)} \\ \frac{\partial X_2(t_0)}{\partial X_1(t_f)} & \frac{\partial X_2(t_0)}{\partial X_2(t_f)} & \frac{\partial X_2(t_0)}{\partial X_3(t_f)} \\ \frac{\partial X_3(t_0)}{\partial X_1(t_f)} & \frac{\partial X_3(t_0)}{\partial X_2(t_f)} & \frac{\partial X_3(t_0)}{\partial X_3(t_f)} \end{pmatrix} \quad (44)$$

The elements of this matrix are obtained by integrating the sensitivity differential equations backwards. All the necessary information for the iteration scheme

$$\bar{\pi}^{k+1} = \bar{\pi}^k - [(1-q_k) \gamma_k' \Gamma(\bar{\pi}^k)^T + q_k \Gamma(\bar{\pi}^k)^{-1}] \bar{R}(\bar{\pi}^k)$$

is now available.

NUMERICAL ASPECTS

In order to solve this problem numerically, the constants are assumed to have the following values:

$a = 70.0$	$C_I = 0.15$
$b = 100.0$	$P_I = 50.0$
$c = 2.0$	$C_A = 1.5$
$N = 150.0$	$X_1^0 = 20.0$
$C_s = 10.0$	$X_2^0 = 20.0$
$t_0 = 0$	$t_f = 1.$

The problem was solved on an IBM 360-50 computer. The Runge Kutta integration formulae were used to integrate the differential equations. The step size used for the numerical integration was 0.01.

To obtain the solutions, the initial approximations given in Table 5.1 were used.

Out of the seven initial sets of approximations tried, only five converged to the optimum. In other two cases overflow occurred even before the first iteration.

The convergence rates of final sales, final inventory, and total cost are given in Tables 5.2, 5.3, and 5.4. The convergence rates of the trajectories for sales, inventory level, cost and advertisement are shown in Figs. (5.1), (5.2), (5.3) and (5.4) respectively. The following initial approximations, step size and weighting factor are used in obtaining these figures:

$$X_1(t_f) = 150$$

$$X_2(t_f) = 90$$

$$X_3(t_f) = 650$$

$$q_k = \frac{k}{10+k}$$

$$\gamma_k = 0.1$$

The effect of step size and weighting factor on convergence rate measured in terms of number of iterations required was also studied. The results are shown in Table 5.6.

RESULTS

The optimal profit in this problem was $J = 584.2151$ and the optimal initial and final values are

Table 5.1. Initial Approximations Used for Advertisement Model

Set No.	$x_1^0(t_f)$	$x_2^0(t_f)$	$x_3^0(t_f)$
1	120.0	75.0	600.0
2	150.0	90.0	650.0
3	100.0	50.0	500.0
4	80.0	40.0	450.0
5	75.0	125.0	300.0
6	45.0	45.0	600.0
7	200.0	55.0	400.0
Optimum	115.6907	66.1579	584.2151

Table 5.2. Convergence Rate of Sales, $X_1(t_f)$

	$q_k = \left[\frac{k}{10+k}\right]$		$\gamma_k = 0.1$				
$x_1^0(t_f)$	120.0	150.0	100.0	80.0	75.0	45.0	200.0
$x_2^0(t_f)$	75.0	90.0	50.0	40.0	125.0	45.0	55.0
$x_3^0(t_f)$	600.0	650.0	500.0	450.0	300.0	600.0	400.0
Iteration							
1	120.0	150.0	100.0	80.0		45.0	
2	119.784	147.997	101.254	82.595		48.671	
3	119.270	144.942	104.003	90.374		113.799	
4	118.591	141.347	107.204	98.862		115.419	
5	117.876	137.446	110.041	105.430		116.486	
6	117.226	133.357	112.186	109.820		116.996	
7	116.699	129.266	113.644	112.521		117.070	
8	116.311	125.453	114.560	114.078		116.880	
9	116.050	122.202	115.099	114.921		116.583	
10	115.886	119.698	115.396	115.349	OVERFLOW	116.291	OVERFLOW
11	115.791	117.965	115.552	115.551		116.059	
12	115.738	116.886	115.628	115.639		115.899	
13	115.718	116.278	115.664	115.674		115.801	
14	115.703	115.962	115.680	115.686		115.745	
15	115.696	115.810	115.686	115.690		115.761	
16	115.693	115.741	115.689	115.6909		115.702	
17	115.6915	115.711	115.690	115.6910		115.696	
18	115.6911	115.699	115.6906	115.6909		115.693	
19	115.6909	115.694	115.6907	115.6909		115.692	
20	115.6908	115.692	115.6907	115.6908		115.691	
21	115.6907	115.691	115.6907	115.6908		115.6909	
22	115.6907	115.6909	115.6907	115.6907		115.6908	
23	—	115.6908	—	115.6907		115.6907	
24	—	115.6907	—	—		115.6907	
25	—	115.6907	—	—		115.6907	

Table 5.3. Convergence Rate of Final Inventory, $X_2(t_f)$

	$q_k = \frac{k}{10+k}, \quad \gamma_k = 0.1$						
$x_1^0(t_f)$	120.0	150.0	100.0	80.0	75.0	45.0	200.0
$x_2^0(t_f)$	75.0	90.0	50.0	40.0	125.0	45.0	55.0
$x_3^0(t_f)$	600.0	650.0	500.0	450.0	300.0	600.0	400.0
Iteration							
1	75.0	90.0	50.0	40.0		45.00	
2	75.049	91.652	50.384	41.865		50.228	
3	74.249	88.672	51.862	44.217		29.970	
4	72.885	82.538	54.128	47.411		36.450	
5	71.310	75.721	56.780	51.370		43.546	
6	69.818	70.159	59.381	55.409		50.095	
7	68.585	66.638	61.602	58.916		55.450	
8	67.667	65.000	63.295	61.605		59.431	
9	67.041	64.631	64.469	63.471		62.166	
10	66.647	64.883	65.218	64.661	OVERFLOW	63.916	OVERFLOW
11	66.414	65.301	65.663	65.368		64.964	
12	66.285	65.659	65.910	65.762		65.554	
13	66.215	65.895	66.040	65.969		65.867	
14	66.184	66.030	66.104	66.072		66.024	
15	66.169	66.099	66.135	66.120		66.099	
16	66.163	66.132	66.148	66.142		66.133	
17	66.160	66.147	66.154	66.152		66.148	
18	66.159	66.151	66.156	66.155		66.154	
19	66.158	66.155	66.157	66.157		66.156	
20	66.158	66.157	66.1577	66.1576		66.157	
21	66.1579	66.157	66.1578	66.1578		66.1578	
22	66.1579	66.158	66.1579	66.1579		66.1579	
23	—	66.158	—	66.1578		66.1579	
24	—	66.1579	—	—		66.1580	
25	—	66.1580	—	—		66.1580	

Table 5.4. Convergence Rate of Total Profit, $X_3(t_f)$

$$q_k = \frac{k}{10+k}, \gamma_k = 0.1$$

$x_1^0(t_f)$	120.0	150.0	100.0	80.0	75.0	45.0	200.0
$x_2^0(t_f)$	75.0	90.0	50.0	40.0	125.0	45.0	55.0
$x_3^0(t_f)$	600.0	650.0	500.0	450.0	300.0	600.0	400.0

Iteration

1	600.0	650.0	500.0	450.0		600.0	
2	600.010	650.120	499.935	448.845		599.664	
3	600.920	724.571	518.363	449.274		819.035	
4	601.590	818.728	543.937	560.231		817.896	
5	600.841	875.927	565.793	602.818		798.168	
6	598.464	876.115	579.331	620.404		760.906	
7	595.141	832.969	585.398	620.090		715.886	
8	591.783	771.382	586.946	611.761		673.510	
9	589.009	711.242	586.572	602.341		639.890	
10	587.025	662.940	585.757	594.905	OVERFLOW	616.431	OVERFLOW
11	585.755	629.146	585.081	590.012		601.659	
12	585.010	608.023	584.653	587.151		593.115	
13	584.604	596.032	584.420	585.619		588.519	
14	584.397	589.756	584.305	584.854		586.198	
15	584.297	586.686	584.253	584.493		585.088	
16	584.250	585.269	584.230	584.331		584.584	
17	584.230	584.646	584.221	584.262		584.364	
18	584.221	584.384	584.218	584.234		584.273	
19	584.218	584.280	584.216	584.222		584.237	
20	584.216	584.239	584.2158	584.2183		584.224	
21	584.2153	584.224	584.2153	584.2161		584.218	
22	584.2151	584.219	584.2148	584.2151		584.216	
23	—	584.216	—	584.2148		584.215	
24	—	584.2153	—	—		584.2151	
25	—	584.2151	—	—		584.2151	

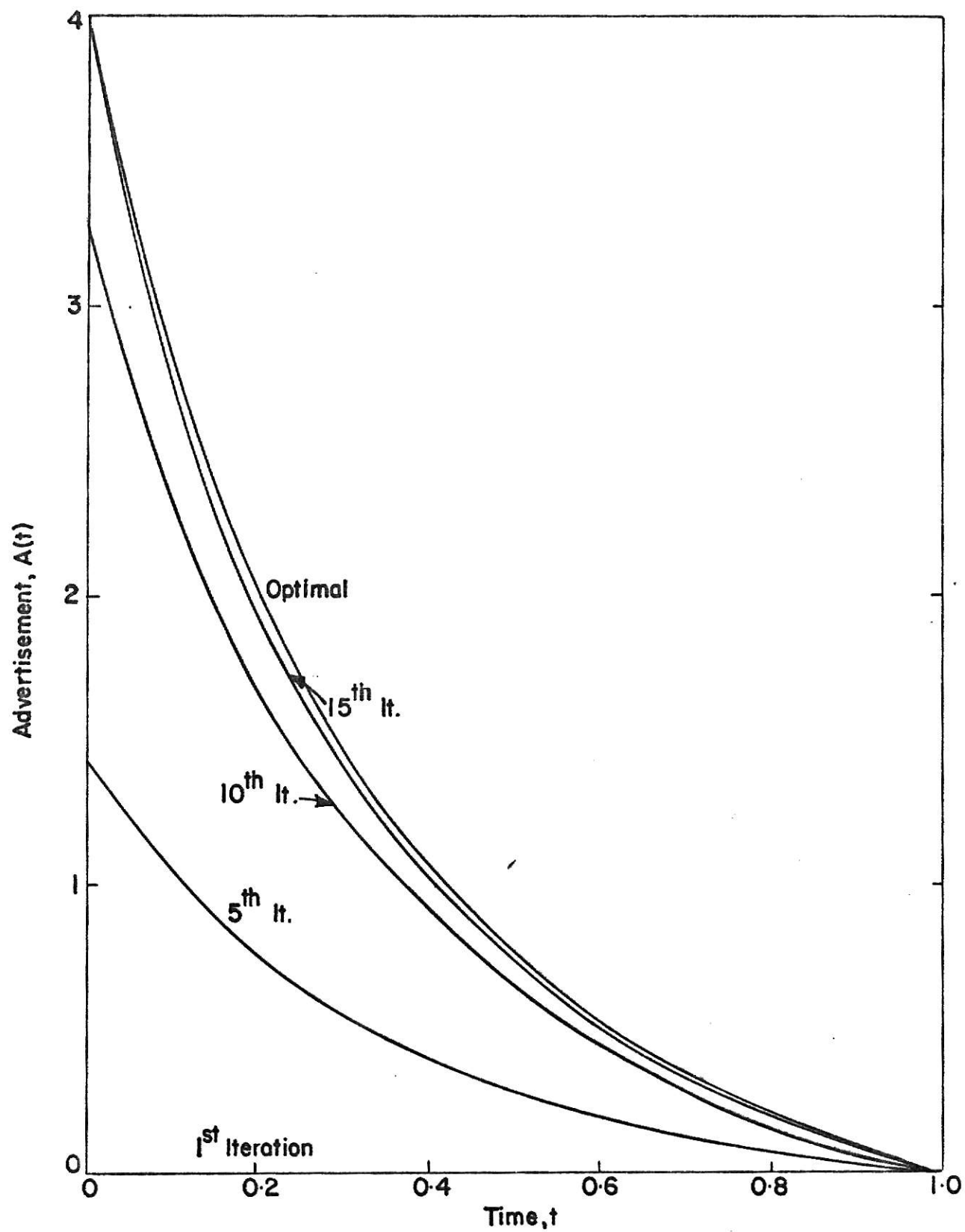


Fig.5.1. Convergence rate of advertisement, $A(t)$.

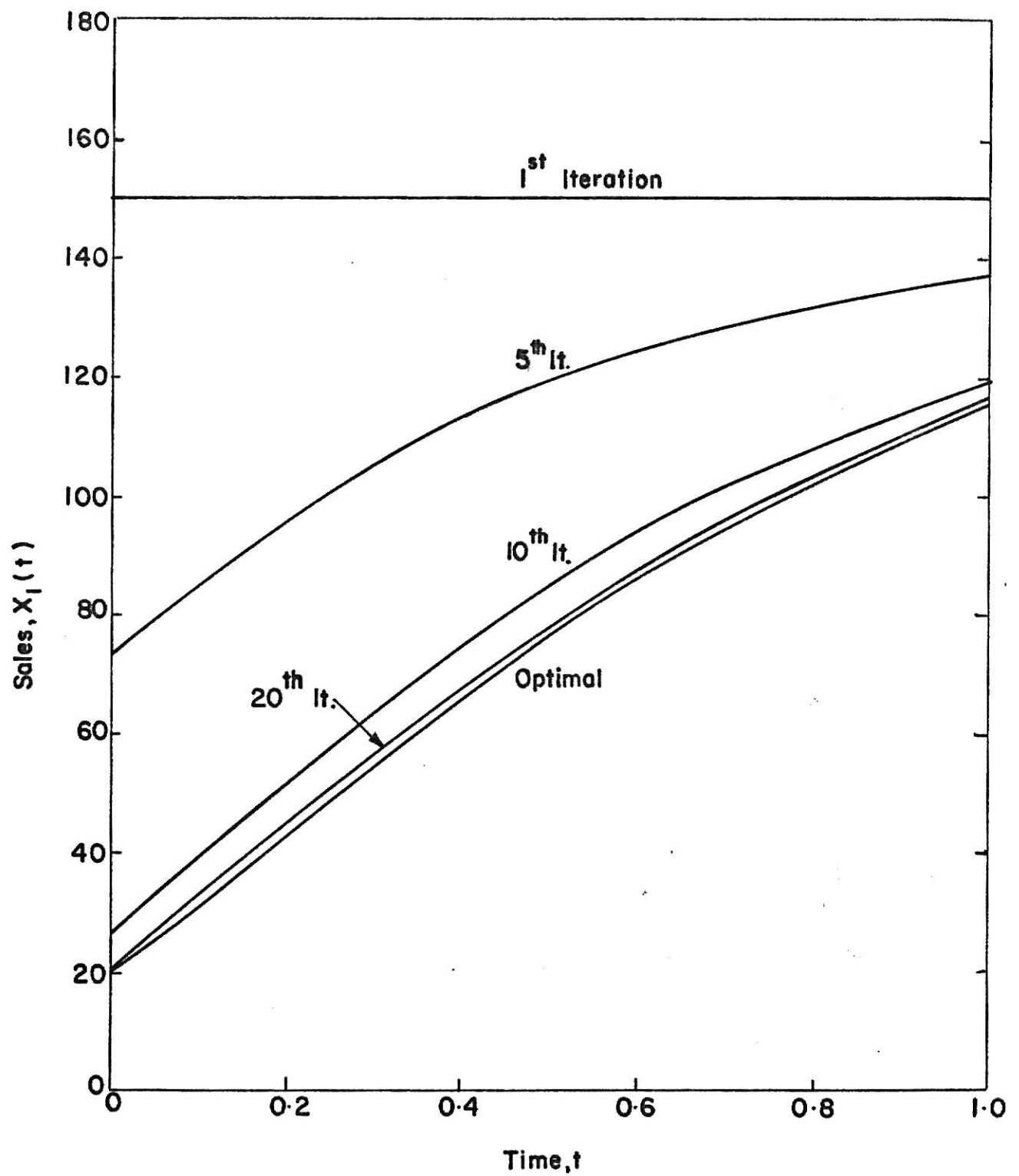


Fig.5.2. Convergence rate of sales, $X_1(t)$ in advertisement model.

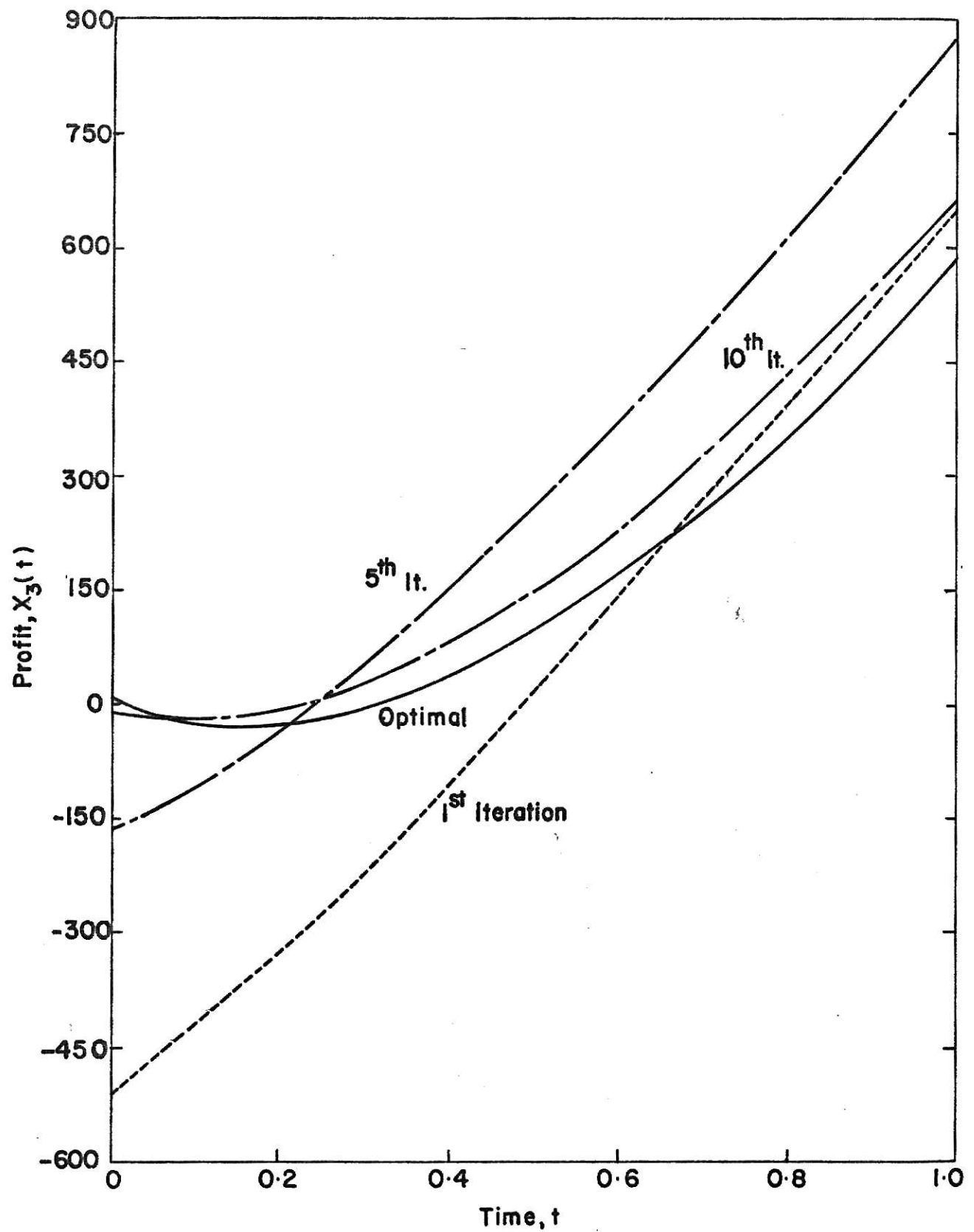


Fig.5.3 Convergence rate of profit, $X_3(t)$, in advertisement model.

