OPTIMIZATION
OF
WATER RESOURCES

by 500

PRATAP DHANJI DAND

B.Tech., Indian Institute of Technology
Bombay, India, 1967

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

1969

Approved by:

E. Stanley Lee

Major Professor

ACKNOWLEDGEMENT

The author wishes to express his sincere gratitude to his Major Professor, Dr. E. S. Lee, for his guidance and helpful assistance during the preparation of this thesis and for the time and energy devoted for counsel.

He also wishes to express his appreciation to Dr. L. E. Erickson, Professor of Chemical Engineering for his suggestions in this work.

This work was supported by U. S. Department of the Interior, Office of Water Resources Research Grant 14-01-0001-1962.

TABLE OF CONTENTS

LIST OF FIGURE

LIST OF TABLES

TABLE | PAGE

CHAPTER 1

INTRODUCTION AND STATEMENT OF PURPOSE

New methods for design and management of water-resource systems are being evolved as part of a general social tendency toward expressing social problems in the formal models that have hitherto been restricted to scientific and engineering problems.

Two general types of models have been fruitful in the field of water-resource development: the simulation model and the analytic model. Simulation is used for highly complex systems involving a large number of design decisions. For less complicated systems, various optimization techniques can be used to obtain the optimal conditions. Simulations are awkward when a wide range of design decisions has to be evaluated; analytic models can not be applied to practical problems without drastically simplifying them. But the two methods can be used in tandem, with analytic models delimiting the range within which simulation is required.

Two basic problems in the analysis of water-resource systems are: the problem of obtaining a realistic mathematical model and the problem of obtaining the best management policy from an analysis of the resulting mathematical model.

Of the different analytical approaches, dynamic programming has been quite useful in optimization of water-resource systems. Linear programming is not generally useful, because individual reservoir utility or cost functions are in general definitely nonlinear. In this work, a nonlinear programming approach is considered.

Significant progress has been made in treating the general nonlinear programming problem during the last few years. The different approaches are generally a combination of gradient or search techniques with a method for treating constraints. The specific purpose of this research is to apply a recently developed optimization technique - Rosen's gradient projection method [24] - to the optimal management and design of water-resource systems.

The gradient projection method is described in Chapter 2. A rather nontheoretical approach is attempted in describing various aspects of the method. The emphasis is on the applicatiion of the method and not on the complex mathematical details.

Two test systems - one connected with water-quality and other connected with water-quantity - are proposed and solved within the framework of an analytical approach.

The basic principles involved in the construction of a mathematical model of dissolved oxygen in a simplified river basin are presented in Chapter 3. An attempt has been made to indicate how the mathematical model can be used to generate useful water quality control information.

Chapter 4 deals with water quantity aspects. A very simplified river basin configuration is selected to illustrate the mathematical programming approach. An optimal plan for development of this system is evolved.

CHAPTER 2

THE GRADIENT PROJECTION METHOD

2.1 INTRODUCTION

In the fields of nonlinear optimization there are many methods [29] which may be used successfully on unconstrained problems. However, most practical problems involve constraints, both general and specific, which must be satisfied in order to obtain meaningful results.

Rosen [24] developed the gradient projection (GP) method for solving the subclass of nonlinear programming problems characterized by linear constraints. He extended this work to handle the more difficult case of nonlinear constraints [15]. The discussion here is confined to the simpler case of linear constraints only.

Briefly, GP is a method for a system of m variables (x's) subject to linear constraints which may consist of inequalities, equalities or both. The objective is to find the maximum value of the objective function while satisfying the constraints. In geometric terms, GP starts with a feasible point (one that satisfies all the constraints) or finds one, if not given, and then a stepwise procedure gives a new feasible point at each step with an increased value of the objective function. This is accomplished by taking steps in the direction of the gradient of the objective function or its projection on the intersection of selected constraints. The maximum is found when any step that would increase the value of the objective function would violate the constraints, or when the gradient goes to zero.

2.2 FORMULATION FOR LINEAR CONSTRAINTS [24]

A problem with m variables, x_i, $i = 1, 2, \ldots, m$, is considered. Geometrically, any specified set of values for the x_i represents a point in a Euclidean m-dimensional space, E_m. It is assumed that the variables are constrained by a set of k linear inequalities or equalities. These constraints form a convex region R in E_m which is assumed to be bounded. A feasible point lies in R. The constraints are of the form

$$\sum_{j=1}^{m} n_{ij}x_j - b_i \geq 0 \qquad i = 1, 2, \ldots, k \tag{1}$$

where the n_{ij} have been normalized, so that

$$\sum_{j=1}^{m} (n_{ij})^2 = 1 \qquad i = 1, 2, \ldots, k \tag{2}$$

Since R is a bounded region there must be at least m+1 constraints, so that $k \geq m+1$.

Corresponding to each of the k constraints a vector n_i is defined

$$n_i = \left\{ n_{i1}, n_{i2}, \ldots, n_{im} \right\} \qquad i = 1, 2, \ldots, k \tag{3}$$

These are unit vectors. The inequalities (1) can now be written in the form

$$x^T n_i - b_i = \lambda_i(x) \geq 0 \qquad i = 1, 2, \ldots, k \tag{4}$$

The (m-1) dimensional manifold defined by $\lambda_i(x) = 0$ is a hyperplane which will be denoted by H_i,

$$H_i \;:\; \lambda_i(x) = 0 \qquad i = 1, 2, \ldots, k \tag{5}$$

The unit vector n_i is orthogonal to H_i. A set of hyperplanes are linearly independent if the corresponding vectors n_i are linearly independent. Clearly there can be at most m linearly independent hyperfplanes in an m-dimensional space.

Define the m x k matrix

$$N_k = [n_1, n_2, \ldots, n_k] \tag{6}$$

and the k-dimensional vector

$$b_k = \{b_1, b_2, \ldots, b_k\} \tag{7}$$

Then (1) or (4) can be written conveniently as

$$N_k^T x - b_k \geq 0 \tag{8}$$

A set of q linearly independent unit vectors n_i, $i = 1, 2, \ldots, q$, with $1 \leq q \leq m$, defines a set of q linearly independent hyperplanes H_i as given by (4) and (5) for any specified values of the b_i. Let

$$N_q = [n_1, n_2, \ldots, n_q] \tag{9}$$

It can be shown that the q x q symmetric matrix $N_q^T N_q$ is nonsingular [10] and therefore its inverse $(N_q^T N_q)^{-1}$ exists. Let Q denote the intersection of the q hyperplanes H_i and let $\overline{Q}$ be the q-dimensional subspace of E_m which is spanned by $n_1, n_2, \ldots, n_q$. Since $E_m = Q + \overline{Q}$, Q is an (m-q) dimensional subspace. The subspaces Q and $\bar{Q}$ are orthogonal.

Define the m x m symmetric matrix

$$\bar{P}_q = N_q (N_q^T N_q)^{-1} N_q^T \tag{10}$$

The matrix $\bar{P}_q$ is a projection matrix which takes any vector in E_m into $\bar{Q}$.

An m x m matrix is now defined by

$$P_q = I - \bar{P}_q \tag{11}$$

The matrix P_q is a projection matrix which takes any vector in E_m into the intersection Q.

In course of an optimization calculation, it is necessary to obtain the projection of the gradient vector on various intersections Q. In other words, the matrix $(N^T N)^{-1}$ is required at each step. Rosen [24] derived two recursion relations which permit a hyperplane to be dropped from or added to $(N^T N)^{-1}$ with considerably less computation. A simple and very useful recursion relation for P_q is also given.

2.3 CONSTRAINED MAXIMUM AND CONVERGENCE [14]

The objective function

$$F(x) = F(x_1, x_2, \ldots, x_n) \tag{12}$$

is defined in R, and is assumed to have continuous and bounded second partial derivatives with respect to the x_i. Also $F(x)$ is a concave function in R. It is desired to find a point x_{max} in

R at which $F(x)$ has a global maximum. If the point x_{max} is on the boundary of R, it is a constrained maximum.

The gradient of $F(x)$ will be denoted by $g(x)$ and is defined by

$$g(x) = \frac{\partial F}{\partial x_1}, \frac{\partial F}{\partial x_2}, \ldots, \frac{\partial F}{\partial x_m} \tag{13}$$

It is well known that for an unconstrained concave function the necessary and sufficient condition that x_o be the maximum point is $g(x_o) = 0$. If such a point exists in the interior of R, it will be called the interior global maximum of $F(x)$. If $g(x)$ does not vanish in the interior of R, the global maximum lies on the boundary and will be called a constrained global maximum. The basic theorem concerning a constrained global maximum is as follows

Theorem:

Let x_o be a boundary point of R which lies on exactly q, $1 \leq q \leq m$, hyperplanes, which are assumed to be linearly independent. Let the intersection of these hyperplanes be the manifold Q. Then the point x_o is a constrained global maximum of $F(x)$ if, and only if,

$$P_q \, g(x_o) = 0 \tag{14}$$

and

$$(N_q^T N_q)^{-1} N_q^T g(x_o) \leq 0 \tag{15}$$

Rosen [14] has also given a theorem to establish the convergence to a global maximum of $F(x)$. For the rigorous mathematical treatment of these aspects of constrained maximum and convergence, the interested reader can consult the original reference [24].

2.4 ALGORITHM [24]

Certain quantities used in the algorithm are defined here without going into complex mathematical details. Let

$$g_0 = g(x_0) \tag{16}$$

$\lambda_i(x_0)$ = normal distance to a constraint

$$r = \left\{r_1,\ r_2,\ \ldots,\ r_q\right\} = (N_q^T N_q)^{-1} N_q^T\, g_0 \tag{17}$$

$$z = \frac{P_q\ g_0}{\|P_q\ g_0\|} = \text{unit vector in direction of step} \tag{18}$$

Υ = selected step size

Now consider a feasible point x_0 which lies on the q hyperplanes in Q, H_i, $i = 1, \ldots, q$. Since it is a feasible point, $\lambda_i(x) > 0$, $i = q+1, \ldots, k$. Then for each of the remaining n-q hyperplanes, H_i, $i = q+1, \ldots, k$, there may exist a value $\Upsilon = \Upsilon_i$ such that $\lambda_i(x) = 0$. Υ_i is the distance from x_0 to the hyperplane H_i along a parallel to z. In particular,

$$\Upsilon_i = \frac{\lambda_i(x_0)}{z^T n_i} \qquad i = q+1, \ldots, k \tag{19}$$

Let Υ_j be the minimum quantity chosen from the set of Υ_i which are positive.

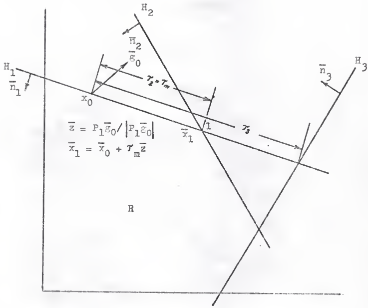


Fig. 1. Determination of Step Length [24]

$$\Upsilon_m = \min\left\{\Upsilon_i > 0\right\} \qquad i = q+1, \ldots, k \tag{20}$$

The distance Υ_m represents the largest step that can be taken from x_0 in the direction z without leaving R (see Fig. 1). If

$$x = x_0 + \Upsilon z \tag{21}$$

then it follows that for any Υ, $0 \leq \Upsilon \leq \Upsilon_m$, x is in R.

The algorithm stated below is for a current, arbitary point x_v in R which lies on the manifold Q formed by the intersection of q linearly independent hyperplanes.

Step 1.

Compute $g_v = g(x_v)$ and $P_q g_v$. If $P_q g_v = 0$ and $r \leq 0$, where r is given by (17), then x_v is a global maximum.

Step 2.

Now, either $\|P_q g_v\| > \max\left\{0, \frac{1}{2} r_q d_{qq}^{-\frac{1}{2}}\right\}$ or $\|P_q g_v\| \leq \frac{1}{2} r_q d_{qq}^{-\frac{1}{2}}$, where $r_q d_{qq}^{-\frac{1}{2}} \geq r_i d_{ii}^{-\frac{1}{2}}$, $i = 1, 2, \ldots, q-1$, and where d_{ii} is the, i^{th} diagonal element of $(N_q^T N_q)^{-1}$. If the former holds, compute z according to (18). If the latter holds, drop H_q from Q and obtain N_{q-1} and the corresponding P_{q-1}. Compute

$$z = \frac{P_{q-1} g_v}{|P_{q-1} g_v|}$$

Step 3.

Compute Υ_m from (19) and (20) using the appropriate z from step 2. Let $x'_{v+1} = x_v + \Upsilon_m z$.

Step 4.

Compute $g'_{v+1} = g(x'_{v+1})$. If $z^T g'_{v+1} \geq 0$, take $x_{v+1} = x'_{v+1}$ and add to Q the H_i corresponding to the minimum τ_i in (20).

Step 5.

If $z^T g'_{v+1} < 0$, let

$$\rho = \frac{z^T g_v}{(z^T g_v - z^T g'_{v+1})} \tag{22}$$

and

$$x_{v+1} = \rho x'_{v+1} + (1 - \rho) x_v \tag{23}$$

The intersection Q remains unchanged.

This is the basic algorithm for a single step from a point x_v (if it is not maximum) to a new point x_{v+1} with an increased value of $F(x)$. At each step, the intersection Q may remain unchanged, a hyperplane may be added or dropped, or one may be dropped and another added. The last step of algorithm is the orthogonal gradient interpolation.

2.5 PROGRAM FLOW

A general computer program known as GP 90 [21] was developed for the purpose of solving a series of engineering design and economic problems. A complete flow chart of the program is given in the Appendix B. This includes the extrapolation methods, matrix computations, re-inversion logic etc.

A brief description of the main program and subroutines follows the explanation of tolerences and limits necessary for the program.

2.5.1 Tolerances and Limits

The gradient tolerance, ε_1, is used to determine when the norm of the gradient is zero, and the problem has reached a maximum. The value of ε_1 is harder to determine for a nonlinear function. In general, the smaller the value of ε_1, the better the "maximum" will be. The price paid for this better answer will be in more machine time. A value of ε_1 about $10^{-3}\ \|g\|$ seems to be reasonable.

The constraint tolerance, ε_2, determines when a point is on a constraint, and is therefore the acceptable error in satisfying the constraints. A value of ε_2 about $10^{-3}\ b_i'$, where b_i' is the largest right hand side of constraints, b_i, divided by the corresponding scaling divisor or 10^{-3} times the largest value of x seems to be reasonable.

The linear dependence tolerance, ε_3, determines when a constraint is linearly dependent and therefore can not be added to the basis. The general value used for ε_3 is 0.005.

τ_{max} is the maximum step length which is used as the initial step length and must be provided when the region is unbounded. A possible choice for τ_{max} is $L\sqrt{m}$, where L is the largest value that any individual x can assume. A value of $\tau_{max} = 10$ is found reasonable for quadratic functions. The program computes a minimum step length, $\tau_{min} = 10^{-4}\ \tau_{max}$, which prevents some interpolating or extrapolating when the distance between the points is less than τ_{min}.

The five limits which must be set are now summarized. The settings for these limits will generally depend on the size and type of problem being solved.

MAXU, maximum number of steps. The number of steps required to reach the maximum is difficult to pre-determine since it depends on such factors as size, type of function, and number of constraints in the basis. It keeps the run to a reasonable length without having to rely on a time limit.

MXRN, maximum number of re-inversions. The number of re-inversions is limited to prevent the problem from re-inverting too often thereby consuming extra time. The program will not re-invert twice in a row to the same basis, but it may re-invert after only one basis change or possibly repeat a series of re-inversions.

β_{max} and γ_{max} limit the number of gradient interpolations which are computed in finding an x for which F has increased. For a quadratic function, β_{max} should be one and γ_{max} should be zero. η_{max} limits the number of points to be saved and used in computing extrapolation. If η_{max} is zero in input, the program will use the theoretical limit, m-q, for η_{max}.

2.5.2 Main Program

The first section reads input and sets the initial conditions for a problem. The constraints are read and normalized to unit vectors. If equalities are indicated, these form the initial basis, and the corresponding inverse is computed.

In the next section, the program tests for feasibility and, if necessary, finds a feasible x or determines that there is no feasible x.

When a feasible x is found, the subprogram is entered to compute F and g. The program is now ready to enter the step procedure. A step includes the projection of the gradient, testing for the maximum, changes in the basis and computation of the step length.

At the beginning of each step, the program has a feasible x for which F and g have been computed. The non-basis constraints are classified into V($|\lambda_i| \leq \varepsilon_2$) and W($|\lambda_i| > \varepsilon_2$). The norm of the gradient is computed and tested for zero. If $\|g\| \leq \varepsilon_1$, this is the maximum.

If $\|g\| > \varepsilon_1$ and $q = 0$, the computed gradient is stored as the projected gradient, and program skips to compute z. If $q \neq 0$, the gradient is projected and the norm of the projected gradient is tested for zero. If $\|Pg\| < \varepsilon_1$, the program tests for a constraint to drop. If there is no constraint to drop, the maximum test is satisfied and the current point is the maximum. If there is at least one, the best one is cropped from the basis, and $\|P_{q-1}g\|$ is calculated and tested.

If $\|Pg\| \geq \varepsilon_1$ and $q < m$, the unit vector z is computed. If $V \neq 0$, $z^T n_i$ is computed and tested for all i in V. If the minimum $z^T n_i$ is negative, the corresponding constraint is added to the basis if it is linearly independent, and $\|Pg\|$ is calculated and tested.

When there are no more constraints in V for which $z^T n_i$ is negative or when $V = 0$, the program tests for basis changes. If the step was interior, the program tests for a constraint drop.

When the program finds no more changes to be made in the basis, it is ready to compute the next step length. If the previous step was not interior, the initial step length is set at Υ_{max}. If the previous step was interior, the initial step length depends on the extrapolation method selected. Two extrapolation methods available are x, G method and 3-point method.

When initial step length is determined and if $W = 0$ this is checked against constraints in W. For all i in W for which $z^T n_i$ is negrative, $\Upsilon_i = -\lambda_i / z^T n_i$ is computed. If the minimum Υ_i is less than the initial Υ, the step is limited by the constraint H_i and Υ_i replaces Υ.

A new x, $x_v = x_{v-1} + \Upsilon z$, is computed. F and g are computed and $z^T n_i$ is computed. If Υ is less than Υ_{min}, the interpolations are skipped. Two kinds of interpolations are available to find the maximum of F in the direction determined by z. The first is computed if the gradient has reversed direction between x_{v-1} and x_v. This prevents the problem from overshooting the maximum. The second is computed if F has not increased.

After completing the necessary gradient interpolations, x is checked for feasibility. Theoretically, x should be feasible, but because of inaccuracies due to round-off in the inverse matrix it is possible to violate the constraints. In that case F and g are recomputed and F is rechecked.

One pass through the step procedure is now complete. The step counter is tested against maximum step limit. If the limit is not reached, the program returns to the beginning of step procedure.

2.5.3 Subroutines

Subrpitone REINV essentially computes the inverse matrix $(N_q^T N_q)^{-1}$ whereas subroutines MATCOM and COMMAT do the matrix computations required by the gradient projection algorithm.