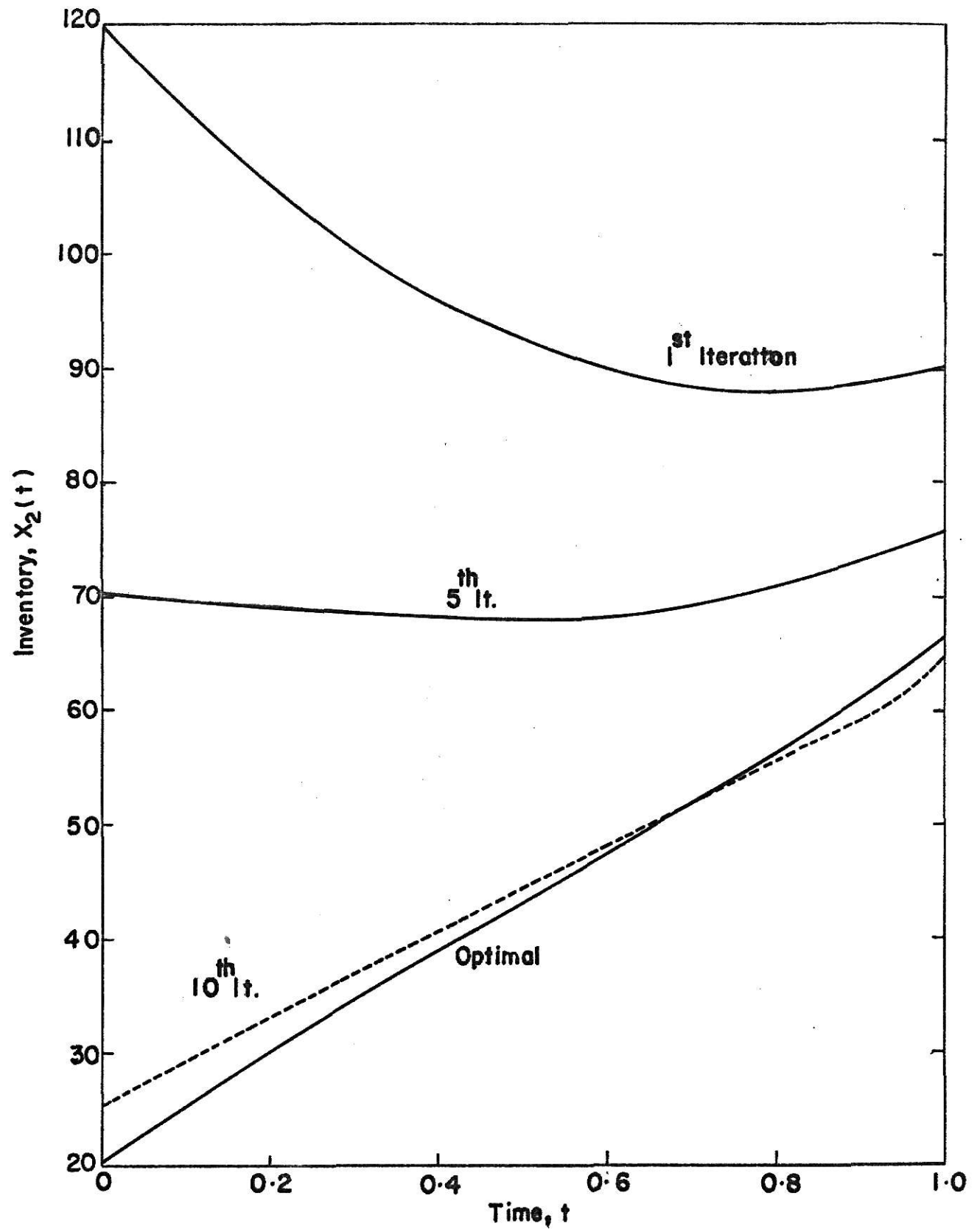


Fig.5.4. Convergence rate of inventory, $X_2(t)$, in advertisement model.

Table 5.5. Convergence Rate of Residuals ψ_1 , ψ_2 , ψ_3 and $||R||$

$$q_k = \left(\frac{k}{10+k}\right), \gamma_k = 0.1$$

Iteration	ψ_1	ψ_2	ψ_3	$ R $	ψ_1	ψ_2	ψ_3	$ R $
$x_1^0(t_f)$	150.0	80.0						
$x_2^0(t_f)$	90.0	40.0						
$x_3^0(t_f)$	650.0	450.0						
1	130.0	99.99	-510.01	535.73	-15.85	-61.51	653.93	657.01
3	94.31	83.12	-363.78	384.89	-13.30	-48.99	479.37	482.05
5	53.93	50.18	-165.60	181.25	-6.49	-27.00	248.67	250.22
7	26.34	23.43	-55.70	65.92	-1.85	-11.14	99.62	100.26
9	10.56	8.68	-16.82	21.68	-2.594x10 ⁻¹	-3.59	30.94	31.15
11	3.34	2.57	-4.33	6.04	2.875x10 ⁻²	-9.359x10 ⁻¹	7.49	7.55
13	8.33x10 ⁻¹	6.08x10 ⁻¹	-8.07x10 ⁻²	1.31	2.766x10 ⁻²	-2.005x10 ⁻¹	1.45	1.46
15	1.68x10 ⁻¹	1.18x10 ⁻¹	-9.33x10 ⁻³	2.26x10 ⁻²	8.865x10 ⁻³	-3.068x10 ⁻²	2.31x10 ⁻¹	2.343x10 ⁻¹
17	2.82x10 ⁻²	1.86x10 ⁻²	-3.31x10 ⁻³	3.39x10 ⁻³	1.953x10 ⁻³	-5.585x10 ⁻³	3.04x10 ⁻²	3.10x10 ⁻²
19	4.67x10 ⁻⁴	2.21x10 ⁻⁴	-5.52x10 ⁻⁴	7.56x10 ⁻⁴	3.20x10 ⁻⁴	-7.02x10 ⁻⁴	3.80x10 ⁻³	3.88x10 ⁻³
21	6.41x10 ⁻⁵	2.59x10 ⁻⁵	-6.50x10 ⁻⁴	9.49x10 ⁻⁴	9.2x10 ⁻⁵	-1.22x10 ⁻⁴	1.51x10 ⁻⁴	2.14x10 ⁻⁴
23	7.6x10 ⁻⁵	9.2x10 ⁻⁵	-4.64x10 ⁻⁴	4.79x10 ⁻⁴	-	-	-	-
25	-1.5x10 ⁻⁵	4.6x10 ⁻⁵	1.33x10 ⁻⁴	1.41x10 ⁻⁴	-	-	-	-

Table 5.6. Effect of Step Size γ_k and Weighting Factor q_k on Convergence Rate

$$x_1^0(t_f) = 100.0 \quad x_2^0(t_f) = 50.0 \quad x_3^0(t_f) = 500.0$$

Step Size γ_k	Weighting Factor q_k	No. of Iterations Required	Time Min.
0.1	$\left(\frac{k}{100+k}\right)^2$ *	Did not converge. After 64 iterations $ R = 0.09588$	12.48
0.1	$\frac{k}{100+k}$	56	10.91
0.1	$\frac{k}{50+k}$	42	8.16
0.1	$\frac{k}{30+k}$	32	6.24
0.1	$\frac{k}{10+k}$	22	4.28
0.1	$\frac{k}{5+k}$	17	3.31
0.1	$\frac{k}{1+k}$	11	2.14
0.1	$\frac{k}{.1+k}$	8	1.56
0.1	$\frac{k}{0+k}$	7	1.37
0.2	$\frac{k}{.1+k}$	8	1.56

* k = iteration no.

$$x_1(0) = 20.0 \qquad x_1(1) = 115.6907$$

$$x_2(0) = 20.0 \qquad x_2(1) = 66.1579$$

$$x_3(0) = 0.0 \qquad x_3(1) = 584.2151$$

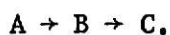
CHAPTER 6

APPLICATION TO A PRODUCTION AND ADVERTISEMENT SCHEDULING MODEL

To study the effectiveness of the computational procedure in obtaining numerical solutions for complex dynamic problems, a production and advertisement problem with four state variables and two control variables is solved in this chapter. To illustrate the versatility of the procedure, Lagrange formulation of the variational problem has been used for the derivation of the necessary equations.

DEVELOPMENT OF THE MODEL

The system under consideration is shown in Fig. (6.1). A manufacturing firm has decided to produce and market a new product B. Since the product is new, advertising is planned to boost its sales. Furthermore, to protect against fluctuations in demand, it is planned to maintain an inventory for the product. Product B is produced in the chemical reactor in which the following first order reaction takes place



During this reaction, a certain amount of B also decomposes into C. B is the most valuable product and C is the least valuable. Since C is less valuable, the decomposition of B into C is undesirable. It is assumed that the raw material containing A, B and C is available in unlimited quantity and that both A and C have an unlimited market at a fixed price and they can be sold as soon as manufactured. Therefore, inventory for A and C is not considered. The amount of B produced and decomposed can be controlled by the reaction temperature T.

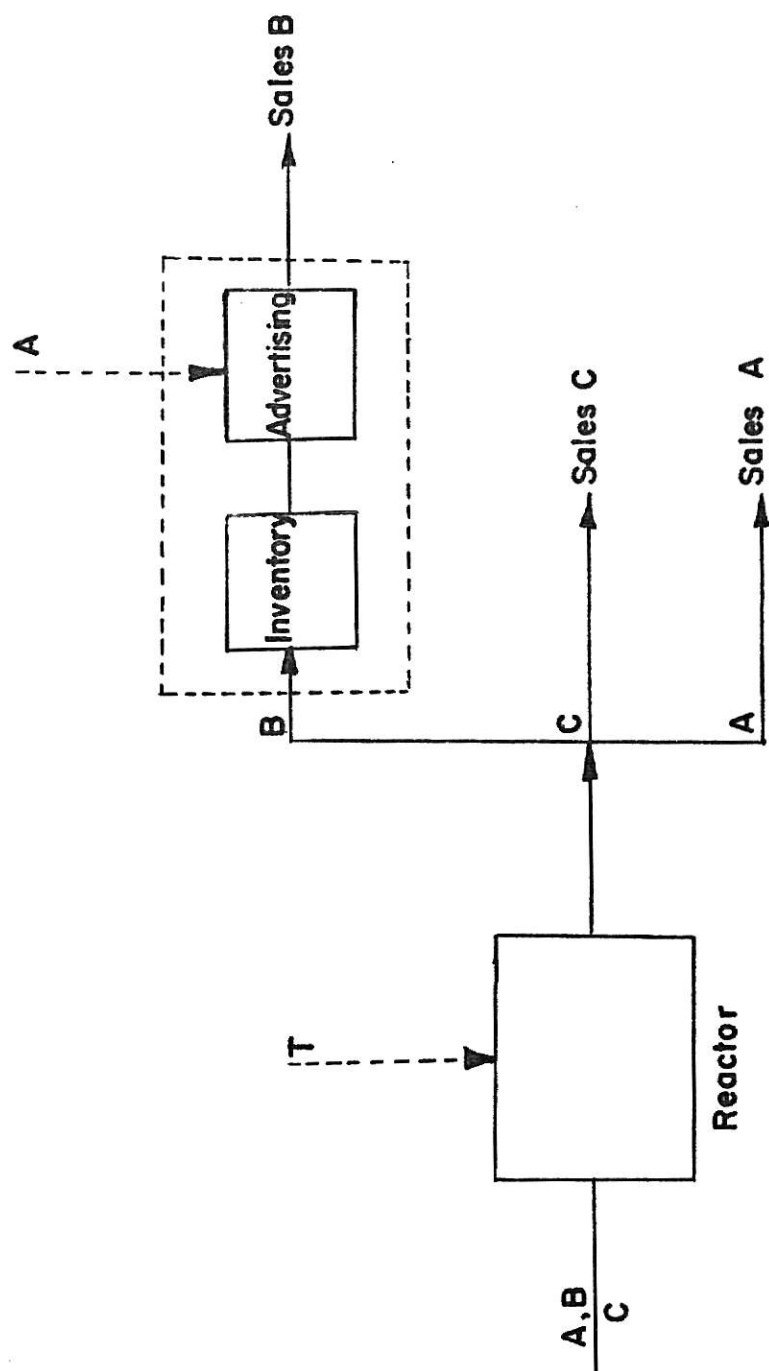


Fig.6.1. A production and advertising model.

Let

V = volume of chemical reactor

q = flow rate

K_a = reaction rate constant of A

K_b = reaction rate constant of B

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

E_a, E_b = activation energies of reactions

R = gas constant

T = temperature in reactor

X_1 = concentration of A in reactor

X_2 = concentration of B in reactor

X_1^0 = raw material concentration of A

X_2^0 = raw material concentration of B.

The production rates in the reactor can be represented by

$$V \frac{dX_1}{dt} = q(X_1^0 - X_1(t)) - VK_a X_1(t) \quad (1)$$

$$V \frac{dX_2}{dt} = q(X_2^0 - X_2(t)) - VK_b X_2(t) + VK_a X_1(t). \quad (2)$$

The reaction rate constants are defined as

$$K_a = G_a \exp \left(- \frac{E_a}{RT} \right) \quad (3)$$

and $K_b = G_b \exp \left(- \frac{E_b}{RT} \right).$

As indicated before, an inventory is desired for the product B and the performance equation for inventory is

Rate of change in = total production - total sales
inventory

or

$$\frac{dX_3(t)}{dt} = qX_2(t) - X_4(t) \quad (4)$$

where $X_3(t)$ is the inventory and $X_4(t)$ is the sales of product B.

The sales rate for product B is again given by the diffusion model of Chapter 5, equation (5). Hence,

$$\frac{dX_4(t)}{dt} = X_4(t) [C_c + A(t)] \left[1 - \frac{X_4(t)}{N}\right] \quad (5)$$

where C_c is the contact co-efficient, $A(t)$ is the advertisement rate and N represents the total number of people in the group. Equations (1) through (5) represent the system. We have four state variables, $X_1(t)$, $X_2(t)$, $X_3(t)$ and $X_4(t)$, and two control variables $T(t)$ and $A(t)$.

The management wishes to select the control variables, such that, considering the costs of advertising and storage of B and the relative sales values of A, B and C, the profit, given by the following function, is maximized.

$$\text{Total Profit} = \int_{t_0}^{t_f} \left[\begin{array}{l} \text{Total revenue of B} + \text{Total revenue of A} + \text{Total revenue of C} \\ - \text{inventory cost} - \text{advertisement cost} - \text{manufacturing cost} \end{array} \right] dt.$$

Mathematically, the objective function can be stated as

$$J = \int_{t_0}^{t_f} [C_1 X_4 + C_2 q X_1 + C_3 q (1 - X_1 - X_2) - C_I (I_m - X_3)^2 - C_u A^2 X_4^2 - C_T (T_m - T)^2] dt \quad (6)$$

where C_1 , C_2 , and C_3 are the revenues from sales of an unit of the

products B, A, and C, respectively. The fourth and fifth terms represent the inventory and advertising costs for product B. The last term represents the manufacturing cost. I_m can be considered as the desirable inventory level and C_I and C_u are the inventory and advertising costs.

DEFINITION OF THE PROBLEM

The optimum control problem is to determine the trajectories $X_1(t)$, $X_2(t)$, $X_3(t)$, and $X_4(t)$, $t_0 \leq t \leq t_f$, and the control functions $A(t)$ and $T(t)$ in such a way that the performance functional

$$J[\bar{X}(t), A(t), T(t)] = \int_{t_0}^{t_f} [C_1 X_4 + C_2 q x_1 + C_3 q(1 - X_1 - X_2) - C_I(I_m - X_3)^2 - C_u A^2 X_4^2 - C_T(T_m - T)^2] dt \quad (7)$$

is maximized, subject to the constraints

$$V \frac{dX_1(t)}{dt} = q(X_1^0 - X_1(t)) - VK_a X_1(t) \quad (8)$$

$$V \frac{dX_2(t)}{dt} = q(X_2^0 - X_2(t)) - VK_b X_2(t) + VK_a X_1(t) \quad (9)$$

$$\frac{dX_3(t)}{dt} = qX_2(t) - X_4(t) \quad (10)$$

$$\text{and} \quad \frac{dX_4(t)}{dt} = X_4(t)[C_c + A(t)][1 - \frac{X_4(t)}{N}] \quad (11)$$

with the boundary conditions

$$X_1(t_0) = X_1^0 \quad (12)$$

$$X_2(t_0) = X_2^0 \quad (13)$$

$$x_3(t_0) = x_3^0 \quad (14)$$

$$\text{and } x_4(t_0) = x_4^0. \quad (15)$$

Note that the final time t_f is explicitly specified in this formulation and it is assumed that all functions have continuous second order derivatives. The problem is assumed to be deterministic and that there are no bounds over the state history except at the initial point. There are no bounds over the temperature $T(t)$ and from physical reasoning, it is clear that advertisement rate $A(t)$ cannot be negative, i.e.

$$A(t) \geq 0, \quad t_0 \leq t \leq t_f. \quad (16)$$

FORMULATION OF THE PROBLEM

The procedure for solution of this problem is the same as for the last two problems. First, the necessary conditions for optimality are in the usual way and then necessary equations are derived for boundary condition iteration.