Subroutine AMDA calculates the λ's while subroutine CLASS classifies the constraints into different catagories. In the program.

u = linearly dependent constraint

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

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

Subroutine FUNCT is added to the program to compute the value of the objective function, $F(x)$, and its gradient, $g(x)$, for any given x within the region.

2.6 DISCUSSION

The technique and program into which it is incorporated are designed to handle general nonlinear problems with linear constraints. The method is intended to apply to a variety of problems, even though in many cases more computation time may be required.

If the function is concave, a global maximum is guaranteed. In order cases, the solution may be only a local maximum, and several widely separated starting points should be tried. If different results are obtained, the best that can be done is to take the maximum value of this set.

As true with different variants of the gradient methods, in general, an infinite number of iterations may be required before the conditions regarding a constrained global maximum are reached. Also, the steps may get too small near the stationary point resulting in a very slow convergence rate [11].

Goldfarb [9] has developed a conjugate gradient method for nonlinear problems with linear constraints. According to him, the method has a faster convergence rate and can navigate through the troublesome regions like "steep ridges" and "narrow curving valleys". At this stage, only a limited amount of computational experience is available for conjugate gradient method.

In conclusion, with this program, as with most nonlinear programs, the results obtained on real-life problems are very dependent on the design of the problem to be solved, as well as the effectiveness of minor adjustments of an algorithm to obtain best results for specific unusual problems.

CHAPTER 3

A MANAGEMENT MODEL FOR WATER QUALITY CONTROL

3.1 INTRODUCTION

The problem of river basin planning for water quality is currently receiving wide attention. The stimulus for this attention is the recognition that water quality problems are not necessarily the result of one recalcitrant polluter but of many. Because of this, the Federal government's deep commitment to restoring stream quality is increasing not only in form of financial assistance, but also in responsibility.

One municipality, one industry or even one state can not always control stream pollution, For this reason governmental agencies having jurisdiction over entire river basins are establishing quality standards for each section of the stream. These stream standards are intended to maintain stream quality by limiting the amount of waste that can be discharged into the stream.

The problem of determining standards for the stream becomes more complex when there are two or more sources of pollution. In these cases, the amount of waste released from one point may mix with the waste released at another point to contribute to the pollution downstream from both points. The quality standard at any point in the stream can be met by many combinations of quantities released at various locations upstream. The problem is to find the combination that results in a minimum total cost.

Systems analysis and its applications have been increasingly applied to this problem of water quality management. A series linear programming models were structured to determine alternative ways of meeting quality standards. Deininger [4] structured the problem as a linear programming model utilizing various approximations of the differential equations used to describe the dissolved oxygen profile of streams. Kerri [17] proposed a dynamic model for achieving and maintaining water quality control. He applied the concept of a critical reach to a simplified version of Willamette River in Oregon. Thomann [28] and Sobel [26] also developed linear programming models for related but different conditions of the Delaware estury. Liebman and Lynn [18] presented a dynamic programming model to study this problem. The model was solved for a simplified example based on data from the Willamette River. Revelle, Loucks and Lynn [19, 23] developed several linear programming models. The difference in each model reflected both the assumption regarding the river basin and the manner in which the standard is specified.

In general, the treatment plant costs are not linear. An attempt has been made to formulate this problem as a multistage decision process with a nonlinear cost function. The model so developed can be readily applied to a variety of river basins with minimal alteration of the basic model. The treatment closely follows one discussed in [19].

3.2 BACKGROUND

A large portion of the municipal and industrial waste released into streams is organic material. These organics become a source of nutrients for many organisms found in streams. Dissolved oxygen (DO) contained is withdrawn by these organisms in the process of utilizing these wastes. The larger the quantity of these bio-degradable wastes the larger is the population of these organisms and the greater is the demand for oxygen.

Fish and other aquatic animals and plants require certain minimum concentrations of DO if they are to survive in the stream. If the DO concentration is completely depleted, the stream becomes anaerobic and may be more reminiscent of a sewer than a stream. Insufficient removal of organics in the wastewater prior to its release into the stream can bring about these conditions. For these reasons stream quality standards usually specify minimum allowable DO concentrations in each section of the stream. The DO parameter is the one most commonly used to measure and limit the amount of pollution resulting from these organic wastes.

The capacity of the stream to assimilate bio-degradable wastes is determined by such factors as stream flow, stream temperature, the waste concentration as measured by its biochemical oxygen demand (BOD), the DO concentration, and the physical and biological properties of the stream that affect settling rates, reaeration, BOD addition due to runoff, scour etc. Two differential equations describing the processes that determine the concentration of DO in the stream have been developed by Camp [2] and Dobbins [5].

The first equation assumes that the rate of change in the

BOD concentration with time, (dB)/(dt), is proportional to the concentration of BOD present, B, and to the rate of BOD addition, R, due to runoff and scour.

$$(dB)/(dt) = -(k_1 + k_3) B + R \tag{1}$$

The terms k_1 and k_3 represent rate constants for deoxygenation and sedimentation, respectively.

The second equation assumes that the rate of change in DO deficit, (dD)/(dt), is proportional to the concentration of BOD present, B; the existing oxygen deficit, D; and the rate of oxygen production or reduction, A, due to plant photosynthesis and respiration.

$$(dD)/(dt) = k_1 B - k_2 D - A \tag{2}$$

The constant, k_2, reflects the rate at which DO is returned to the stream through reaeration.

The equations, at best, grossly describe the effects of the introduction of unstable oxygen-demanding substances upon the oxygen resources of the stream. They do not adequately describe the complex biological, physical and chemical phenomena of streams. But the formulations are currently used by state and federal officials to prescribe levels of wastewater treatment.

By integrating equation (1), the BOD concentration, B_t, at any point (corresponding to a time, t) downstream from an initial BOD concentration, B_o, can be determined.

$$B_t = (B_o - \frac{R}{k_1 + k_3})(e^{-(k_1+k_3)t}) + \frac{R}{k_1 + k_3} \tag{3}$$

Using equation (3), equation (2) can be integrated to determine the oxygen deficit, D_t, at any time, t, downstream from an initial oxygen deficit, D_o.

$$D_t = \frac{k_1}{k_2 - (k_1 + k_3)} \left(B - \frac{R}{k_1 + k_3}\right)\left(e^{-(k_1+k_3)t} - e^{-k_2 t}\right) + \frac{k_1}{k_2}\left(\frac{R}{k_1 + k_3} - \frac{A}{k_1}\right)\left(1 - e^{-k_2 t}\right) + D_o\, e^{-k_2 t} \tag{4}$$

At any time, t, the DO saturation concentration, CS, minus the deficit, D_t, yields the DO concentration, C_t.

$$C_t = CS - D_t \tag{5}$$

Equation (4) represents the "oxygen sag" curve as shown in Fig. 2. The critical deficit, D_c, and critical time, t_c, occur when DO concentration is at its lowest value, C_c. The critical deficit, D_c, is the difference between saturation concentration, CS, and the actual concentration at the critical time. In Region I the rate of deoxygenation exceeds the reaeration rate. In Region II the reverse is true.

It is recognised that methods for measuring some of parameters, namely A, k_3 and R, have not been perfected and in most cases these are unavailable. If A, k_3 and R are assumed to be zero, the more general oxygen sag eqation (4) becomes the simpler Streeter-Phelps [27] sag equation.

$$D_t = \frac{k_1}{k_2 - k_1} B_o\left(e^{-k_1 t} - e^{-k_2 t}\right) + D_o\, e^{-k_2 t} \tag{6}$$

and equation (3) is simplified to

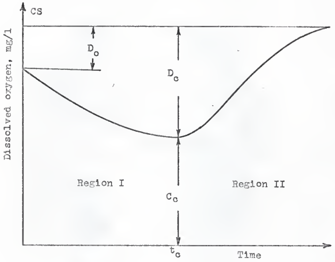


Fig. 2. Oxygen Sag Curve

$$B_t = B_o e^{-k_1 t} \tag{7}$$

The critical time, t_c, for this system is given by

$$t_c = \frac{1}{k_2 - k_1} \ln \frac{k_2}{k_1} \left(1 - \frac{(k_2 - k_1)D_o}{k_1 B_o}\right) \tag{8}$$

and the resulting maximum deficit is given by

$$D_{t_c} = \frac{k_1 B_o}{k_2 e^{-k_1 t_c}} \tag{9}$$

3.3 MATHEMATICAL FORMULATION

The purpose of this section is to illustrate how the non-linear oxygen sag equation (4) can be recast into a series of linear constraints. These linear constraints can then be incorporated into a mathematical model for determining minimum cost solutions of various water quality control poticies in a river basin.

3.3.1 Assumptions

1. A stream in which there are N waste dischanges is considered in the model as being divided into N reaches, the reach being defined as the stretch of stream between i^{th} and $(i+1)^{st}$ discharge.
2. Tributaries, if any are assumed to enter the top of a reach.
3. The amount of flow from each discharge, the DO concentrations, and the raw BOD loadings are assumed to be known.
4. The parameters of the equation relating DO to waste loading

are constant in each reach and known.

5. The stream flow in each reach is considered deterministic and known.
6. The standards for minimum DO concentration in each reach are specified.
7. A complete mixing is assumed at all points where a tributary or wastewater effluent enters a stream.
8. The design flow used for determining the capacity of the stream to assimilate wastes is usually the minimum average consecutive seven-day flow expected once in 10 years on the average [19].

3.3.2 Constraints

It is necessary to compute the BOD and DO concentrations at the beginning and end of each reach. In the inventory equations that follow, the subscript r denotes a particular reach.

In general the total flow, QS_r, in reach r is the sum of the flow in the previous reach, QS_{r-1}; the tributary flow entering the reach, QT_r; and the wastewater flow discharged into the reach, QW_r.

$$QS_r = QS_{r-1} + QT_r + QW_r \tag{10}$$

Assuming complete mixing, the BOD concentration at the beginning of each reach, BB_r, is equal to the sum of the BOD concentrations at the end of the previous reach, BE_{r-1}; in the tributary, BT_r; and in the wastewater effluent, BW_r, times their respectivs flows divided by the total flow.

$$BB_r = \frac{BE_{r-1}\ QS_{r-1} + BT_r\ QT_r + BW_r\ QW_r}{QS_r} \tag{11}$$

Similarly, the DO concentration at the beginning of each reach, CB_r, can be determined from the concentration at the end of the previous reach, CE_{r-1}; the tributary concentration CT_r; and DO concentration in the wastewater effluent, CW_r.

$$CB_r = \frac{CE_{r-1}\ QS_{r-1} + CT_r\ QT_r + CW_r\ QW_r}{QS_r} \tag{12}$$

The DO deficit at the beginning of each reach, DB_r, is the difference between the saturation concentration CS_r, and the initial DO concentration, CB_r.

$$DB_r = CS_r - CB_r \tag{13}$$

The BOD concentration and DO deficit at the end of each reach can be computed from the initial BOD concentration and DO deficit, using either equations (3) and (4) or equations (6) and (7). The time, t, in these equations is understood to be equal to the time required for water to flow from the beginning of each reach to the end of that reach, T_r. Thus, BE_r, the BOD concentration at the end of reach r and DE_r, the DO deficit at the end of reach r, can be determined.

Using these equations it is possible to write constraints that define the minimum allowable DO concentration within each reach.

The DO deficit, D_{rt}, at various points t along each reach r could be constrained to be less than or equal to the maximum allowable deficit in that reach, D_r^{max}.

$$D_{rt} \leq D_r^{max} \quad \text{for various } t : \; 0 \leq t \leq T_r \tag{14}$$

By reducing the interval between successive t's, the possibility of violation between these points is reduced. If the time of flow, T_r, is less than the critical time, t_c, only one quality constraint is necessary for that reach, namely

$$D_{rT_r} \leq D_r^{max} \tag{15}$$

A few trial solutions will enable one to eventually place these quality constraints, equation (14), at the points having the lowest DO concentration. With these constraints, the solution will yield the maximum amounts of BOD that can be released into each reach without violating the standard for any reach.

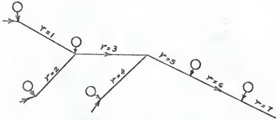
Sometimes an additional constraint may be desired if each treatment facility is required to remove the same fraction of BOD from the wastewater influent. Such a requirement can be expressed by equating for each reach the ratios of the BOD concentration released into a reach, BW_r, over the total amount available, BW_r^{max}.

$$\frac{BW_r}{BW_r^{max}} = \frac{BW_{r+1}}{BW_{r+1}^{max}} \tag{16}$$

or

$$P_r = P_{r+1} \tag{17}$$

where



r = Reach number

O = Wastewater treatment facility

Fig. 3. Hypothetical River Basin [19]

P_r = percentage treatment at r^{th} plant

Also, in general, P_r is constrained between two specified limits.

$$P_r^{min} \leq P_r \leq P_r^{max} \tag{18}$$

3.3.3 Objective Function

Objective function should enable one to specify amount of wastewater treatment required to meet at minimum cost a set of quality standards for the river basin. The objective is to minimize the total cost of wastewater treatment. mathematically, it may be stated as

$$\text{MINIMIZE} \sum_{r=1}^{N} C_r(P_r) \tag{19}$$

where

$C_r(P_r)$ = the function representing the total cost of providing the treatment P_r at the r^{th} discharge, given as an annual cost including amorization and operating costs.

The objective is constrained by the quality standards as expressed by (14) and (15) and the inventory equations discussed earlier.

3.4 DESCRIPTION OF SYSTEM

A hypothetical river basin shown in Fig. 3 was used by Loucks, Revelle and Lynn [19] to establish and evaluate various water quality control policies. The same basin with all the necessary data is used here. As a matter of fact this hypothetical system was derived from a highly simplified representation of Willamette River in Oregon. The main contributors of pollutents on this stream are municipalities and pulp and paper industries.

In this basin the quality of water in seven reaches is affected by the amount of BOD released from six wastewater treatment facilities. It is assumed that these facilities exist and are currently removing a sufficient amount of BOD to satify the stream quality standards. It is anticipated, however, that by 1980 the BOD load will increase considerably. This will obviously necessitate additional treatment. It is required to determine the additional treatment necessary and the minimum cost required to meet the standards in 1980.

Table 1 provides all the necessary stream and wastewater data. Table 2 gives wastewater treatment data. For all treatment facilities, a minimum removal of 35% has been imposed. The intent of this constraint is to require each plant to provide at least primary treatment, thus ensuring the abscence of floating solids in the stream. The presence of these solids is usually considered objectionable even though they may not reduce the oxygen concentration below the minimum acceptable level. Because of technological difficulties in construction of facilities that can remove over 90% of the BOD with certainty, 90% will be assumed to be the maximum treatment required.

The treatment plant costs are usually convex within the range from 35% to 90% BOD removal. A typical cost curve is shown in Fig. 4A. Loucks, Revelle and Lynn [19] assumed that costs are

Table 1. Stream and Wastewater Data [19]

Reach No.	T_r (Days)	QW_r (MGD)	QT_r (MGD)	QS_r (MGD)	CS_r (mg/l)	D_r^{max} (mg/l)	CW_r (mg/l)
1	.235	5	1355	1360	10.20	3.20	1.0
2	1.330	37	1290	1327	9.95	2.45	1.0
3	1.087	8	1360	2695	9.00	2.00	1.0
4	2.067	14	296	310	9.54	3.54	1.0
5	.306	0	310	3005	9.00	2.50	-
6	1.050	26	0	3031	8.35	2.35	1.0
7	6.130	41	0	3072	8.17	4.17	1.0

Table 1. (Con'd)

Reach No.	CT_r (mg/l)	BT_r (mg/l)	K_1 $days^{-1}$	K_2 $days^{-1}$	K_3 $days^{-1}$	A (mg/l/day)	R (mg/l/day)
1	9.50	1.66	.31	1.02	.02	.85	.15
2	8.00	0.68	.41	.60	.03	.14	.14
3	?	?	.36	.63	.04	.18	.14
4	9.70	1.0	.35	.09	.04	.05	.11
5	?	?	.34	.72	.05	.39	.11
6	-	-	.35	.14	.06	.07	.13
7	-	-	.30	.02	.00	.00	.00

Table 2. Wastewater Treatment Data [19]

Reach No.	1980 BOD Load (mg/1)	Present % Removal 1980 Load	Annual Costs of Various 1980 BOD Removals Dollars					
			35%	50%	60%	75%	85%	90%
1	248	67	0	0	0	22,100	77,500	120,000
2	408	10	546.000	552,000	630,000	780,000	987,000	1,170,000
3	240	26	160,000	170,000	210,000	277,500	323,000	378,000
4	1440	24	324,000	339,000	413,000	523,000	626,000	698,000
6	2180	12	385,000	408,000	500,000	638,000	790,000	900,000
7	279	26	670,000	690,000	840,000	1,072,000	1,232,500	1,350,000

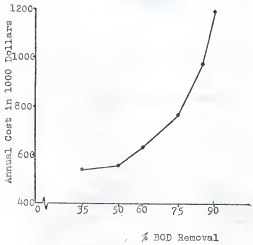


Fig. 4A. Annual Cost of Wastewater Treatment for Plant 2 [19]

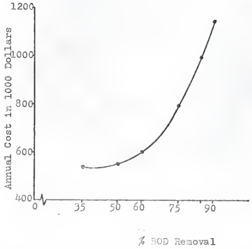


Fig. 4B. Fitted Annual Cost of Watewater Treatment for Plant 2

Table 3. Coefficients of Fitted Quadratic Curves for Treatment Plant Costs

Plant No.	a $\times 10^6$ ($)	b $\times 10^6$ ($)	c $\times 10^6$ ($)
1	0.6677246	-2.1281730	1.6879880
2	1.086456	-2.5185540	2.8623040
3	0.2358398	-0.4570770	0.67469790
4	0.4561768	-0.8029786	1.1875000
6	0.6172485	-1.2800290	1.7590330
7	0.8294067	-1.1845700	1.9653320

linear within each segment. In this work, a quadratic fit was obtained for treatment costs at each plant using the standard regression analysis procedure [7]. A typical quadratic fit is shown in Fig. 4B. The general cost equation can be written as

$$a + bP + cP^2 \tag{20}$$

where P = percent treatment provided at a plant

The coefficients a, b and c for various plants are given in Table 3. This particular aspect is a significant change from [19] and represents a more realistic situation. The problem no longer can be solved by linear programming. It becomes one of nonlinear programming where a nonlinear objective function is subjected to linear constraints.

The objective function can now be written for the river basin illustrated in Fig. 3.

$$\text{MINIMIZE} \quad \sum_{\substack{r=1 \\ r \neq 5}}^{7} a_r + b_r P_r + c_r P_r^2 \tag{21}$$

The constraints must bound all P_r's, define the initial and final BOD and DO concentrations in each reach and limit the BOD concentrations at the beginning of each reach so that the stream standard is not violated.

3.5 SOLUTION

The problem formulated above was a nonlinear programming problem with linear constraints. Application of the gradient projection method seems quite justifiable to obtain the solution of this problem.