Following the classical treatment in Calculus of Variations, let us introduce the set of Lagrange multipliers

$$\lambda_i(t), \quad i = 1, \dots, 4, \quad t_0 \leq t \leq t_f \quad (17)$$

and the set of constant multipliers

$$\mu_j, \quad j = 1, \dots, 4. \quad (18)$$

Define the functions

$$\begin{aligned} F(\bar{X}, \dot{\bar{X}}, T(t), A(t), \bar{\lambda}) = & \lambda_1 [\dot{X}_1 - \frac{q}{V} (X_1^0 - X_1) + K_a X_1] + \lambda_2 [\dot{X}_2 - \frac{q}{V} (X_2^0 - X_2) \\ & + K_b X_2 - K_a X_1] + \lambda_3 [\dot{X}_3 - qX_2 + X_4] + \lambda_4 [\dot{X}_4 - X_4 (C_c + A) (1 - \frac{X_4}{N})] \\ & + [C_1 X_4 + C_2 q X_1 + C_3 q (1 - X_1 - X_2) - C_1 (I_m - X_3)^2 - C_u A^2 X_4^2 - C_T (T_m - T)^2] \end{aligned} \quad (19)$$

$$\text{and } G(\bar{X}^0, \bar{\mu}) = \mu_1[x_1(t_0) - x_1^0] + \mu_2[x_2(t_0) - x_2^0] + \mu_3[x_3(t_0) - x_3^0] + \mu_4[x_4(t_0) - x_4^0] \quad (20)$$

The Euler-Lagrange equations [15],

$$\frac{d}{dt} \left(\frac{\partial F}{\partial \dot{x}_i} \right) - \frac{\partial F}{\partial x_i} = 0, \quad i = 1, \dots, 4 \quad (21)$$

$$\text{and } \frac{\partial F}{\partial T} = 0, \quad \frac{\partial F}{\partial A} = 0 \quad (22)$$

can now be applied to equation (19) to obtain the following relationships

$$\frac{d\lambda_1(t)}{dt} = q \left[\frac{\lambda_1}{V} + C_2 - C_3 \right] - K_a [\lambda_2 - \lambda_1] \quad (23)$$

$$\frac{d\lambda_2(t)}{dt} = q \left[\frac{\lambda_2}{V} - \lambda_3 - C_3 \right] + \lambda_2 K_b \quad (24)$$

$$\frac{d\lambda_3(t)}{dt} = 2C_I (I_m - x_3) \quad (25)$$

$$\text{and } \frac{d\lambda_4(t)}{dt} = \lambda_3 - \lambda_4 (C_c + A) \left(1 - \frac{2x_4}{N} \right) + C_1 - 2C_u A^2 x_4. \quad (26)$$

Application of equation (22) to equation (19) yields

$$A(t) = \lambda_4 (x_4 - N) / 2C_u N x_4 \quad (27)$$

$$\text{and } \frac{K_a x_1 E_a}{RT^2} [\lambda_2 - \lambda_1] + \frac{K_b x_2 E_b \lambda_2}{RT^2} + 2C_I (T_m - T) = 0. \quad (28)$$

Equation (27) gives an explicit expression for control variable $A(t)$

and hence can be eliminated from all the performance equations. However,

equation (28) gives an implicit equation for control variable $T(t)$ and therefore DSV method has to be used. It will be observed that all the equations are non-linear differential equations. For the 8 differential equations [equations (8) through (11) and equations (23) through (26)] we have only four boundary conditions given by equations (12) through (15). Additional four boundary conditions can be obtained by applying transversality conditions [15].

$$\left. \frac{\partial G}{\partial X_1} \right|_{t=t_f} + \left. \frac{\partial F}{\partial \dot{X}_1} \right|_{t=t_f} = 0 \quad (29)$$

to equations (19) and (20). Hence, we obtain

$$\lambda_1(t_f) = 0 \quad (30)$$

$$\lambda_2(t_f) = 0 \quad (31)$$

$$\lambda_3(t_f) = 0 \quad (32)$$

$$\lambda_4(t_f) = 0 \quad (33)$$

The boundary conditions given by equations (12) through (15) and equations (30) through (33) make this system, a two point boundary value problem.

BOUNDARY CONDITION ITERATION

The procedure for the solution of the problem is the same as that used in previous two models. The final values of the state variables are estimated using judgment and knowledge of the process. These estimates are then improved till the boundary conditions at the initial point are satisfied using the combined gradient and Newton-Raphson technique and employing the sensitivity information. In order to obtain the necessary equations in vector form, we define the generalized Hamiltonian

$$\begin{aligned}
H = & \lambda_1 \left[\frac{q}{V} [X_1^0 - X_1] - K_a X_1 \right] + \lambda_2 \left[\frac{q}{V} [X_2^0 - X_2] - K_b X_2 + K_a X_1 \right] \\
& + \lambda_3 [qX_2 - X_4(t)] + \lambda_4 \left[(C_c + A) \left(X_4 - \frac{X_4^2}{N} \right) \right. \\
& \left. - [C_1 X_4 + C_2 q X_1 + C_3 q (1 - X_1 - X_2) - C_I (I_m - X_3)^2 - C_u A^2 X_4^2 \right. \\
& \left. - C_T (T_m - T)^2] \right].
\end{aligned} \quad (34)$$

The process and adjoint equations can therefore be written as

$$\dot{\bar{X}}(t) = H_{\bar{X}}[\bar{X}, A(t), T(t)] \quad (35)$$

$$\text{and } \dot{\bar{\lambda}}(t) = -H_{\bar{\lambda}}[\bar{X}, \bar{\lambda}, A(t), T(t)] \quad (36)$$

and equations (27) and (28) can be written as

$$H_A[\bar{X}, \bar{\lambda}, A(t)] = 0 \quad (37)$$

$$H_T[\bar{X}, \bar{\lambda}, T(t)] = 0 \quad (38)$$

As discussed in Chapter 3, we need the sensitivity measures

$$\frac{\partial \bar{X}(t)}{\partial \bar{X}(t_f)} \text{ and } \frac{\partial \bar{\lambda}(t)}{\partial \bar{X}(t_f)}. \text{ The differential sensitivity equations for}$$

these measures may be derived as indicated in Chapter 3 and are given by

$$\frac{d}{dt} \left(\frac{\partial \bar{X}(t)}{\partial \bar{X}(t_f)} \right) = A(t) \frac{\partial \bar{X}(t)}{\partial \bar{X}(t_f)} + B(t) \frac{\partial \bar{\lambda}(t)}{\partial \bar{X}(t_f)} \quad (39)$$

$$\left. \frac{\partial \bar{X}(t)}{\partial \bar{X}(t_f)} \right|_{t=t_f} = I \quad (40)$$

$$\frac{d}{dt} \left(\frac{\partial \bar{\lambda}(t)}{\partial \bar{X}(t_f)} \right) = C(t) \frac{\partial \bar{X}(t)}{\partial \bar{X}(t_f)} + D(t) \frac{\partial \bar{\lambda}(t)}{\partial \bar{X}(t_f)} \quad (44)$$

$$\left. \frac{\partial \bar{\lambda}(t)}{\partial \bar{X}(t_f)} \right|_{t=t_f} = 0 \quad (42)$$

where I is a (4×4) identity matrix and the matrices $A(t)$, $B(t)$, $C(t)$ and $D(t)$ are given by

$$A(t) = H_{\bar{\lambda}, \bar{X}} - H_{\bar{\lambda}, T} H_{T, T}^{-1} H_{T, \bar{X}} - H_{\bar{\lambda}, A} H_{A, A}^{-1} H_{A, \bar{X}} \quad (43)$$

$$B(t) = - [H_{\bar{\lambda}, T} H_{T, T}^{-1} H_{T, \bar{\lambda}} + H_{\bar{\lambda}, A} H_{A, A}^{-1} H_{A, \bar{\lambda}}] \quad (44)$$

$$C(t) = - H_{\bar{X}, \bar{X}} + H_{\bar{X}, T} H_{T, T}^{-1} H_{T, \bar{X}} + H_{\bar{X}, A} H_{A, A}^{-1} H_{A, \bar{X}} \quad (45)$$

and

$$D(t) = - [A(t)]^T \quad (46)$$

where subscripts denote partial differentiation and superscript T denotes matrix transpose. For the particular problem the elements of the (4×4) matrices are given by

$$A_{11} = - \left[\frac{q}{V} + K_a \right] + \frac{X_1 K_a^2 E_a^2 (\lambda_2 - \lambda_1)}{R^2 T^4 H_{T, T}}$$

$$A_{12} = \frac{-X_1 \lambda_2 K_a K_b E_a E_b}{R^2 T^4 H_{T, T}}$$

$$A_{13} = 0$$

$$A_{14} = 0$$

$$A_{21} = K_a - \frac{K_a E (\lambda_2 - \lambda_1) (X_1 K_a E_a - X_2 K_b E_b)}{R^2 T^4 H_{T,T}}$$

$$A_{22} = - \left[\frac{q}{V} + K_b \right] + \frac{\lambda_2 K_b E_b [X_1 K_a E_a - \lambda_2 K_b E_b]}{R^2 T^4 H_{T,T}}$$

$$A_{23} = 0$$

$$A_{24} = 0$$

$$A_{31} = 0$$

$$A_{32} = q$$

$$A_{33} = 0$$

$$A_{34} = -1$$

$$A_{41} = 0$$

$$A_{42} = 0$$

$$A_{43} = 0$$

$$A_{44} = (C_c + A) \left(1 - \frac{2X_4}{N}\right) - \frac{X_4}{H_{A,A}} \left(1 - \frac{X_4}{N}\right) \left[\lambda_4 \left(1 - \frac{2X_4}{N}\right) + 4C_u A X_4\right] \quad (47)$$

$$B_{11} = \frac{-X_1^2 K_a^2 E_a^2}{R^2 T^4 H_{T,T}}$$

$$B_{12} = \frac{X_1 K_a E_a}{R^2 T^4 H_{T,T}} [X_1 K_a E_a - X_2 K_b E_b]$$

$$B_{13} = 0$$

$$B_{14} = 0$$

$$B_{21} = B_{12}$$

$$B_{22} = \frac{-1}{R^2 T^4 H_{T,T}} [X_1 E_a K_a - X_2 K_b E_b]^2$$

$$B_{23} = 0$$

$$B_{24} = 0$$

$$B_{31} = 0$$

$$B_{32} = 0$$

$$B_{33} = 0$$

$$B_{34} = 0$$

$$B_{41} = 0$$

$$B_{42} = 0$$

$$B_{43} = 0$$

$$B_{44} = -\frac{X_4^2}{H_{A,A}} \left(1 - \frac{X_4}{N}\right)^2 \quad (48)$$

$$C_{11} = \frac{K_a^2 E_a^2 (\lambda_2 - \lambda_1)^2}{R^2 T^4 H_{T,T}}$$

$$C_{12} = \frac{-\lambda_2 K_a K_b E_a E_b (\lambda_2 - \lambda_1)}{R^2 T^4 H_{T,T}}$$

$$c_{13} = 0$$

$$c_{14} = 0$$

$$c_{21} = c_{12}$$

$$c_{22} = \frac{(\lambda_2 K_b E_b)^2}{R^2 T^4 H_{T,T}}$$

$$c_{23} = 0$$

$$c_{24} = 0$$

$$c_{31} = 0$$

$$c_{32} = 0$$

$$c_{33} = -2C_I$$

$$c_{34} = 0$$

$$c_{41} = 0$$

$$c_{42} = 0$$

$$c_{43} = 0$$

$$c_{44} = \frac{2\lambda_4}{N} (C_c + A) + 2C_u A^2 - \frac{1}{H_{A,A}} [\lambda_4 (1 - \frac{2X_4}{N}) + 4C_u A X_4]^2 \quad (49)$$

$$\text{and } D(t) = - [A(t)]^T \quad (50)$$

$$\text{where } H_T = \frac{X_1 K_a E_a}{R T^2} (\lambda_2 - \lambda_1) - \frac{-\lambda_2 K_b E_b X_2}{R T^2} - 2C_T (T_m - T) \quad (51)$$

$$H_{T,T} = \frac{-2}{T} H_T + \frac{K_a X_1 E_a^2}{R^2 T^4} (\lambda_2 - \lambda_1) - \frac{K_b E_b^2 X_2^2}{R^2 T^4} + 2C_T \quad (52)$$

$$\text{and } H_{A,A} = -2C_u X_4^2. \quad (53)$$

The residual norm vector \bar{R} is defined as

$$\bar{R} = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \psi_4 \end{pmatrix} \quad (54)$$

$$= \begin{pmatrix} X_1(t_0) - X_1^0 \\ X_2(t_0) - X_2^0 \\ X_3(t_0) - X_3^0 \\ X_4(t_0) - X_4^0 \end{pmatrix}. \quad (55)$$

The constrained Jacobian, $\frac{d\bar{R}}{d\pi}$ is given explicitly by

$$\Gamma(\bar{\pi}) = \begin{pmatrix} \frac{\partial X_1(t_0)}{\partial X_1(t_f)} & \frac{\partial X_1(t_0)}{\partial X_2(t_f)} & \frac{\partial X_1(t_0)}{\partial X_3(t_f)} & \frac{\partial X_1(t_0)}{\partial X_4(t_f)} \\ \frac{\partial X_2(t_0)}{\partial X_1(t_f)} & \frac{\partial X_2(t_0)}{\partial X_2(t_f)} & \frac{\partial X_2(t_0)}{\partial X_3(t_f)} & \frac{\partial X_2(t_0)}{\partial X_4(t_f)} \\ \frac{\partial X_3(t_0)}{\partial X_1(t_f)} & \frac{\partial X_3(t_0)}{\partial X_2(t_f)} & \frac{\partial X_3(t_0)}{\partial X_3(t_f)} & \frac{\partial X_3(t_0)}{\partial X_4(t_f)} \\ \frac{\partial X_4(t_0)}{\partial X_1(t_f)} & \frac{\partial X_4(t_0)}{\partial X_2(t_f)} & \frac{\partial X_4(t_0)}{\partial X_3(t_f)} & \frac{\partial X_4(t_0)}{\partial X_4(t_0)} \end{pmatrix}. \quad (56)$$

The elements of the matrix $\Gamma(\bar{\pi})$ are obtained by integrating the sensitivity differential equations (39) and (41) backwards. All the necessary information is now available for iterating on the boundary conditions using the iteration formula

$$\bar{\pi}^{K+1} = \bar{\pi}^K - [(1 - q_k)\gamma'_k \Gamma(\bar{\pi}^K)^T + q_k \Gamma(\bar{\pi}^K)^{-1}] \bar{R}(\bar{\pi}^K). \quad (57)$$

As before, q_k is the weighting factor and γ'_k is the normalized step size for the gradient technique.

NUMERICAL ASPECTS

In order to obtain numerical solution of this problem, the constants were assumed to have the following values. The values were selected so as to make the system as close as possible to the one solved by Lee and Shah [17].

$$G_a = 0.535 \times 10^{11} \text{ per unit time}$$

$$G_b = 0.461 \times 10^{18} \text{ per unit time}$$

$$E_a = 18000 \text{ cal/mole}$$

$$E_b = 30000 \text{ cal/mole}$$

$$R = 2 \text{ cal/mole } ^\circ\text{K}$$

$$V = 24$$

$$2 = 60$$

$$I_m = 10$$

$$T_m = 340^\circ\text{K}$$

$$N = 100$$

$$C_c = 1$$

$$C_T = 0.001$$

$$c_u = 0.01$$

$$c_1 = 5.0$$

$$c_2 = c_3 = 0.0$$

$$c_I = 1.0$$

$$x_1^0 = 0.53$$

$$x_2^0 = 0.43$$

$$x_3^0 = 1.0$$

$$x_4^0 = 0.1$$

$$t_0 = 0.0$$

$$t_f = 1.0$$

Since an explicit solution for $T(t)$ cannot be obtained, it is necessary to assume an initial trajectory for this control variable. For all numerical work, the following trajectory was assumed:

$$T(t) = 340^\circ\text{K}, \quad t_0 \leq t \leq t_f. \quad (58)$$

With the assumed initial function, equation (58), the process and adjoint equations are integrated. To obtain a solution of the equation (28) in conjunction with the process and adjoint equations, direct second variational method outlined in Appendix (1) was used. The iteration cycle for DSV method is nested inside the boundary condition iteration cycle in the computer program.

Since only the final conditions for adjoint variables $\bar{\lambda}(t)$ are given at the final time, it is necessary to assume the boundary conditions of the state variables at the final time t_f . The different

sets of values assumed for the state variables at the final time are given in Table 6.1.

In order to ensure rapid and uniform convergence to the optimum, the weighting factor q_k and the step size γ_k have to be suitably selected. Because of the nature of the process equations, it was observed during computer runs that the problem was very sensitive to the values assumed for the control variable $T(t)$, the state variables $\bar{X}(t_f)$, the weighting factor q_k and the step size γ_k . The results have been documented and discussed in detail in the next section for each set separately.

RESULTS

The initial function assumed for the control variable $T(t)$, given by equation (58) was the same for all the sets. Hence the sets have been grouped according to the values assumed for the state variables $\bar{X}(t_f)$. This problem was solved on an IBM-360/50 computer. Runge-Kutta integration method was used to integrate the systems of non-linear differential equations. The step size used for the numerical integration was 0.01.

Set 1.

The different combinations of values of the weighting factor q_k and step size γ_k tried are given in Table 6.2. The assumed values for $\bar{X}(t_f)$ were close to the solution of the problem solved by Lee and Shah [17].

Set 1A which corresponds to pure Newton-Raphson method failed after two iterations. Sets 1B and 1C, in which the effect of gradient technique was gradually increased, also failed to converge. In set 1D,

TABLE 6.1. Initial Approximations Used for the Production and Advertisement Scheduling Model

Initial Approximation of Temperature $T(t) \approx 340^{\circ}$, $t_0 \leq t \leq t_f$.

Set No.	$x_1(t_f)$	$x_2(t_f)$	$x_3(t_f)$	$x_4(t_f)$
1	0.42	0.47	2.5	50.0
2	0.42	0.47	2.5	80.0
3	0.4	0.45	2.5	50.0

TABLE 6.2. Effect of Weighting Factor q_K and Step Size v_K on Convergence Rate for Set 1

$$X_1(t_f) = 0.42, \quad X_2(t_f) = 0.47, \quad X_3(t_f) = 2.5, \quad X_4(t_f) = 50.0$$

No.	Weighting Factor and Step Size	No. of Iterations	Initial Value of Norm $ R $	Final Value of Norm $ R $	Computer Time, Min.	Remarks
1A	$q_K = 1$	2	1.572870	1.606681	1.74	Overflows after two iterations
	$v_K = 0.1$					
1B	$q_K = \frac{K}{1+K}$	42	1.572967	0.100492	31.2	Converges uniformly in first 14 iterations and then oscillates
	$v_K = 0.1$					
1C	$q_K = \frac{K}{5+K}$	111	1.572967	0.100492	56.4	Converges uniformly in first 14 iterations & then oscillates
	$v_K = 0.1$					
1D	$q_K = \frac{K}{20+K}$	28	1.572870	0.000079	25.3	Converges uniformly to the optimum
	$v_K = 0.5$					

* K = Iteration number

the effect of gradient technique was increased further and the solution was obtained in 28 iterations.

The rates of convergence of the concentrations of A and B are shown in Fig. (6.2). The convergence rate of inventory, $X_3(t)$, is shown in Fig. (6.3). The optimal trajectory is quite close to the trajectory at the first iteration and the convergence is uniform. The convergence rates of sales $X_4(t)$, the temperature $T(t)$, the advertisement rate $A(t)$ and the profit J are shown in Fig. (6.4), Fig. (6.5), Fig. (6.6) and Fig. (6.7) respectively.

In obtaining the Figs. (6.1) through (6.7), the following approximations of the state variables at the final time were used

$$X_1(t_f) = 0.42$$

$$X_2(t_f) = 0.47$$

$$X_3(t_f) = 2.5$$

$$X_4(t_f) = 50.0 .$$

The optimal profit in this problem was 94.23 and the optimal values of state variables at final time are

$$X_1(1) = 0.4437$$

$$X_2(1) = 0.4753$$

$$X_3(1) = 2.4021$$

$$X_4(1) = 48.1296.$$

Set 2.

The values assumed for the state variables at final time were

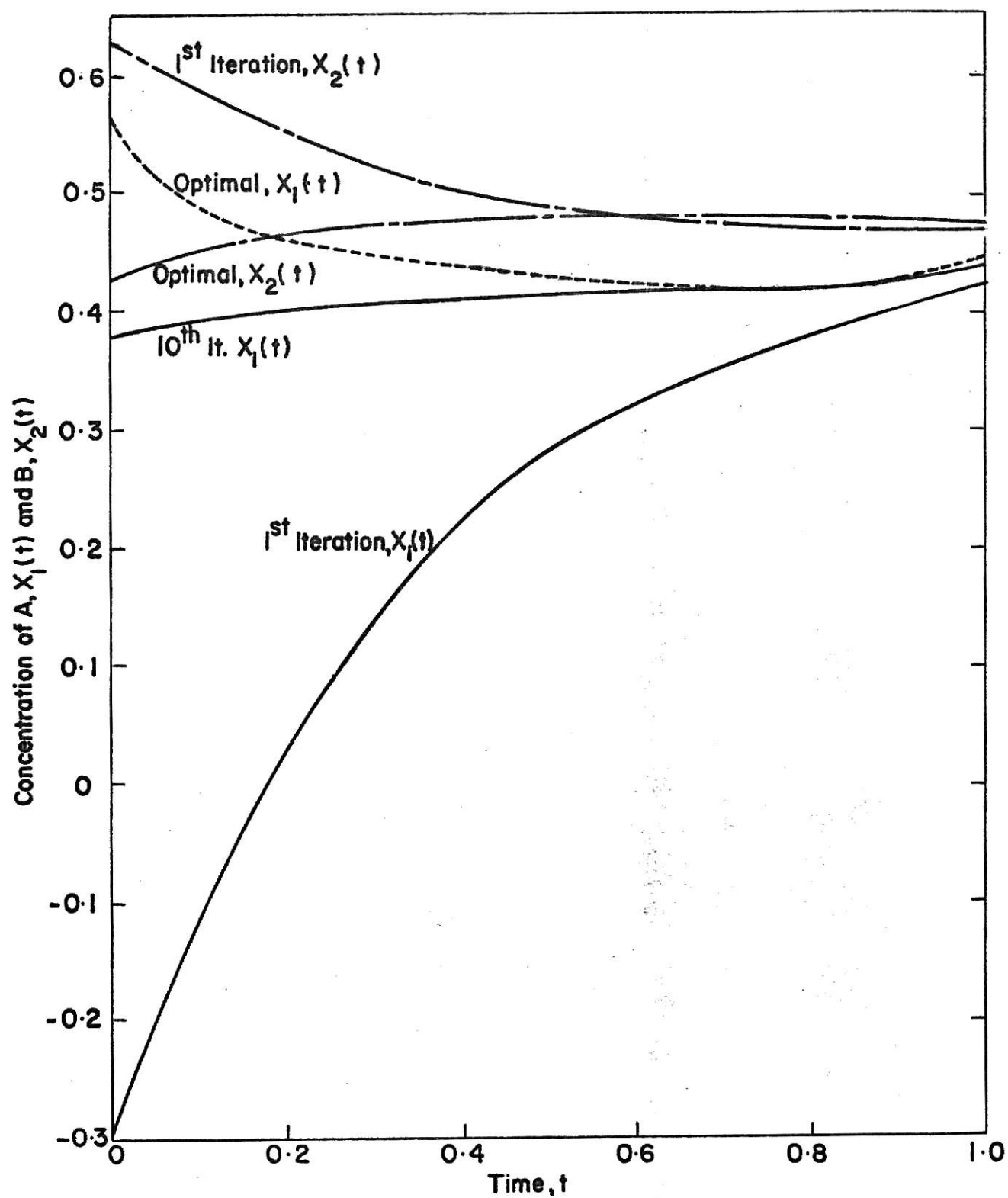


Fig.6.2. Convergence rates of concentration of A and B for set I D.

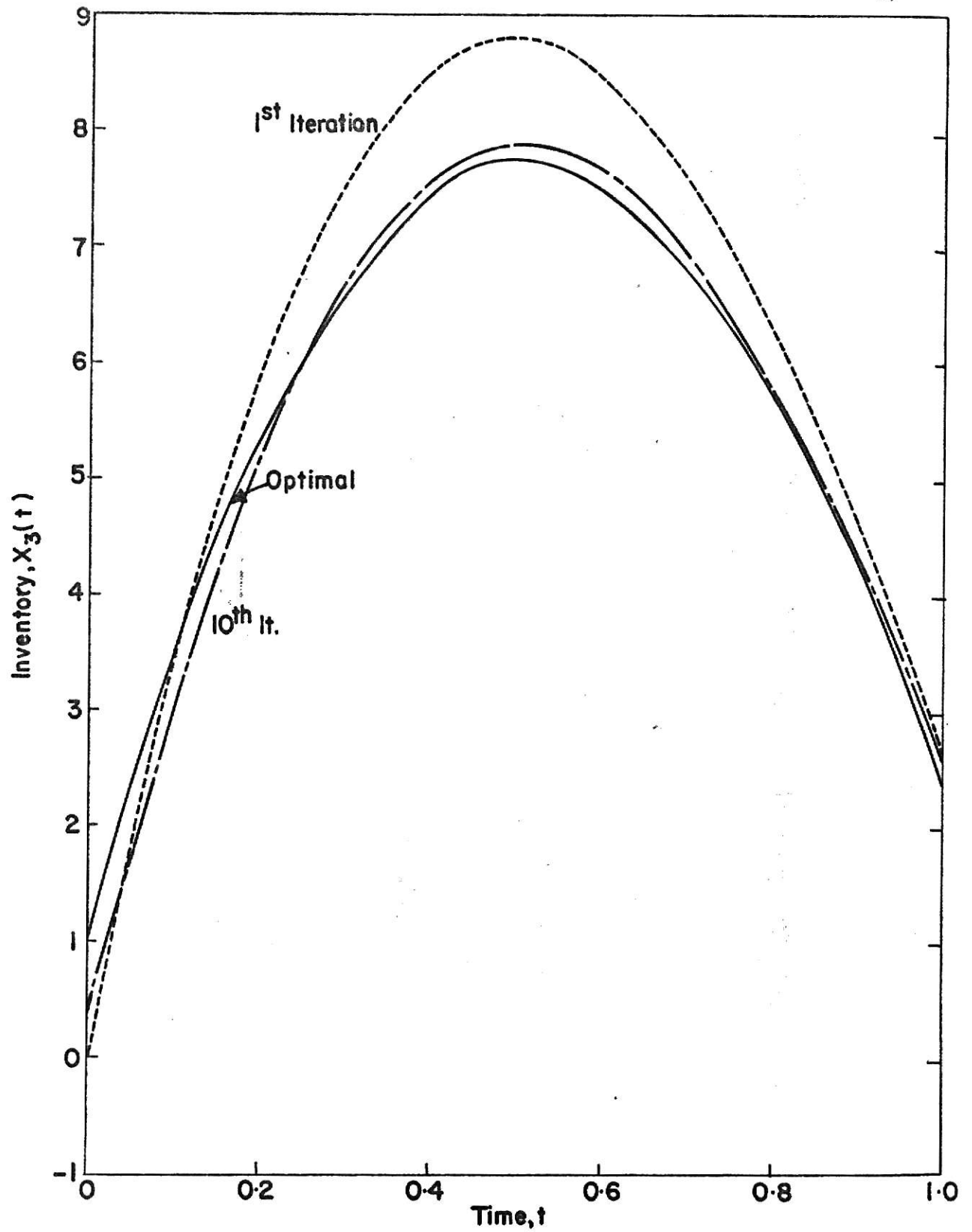


Fig.6.3. Convergence rate of inventory, $X_3(t)$.

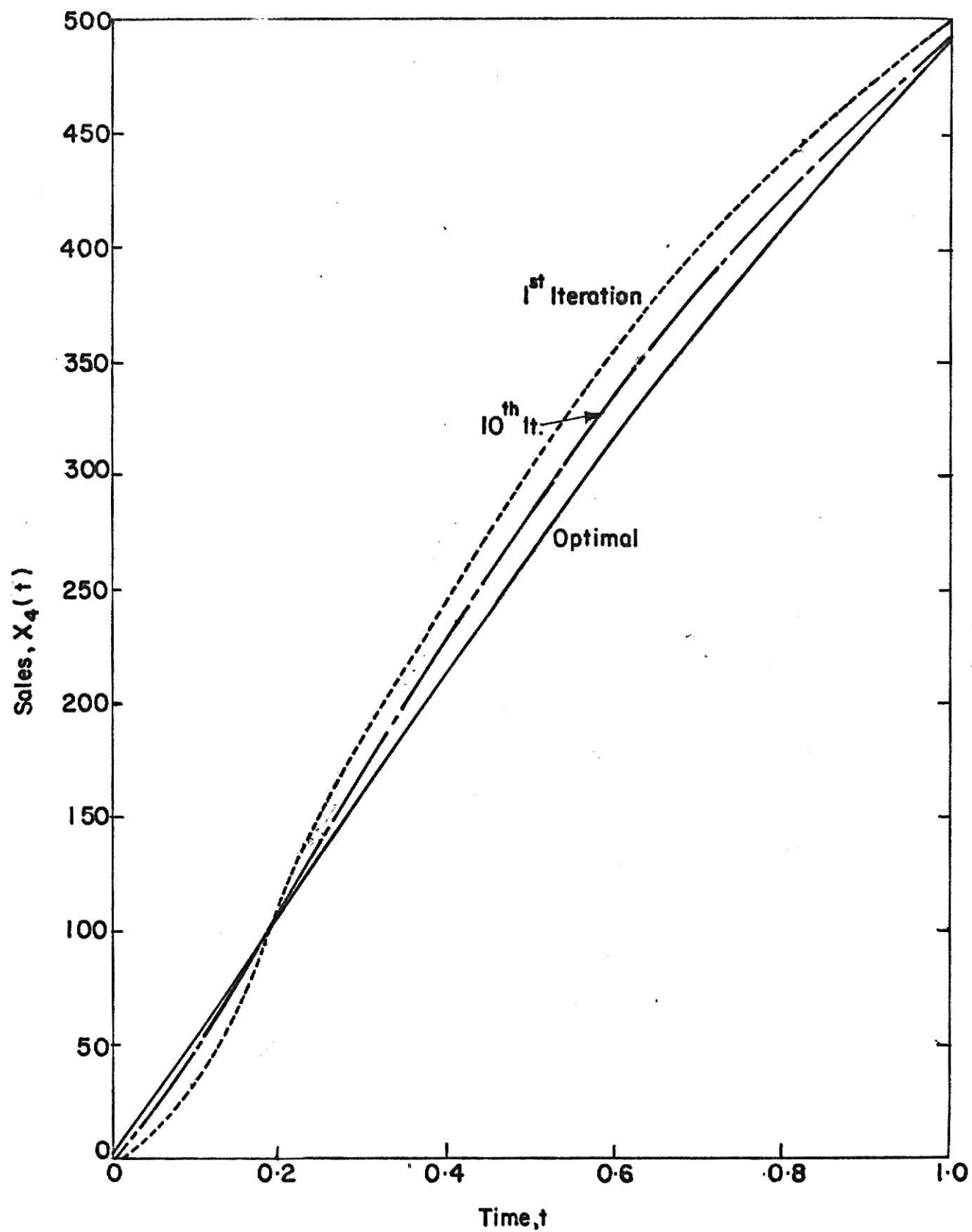


Fig.6.4. Convergence rate of sales, $X_4(t)$.

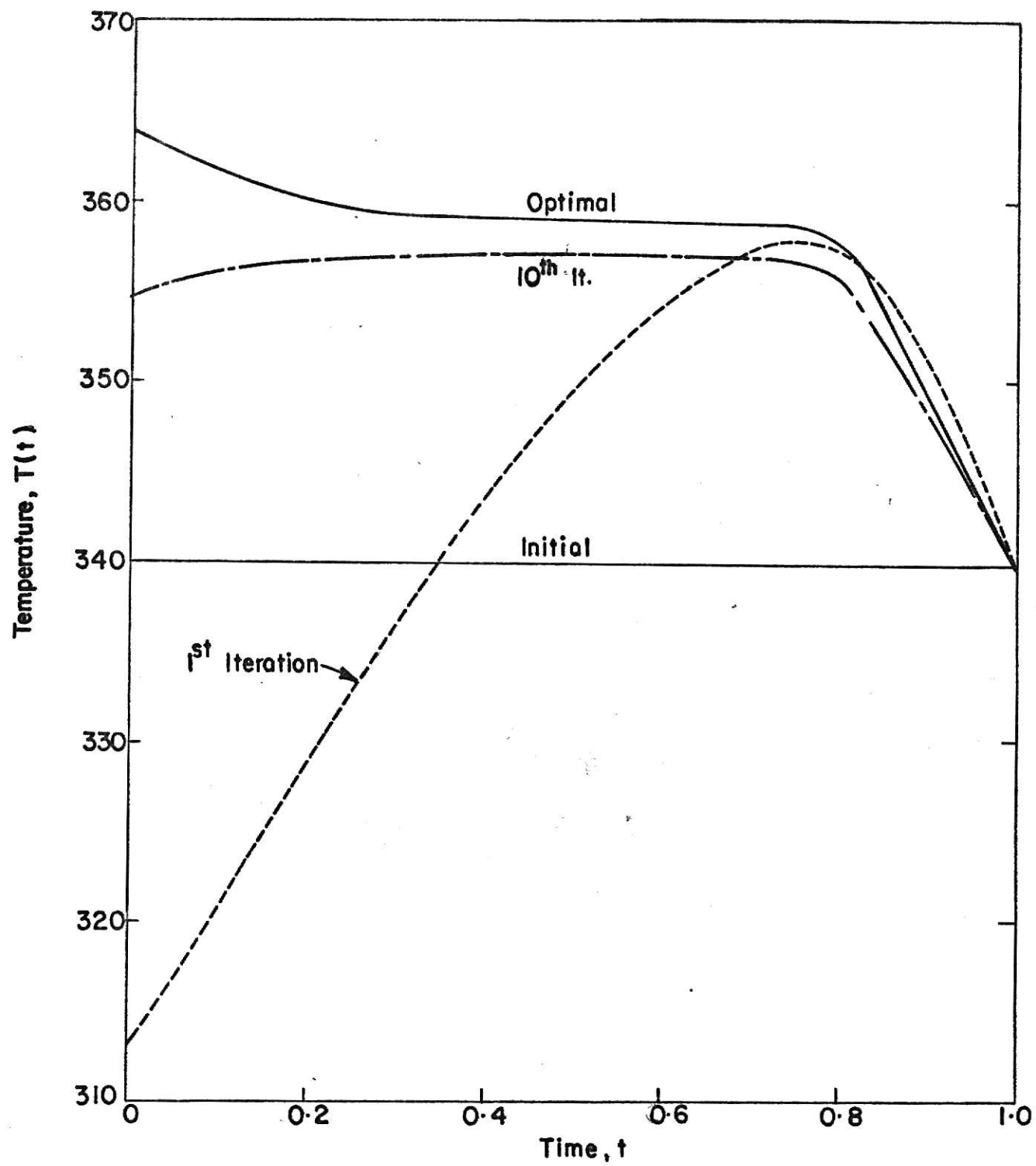


Fig.6.5. Convergence rate of temperature $T(t)$.