Two different runs were made with the data of Tables 1 and 2. For each run, both Streeter-Phelps and Camp-Dobbins formulations were used. Run 1 was used to find the optimum (minimum) cost configuration of plants that will just meet the DO standards specified for the stream. Run 2 duplicated the conditions of Run 1, except that the minimum DO standards in each reach had been reduced by 0.5 mg/l.

A brief description of different computational aspects involved in solution by the gradient projection method is not out of place.

The various tolerances and limits are to be judiciously selected. A very small value of ε_1 may require a considerably more computation time without actually contributing much to the improvement of functional value. In general, the following approach was used. Firstly, using a relatively large value of ε_1(0.005), the solutions were obtained starting with different initial points. Using this information, a new starting point very close to optimal values of control variables was established. The value of ε_1 was then considerably reduced (.0001) and an accurate optimal solution was obtained.

The following values were assumed

ε_1 = .005 and .0001	MANU = 20	$\gamma_{max} = 0$
ε_2 = .005	MXRN = 3	$\eta_{max} = 0$
ε_3 = .005	$\beta_{max} = 1$	$\tau_{max} = 10$

The problem being one of minimization was solved by the gradient projection maximization algorithm by maximizing the negative of the original objective function. The constant term appearing in the objective function given by (21) was not considered in optimization. Thus,

$$F = \sum_{\substack{r=1 \\ r \neq 5}}^{7} b_r P_r + c_r P_r^{\,2} \tag{22}$$

while, the actual total minimum cost is given by

$$F' = \sum_{\substack{r=1 \\ r \neq 5}}^{7} a_r \; - \; \text{maximum } (-F) \tag{23}$$

Five different initial points were used for each case. This information is shown in Tables 4 through 7 along with details about execution times, number of iterations and number of functional evaluations. The values of $\|P_q g\|$ at optimal are also given. Tables 8 through 11 indicate the convergence rates obtained for certain specific cases. The same information is illustrated in Fig. 5 for one selected case.

The optimal (minimum) cost solutions obtained for each case are presented in Tables 12 through 15. These tables also provide the information regarding the maximum amount of BOD that can be released to the stream and the resulting DO concentrations. Tables 16 and 17 indicate the linear programming solutions obtained by Loucks, Revelle and Lynn [19].

3.6 DISCUSSION

The model presented in this work can be used to determine the minimum total cost associated with any particular set of

Table 4. Optimal F obtained with Different Starting Control Variable Values Streeter-Phelps Formulation

Serial No.	Initial Conditions	$-F \times 10^6$ (\$)	Actual Cost $F' \times 10^6$ (\$)	$\|P_q\, g\|$ at Optimal	Total No. of Functional Evaluations	Total No. of Iterations	Execution Time in Seconds
1	All 0.50	0.630784	3.262067	0.000076	39	25	28.39
2	All 0.60	0.630696	3.262155	0.000069	32	20	21.62
3	All 0.70	0.630824	3.262028	0.000074	41	29	31.28
4	All 0.80	0.630563	3.262288	0.000099	42	27	28.50
5	Near Optimal	0.631601	3.261250	0.000064	7	4	6.54

Table 5. Optimal F obtained with Different Starting Control Variable Values. Streeter-Phelps Formulation for Reduced DO standards

Serial No.	Initial Conditions	-F $\times 10^6$ (\$)	Actual Cost F' $\times 10^6$ (\$)	$\|P_q\ g\|$ at Optimal	Total No. of Functional Evaluations	Total No. of Iterations	Execution Time in Seconds
1	All 0.50	0.820330	3.072521	0.001648	9	5	7.18
2	All 0.60	0.820375	3.072475	0.001106	10	6	8.37
3	All 0.70	0.820347	3.072504	0.000120	11	6	7.92
4	All 0.80	0.820334	3.072516	0.000500	12	7	9.62
5	Near Optimal	0.820386	3.072465	0.000047	6	4	6.74

Table 6. Optimal F obtained with Different Starting Control Variable Values Camp-Dobbins Formulation

Serial No.	Initial Conditions	$-F \times 10^6$ (\$)	Actual Cost F' $\times 10^6$ (\$)	$\|P_q\ g\|$ at Optimal	Total No. of Functional Evaluations	Total No. of Iterations	Execution Time in Seconds
1	All 0.50	0.670485	3.222366	0.000400	76	50	52.13
2	All 0.60	0.670253	3.222597	0.000090	50	34	33.19
3	All 0.70	0.668591	3.224259	0.000060	34	22	24.71
4	All 0.80	0.669118	3.223733	0.000050	34	23	23.98
5	Near Optimal	0.671553	3.221297	0.000106	2	1	3.72

Table 7. Optimal F obtained with Different Starting Control Variable Values Camp-Dobbins Formulation for Reduced DO Standards

Serial No.	Initial Conditions	$-F \times 10^6$ (\$)	Actual Cost $F' \times 10^6$ (\$)	$\|P_q\ g\|$ at Optimal	Total No. of Functional Evaluations	Total No. of Iterations	Execution Time in Seconds
1	All 0.50	0.848899	3.046172	0.003000	9	5	7.59
2	All 0.60	0.851298	3.041553	0.000385	30	20	22.45
3	All 0.70	0.847289	3.045562	0.000200	14	8	10.93
4	All 0.75	0.848977	3.043874	0.000219	25	16	17.00
5	Near Optimal	0.856360	3.036491	0.000211	2	1	3.34

Table 7A. Starting Control Variable Values for Serial No. 5 in Tables 4 through 7

Variable No.	Table No.			
	4	5	6	7
1	0.665400	0.653646	0.661433	0.649500
2	0.625900	0.563028	0.601821	0.542400
3	0.480000	0.432987	0.464417	0.417900
4	0.900000	0.900000	0.900000	0.900000
5	0.900000	0.900000	0.900000	0.900000
6	0.638000	0.524813	0.628192	0.500000

Table 8. A Typical Convergence Rate
Streeter-Phelps Formulation (Table 4, S. No. 1)

Iteration No.	$-F$	$\|P_q\ g\|$	Total No. of Functional Evaluations	No. of Constraints in Basis
0	0.589375	0.518968	1	3
1	0.629450	0.038134	3	3
2	0.629820	0.026254	4	3
5	0.629944	0.001224	10	3
10	0.630128	0.001783	17	3
15	0.630222	0.000246	25	3
20	0.630594	0.000725	32	3
25	0.630784	0.000076	39	3

Table 9. A Typical Convergence Rate
Streeter-Phelps Formulation for Reduced DO Standards
(Table 5, S. No. 1)

Iteration No.	$-F$	$\lVert P_q g \rVert$	Total No. of Functional Evaluations	No. of Constraints in Basis
0	0.748938	0.934905	1	2
1	0.789650	0.445582	2	2
2	0.818794	0.081293	4	3
3	0.819835	0.046074	6	3
4	0.820168	0.026164	8	3
5	0.820330	0.001648	9	3

Table 10. A Typical Convergence Rate
Camp-Dobbins Formulation (Table 6, S. No. 1)

Iteration No.	$-F$	$\lVert P_q\, g \rVert$	Total No. of Functional Evaluations	No. of Constraints in Basis
0	0.626197	0.513754	1	3
1	0.665749	0.044745	3	3
5	0.666663	0.009781	9	3
10	0.667031	0.006826	16	3
20	0.667910	0.003167	31	3
30	0.668808	0.002297	46	3
40	0.669648	0.002064	61	3
50	0.670485	0.000400	76	3

Table 11. A Typical Convergence Rate
Camp-Dobbins Formulation for Reduced DO Standards
(Table 7, S. No. 2)

Iteration No.	-F	$\|P_q g\|$	Total No. of Functional Evaluations	No. of Constraints in Basis
0	0.652358	1.343425	1	2
1	0.831557	0.511353	2	2
2	0.843963	0.158165	4	3
5	0.848863	0.012325	9	3
10	0.849896	0.080906	16	3
15	0.851011	0.004447	22	3
20	0.851298	0.000385	30	3

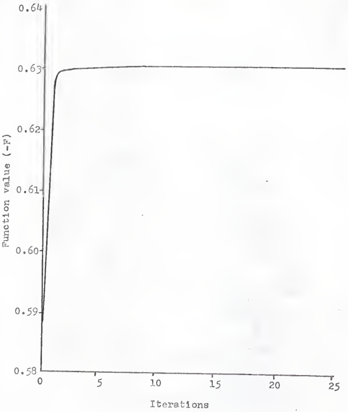


Fig. 5. A Typical Convergence Rate (Table 4, S. No. 1)

Table 12. Minimum Cost Solution for Maintaining DO Standards Streeter-Phelps Formulation

Reach No.	% BOD Removal	Annual Cost ($)	Maximum BOD Release to Stream (mg/1/day)	Minimum DO in Reach (mg/1)	Minimum Allowable DO in Reach (mg/1)
1	66.5	0	83.00	9.50	7.00
2	63.0	631,200	152.70	7.61	7.50
3	48.0	171,921	124.80	8.15	7.00
4	90.0	695,350	144.00	6.01	6.00
5*	-	-	-	7.95	6.50
6	90.0	890,057	218.00	6.91	6.00
7	63.8	873,718	101.00	4.00	4.00
Total		3,262,246	823.50		

* No wastewater effluent is released into Reach 5.

Table 13. Minimum Cost Solution When DO Standards Are Reduced by 0.5 mg/l
Streeter-Phelps Formulation

Reach No.	% BOD Removal	Annual Cost ($)	Maximum BOD Release to Stream (mg/l/day)	Minimum DO in Reach (mg/l)	Minimum Allowable DO in Reach (mg/l)
1	65	0	85.90	9.50	6.50
2	56	575,890	178.20	7.41	7.00
3	43	164,365	136.20	8.04	6.50
4	90	695,360	144.00	6.01	5.50
5*	-	-	-	7.86	6.00
6	90	890,042	218.00	6.80	5.50
7	52	748,963	132.60	3.50	3.50
Total		3,074,620	894.90		

* No wastewater effluent is released into Reach 5.

Table 14. Minimum Cost Solution for Maintaining DO Standards Camp-Dobbins Formulation

Reach No.	% BOD Removal	Annual Cost ($)	Maximum BOD Relase to Stream (mg/1/day)	Minimum DO in Reach (mg/1)	Minimum Allowable DO in Reach (mg/1)
1	66	0	84.0	9.50	7.00
2	60	607,500	162.4	7.66	7.50
3	46.4	169,086	128.5	8.32	7.00
4	90	695,368	144.0	6.04	6.00
5*	-	-	-	8.19	6.50
6	90	890,020	218.0	7.17	6.00
7	62.8	860,762	103.8	4.12	4.00
Total		3,222,736	840.7		

* No wastewater effluent is released into Reach 5.

Table 15. Minimum Cost Solution When DO Standards Are Reduced by 0.5 mg/1
Camp-Dobbins Formulation

Reach No.	% BOD Removal	Annual Cost ($)	Maximum BOD Release to Stream (mg/1/day)	Minimum DO in Reach (mg/1)	Minimum Allowable DO in Reach (mg/1)
1	65	0	86.92	9.50	6.50
2	54	560,714	187.95	7.47	7.00
3	42	162,607	139.82	8.22	6.50
4	90	695,368	144.00	6.04	5.50
5*	-	-	-	8.10	6.00
6	90	890,022	218.00	7.06	5.50
7	50	730,226	138.67	3.61	3.50
Total		3,038,937	915.56		

* No wastewater effluent is released into Reach 5.

Table 16. Minimum Cost Solution for Maintaining Dissolved Oxygen Standards by Loucks et.al. [19]

Reach No.	% BOD Removal	Annual Cost ($)	BOD Released to Stream (mg/l/day)	Minimum DO in Reach (mg/l)	Minimum Allowable DO (mg/l)
1	67	0	82	9.5	7.0
2	55	608,500	183	7.5	7.5
3	50	170,000	120	8.3	7.0
4	90	690,000	144	6.0	6.0
5*	-	-	-	8.0	6.5
6	90	900,000	218	7.1	6.0
7	64	902,000	102	4.0	4.0
Total		3,270,500	849		

* No wastewater effluent is released into Reach 5.

Table 17. Minimum Cost Solution When Dissolved Oxygen Standards Are Reduced by 0.5 mg/l by Loucks et. al. [19]

Reach No.	% BOD Removal	Annual Cost ($)	BOD Released to Stream (mg/l/day)	Minimum DO in Reach (mg/l)	Minimum Allowable DO (mg/l)
1	67	0	82	9.5	6.5
2	50	522,000	204	7.3	7.0
3	50	170,000	120	8.2	6.5
4	90	690,000	144	6.0	5.5
5	-	-	-	7.9	6.0
6	90	900,000	218	7.0	5.5
7	50	690,000	139	3.5	3.5
Total		3,002,000	907		

minimum allowable DO concentrations in a river basin. This model also provides an useful tool to determine sensitivity of both the cost and the actual minimum DO concentrations in reaches to changes in the minimum allowable DO concentration in any particular reach.

As the data for the problem was taken from the work of Loucks, Revelle and Lynn [19], a comparison of the results is not out of place. Discussion here is confined to only Camp-Dobbins formulation, but the same applies to Streeter-Phelps formulation. The results of Loucks, Revelle and Lynn are summarized in Tables 16 and 17. These correspond to Tables 14 and 15, respectively of this work.

Note that whereas Plants 2 and 3 provided 60% and 46.5% treatments (Table 14), these plants provided 55% and 50% (Table 16) in the linear programming solution presented by them. But it is worth noting that the sum of costs at Plants 2 and 3 was nearly identical in both cases. Note, also, that the costs for the two solutions were nearly identical, this solution costing $47,764 less than the linear programming solution. This small difference may well stem from the fact that they used linearized cost curves while a quadratic fit was used in this solution.

Table 15 presents the optimal (minimum) cost solution when the minimum allowable DO concentration is reduced by 0.5 mg/l in each reach. This reduction in DO standards results in an annual cost savings of $183,800 or about 6 percent of the total cost.

From Tables 14 and 15, it is clear that the DO standards for Reaches 2 and 7 dictate the actual concentrations in the entire basin. In other words, a reduction in the minimum DO concentrations in any but Reaches 2 and 7 would neither decrease the minimum total cost nor the actual DO concentrations. Table 15 shows that once the DO standards have been reduced by 0.5 mg/l in each reach, only the Reach 7, can be regarded as critical and hence it can be said to determine the required treatment, and therefore, the cost, throughout the basin.

It is evident from the results that a change in the minimum DO concentration standards in several reaches may have no effect on the DO concentrations in these reaches. Conversely, a change in the minimum allowable DO concentration in a single reach may affect the DO concentrations in every other reach.

A decrease in the minimum DO concentration by 0.5 mg/l in each reach of the hypothetical basin only reduces the actual minimum DO concentrations by 0.2 mg/l in Reach 2; 0.1 mg/l in Reaches 3, 5 and 6 and 0.5 mg/l in Reach 7. The change in annual benefits resulting from these lower oxygen concentrations can be compared to the annual cost savings of $185,875 in order to determine the desirability of this reduced standard.

In many basins, under present legislation, uniform standards in terms of a final percentage removal of waste material are imposed upon polluters. If such a standard is applied to the every facility in this basin, it is evident that every facility will have to remove the same percentage as is required by the treatment facility on Reach 4, namely 90%. Anything less than 90% waste removal from the effluent entering Reach 4 would result in a minimum DO concentration less than the minimum allowable DO.

The same conclusion can not be drawn from noting that 90% treatment is required on Reach 6. Since in this case there exists some control over the concentration of waste and DO in the water entering that reach.

The gradient projection method did not encounter any trouble in solution of the problem. The efficiency of method, of course, depends upon a judicious choice of various limits and tolerances.

CHAPTER 4

OPTIMAL DESIGN AND CONTROL OF RESERVOIR SYSTEMS

4.1 INTRODUCTION

The mathematical models which are used to describe water-resource systems often contain nonlinear mathematical relationships that are difficult to analyze and optimize. Furthermore, a large water-resource system is generally a multidimensional problem. Pioneering work in water-resource systems and optimization analysis has been carried out in the Harvard Water Program [16, 20]. Indications of recent research [16] imply that simulation is still being used in the detailed, final-stage optimization of a given water-resource system.

Hall and others [13, 14] were the first to propose the application of dynamic programming to the optimization of reservoir systems. Meyer [22] successfully used dynamic programming approach in optimizing the operation of a multiple-purpose reservoir. Dynamic programming has the advantage of effecting the decomposition of a highly complex problem into a series of far less complex problems. However, due to the dimensionality difficulties, this procedure can not be extended to more practical problems.

Linear programming is not generally useful, because individual reservoir utility functions are in general definitely nonlinear. Quite often, many of the restrictions on the operation of a water-resource system are linear. This situation arising from a nonlinear objective or utility function subjected to linear constraints provides an useful field for nonlinear programming techniques such as the gradient projection method.

Specifically, the gradient projection method discussed earlier, is applied here to solve two simple problems. The stress is on the method that how a general nonlinear objective can be treated without going into tedious details of linearization. Both hypothetical systems are taken from [20] along with all the relavent data needed to solve them.

Initially, the artificial system from which these two hypothetical sytems are derived, is described. This includes a brief description of streamflow data and certain basic assumptions regarding irrigation and water power. Following this, both models are described and solved by the gradient projection method.

4.2 DESCRIPTION OF SYSTEM

The simplified system based on the Clearwater River Basin in Idaho is described in detail in the Harvard Water Program [20]. For this system, at least one of each major kind of output of a water-resource system was developed. These were: a withdrawal - consumptive use; a nonwithdrawal, essentially non-consumptive use; and a retardation or withholding use. For these purposes, the irrigation of crops, the development of water power and reduction of flood damage were considered.

4.2.1 Physical Layout of the System

The physical layout of the system chosen for development is

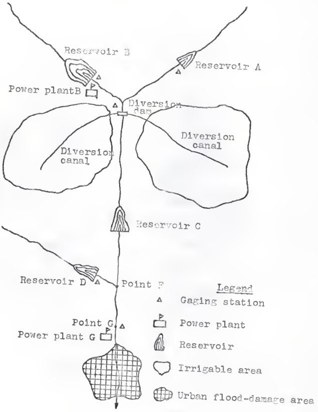


Fig. 6. Sketch of the Simplified River-Basin System [20]

shown in Fig. 6. There are four impounding reservoirs A, B, and C and D; two power plants, one at reservoir B and other at point G. Also there is an irrigation diversion dam at point E. Only reservoirs A and B can provide releases for irrigation. Reservoir D lies on a tributary stream which joins the main stem at point F. Irrigable areas lie on both banks of the main river downstream from reservoirs A and B. Return flow reenters the river channel below E but upstream from reservoir C; none of it can be reused within the irrigated areas. A flood damage zone is situated just below point G.

The additional details have been purposely avoided in the development of the system because these may unnecessarily complicate operation studies, the computer programming and the general analysis.