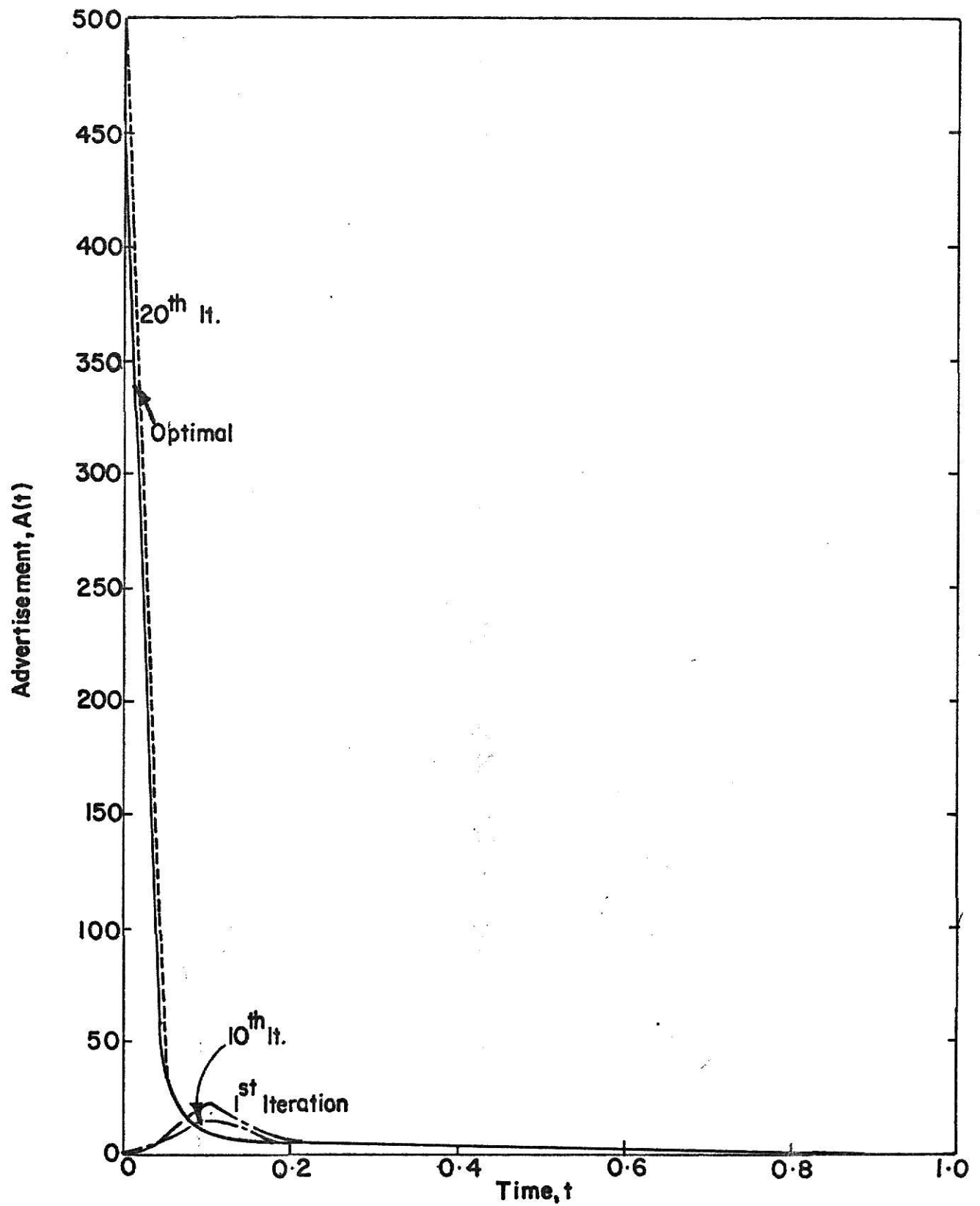


Fig.6.6. Convergence rate of advertisement $A(t)$.

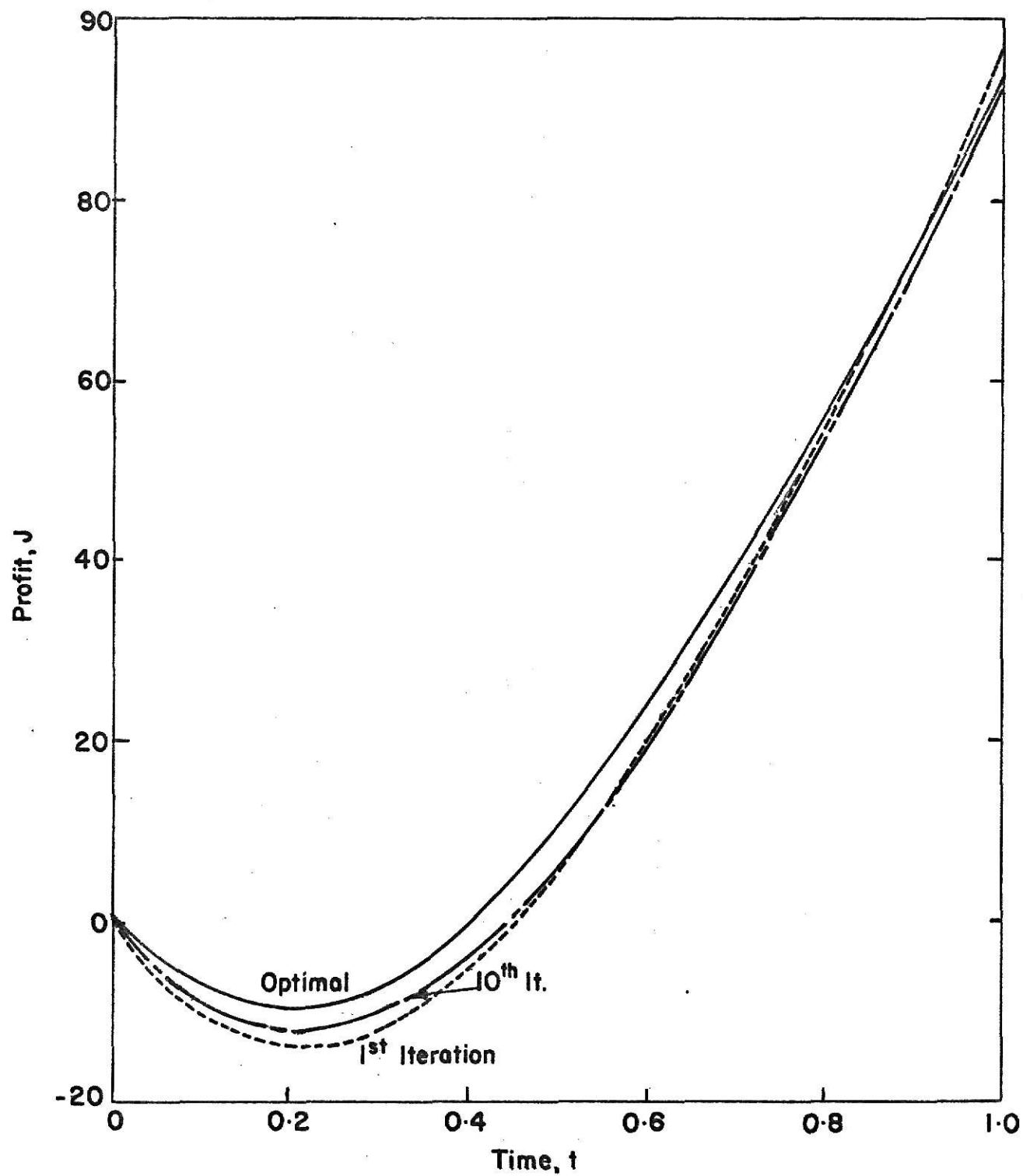


Fig.6.7. Convergence rate of profit J .

$$X_1(t_f) = 0.42$$

$$X_2(t_f) = 0.47$$

$$X_3(t_f) = 2.5$$

$$X_4(t_f) = 80.0$$

For this set of values, overflow occurred during the initial integration of process and adjoint equations. Hence this set was not explored further. However, from this set we get a clue to the maximum value of $X_4(t_f)$ that can be used in the initial guesses.

Set 3.

The different combinations of values of the weighting factor q_K and step size γ_K tried are given in Table 6.3. The assumed values of $X_1(t_f)$ and $X_2(t_f)$ were different from those assumed in Set 1.

Sets 3A through 3E failed to converge. The results of these sets are given in Table 6.3. In our considered opinion, this failure was mainly caused by the discontinuity in advertisement rate curve near the initial boundary. A near discontinuity in the function can make the Newton-Raphson method fail to converge. The discontinuity in advertisement rate function is caused by the sales which remains negative at $t = 0$ in all the cases.

TABLE 6.3. Effect of Weighting Factor q_K and Step Size γ_K on Convergence Rate for Set 3.

$$X_1(t_f) = 0.4, \quad X_2(t_f) = 0.45, \quad X_3(t_f) = 2.5, \quad X_4(t_f) = 50.0$$

No.	Weighting Factor and Step Size	No. of Iterations	Initial Value of Norm R	Final Value of Norm R	Computer Time, Min.	Remarks
3A	$q_K = \frac{K}{20+K}$	35	2.610972	0.207509	25.6	Did not converge. Sales negative at $t = 0$.
	$\gamma_K = 0.5$					
3B	$q_K = \frac{K}{30+K}$	78	2.610972	0.169049	50.5	Did not converge. Sales remains negative at $t = 0$.
	$\gamma_K = 0.4$					
3C	$q_K = \frac{K}{50+K}$	74	2.610972	0.153814	50.6	Did not converge. Sales remains negative at $t = 0$.
	$\gamma_K = 0.2$					
3D	$q_K = \frac{K}{50+K}$	20	2.610972	0.500829	18.0	Converges uniformly to 0.181243 in 19 iterations and then diverges.
	$\gamma_K = 0.3$					
3E	$q_K = \frac{K}{100+K}$	68	2.610972	0.124791	45.0	Did not converge. Sales remains negative at $t = 0$.
	$\gamma_K = 0.1$					

CHAPTER 7

CONCLUSION

The usefulness of boundary condition iteration based on differential sensitivity analysis has been shown. The method converges to the solution even with crude initial approximations of the state variables in most of the cases. In some cases, depending on the nature of the equations, it may fail to converge from points not close to the optimum. On the other hand, the rate of convergence depends heavily on the choice of the weighting factor and the step size used. A suitable combination of these factors has to be determined by trial and error for each problem. However, it is suggested that initially gradient technique be used to bring the approximations near the optimum and then utilize the quadratic convergence property of the Newton-Raphson method to obtain the optimum.

This technique in conjunction with direct second variational method has proved successful even for those problems where explicit solutions for control variables could not be obtained. It has even proved successful in closed-loop problems where other techniques like Quasilinearization failed.

Differential sensitivity analysis has a major disadvantage. The number of equations to be integrated increases rapidly as the number of state variables increase and therefore limit its application to problems of small dimension.

While evaluating the merits and demerits of this problem, it should be considered that no single optimization technique is suitable for all classes of problems that are encountered in practical situations.

The selection of the technique to be used depends upon the nature of the problem and the nature of the application and is best left to the analyst solving the problem.

REFERENCES

1. Bliss, G. A., Lectures on the Calculus of Variations, University of Chicago Press, Chicago, 1946.
2. Bryson, A. E., Denham, W. F., Carroll, F. J., and Mikani, K., Determination of Lift or Drag Programs to Minimize Reentry Heating, Journal of Aerospace Sciences, Vol. 29, 1962, p. 540.
3. Bryson, A. E., and Denham, W. F., A Steepest-Ascent Method for Solving Optimum Programming Problems, Journal of Applied Mechanics, Vol. 29, 1962.
4. Conrad, D. A., Quasilinearization Extended and Applied to the Computation of Lunar Landings, Proceedings, 15th International Astronautical Congress, Warsaw, Poland, 1964.
5. Denn, M. M., and Aris, R., Green's Functions and Optimal Systems. Gradient Direction in Decision Space, Ind. & Engg. Chem. Fundamentals, Vol. 4, 1965, p. 213.
6. Fine, F. A., and Bankoff, S. G., Control Vector Iteration in Chemical Plant Optimization. Accessory Minimization Problem of Jacobi, Ind. & Engg. Chem. Fundamentals, Vol. 6, 1967, p. 288.
7. Hestenes, M. R., Numerical Methods of Obtaining Solution of Fixed End Point Problems in Calculus of Variations, The RAND Corporation, Memo. No. RM-102, 1949.
8. Kalaba, R., On Non-linear Differential Equations, the Maximum Operation and Monotone Convergence, Journal of Mathematics and Mechanics, Vol. 8, 1959, p. 519.
9. Kelley, H. J., Methods of Gradients, Optimization Techniques, Chapter 6, Academic Press, New York, 1962.
10. Kelley, H. J., Kopp, R. E., and Moyer, H. G., Successive Approximation Techniques for Trajectory Optimization, Proceedings of the IAS Symposium on Vehicle Systems Optimization, Garden City, New York, 1961.
11. Kelley, H. J., Kopp, R. E., and Moyer, H. G., A Trajectory Optimization Technique Based Upon the Theory of Second Variations, Progress in Astronautics and Aeronautics, Chapter 5, Vol. 14, Academic Press, New York, 1964.
12. Kenneth, P., and McGill, R., Two-Point Boundary Value Problem Techniques, Advances in Control Systems, Chapter 2, Vol. 3, Academic Press, New York, 1966.
13. Kopp, R. E., and Moyer, H. G., Trajectory Optimization Techniques, Advances in Control Systems, Chapter 3, Vol. 4., Academic Press, New York, 1966.

14. Lee, E. S., Quasilinearization and Invariant Imbedding, Academic Press, New York, 1968.
15. Lee, E. S., Quasilinearization in Optimization: A Numerical Study, A.I.Ch.E. Journal, Vol. 13, 1967, p. 1043.
16. Lee, E. S., and Shaikh, Optimal Production Planning by a Gradient Technique, to appear in Management Science.
17. Lee, E. S., and Shah, P. D., Optimization of Production Planning by the Generalized Newton-Raphson Method, presented at the 20th Annual AIIE Conference and Convention, Houston, Texas, May 14-17, 1969.
18. Levine, M. D., A Steepest Descent Method for Synthesizing Optimal Control Programmes, Institute of Mechanical Engineers Convention on Advances in Automatic Control, University of Nottingham, 1965.
19. Levine, M. D., Trajectory Optimization Using Newton-Raphson Method, Automatica, Vol. 3, 1966, p. 203.
20. Lewallen, J. M., A Modified Quasilinearization Method for Solving Trajectory Optimization Techniques, AIAA Journal, Vol. 5., 1967, p. 962.
21. Long, R. S., Newton-Raphson Operator; Problems with Undetermined End Points, AIAA Journal, Vol. 3, 1965, p. 1351.
22. Luus, R., and Lapidus, L., The Control of Nonlinear Systems: Part II. Convergence by Combined First and Second Variations, A.I.Ch.E. Journal, Vol. 13, 1967, p. 108.
23. Merriam, C. W., Optimization Theory and the Design of Feedback Control Systems, McGraw Hill, New York, 1964.
24. Mitter, S. K., Successive Approximation Methods for the Solution of Optimal Control Problems, Automatica, Vol. 3, 1966, p. 135.
25. Moyer, H. G., and Pinkham, G., Several Trajectory Optimization Techniques, Part II, Computing Methods in Optimization Problems, Chapter 3, Academic Press, New York, 1964.
26. Padmanabhan L., and Bankoff, S. G., Application of Differential Sensitivity Analysis to Non-Linear Boundary Value Problems: Part I - An Algorithm for Recycle Reactor Optimization,
27. Padmanabhan, L., and Bankoff, S. G., A Direct-Second-Variational Method for Free-End Point, Free-Time Optimal Control Problems by Newton's Method in Function Space, presented at the 1969 Joint Automatic Control Conference, Boulder, Colorado.

28. Padmanabhan L., and Bankoff, S. G., Application of Differential Sensitivity Analysis to Non-linear Boundary Value Problems: Part II - Computational Results and Convergence Studies,
29. Teichroew, D., An Introduction to Management Science - Deterministic Models, John Wiley, New York, 1964.

APPENDIX 1

DIRECT SECOND - VARIATIONAL METHOD

Direct Second Variational Method is a useful technique for obtaining numerical solution of the control variable $T(t)$ from the Euler-Lagrange equation

$$H_T(\bar{X}(t), \bar{\lambda}(t), T(t)) = 0 \quad (1)$$

Assuming that there is only one control variable in question and that the optimal control $T^*(t)$ is unique, we can apply Newton's method to equation (1). Consider a non-optimal path $(\bar{X}(t), \bar{\lambda}(t), T(t))$ lying in an ϵ neighborhood of the optimal path, $(\bar{X}^*(t), \bar{\lambda}^*(t), T^*(t))$. A first order expansion of $H_T(\bar{X}^*, \bar{\lambda}^*, T^*)$ around $(\bar{X}, \bar{\lambda}, T)$ yields

$$\begin{aligned} H_T(\bar{X}^*, \bar{\lambda}^*, T^*) &= H_T(\bar{X}, \bar{\lambda}, T) + H_{T,\bar{X}}(\bar{X}, \bar{\lambda}, T) \Delta\bar{X}(t) \\ &\quad + H_{T,\bar{\lambda}}(\bar{X}, \bar{\lambda}, T) \Delta\bar{\lambda}(t) + H_{T,T}(\bar{X}, \bar{\lambda}, T) \Delta T(t) + O(\epsilon^2) \end{aligned} \quad (2)$$

where $\Delta\bar{X}(t) = \bar{X}^*(t) - \bar{X}(t)$

$$\Delta\bar{\lambda}(t) = \bar{\lambda}^*(t) - \bar{\lambda}(t)$$

$$\Delta T(t) = T^*(t) - T(t) \quad (3)$$

If $(\bar{X}^*(t), \bar{\lambda}^*(t), T^*(t))$ is the solution, then the left hand side of equation (1) is zero, so that after discarding terms of $O(\epsilon^2)$ we get

$$\Delta T(t) = -H_{T,T}^{-1} [H_T + H_{T,\bar{X}} \Delta\bar{X}(t) + H_{T,\bar{\lambda}} \Delta\bar{\lambda}(t)] \quad (4)$$

Equation (4) forms the basis of the algorithm. Letting $\bar{X} = \bar{X}^k$, $\bar{\lambda} = \bar{\lambda}^k$, $T = T^k$, then equation (4) can be rewritten to obtain the $(k+1)^{st}$ estimate of the control vector:

$$\begin{aligned} \Delta T^k(t) = T^{k+1}(t) - T^k(t) = \\ - \{H_{T,T}^k\}^{-1} \{H_T^k + H_{T,\bar{X}}^k [\bar{X}^{k+1}(t) - \bar{X}^k(t)] + H_{T,\bar{\lambda}}^k [\bar{\lambda}^{k+1} - \bar{\lambda}^k]\} \end{aligned} \quad (5)$$

Since $\bar{X}^k(t)$ is the state of the system at time t under an input $T^k(t)$ and is determined by integrating the system and adjoint equations using the known $T^k(t)$, Equation (5), as it stands, is not useful as an iterative formula, since it involves $\bar{X}^{k+1}(t)$ and $\bar{\lambda}^{k+1}(t)$ which are unknowns. If we assume that variations in $\Delta \bar{X}(t)$ and $\Delta \bar{\lambda}(t)$ between successive iterations are of same order of magnitude, we can modify equation (5) into the following computationally feasible scheme:

$$\begin{aligned} T^{k+1}(t) = T^k(t) - \{H_{T,T}^k\}^{-1} \{H_T^k + H_{T,\bar{X}}^k [\bar{X}^k - \bar{X}^{k-1}] + H_{T,\bar{\lambda}}^k [\bar{\lambda}^k - \bar{\lambda}^{k-1}]\} \\ k \geq 1 \end{aligned} \quad (6)$$

$$\text{and } T^1(t) = T^0(t) - (H_{T,T}^{-1})^0 [H_T^0] \quad k = 0 \quad (7)$$

where $T^0(t)$ is the initial assumed policy.

Normally it is necessary to limit the step sizes in $T(t)$ while using equations (6) and (7), especially if the initial guess is far from the optimum. To this end, a factor $\eta^k \in (0, 1)$ is introduced in equation (6) which now becomes,

$$\begin{aligned} T^{k+1}(t) = T^k(t) - \eta^k \{H_{T,T}^k\}^{-1} \{H_T^k + H_{T,\bar{X}}^k [\bar{X}^k - \bar{X}^{k-1}] + H_{T,\bar{\lambda}}^k [\bar{\lambda}^k - \bar{\lambda}^{k-1}]\} \\ (8) \end{aligned}$$

The computational scheme is as follows:

1. Choose a nominal $T^0(t)$, which is not far from the optimum, and integrate the system and adjoint equations to determine \bar{X}^0 & $\bar{\lambda}^0$.

2. Compute $T^1(t)$ from Equation (7) for the first iteration and $T^k(t)$ by equation (8) when $k \geq 1$.
3. If $||H_T^k||$ is less than some preassigned tolerance, then $T^{k+1}(t)$ is taken to be the optimal policy. Otherwise go to step 2.

This method, for inventory model has yielded the solution in an average of 10-12 iterations. The number of iterations required increase if the initial policy is far from optimum. To improve the convergence, gradient technique may be employed to get close to the optimum and then use the second-variation method outlined above.

APPENDIX 2

COMPUTER PROGRAM FOR THE INVENTORY MODEL

PROBLEM1--INVENTORY MODEL

```

COMMON/ARRAY1/CP,CI,PM,OIM,II
COMMON/ARRAY2/Y(101),ALAG(2,101),KOUNT
COMMON/ARRAY3/AMDA(2,101),PROD(101)
COMMON/ARRAY4/EX(2,101),HTT(101),A21(101),B11(101),B12(101),B21(10
11),B22(101),C11(101),D12(101)
COMMON/ARRAY5/HT(101)
DIMENSION VECZ(2),PP(8,4),RESVEC(2),SENCO(8,101),SEND(8),Q(4),R(4)
1,L(4),M(4),RI(4),RA(2),RM(2)
100 FORMAT(8F10.7)
200 FORMAT(3F10.4,2E15.8)
201 FORMAT(4F6.2,2I3)
400 FORMAT(1H ,3E18.8)
500 FORMAT(1H1,' ITERATION NUMBER ',I4/)
600 FORMAT(1H1, 10X,' THESE ARE FINAL ANSWERS ' )
998 FORMAT(1H ,F8.4,7X,F8.4,4X,F9.4)
999 FORMAT(1H , 'PRODUCTION      INVENTORY      COST ')
KPROB=0

```

READ IN THE DATA

```

READ 100,A,B,C,CI,OIM,CP,PM,TFINAL
886 READ 200,X1,X2,GS,EPS,EPSLN
READ 201,AA,BB,CC,DD,MM,KP

```

PRINT OUT THE DATA

```

PRINT 100,A,B,C,CI,OIM,CP,PM,TFINAL
PRINT 200,X1,X2,GS,EPS,EPSLN
PRINT 201,AA,BB,CC,DD,MM,KPROB
ITER=1
NINT=TFINAL/GS
II=NINT+1
NINT=100
II=101

```

INITIALIZE

```

EX(1,II)=X1
EX(2,II)=X2
VECZ(1)=X1
VECZ(2)=X2

```

ASSUME THE CONTROL FUNCTION

```

DO 1 I=1,II
AI=I
T=(AI-1.)*GS

```

```

PROD(I)=AA-BB*T
1 CONTINUE

```

COMPUTE THE BOUNDARY CONDITIONS ON ADJOINT VARIABLES

```

AMDA(1,II)=0.
AMDA(2,II)=-1.

```

INTEGRATE THE STATE AND ADJOINT EQS. BACKWARDS BY FOURTH
R RUNGE-KUTTA METHOD

```

81 KOUNT=1
8 DO 2 I=1,NINT
  IT=II+1-I
  AI=IT
  T=(AI-1.)*GS
  AT=T
  X1=EX(1,IT)
  X2=EX(2,IT)
  AMD1=AMDA(1,IT)
  AMD2=AMDA(2,IT)
  ALP=(PM-PROD(IT))**2
  DO 3 J=1,4
    PP(1,J)=-GS*(PROD(IT)-A-B*AT)
    PP(2,J)=-GS*(CI*(OIM-X1)**2+CP*EXP(ALP))
    PP(3,J)=-GS*(OIM-X1)*CI*AMD2*2.
    CJ=1.
    IF(J.EQ.3) CJ=2.
    AT=T-GS*CJ/2.
    X1=EX(1,IT)+PP(1,J)*CJ/2.
    X2=EX(2,IT)+PP(2,J)*CJ/2.
    AMD1=AMDA(1,IT)+PP(3,J)*CJ/2.
3  AMD2=AMDA(2,IT)
  EX(1,IT-1)=EX(1,IT)+1./6.*(PP(1,1)+2.*PP(1,2)+2.*PP(1,3)+PP(1,4))
  EX(2,IT-1)=EX(2,IT)+1./6.*(PP(2,1)+2.*PP(2,2)+2.*PP(2,3)+PP(2,4))
  AMDA(1,IT-1)=AMDA(1,IT)+1./6.*(PP(3,1)+2.*PP(3,2)+2.*PP(3,3)+PP(3,
14))
  AMDA(2,IT-1)=AMDA(2,IT)
2 CONTINUE
  CALL H1

```

TEST IF PRESENT CONTROL SATISFIES HT(T)=0

```

DO 5 I=1,II
  IF(ABS(HT(I)).GT.EPS) GO TO 6
5 CONTINUE
  GO TO 7
6 CONTINUE
  CALL H2
  CALL H3

```

IMPROVE THE CONTROL BY DSV METHOD

```

QQQ=KOUNT
STEP=CQQ/(CC+QQQ)
DO 4 I=1,II
  PROD(I)=PROD(I)-STEP*(HT(I)+Y(I))/HTT(I)
4 CONTINUE
  KOUNT=KOUNT+1
  DO 50 I=1,II
    ALAG(1,I)=AMDA(1,I)
    ALAG(2,I)=AMDA(2,I)
50 CONTINUE
  GO TO 8

```

CALCULATE THE RESIDUAL VECTOR AND NORM R

```

7 RESVEC(1)=EX(1,1)-C
  RESVEC(2)=EX(2,1)
  RESV=0.
  DO 9 I=1,2
    RESV=RESV+      (RESVEC(I)**2)
9 CONTINUE
  RESV=ABS(SQRT(RESV))

```

TEST FOR CONVERGANCE

```

IF(RESV.EQ.EPS) GO TO 101
PRINT 500,ITER
PRINT 400,(RESVEC(I),I=1,2),RESV
PRINT 9999
DO 9997 I=1,II
  PRINT 9998,PROD(I),EX(1,I),EX(2,I)
997 CONTINUE

```

COMPUTE SENSITIVITY COEFFICIENTS AT FINAL TIME

```

SENCO(1,II)=1.0
SENCO(2,II)=0.0
SENCO(3,II)=0.0
SENCO(4,II)=1.0
DO 91 I=5,8
91 SENCO(I,II)=0.
  CALL H3
  CALL H4

```

INTEGRATE DIFFERENTIAL SENSITIVITY EQUATIONS BACKWARDS
BY FOURTH ORDER RUNGE KUTTA METHOD

```

DO 10 I=1,NINT

```

```

      IT=II+1-I
      DO 11 K=1,8
11  SEND(K)=SENCO(K,IT)
      DO 12 J=1,4
      PP(1,J)=-(B12(IT)*SEND(7)+B11(IT)*SEND(5))*GS
      PP(2,J)=-(B12(IT)*SEND(8)+B11(IT)*SEND(6))*GS
      PP(3,J)=-(A21(IT)*SEND(1)+B22(IT)*SEND(7)+B21(IT)*SEND(5))*GS
      PP(4,J)=-(A21(IT)*SEND(2)+B22(IT)*SEND(8)+B21(IT)*SEND(6))*GS
      PP(5,J)=-(C11(IT)*SEND(1)+D12(IT)*SEND(7))*GS
      PP(6,J)=-(C11(IT)*SEND(2)+D12(IT)*SEND(8))*GS
      PP(7,J)=0.
      PP(8,J)=0.
      CJ=1.
      IF(J.EQ.3) CJ=2.
      DO 12 K=1,8
12  SEND(K)=SENCO(K,IT)+PP(K,J)*CJ/2.
      DO 13 K=1,8
13  SENCO(K,IT-1)=SENCO(K,IT)+1./6.*(PP(K,1)+2.*PP(K,2)+2.*PP(K,3)+PP(
1K,4))
10  CONTINUE

```

CALCULATE THE ELEMENTS OF CONSTRAINED JACOBIAN MATRIX

```

Q(1 )=SENCO(1 ,1)
Q(2 )=SENCO(3 ,1)
Q(3 )=SENCO(2 ,1)
Q(4 )=SENCO(4 ,1)
N=2

```

PERFORM MATRIX OPERATIONS FOR BOUNDARY CONDITION ITERATION

```

CALL GMTRA(Q,R,N,N)
CALL MINV(Q,N,D,L,M)
QQQ=ITER-1
QK=(QQQ/(QQQ+DD ))**MM
CALL SMPY(Q,QK,RI,N,N,0)
QK1=0.
CALL GMPRD(R,RESVEC,RA,N,N,1)
DO 29 I=1,N
29 QK1=QK1+RA(I)**2
QKI=(1.-QK)*0.5/QK1*RESV**2
CALL SMPY(R,QKI,Q ,N,N,0)
CALL GMADD(RI,Q ,R ,N,N)
CALL GMPRD(R ,RESVEC,RA,N,N,1)
CALL GMSUB(VECZ,RA,RM,N,1)
DO 25 I=1,N
25 VECZ(I)=RM(I)
EX(1,II)=VECZ(1)
EX(2,II)=VECZ(2)
ITER=ITER+1

```

```
GO TO 81
101 PRINT 600
PRINT 400,(RESVEC(I),I=1,2),RESV
PRINT 9999
DO 8888 I=1,II
PRINT 9998,PROD(I),EX(1,I),EX(2,I)
888 CONTINUE
KPROB=KPROB+1
IF(KPROB.EQ.KP) GO TO 8887
GO TO 8886
887 STOP
END
```

SUBROUTINE H1

SUBROUTINE H1 CALCULATES HT(T)

```

COMMON/ARRAY1/CP,CI,PM,OIM,II
COMMON/ARRAY3/AMDA(2,101),PROD(101)
COMMON/ARRAY5/HT(101)
DO 1 I=1,II
1 HT(I)=AMDA(1,I)-2.*AMDA(2,I)*CP*(PM-PROD(I))*EXP((PM-PROD(I))*2)
RETURN
END

```

SUBROUTINE H2

SUBROUTINE H2 CALCULATES Y(T) FOR USE IN DSV METHOD

```

COMMON/ARRAY1/CP,CI,PM,OIM,II
COMMON/ARRAY2/Y(101),ALAG(2,101),KOUNT
COMMON/ARRAY3/AMDA(2,101),PROD(101)
IF(KOUNT.EQ.1) GO TO 2
DO 1 I=1,II
Y(I)=AMDA(1,I)-ALAG(1,I)
1 CONTINUE
RETURN
2 DO 3 I=1,II
3 Y(I)=0.
RETURN
END

```

SUBROUTINE H3

SUBROUTINE H3 CALCULATES HTT(T)

```

COMMON/ARRAY1/CP,CI,PM,OIM,II
COMMON/ARRAY3/AMDA(2,101),PROD(101)
COMMON/ARRAY4/EX(2,101),HTT(101),A21(101),B11(101),B12(101),B21(10
11),B22(101),C11(101),D12(101)
DO 1 I=1,II
1 HTT(I)=2.*AMDA(2,I)*CP*EXP((PM-PROD(I))**2)*(1.+2.*(PM-PROD(I))**2
1)
RETURN
END

```

SUBROUTINE H4

SUBROUTINE H4 CALCULATES MATRICES A,B,C, AND D

```

COMMON/ARRAY1/CP,CI,PM,OIM,II
COMMON/ARRAY3/AMDA(2,101),PROD(101)
COMMON/ARRAY4/EX(2,101),HTT(101),A21(101),B11(101),B12(101),B21(10
11),B22(101),C11(101),D12(101)
DO 1 I=1,II
A21(I)=-2.*CI*(OIM-EX(1,I))
B11(I)=-1./HTT(I)
B12(I)=+2.*CP*(PM-PROD(I))*EXP((PM-PROD(I))**2)/HTT(I)
B21(I)=B12(I)
B22(I)=(B12(I)**2)*HTT(I)
B22(I)=-B22(I)
C11(I)=-2.*AMDA(2,I)*CI
D12(I)=2.*CI*(OIM-EX(1,I))
1 CONTINUE
RETURN
END

```

APPENDIX 3

COMPUTER PROGRAM FOR THE INVENTORY AND ADVERTISEMENT SCHEDULING MODEL


```

C
C      PROBLEM 2 -- INVENTORY AND ADVERTISEMENT SCHEDULING
C      PROBLEM
C
COMMON/BLOCK/II,EX(3,101),CA,C,ADTV(101),ON,AMDA(2,101),F,CI,PI,
1A11(101),A21(101),A31(101),A32(101),B11(101)          ,B31(101),
2      C11(101),C22(101),D11(101),D12(101)
DIMENSION VECZ(3),PP(15,4),RESVEC(3),SENCO(15,101),SEND(15),Q(9),
1R(9),RI(9),RA(3),RM(3),L(9),M(9)
10 FORMAT(10F8.4)
20 FORMAT(4F8.4,E10.4)
21 FORMAT(4F10.4,E10.4)
22 FORMAT(2F7.2,2I4)
30 FORMAT(1H , ' ITERATION NUMBER= 'I6)
40 FORMAT(1H , 4F12.6)
60 FORMAT(1H , 20X, ' THESE ARE FINAL ANSWERS ')
80 FORMAT(1H , ' ADVERTISEMENT      SALES      INVENTORY      PROFIT ')
90 FORMAT(1H , 4F12.6)

C
C      READ IN THE DATA
C
READ 10,A,B,C,ON,F,CI,PI,CA,QO,OI
8887 READ 20,X1,X2,X3,GS,EPS
READ 22,AA,BB,MM,KP

C
C      PRINT OUT THE DATA
C
PRINT 10,A,B,C,ON,F,CI,PI,CA,QO,OI
PRINT 21,X1,X2,X3,GS,EPS
PRINT 22,AA,BB,MM,KPROB

C
C      INITIALIZE THE VARIABLES
C
KPROB=0
ITER=1
NINT=100
II=101
EX(1,II)=X1
EX(2,II)=X2
EX(3,II)=X3

C
C      INITIALIZE VECTOR PI
C
VECZ(1)=X1
VECZ(2)=X2
VECZ(3)=X3

C
C      CALCULATE BOUNDARY CONDITIONS ON ADJOINT VARIABLES AT T=TF
C
AMDA(1,II)=0.

```

AMDA(2,II)=0.