4.2.2 Streamflow Data

A detailed description of the streamflow data is given in [20]. Table 18 identifies the magnitudes of mean monthly flows observed at point E. The observed pattern is typical of the hydrology of a catchment area in which the melting of winter snows produces large spring runoffs, followed by low summer and fall discharges. Such basins are found in wide regions of the wastern United States. Data for other points also indicate the same pattern.

4.2.3 Irrigation

The consumptive use and diversion requirement for irrigation are based on climatological data and irrigation practices in the region. The region is semiarid with a moderately long growing season of 205 days.

The unit irrigation-diversion requirement was assumed to be 5.0 ft(or 5.0 acre ft per acre) yearly. This is distributed by months as shown in Table 19.

The estimate of return flow from irrigation diversion was based on certain assumptions. No water would be lost by evaporation from drainage and wasteway channels, that no return flow would be consumed on nonirrigated land, and that no water escape into neighbouring basins. Return flow would therefore equal the difference between the irrigation diversion requirement and the set consumptive use of irrigation water. In general, it comes to about 50 to 60 percent of total irrigation diversion requirement. The monthly distribution of this return flow in years of full irrigation supply, shown in Table 20, is in accordance with observations in several Bureau of Reclamation projects.

Irrigation output of the system was set at 6×10^6 acre ft. Unit annual gross irrigation benefits were taken as decreasing from about \$ 6.50 per acre feet at low levels of development to about \$ 5.00 per acre feet at maximum development. Figure 7 shows the unit gross benefit function.

The losses occuring from shortages are not discussed here. The estimates of capital costs and costs of operation, maintainance, and replacement (OMR) for irrigation-diversion works are given in [20]. They are based on data from Bureau

Table 18. Magnitude of Mean Monthly Flows at Point E [20]

Month	Flow (10^2 acre ft)	Percentage of Mean Annual Runoff
January	1,698	3.00
February	1,778	3.10
March	3,066	5.40
April	8,939	15.80
May	18,100	32.00
June	12,618	22.10
July	3,469	6.20
August	1,079	1.80
September	845	1.50
October	1,264	2.20
November	1,813	3.30
December	1,989	3.60
Total	56,658	100.00

Table 19. Assumed Monthly Distribution of Annual Irrigation Diversion Requirement [20]

Month	Percentage of total annual diversion requirement
April	12.4
May	14.6
June	16.6
July	19.0
August	18.0
September	12.4
October	7.0
November-March	0.0
	100.0

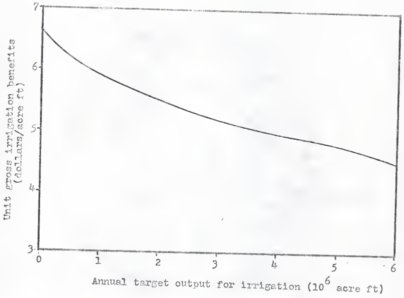


Fig. 7. Assumed Unit Gross Irrigation Benefit Function [20]

Table 20. Assumed Monthly Distribution of Annual Return Flow with Full Irrigation Supply [20]

Month	Percentage of total annual return flow
November	8
December	7
January	5
February	4
March	4
April	6
May	8
June	10
July	12
August	14
September	12
October	10
	100

Table 21. Assumed Monthly Distribution of Annual Energy Requirement [20]

Month	Percentage of total annual energy requirement
November	8.1
December	8.3
January	8.2
February	7.5
March	7.3
April	7.7
May	8.3
June	8.9
July	9.1
August	9.3
September	9.1
October	8.2
	100.0

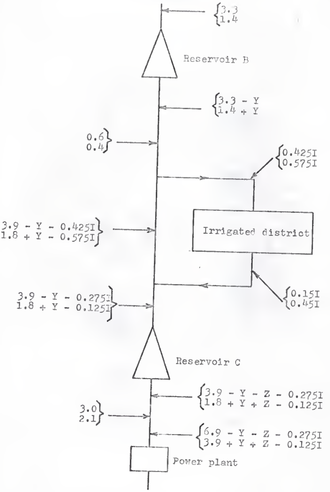


Fig. 8. Sketch of Configuration Used in Two-Period Problem [20]

of Reclamation projects.

4.2.4 Water Power

The power-requirement of the system was selected to be representative of a diversified agricultural-industrial, rural-urban economy. The assumed monthly distribution of the load is listed in Table 21. The maximum yearly target output for energy for the system was set at 4×10^9 kw hr.

Unit energy benefits were assumed to be constant for all levels of putput, namely, 9 mills per kw hr. of firm energy and 1.5 mills per kw hr. of nonfirm energy. Assumed capital costs and OMR costs for power plants are given in [20].

4.3 A MODEL WITH TWO SEASONS AND PREDICTABLE HYDROLOGY

This example deals with a relatively simple configuration of uses and installations under conditions in which the pattern of inflows repeats itself each year with certainty so that overyear storage is not required. This is the simplest problem that could be devised while retaining enough suvstance to present some challenge.

The configuration shown in Fig. 8 has been abstracted from simplified river basin discussed earlier. This configuration retains two reservoir sites, an irrigation project, and a run-of-the river power plant.

Most of the data are contained in the figure. The numbers in braces give mean flows in the river in two seasons of the year the wet season (top figure) and the dry (bottom figure).

The first topographic feature the river encounters is the reservoir B; its active capacity, denoted by Y, is one of the unknowns of the problem. The amount of water, Y, will be retained in the wet season and released in the dry.

Just below the confluence of the west branch with the main stem an irrigation diversion canal takes off water to an irrigated area lying to the east. The total amount of irrigation, I, is the second unknown of the problem. It is assumed that, whatever I may be, 42.5 percent of irrigation water must be provided in the wet season and 57.5 percent in the dry. The resulting return flows from the irrigated area are assumed to be 15 percent of I in the wet season and 45 percent in the dry.

It is also necessary to determine the usable capacity of reservoir C, denoted by Z. The fourth and final variable is the energy output of power plant, E. Half the annual output of energy generated is assumed to be required in the wet season and half in the dry.

4.3.1 Constraints

The first group of constraints requires simply that none of the four decision variables be negative.

$$Y \geq 0 \tag{1}$$

$$Z \geq 0 \tag{2}$$

$$I \geq 0 \tag{3}$$

$$E \geq 0 \tag{4}$$

The second group of constraints states that the flows in all reaches of the system must be nonnegative. From the map these constraints can be written down.

$$3.3 - Y \geq 0 \quad (5)$$

$$3.9 - Y - 0.425I \geq 0 \quad (6)$$

$$1.8 + Y - 0.575I \geq 0 \quad (7)$$

$$3.9 - Y - Z - 0.275I \geq 0 \quad (8)$$

Although there are other six flow constraints involving decision variables, these are autometically satisfied if above four constraints are satisfied.

The third group of constraints asserts that the flow at the power plant must be adequate in both the wet and the dry seasons to generate the amount of power that has been decided on.

The technical relationship between flow and energy output is taken to be

$$E = 0.144F$$

where

$E =$ energy generated in any period in 10^9 kw hrs.

and $F =$ flow through turbines in 10^6 acre feet

As equal amount of energy is reguired in both seasons, the two power constraints are

$$6.9 - Y - Z - 0.275I \geq 0.5E/0.133 = 3.47E \quad (9)$$

and

$$3.9 + Y - Z - 0.125I \geq 0.5E/0.144 = 3.47E \quad (10)$$

Rearranging terms, these power constraints for wet and dry seasons, respectively can be written as

$$Y + Z + 0.275I + 3.47E \leq 6.9 \tag{11}$$

$$-Y - Z + 0.125I + 3.47E \leq 3.9 \tag{12}$$

The design sought is assumed to be the one which, while satisfying these constraints, yields the greatest possible present value of net benefits.

4.3.2 Objective Function

The objective function to be maximized is [20]

$$\pi = B_1(E) + B_2(I) - C_1(Y) - C_2(Z) - C_3(E) - C_4(I) \tag{13}$$

where

π = the present value of net benefits in 10^6 dollars

$B_1(E)$ = the present value of an output $E \times 10^9$ kw hr per year in 10^6 dollars

$B_2(I)$ = the present value of an irrigation supply of $I \times 10^6$ acre ft per year in 10^6 dollars

$C_1(Y)$ = the capital cost of building reservoir B to capacity Y in 10^6 dollars

$C_2(Z)$ = the capital cost of building reservoir C to capacity Z in 10^6 dollars

$C_3(E)$ = the capital cost of building the power plant to capacity E per year in 10^6 dollars

and

$C_4(I)$ = the capital cost of building the irrigation system to capacity I per year in 10^6 dollars

The data for all these functions are given. Firstly, capital cost functions are discussed.

$$C_1(Y) = 43Y/(1 + 0.2Y) \tag{14}$$

$$C_2(Z) = 47Z/(1 + 0.3Z) \tag{15}$$

$$C_3(E) = 20.6E - E^2 \tag{16}$$

It can be seen that the larger the reservoir capacity, the higher is the cost. E is obviously restricted to a higher value pf 4×10^9 kw hr as indicated earlier. The derivation of $C_4(I)$ is a bit complicated because of the assumption that only 3×10^6 can be taken for irrigation without pumping. A pumping plant is required if more than this amount is required for irrigation. The data to be assumed are as follows.

The basic cost of the diversion works is \$4,500,000 plus \$44,000,000 per 10^6 acre ft of irrigation water. For $I >$ 3×10^6 acre ft, a pumping plant must be constructed at a cost of \$500,000 plus \$20,000,000 per 10^6 acre ft of water to be pumped. Now if, I_1 denotes non pumped portion and I_2 denotes the pumped portion, then the capital-cost function for the irrigation works becomes -

$$C_4(I) = 44I_1 + 64I_2 + 4.5I_1^* + 0.5I_2^* \tag{17}$$

where

$$I_1 + I_2 = I$$
$$I_1 \leq 3$$

$$I_1^* = 1 \quad \text{if} \quad I_1 > 0$$

and

$$I_2^* = 1 \quad \text{if} \quad I_2 > 0$$

The next step is to formulate the benefit functions $B_1(E)$ and $B_2(I)$. The calculation of each of these proceeds in four stages.