INTEGRATE PROCESS AND ADJOINT DIFF. EQUATIONS BY 4TH ORDER RUNGE
KUTTA METHOD

```

12 DO 1 I=1,NINT
  IT=II+1-I
  AI=IT
  T=(AI-1.)*GS
  AT=T
  X1=EX(1,IT)
  X2=EX(2,IT)
  X3=EX(3,IT)
  AMD1=AMDA(1,IT)
  AMD2=AMDA(2,IT)
  DO 2 J=1,4
    ADT=-AMD1*(1.-X1/ON)/(2.*CA)
    IF(ADT.LE.0.) ADT=0.
    PP(1,J)=-GS*(X1*C+X1*ADT-C*(X1**2)/ON-(X1**2)*ADT/ON)
    PP(2,J)=-GS*(A+B*AT-X1)
    PP(3,J)=-GS*(F*X1-CI*(PI-X2)**2-CA*X1*(ADT)**2)
    PP(4,J)=-GS*(-AMD1*(C+ADT)*(1.-2.*X1/ON)+AMD2+F-CA*(ADT)**2)
    PP(5,J)=-GS*(2.*CI*(PI-X2))
    CJ=1.
    IF(J.EQ.3) CJ=2.
    AT=T-GS*CJ/2.
    X1=EX(1,IT)+PP(1,J)*CJ/2.
    X2=EX(2,IT)+PP(2,J)*CJ/2.
    X3=EX(3,IT)+PP(3,J)*CJ/2.
    AMD1=AMDA(1,IT) +PP(4,J)*CJ/2.
    AMD2=AMDA(2,IT) +PP(5,J)*CJ/2.
  2 CONTINUE
  EX(1,IT-1)=EX(1,IT)+(PP(1,1)+PP(1,2)*2.+2.*PP(1,3)+PP(1,4))/6.
  EX(2,IT-1)=EX(2,IT)+(PP(2,1)+PP(2,2)*2.+2.*PP(2,3)+PP(2,4))/6.
  EX(3,IT-1)=EX(3,IT)+(PP(3,1)+PP(3,2)*2.+2.*PP(3,3)+PP(3,4))/6.
  AMDA(1,IT-1)=AMDA(1,IT)+(PP(4,1)+2.*PP(4,2)+2.*PP(4,3)+PP(4,4))/6.
  AMDA(2,IT-1)=AMDA(2,IT)+(PP(5,1)+2.*PP(5,2)+2.*PP(5,3)+PP(5,4))/6.
1 CONTINUE

```

GENERATE THE CONTROL PROGRAM

```

DO 3 I=1,II
  ADTV(I)=-AMDA(1,I)*(1.-EX(1,I)/ON)/(2.*CA)
  IF(ADTV(I).LE.0.) ADTV(I)=0.
3 CONTINUE

```

CALCULATE THE RESIDUAL VECTOR AND NORM R

```

RESVEC(1)=EX(1,1)-QO
RESVEC(2)=EX(2,1)-OI

```

```

RESVEC(3)=EX(3,1)
RESV=0.
DO 4 I=1,3
4 RESV=RESV+RESVEC(I)**2
RESV=ABS(SQRT(RESV))
PRINT 30,ITER

```

C
C
C

TEST FOR CONVERGANCE

```

IF(RESV.LE.EPS) GO TO 101
PRINT 40,(RESVEC(I),I=1,3),RESV
PRINT 80
DO 91 I=1,II
91 PRINT 90,ADTV(I),(EX(JJ,1),JJ=1,3)

```

C
C
C

COMPUTE SENSITIVITY COEFFICIENTS AT FINAL TIME

```

SENCO(1,II)=1.
SENCO(5,II)=1.
SENCO(9,II)=1.
SENCO(2,II)=0.
SENCO(3,II)=0.
SENCO(4,II)=0.
SENCO(6,II)=0.
SENCO(7,II)=0.
SENCO(8,II)=0.
DO 5 I=10,15
5 SENCO(I,II)=0.

```

C
C
C

SUBROUTINE H1 CALCULATES THE MATRICES A,B,C AND D

```
CALL H1
```

C
C
C
C

INTEGRATE DIFFERENTIAL SENSITIVITY EQUATIONS BACKWARDS BY FOURTH ORDER RUNGE KUTTA METHOD

```

DO 6 I=1,NINT
IT=II+1-I
DO 7 K=1,15
7 SEND(K)=SENCO(K,IT)
DO 8 J=1,4
PP(1,J)=-(A11(IT)*SEND(1)+B11(IT)*SEND(10)) *GS
PP(2,J)=-(A11(IT)*SEND(2)+B11(IT)*SEND(11)) *GS
PP(3,J)=-(A11(IT)*SEND(3)+B11(IT)*SEND(12)) *GS
PP(4,J)=-(A21(IT)*SEND(1))*GS
PP(5,J)=-(A21(IT)*SEND(2))*GS
PP(6,J)=-(A21(IT)*SEND(3))*GS
PP(7,J)=-(A31(IT)*SEND(1)+A32(IT)*SEND(4)+B31(IT)*SEND(10)) *GS
PP(8,J)=-(A31(IT)*SEND(2)+A32(IT)*SEND(5)+B31(IT)*SEND(11)) *GS
PP(9,J)=-(A31(IT)*SEND(3)+A32(IT)*SEND(6)+B31(IT)*SEND(12)) *GS

```

```

PP(10,J)=- (C11(IT)*SEND(1)+D11(IT)*SEND(10)+D12(IT)*SEND(13))*GS
PP(11,J)=- (C11(IT)*SEND(2)+D11(IT)*SEND(11)+D12(IT)*SEND(14))*GS
PP(12,J)=- (C11(IT)*SEND(3)+D11(IT)*SEND(12)+D12(IT)*SEND(15))*GS
PP(13,J)=- (C22(IT)*SEND(4))*GS
PP(14,J)=- (C22(IT)*SEND(5))*GS
PP(15,J)=- (C22(IT)*SEND(6))*GS
CJ=1.
IF(J.EQ.3) CJ=2.
DO 8 K=1,15
8 SEND(K)=SENCO(K,IT)+PP(K,J)*CJ/2.
DO 9 K=1,15
9 SENCO(K,IT-1)=SENCO(K,IT)+(PP(K,1)+2.*PP(K,2)+2.*PP(K,3)+PP(K,4))/
16.
6 CONTINUE

```

C
C
C

CALCULATE THE ELEMENTS OF CONSTRAINED JACOBIAN MATRIX

```

Q(1)=SENCO(1,1)
Q(2)=SENCO(4,1)
Q(3)=SENCO(7,1)
Q(4)=SENCO(2,1)
Q(5)=SENCO(5,1)
Q(6)=SENCO(8,1)
Q(7)=SENCO(3,1)
Q(8)=SENCO(6,1)
Q(9)=SENCO(9,1)
N=3

```

C
C
C

PERFORM MATRIX OPERATIONS FOR BOUNDARY CONDITION ITERATION

```

CALL GMTRA(Q,R,N,N)
CALL MINV(Q,N,D,L,M)
QQQ=ITER-1
QK=(QQQ/(AA +QQQ))*MM
CALL SMPY(Q,QK,RI,N,N,0)
QK1=0.
CALL GMPRD(R,RESVEC,RA,N,N,1)
DO 13 I=1,N
13 QK1=QK1+RA(I)**2
QKI=(1.-QK)*BB /QK1*RESV**2
CALL SMPY(R,QKI,Q,N,N,0)
CALL GMADD(RI,Q,R,N,N)
CALL GMPRD(R,RESVEC,RA,N,N,1)
CALL GMSUB(VECZ,RA,RM,N,1)
DO 11 I=1,N
11 VECZ(I)=RM(I)
EX(1,I)=VECZ(1)
EX(2,I)=VECZ(2)
EX(3,I)=VECZ(3)
ITER=ITER+1

```

SUBROUTINE H1

THIS SUBROUTINE CALCULATES MATRICES A,B,C AND D

```

COMMON/BLOCK/II,EX(3,101),CA,C,ADTV(101),ON,AMDA(2,101),F,CI,PI,
1A11(101),A21(101),A31(101),A32(101),B11(101),B31(101),
2C11(101),C22(101),D11(101),D12(101)
DIMENSION HTT(101)
DO 1 I=1,II
HTT(I)=2.*EX(1,I)*CA
A11(I)=C+ADTV(I)-2.*C*EX(1,I)/ON-2.*ADTV(I)*EX(1,I)/ON-EX(1,I)*(1.
1-EX(1,I)/ON)*(AMDA(1,I)-2.*AMDA(1,I)*EX(1,I)/ON+2.*CA*ADTV(I))/HTT
2(I)
A21(I)=-1.
A31(I)=F-CA*ADTV(I)**2+2.*CA*ADTV(I)*EX(1,I)*(AMDA(1,I)-2.*AMDA(1,
1I)*EX(1,I)/ON+2.*CA*ADTV(I))/HTT(I)
A32(I)=2.*CI*(PI-EX(2,I))
B11(I)=-(EX(1,I)*(1.-EX(1,I)/ON))**2/HTT(I)
B31(I)=2.*CA*ADTV(I)*(EX(1,I)**2*(1.-EX(1,I)/ON)/HTT(I)
C11(I)=2.*AMDA(1,I)*(C+ADTV(I))/ON+(AMDA(1,I)*(1.-2.*EX(1,I)/ON)+2
1.*CA*ADTV(I))**2/HTT(I)
C22(I)=-2.*CI
D11(I)=-A11(I)
D12(I)=-A21(I)
1 CONTINUE
RETURN
END

```

```
      GO TO 12
101 PRINT 60
      PRINT 40,(RESVEC(I),I=1,3),RESV
      PRINT 80
      DO 92 I=1,II
92 PRINT 90,ADTV(I),(EX(JJ,I),JJ=1,3)
      KPROB=KPROB+1
      IF(KPROB.EQ.KP) GO TO 8888
      GO TO 8887
8888 STOP
      END
```

APPENDIX 4

COMPUTER PROGRAM FOR THE PRODUCTION AND ADVERTISEMENT SCHEDULING MODEL

C
C
C

PROBLEM NO. 3--PRODUCTION AND ADVERTISEMENT SCHEDULING

```

COMMON/ARR/KOUNT,ALAG(2,101),AX(2,101)
COMMON/BLOCK/II,GA,EA,GR,TEMP(101),GB,EB,EX(4,101),AMDA(4,101),CU,
1Q,V,C,ADTV(101),OL,CI,A11(101),A12(101),A21(101),A22(101),A32(101)
2,A34(101),A44(101),B11(101),B12(101),B21(101),B22(101),B44(101),
3C11(101),C12(101),C21(101),C22(101),C33(101),C44(101),D11(101),D12
4(101),D22(101),D23(101),D43(101),D44(101),D21(101),HT(101),HTT(101
5),CT,TM,Y(101)
DIMENSION VECZ(4),PP(32,4),RESVEC(4),          Z(101),SE(32,101),SEND
1(32),QQ(16),R(16),RI(16),RA(4),L(16),M(16),RM(4)
10 FORMAT(10F8.4)
14 FORMAT(2F7.1,2E9.3,2F7.3)
15 FORMAT(1H ,5X,'EA  ='F9.1/5X,'EB  ='F9.1/5X,'GA  ='E11.3/5X,'GB
1  ='E11.3/5X,'CT  ='F8.4/5X,'TM  ='F8.4)
20 FORMAT(5F8.4,E10.4,E10.4)
21 FORMAT(1H ,5X,'INITIAL ESTIMATES OF STATE VARIABLES'/5X,'X1(1)='F1
10.4/5X,'X2(1)='F10.4/5X,'X3(1)='F10.4/5X,'X4(1)='F10.4/5X,'GS  ='
2F10.4/5X,'EPS  ='E10.4/5X,'EPSL ='E10.4)
22 FORMAT(2F7.2,2I4,F7.1)
24 FORMAT(1H ,NO. OF ITER TO SOLVE HT =  'I4)
25 FORMAT(1H1,20X,'ADVERTISEMENT AND PRODUCTION PROBLEM '/5X,'C  ='
1F8.4/5X,'C1  ='F8.4/5X,'C2  ='F8.4/5X,'C3  ='F8.4/5X,'N  ='F8
2.4/5X,'V  ='F8.4/5X,'Q  ='F8.4/5X,'X1(0)='F8.4/5X,'X2(0)='F8.4
3/5X,'X3(0)='F8.4/5X,'X4(0)='F8.4/5X,'CU  ='F8.4/5X,'R  ='F8.4/5
4X,'CI  ='F8.4/5X,'IM  ='F8.4)
26 FORMAT(1H ,5X,'AA  ='F7.2/5X,'BB  ='F7.2/5X,'MM  ='I4/5X,'KP
3='I4/5X,'CC  ='F7.1)
30 FORMAT(1H , ' ITERATION NUMBER= 'I6)
40 FORMAT(1H ,5F12.6)
60 FORMAT(1H ,20X,' THESE ARE FINAL ANSWERS ')
80 FORMAT(1H ,4X,'TIME      CONC. OF A  CONC. OF B  INVENTORY      SAL
1ES      ADVERTISEMENT      TEMPRATURE      PROFIT ')
90 FORMAT(1H ,5F12.6,2X,F12.6,8X,F12.6,5X,F12.6)
KPROB=0

```

C
C
C

READ IN THE DATA

```

READ 10,C,C1,C2,C3,OL,V,Q,X10,X20,X30,X40,CU          ,GR,CI,OI
READ 14,EA,EB,GA,GB,CT,TM
8887 READ 20,X1,X2,X3,X4,GS,EPS,EPSL
READ 22,AA,BB,MM,KP,CC

```

C
C
C

PRINT OUT THE DATA

```

PRINT 25,C,C1,C2,C3,OL,V,Q,X10,X20,X30,X40,CU,          GR,CI,OI
PRINT 15,EA,EB,GA,GB ,CT,TM
PRINT 21,X1,X2,X3,X4,GS,EPS,EPSL
PRINT 26,AA,BB,MM,KP      ,CC

```


INITIALIZE THE VARIABLES

```

ITER=1
NINT=100
II=101
EX(1,II)=X1
EX(2,II)=X2
EX(3,II)=X3
EX(4,II)=X4

```

INITIALIZE VECTOR PI

```

VECZ(1)=X1
VECZ(2)=X2
VECZ(3)=X3
VECZ(4)=X4

```

ASSUME THE CONTROL FUNCTION

```

DO 5 I=1,II
5 TEMP(I)=340.00

```

CALCULATE BOUNDARY CONDITIONS ON ADJOINT VARIABLES AT T=TF

```

AMDA(1,II)=0.
AMDA(2,II)=0.
AMDA(3,II)=0.
AMDA(4,II)=0.

```

INTEGRATE PROCESS AND ADJOINT DIFF. EQUATIONS BY 4TH ORDER RUNGE KUTTA METHOD

```

E1=EA/GR
E2=EB/GR
81 KDUNT=1
12 DO 1 I=1,NINT
  IT=II+1-I
  X1=EX(1,IT)
  X2=EX(2,IT)
  X3=EX(3,IT)
  X4=EX(4,IT)
  AMD1=AMDA(1,IT)
  AMD2=AMDA(2,IT)
  AMD3=AMDA(3,IT)
  AMD4=AMDA(4,IT)
  T=TEMP(IT)
  DO 2 J=1,4
    ADT=AMD4*(X4-OL)/(2.*CU*OL*X4)
    IF(ADT.LE.0.0) ADT=0.0
    AK1=GA*EXP(-E1/T)

```

```

AK2=GB*EXP(-E2/T)
PP(1,J)=-GS*(Q*(X10-X1)/V-AK1*X1)
PP(2,J)=-GS*(Q*(X20-X2)/V-AK2*X2+AK1*X1)
PP(3,J)=-GS*(Q*X2-X4)
PP(4,J)=-GS*(X4*(1.-X4/OL)*(C+ADT))
PP(5,J)=-GS*(Q*(AMD1/V+C2-C3)-AK1*(AMD2-AMD1))
PP(6,J)=-GS*(Q*(AMD2/V-AMD3-C3)+AMD2*AK2)
PP(7,J)=-GS*(2.*CI*(OI-X3))
PP(8,J)=-GS*(AMD3+C1-AMD4*(C+ADT)*(1.-2.*X4/OL)-2.*CU*X4*(ADT**2))
CJ=1.

```

```

IF(J.EQ.3) CJ=2.
X1=EX(1,IT)+PP(1,J)*CJ/2.
X2=EX(2,IT)+PP(2,J)*CJ/2.
X3=EX(3,IT)+PP(3,J)*CJ/2.
X4=EX(4,IT)+PP(4,J)*CJ/2.
AMD1=AMDA(1,IT)+PP(5,J)*CJ/2.
AMD2=AMDA(2,IT)+PP(6,J)*CJ/2.
AMD3=AMDA(3,IT)+PP(7,J)*CJ/2.
AMD4=AMDA(4,IT)+PP(8,J)*CJ/2.

```

```

2 CONTINUE

```

```

EX(1,IT-1)=EX(1,IT)+(PP(1,1)+PP(1,2)*2.+2.*PP(1,3)+PP(1,4))/6.
EX(2,IT-1)=EX(2,IT)+(PP(2,1)+PP(2,2)*2.+2.*PP(2,3)+PP(2,4))/6.
EX(3,IT-1)=EX(3,IT)+(PP(3,1)+PP(3,2)*2.+2.*PP(3,3)+PP(3,4))/6.
EX(4,IT-1)=EX(4,IT)+(PP(4,1)+PP(4,2)*2.+2.*PP(4,3)+PP(4,4))/6.

```

```

8886 AMDA(1,IT-1)=AMDA(1,IT)+(PP(5,1)+2.*PP(5,2)+2.*PP(5,3)+PP(5,4))/6.
AMDA(2,IT-1)=AMDA(2,IT)+(PP(6,1)+2.*PP(6,2)+2.*PP(6,3)+PP(6,4))/6.
AMDA(3,IT-1)=AMDA(3,IT)+(PP(7,1)+2.*PP(7,2)+2.*PP(7,3)+PP(7,4))/6.
AMDA(4,IT-1)=AMDA(4,IT)+(PP(8,1)+2.*PP(8,2)+2.*PP(8,3)+PP(8,4))/6.

```

```

1 CONTINUE

```

```

CALL H2

```

C
C
C

```

TEST IF THE PRESENT CONTROL SATISFIES HT=0

```

```

DO 16 I=1,II
IF(ABS(HT(I)).GT.EPSL) GO TO 17

```

```

16 CONTINUE

```

```

GO TO 18

```

```

17 CONTINUE

```

```

CALL H3

```

```

CALL H4

```

C
C
C

```

IMPROVE THE CONTROL BY DIRECT SECOND VARIATION METHOD

```

```

QQQ=KOUNT
STEP=QQQ/(CC+QQQ)

```

```

DO 19 I=1,II

```

```

19 TEMP(I)=TEMP(I)-STEP*(HT(I)+Y(I))/HTT(I)

```

```

KOUNT=KOUNT+1

```

```

DO 23 I=1,II

```

```

AX(1,I)=EX(1,I)

```

```

      AX(2,I)=EX(2,I)
      ALAG(1,I)=AMDA(1,I)
      ALAG(2,I)=AMDA(2,I)
23  CONTINUE
      GO TO 12

```

C
C
C

GENERATE ADVERTISEMENT RATE

```

18  DO 3 I=1,II
      ADTV(I)=AMDA(4,I)*(EX(4,I)-OL)/(2.*CU*OL*EX(4,I))
      IF(ADTV(I).LE.0.0) ADTV(I)=0.0
3   CONTINUE
      PRINT 24,KOUNT

```

C
C
C

CALCULATE THE RESIDUAL VECTOR

```

      RESVEC(1)=EX(1,1)-X10
      RESVEC(2)=EX(2,1)-X20
      RESVEC(3)=EX(3,1)-X30
      RESVEC(4)=EX(4,1)-X40
      RESV=0.
      DO 4 I=1,4
4   RESV=RESV+RESVEC(I)**2
      RESV=ABS(SQRT(RESV))
      PRINT 30,ITER

```

C
C
C

TEST FOR CONVERGENCE OF THE PROBLEM

```

      IF(RESV.EQ.EPS) GO TO 101
      PRINT 40,(RESVEC(I),I=1,4),RESV
      PRINT 80

```

C
C
C

CALCULATE THE PROFIT

```

      DO 31 I=1,II
31  Y(I)=C1*EX(4,I)+C2*Q*EX(1,I)+C3*Q*(1.-EX(1,I)-EX(2,I))-CI*(OI-EX(3
      I,I))**2-CU*(ADTV(I)*EX(4,I))**2-CT*(TM-TEMP(I))**2
      CALL QSF(GS,Y,Z,II)
      DO 91 I=1,II,10
      AI=I
      TIME=(AI-1.)*GS
91  PRINT 90,TIME,(EX(JJ,I),JJ=1,4),ADTV(I),TEMP(I),Z(I)

```

C
C
C

CALCULATE BOUNDARY CONDITIONS OF SENSITIVITY COEFFICIENTS

```

      DO 51 I=1,16,5
51  SE(I,II)=1.0
      DO 52 I=2,5
      SE(I,II)=0.0
      SE(I+5,II)=0.0

```

```

      SE(I+10,II)=0.0
52 SE(I+27,II)=0.0
      DO 53 I=17,28
53 SE(I,II)=0.0
      CALL H3
      CALL H1

```

C
C
C
C

INTEGRATE SENSITIVITY EQUATIONS BY FOURTH ORDER RUNGE KUTT METHOD

```

      DO 6 I=1,NINT
      IT=II+1-I
      DO 7 K=1,32
7 SEND(K)=SE (K,IT)
      DO 8 J=1,4
      PP(1,J)=-GS*(A11(IT)*SEND(1)+A12(IT)*SEND(5)+B11(IT)*SEND(17)+B12(
1IT)*SEND(21))
      PP(2,J)=-GS*(A11(IT)*SEND(2)+A12(IT)*SEND(6)+B11(IT)*SEND(18)+B12(
1IT)*SEND(22))
      PP(3,J)=-GS*(A11(IT)*SEND(3)+A12(IT)*SEND(7)+B11(IT)*SEND(19)+B12(
1IT)*SEND(23))
      PP(4,J)=-GS*(A11(IT)*SEND(4)+A12(IT)*SEND(8)+B11(IT)*SEND(20)+B12(
1IT)*SEND(24))
      PP(5,J)=-GS*(A21(IT)*SEND(1)+A22(IT)*SEND(5)+B21(IT)*SEND(17)+B22(
1IT)*SEND(21))
      PP(6,J)=-GS*(A21(IT)*SEND(2)+A22(IT)*SEND(6)+B21(IT)*SEND(18)+B22(
1IT)*SEND(22))
      PP(7,J)=-GS*(A21(IT)*SEND(3)+A22(IT)*SEND(7)+B21(IT)*SEND(19)+B22(
1IT)*SEND(23))
      PP(8,J)=-GS*(A21(IT)*SEND(4)+A22(IT)*SEND(8)+B21(IT)*SEND(20)+B22(
1IT)*SEND(24))
      PP(9,J)=-GS*(A32(IT)*SEND(5)+A34(IT)*SEND(13))
      PP(10,J)=-GS*(A32(IT)*SEND(6)+A34(IT)*SEND(14))
      PP(11,J)=-GS*(A32(IT)*SEND(7)+A34(IT)*SEND(15))
      PP(12,J)=-GS*(A32(IT)*SEND(8)+A34(IT)*SEND(16))
      PP(13,J)=-GS*(A44(IT)*SEND(13)+B44(IT)*SEND(29))
      PP(14,J)=-GS*(A44(IT)*SEND(14)+B44(IT)*SEND(30))
      PP(15,J)=-GS*(A44(IT)*SEND(15)+B44(IT)*SEND(31))
      PP(16,J)=-GS*(A44(IT)*SEND(16)+B44(IT)*SEND(32))
      PP(17,J)=-GS*(C11(IT)*SEND(1)+C12(IT)*SEND(5)+D11(IT)*SEND(17)+D12
1(IT)*SEND(21))
      PP(18,J)=-GS*(C11(IT)*SEND(2)+C12(IT)*SEND(6)+D11(IT)*SEND(18)+D12
1(IT)*SEND(22))
      PP(19,J)=-GS*(C11(IT)*SEND(3)+C12(IT)*SEND(7)+D11(IT)*SEND(19)+D12
1(IT)*SEND(23))
      PP(20,J)=-GS*(C11(IT)*SEND(4)+C12(IT)*SEND(8)+D11(IT)*SEND(20)+D12
1(IT)*SEND(24))
      PP(21,J)=-GS*(C21(IT)*SEND(1)+C22(IT)*SEND(5)+D21(IT)*SEND(17)+D22
1(IT)*SEND(21)+D23(IT)*SEND(25))
      PP(22,J)=-GS*(C21(IT)*SEND(2)+C22(IT)*SEND(6)+D21(IT)*SEND(18)+D22

```

```

1(IT)*SEND(22)+D23(IT)*SEND(26))
  PP(23,J)=-GS*(C21(IT)*SEND(3)+C22(IT)*SEND(7)+D21(IT)*SEND(19)+D22
1(IT)*SEND(23)+D23(IT)*SEND(27))
  PP(24,J)=-GS*(C21(IT)*SEND(4)+C22(IT)*SEND(8)+D21(IT)*SEND(20)+D22
1(IT)*SEND(24)+D23(IT)*SEND(28))
  PP(25,J)=-GS*C33(IT)*SEND(9 )
  PP(26,J)=-GS*C33(IT)*SEND(10)
  PP(27,J)=-GS*C33(IT)*SEND(11)
  PP(28,J)=-GS*C33(IT)*SEND(12)
  PP(29,J)=-GS*(D44(IT)*SEND(29)+D43(IT)*SEND(25)+C44(IT)*SEND(13))
  PP(30,J)=-GS*(D44(IT)*SEND(30)+D43(IT)*SEND(26)+C44(IT)*SEND(14))
  PP(31,J)=-GS*(D44(IT)*SEND(31)+D43(IT)*SEND(27)+C44(IT)*SEND(15))
  PP(32,J)=-GS*(D44(IT)*SEND(32)+D43(IT)*SEND(28)+C44(IT)*SEND(16))
  CJ=1.
  IF(J.EQ.3) CJ=2.
  DO 8 K=1,32
8 SEND(K)=SE (K,IT)+PP(K,J)*CJ/2.
  DO 9 K=1,32
9 SE (K,IT-1)=SE (K,IT)+{PP(K,1)+2.*PP(K,2)+2.*PP(K,3)+PP(K,4)}/
  16.
6 CONTINUE

```

C
C
C

CALCULATE CONSTRAINED JACOBIAN MATRIX

```

QQ(1 )=SE(1 ,1)
QQ(2 )=SE(5 ,1)
QQ(3 )=SE(9 ,1)
QQ(4 )=SE(13,1)
QQ(5 )=SE(2 ,1)
QQ(6 )=SE(6 ,1)
QQ(7 )=SE(10,1)
QQ(8 )=SE(14,1)
QQ(9 )=SE(3 ,1)
QQ(10)=SE(7 ,1)
QQ(11)=SE(11,1)
QQ(12)=SE(15,1)
QQ(13)=SE(4 ,1)
QQ(14)=SE(8 ,1)
QQ(15)=SE(12,1)
QQ(16)=SE(16,1)
N=4

```

C
C
C

PERFORM MATRIX OPERATIONS FOR BOUNDARY CONDITION ITERATION

```

CALL GMTRA(QQ,R,N,N)
CALL MINV(QQ,N,D,L,M)
QQQ=ITER-1
QK=(QQQ/(AA +QQQ))*MM
CALL SMPY(QQ,QK,RI,N,N,0)
QK1=0.

```

```

      CALL GMPRD(R,RESVEC,RA,N,N,1)
      DO 13 I=1,N
13    QK1=QK1+RA(I)**2
      QKI=(1.-QK)*BB /QK1*RESV**2
      CALL SMPY(R,QKI,QQ,N,N,0)
      CALL GMADD(RI,QQ,R,N,N)
      CALL GMPRD(R,RESVEC,RA,N,N,1)
      CALL GMSUB(VECZ,RA,RM,N,1)
      DO 11 I=1,N
11    VECZ(I)=RM(I)
      EX(1,II)=VECZ(1)
      EX(2,II)=VECZ(2)
      EX(3,II)=VECZ(3)
      EX(4,II)=VECZ(4)
      ITER=ITER+1
      GO TO 81
101  PRINT 60
      PRINT 40,(RESVEC(I),I=1,4),RESV
      PRINT 80
      DO 102 I=1,II
102  Y(I)=C1*EX(4,I)+C2*Q*EX(1,I)+C3*Q*(1.-EX(1,I)-EX(2,I))-CI*(DI-EX(
      1,I))**2-CU*(ADTV(I)*EX(4,I))**2-CT*(TM-TEMP(I))**2
      CALL QSF(GS,Y,Z,II)
      DO 92 I=1,II,10
      AI=I
      TIME=(AI-1.)*GS
92  PRINT 90,TIME,(EX(JJ,I),JJ=1,4),ADTV(I),TEMP(I),Z(I)
      KPROB=KPROB+1
      IF(KPROB.EQ.KP) GO TO 8888
      GO TO 8887
8888 STOP
      END

```

SUBROUTINE H1

C
C
C

SUBROUTINE H1 CALCULATES COMPONENTS OF MATRICES A,B,C,D

```

COMMON/BLOCK/II,GA,EA,GR,TEMP(101),GB,EB,EX(4,101),AMDA(4,101),CU,
1Q,V,C,ADTV(101),OL,CI,A11(101),A12(101),A21(101),A22(101),A32(101)
2,A34(101),A44(101),B11(101),B12(101),B21(101),B22(101),B44(101),
3C11(101),C12(101),C21(101),C22(101),C33(101),C44(101),D11(101),D12
4(101),D22(101),D23(101),D43(101),D44(101),D21(101),HT(101),HTT(101
5),CT,TM,Y(101)
DIMENSION HAA(101)
DO 1 I=1,II
AK1=GA*EXP(-EA/(GR*TEMP(I)))
AK2=GB*EXP(-EB/(GR*TEMP(I)))
DENOM=(GR*TEMP(I)**2)**2
HAA(I)= 2.*CU*(EX(4,I)**2)
DENOM=DENOM*HTT(I)
A11(I)=-(Q/V+AK1)+EX(1,I)*((AK1*EA)**2)*(AMDA(2,I)-AMDA(1,I))/DENO
1M
A12(I)=-EX(1,I)*AMDA(2,I)*AK1*AK2*EA*EB /DENOM
GREG=EX(1,I)*EA*AK1-EX(2,I)*EB*AK2
A21(I)=AK1-AK1*EA*(AMDA(2,I)-AMDA(1,I))*GREG/DENOM
A22(I)=-(Q/V+AK2)+AMDA(2,I)*AK2*EB*GREG/DENOM
A32(I)=Q
A34(I)=-1.0
A44(I)=(C+ADTV(I))*(1.-2.*EX(4,I)/OL)-EX(4,I)*(1.-EX(4,I)/OL)*(AMD
1A(4,I)*(1.-2.*EX(4,I)/OL)+4.*CU*ADTV(I)*EX(4,I))/HAA(I)
B11(I)=-((EX(1,I)*AK1*EA)**2)/DENOM
B12(I)=EX(1,I)*AK1*EA*GREG/DENOM
B21(I)=B12(I)
B22(I)=-(GREG**2)/DENOM
B44(I)=-((EX(4,I)-(EX(4,I)**2)/OL)**2)/HAA(I)
C11(I)=((AMDA(2,I)-AMDA(1,I))*AK1*EA)**2)/DENOM
C12(I)=-AMDA(2,I)*AK1*AK2*EA*EB*(AMDA(2,I)-AMDA(1,I))/DENOM
C21(I)=C12(I)
C22(I)=((AMDA(2,I)*AK2*EB)**2)/DENOM
C33(I)=-2.*CI
C44(I)=2.*AMDA(4,I)*(C+ADTV(I))/OL-2.*CU*(ADTV(I)**2)+((AMDA(4,I)*
1(1.-2.*EX(4,I)/OL)+4.*CU*ADTV(I)*EX(4,I))**2)/HAA(I)
D11(I)=-A11(I)
D12(I)=-A21(I)
D21(I)=-A12(I)
D22(I)=-A22(I)
D23(I)=-A32(I)
D43(I)=-A34(I)
1 D44(I)=-A44(I)
RETURN
END

```

SUBROUTINE H2

C
C
C

SUBROUTINE H2 CALCULATES HT%T<

```

COMMON/BLOCK/II,GA,EA,GR,TEMP(101),GB,EB,EX(4,101),AMDA(4,101),CU,
1Q,V,C,ADTV(101),OL,CI,A11(101),A12(101),A21(101),A22(101),A32(101)
2,A34(101),A44(101),B11(101),B12(101),B21(101),B22(101),B44(101),
3C11(101),C12(101),C21(101),C22(101),C33(101),C44(101),D11(101),D12
4(101),D22(101),D23(101),D43(101),D44(101),D21(101),HT(101),HTT(101
5),CT,TM,Y(101)
DO 1 I=1,II
1 HT(I)=GA*EXP(-EA/(GR*TEMP(I)))*EX(1,I)*EA*(AMDA(2,I)-AMDA(1,I))/(G
1R*TEMP(I)**2)-GB*EXP(-EB/(GR*TEMP(I)))*EB*EX(2,I)*AMDA(2,I)/(GR*TE
2MP(I)**2)-2.*CT*(TM-TEMP(I))
RETURN
END

```

SUBROUTINE H3

C
C
C

SUBROUTINE H3 CALCULATES HTT%T<

```

COMMON/BLOCK/II,GA,EA,GR,TEMP(101),GB,EB,EX(4,101),AMDA(4,101),CU,
1Q,V,C,ADTV(101),OL,CI,A11(101),A12(101),A21(101),A22(101),A32(101)
2,A34(101),A44(101),B11(101),B12(101),B21(101),B22(101),B44(101),
3C11(101),C12(101),C21(101),C22(101),C33(101),C44(101),D11(101),D12
4(101),D22(101),D23(101),D43(101),D44(101),D21(101),HT(101),HTT(101
5),CT,TM,Y(101)
DO 1 I=1,II
HTT(I)=-2.*HT(I)/TEMP(I)+GA*EXP(-EA/(GR*TEMP(I)))*EX(1,I)*EA*EA*(A
1MDA(2,I)-AMDA(1,I))/((GR*TEMP(I)**2)**2)+2.*CT-GB*EXP(-EB/(GR*TEMP
2(I)))*EB*EB*EX(2,I)*AMDA(2,I)/((GR*TEMP(I)**2)**2)-4.*CT*(TM-TEMP(
3I))/TEMP(I)
1 CONTINUE
RETURN
END

```


SUBROUTINE H4

C
C
C

SUBROUTINE H4 CALCULATES Y%IK REQUIRED FOR DSV METHOD

```

COMMON/ARR/KOUNT,ALAG(2,101),AX(2,101)
COMMON/BLOCK/II,GA,EA,GR,TEMP(101),GB,EB,EX(4,101),AMDA(4,101),CU,
1Q,V,C,ADTV(101),OL,CI,A11(101),A12(101),A21(101),A22(101),A32(101)
2,A34(101),A44(101),B11(101),B12(101),B21(101),B22(101),B44(101),
3C11(101),C12(101),C21(101),C22(101),C33(101),C44(101),D11(101),D12
4(101),D22(101),D23(101),D43(101),D44(101),D21(101),HT(101),HTT(101
5),CT,TM,Y(101)
IF(KOUNT.EQ.1) GO TO 2
DO 1 I=1,II
AK1=GA*EXP(-EA/(GR*TEMP(I)))
AK2=GB*EXP(-EB/(GR*TEMP(I)))
Y(I)=AK1*EA*(AMDA(2,I)-AMDA(1,I))/(GR*TEMP(I)**2)*(EX(1,I)-AX(1,I)
1)-AMDA(2,I)*AK2*EB*(EX(2,I)-AX(2,I))/(GR*TEMP(I)**2)-EX(1,I)*AK1*E
2A*(AMDA(1,I)-ALAG(1,I))/(GR*TEMP(I)**2)+(EX(1,I)*AK1*EA-EX(2,I)*AK
32*EB)*(AMDA(2,I)-ALAG(2,I))/(GR*TEMP(I)**2)
1 CONTINUE
RETURN
2 DO 3 I=1,II
3 Y(I)=0.0
RETURN
END

```

OPTIMIZATION OF MANAGEMENT SYSTEMS

BY SENSITIVITY ANALYSIS

by

RAVINDRA KUMAR

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

Bombay, India, 1966

AN ABSTRACT OF 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

ABSTRACT

It is by now well known that many of the problems involved in the optimization of complex non-linear dynamic systems are of computational nature. Dynamic programming suffers from the dimensionality difficulty while the two-point boundary value difficulty limits the use of calculus of variations and maximum principle. The methods of steepest descent and other techniques like quasilinearization overcome the two-point boundary value difficulty in many cases.

Differential sensitivity analysis is a recently developed technique which also helps overcome the two-point boundary value difficulty in complex dynamic systems. It is a boundary condition iteration technique and has been successfully used in many complex problems.

The purpose of this work is to investigate the computational features of this technique in solving various management problems.

First, the method of differential sensitivity analysis is discussed. Then its application to three problems in the field of production and inventory control is discussed in detail.

The first is a simple inventory model with one state variable and one control variable. The second application is an inventory and advertising model with two state variables and one control variable. The last application is that of a production and advertisement scheduling problem having four state variables and two control variables. On the basis of computational experience with these examples it was concluded that:

1. Convergence is not contingent upon the initial approximations of the state variables in most of the problems.

2. Convergence rate depends on the choice of parameters used to control the iteration cycle. However, in most problems it is quite rapid.
3. The number of equations to be integrated increases rapidly as the number of state variables increase and this tends to suppress its advantage of being able to converge from a crude estimate of the state variables.

The merits and demerits of this technique in comparison to other techniques should be evaluated in the light of the nature of application and the type of problem. However, it is a useful technique for the systems analyst for attacking optimization problems.