1. Express annual gross benefits as a function of E or I.
2. Express annual operation, maintainance, and replacement (OMR) costs as a function of E or I.
3. Compute annual net benefits by substraction.
4. Compute the present value of net benefits by applying an appropriate present-value factor.

Assuming a planning period of 50 years and a discount rate of 2 1/2 percent, a present value factor of 28.4 is obtained. That is, under these assumptions the present value of a net benefit of \$1 per year for 50 years is \$28.40.

$B_1(E)$ is derived as follows. Assuming that all the energy is on demand, a price of 9 mills per kw hr can be assumed. Therefore, the gross benefits in 10^6 dollars are 9E. OMR costs are assumed to be 0.2E per year. This leads to annual net benefits of 8.8E from electric power. It follows that the present value of electric power operations is

$$B_1(E) = 28.4 \times 8.8E = 250E \tag{18}$$

The calculation of $B_2(I)$ is comparatively more complicated because of following reasons. The introduction of pumping plant causes discontinuity and also marginal value of irrigation water can not be regarded as constant. The datum assumed for computing the gross benefit from irrigation is

$$\text{Marginal gross benefit} = 2.1 + 3.2/(1 + 0.2I) \qquad (19)$$

Integrating this from 0 to I, total gross benefits in 10^6 dollars can be obtained.

$$\text{Total gross benefits:} \quad 2.1I + 36.8\log(1 + 0.2I) \qquad (20)$$

The OMR costs can be assumed to be $0.5I_1 + 1.56I_2$. Thus,

$$\text{Annual net benefits} = 1.6I_1 + 0.54I_2 + 36.8 \log (1 + 0.2I) \qquad (21)$$

Finally, applying the present-value factor:

$$B_2(I) = 45.4I_1 + 15.3I_2 + 1045\log(1 + 0.2I) \qquad (22)$$

The objective function can now be computed by adding expressions (14), (15), (16), (17), (18) and (22). Hence:

$$\begin{aligned} \pi = {} & 229.4E + E^2 + 1.4I_1 - 48.7I_2 + 1045\log(1 + 0.2I) \\ & - 4.5I_1^* - 0.5I_2^* - 43Y/(1 + 0.2Y) - 47Z/(1 + 0.3Z) \qquad (23) \end{aligned}$$

4.3.3 Solution

This problem requires finding the maximum of a nonlinear, and indeed discontinuous, function of some decision variables that are related by a number of linear constraints. Though simple, such a formalization is appropriate for the initial analysis of many water-resource design problems.

Apart from the complexity of the objective function, a problem of this sort can be solved straight forwardly by the well-known computational technique of linear programming. As the objective function is 'separable', the replacement of nonlinear expressions by linear segments is possible. This will, of course, result in introduction of many new variables. Such a procedure, though useful for an ultimate application of linear programming is not recommended because that will considerably increase the computation.

Instead of attempting the simplification of the objective function, the applicability of the gradient projection method was tested. A brief description of computational aspects follows.

The various tolerances and limits required for the application of the method were selected in the same manner as discussed earlier. Specifically, the following values were assumed.

$\varepsilon_1 = .0005$	MXNU = 20	$\gamma_{max} = 0$
$\varepsilon_2 = .005$	MXRN = 3	$\eta_{max} = 0$
$\varepsilon_3 = .005$	$\beta_{max} = 1$	$\Upsilon_{max} = 10$

The problem being one of maximization was solved by the gradient projection maximization algorithm directly.

Number of runs were tried, each having different initial values for control variables. In selecting these initial starting values, one has to be quite reasonable. If values selected are quite far off the optimal values, a constraint violation may result. This would require a re-inversion to continue and more the number of re-inversions, less accurate is the solution.

Therefore, the following approach was adopted. For each variable a range of values was selected. An initial point then was selected having the value for each variable within its specified range. This approach turned out to be more succeful than one of giving uniform values to all variables. Results of these different runs are summarized in Table 22.

The optimal solution is presented in Table 23. The solution to the same problem was obtained in [20], by the method of chordal approximation. Table 24 indicates the details of that solution.

The gradient projection method did not encounter any difficulty in reaching the optimal solution, though the objective function involved logarithmic expressions. The method of chordal approximation reduces the accuracy of the overall model, because approximate functions are introduced. Of course, the loss of accuracy can be compensated for by employing a finer grid for the straightline approximations. No doubt, the better approach will be to treat the nonlinearity directly and this is done quite efficiently by the gradient projection method.

4.4 A MODEL WITH MORE SEASONS AND PREDICTABLE HYDROLOGY

The problem in the previous section indicates that mathematical programming is quite helpful in finding the optimal designs and operating procedures for a fairly complex water-resource system provided a short segment of time can be considered in isolation. This limitation arises from the fact that the programming computations rapidly become more difficult as the number of constraints to be handled increases. An increase in

Table 22. Optimal π Obtained with Different Starting Control Variable Values Two-Period Problem

Serial No.	Initial Conditions	π $\times 10^6$ (\$)	$\lVert P_q\ g \rVert$ at Optimal	Total No. of Functional Evaluations	Total No. of Iterations	Execution Time in Seconds
1	$E = 1.000$ $I_1 = 1.000$ $I_2 = 1.000$ $Y = 1.000$ $Z = 1.000$	488.7595	0.000375	4	3	5.4[illegible]
2	$E = 1.400$ $I_1 = 2.800$ $I_2 = 0.000$ $Y = 0.000$ $Z = 1.000$	488.8337	0.000503	3	2	4.67
3	$E = 2.000$ $I_1 = 2.000$ $I_2 = 0.000$ $Y = 0.000$ $Z = 2.000$	488.8334	0.000472	2	1	6.07

Table 22. (Cont'd)

Serial No.	Initial Conditions	π_6 x 10^6 (\$)	$\|P_q g\|$ at Optimal	Total No. of Functional Evaluations	Total No. of Iterations	Execution Time in Seconds
4	$E = 1.800$ $I_1 = 1.800$ $I_2 = 1.800$ $Y = 1.800$ $Z = 1.800$	488.7775	0.000377	3	2	4.70
5	$E = 1.200$ $I_1 = 1.200$ $I_2 = 1.200$ $Y = 1.200$ $Z = 1.200$	488.7092	0.000610	3	2	5.45

Table 23. Optimal Solution to the Two-Period Problem

Variable	Description	Optimal Value
Y	Capacity of reservoir B	0 acre-ft
Z	Capacity of reservoir C	1.2750×10^6 acre-ft
E	Output of energy from the power plant	1.3831×10^9 kw hr
I	Supply of irrigation water from the system	3.000×10^6 acre-ft

Maximum Net Benefits = $488,833,700

Table 24. Optimal Solution to the Two-Period Problem by Maass et.al. [20]

Variable	Description	Optimal value
Y	Capacity of reservoir B	0.0000 acre-ft.
Z	Capacity of reservoir C	1.2750×10^6 acre-ft.
E	Output of energy from the power plant	1.3834×10^9 kw-hr.
I	Supply of irrigation water from the system	3.0000×10^6 acre-ft.

Maximum Net Benefits = $494,600,000

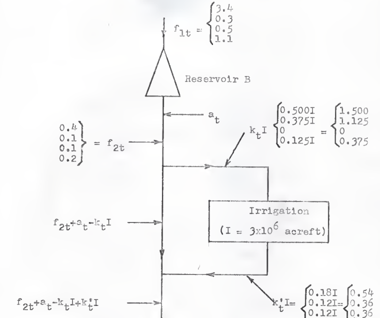


Fig. 9. Sketch of Configuration used in Four-Period Problem [20]

number of periods increases the number of constraints.

In the present section a problem involving four periods and requiring an overyear storage is considered. This latter feature adds considerable complication to the analysis. Such a situation is found in regions in which natural inflow is likely to be deficient in some years, so that overyear storage is required for efficient operation. Unpredictibility of flows is not considered. In other words, the problem is of designing a system that makes efficient use of a sequence of unequal but predictable flows.

The configuration shown in Fig. 9 is a modification of one dealt with in the previous section. In order to emphasize the effect of more periods, certain complicating features of last section are dropped. The schematic representation indicates the reservoir C has been suppressed and the irrigation supply has been set at 3×10^6 acre ft per year.

4.4.1 Constraints

The first set of constraints requires that the volume of water released from the reservoir must be sufficient to meet the period's irrigation demand, where the latter is a preassigned proportion, k_t, of the annual irrigation demand. This requirment can be written as

$$a_t + f_{2t} \geq k_t I \qquad t = 1, 2, 3, 4 \tag{24}$$

where f_{2t} denotes the flow from the tributary which joins the main stem just above irrigation-diversion canal. The natual flows are indicated in the figure while values of k_t are given in

Table 25. Proportions of Irrigation Flow and Energy Demand for Each Period in the Four-Period Problem [20]

Period t	Irrigation demand k_t	Irrigation return flow k'_t	Energy demand c_t
1	0.500	0.18	0.26
2	0.375	0.12	0.29
3	0	0.12	0.24
4	0.125	0.08	0.21
Total	1.000	0.50	1.00

Table 25.

The second constraint is that the volume of water released during any period can not exceed the contents of the reservoir at the beginning of the period plus the flow in reservoir during the period. Let S_t denote the contents of the reservoir at the beginning of period I. Then the constraint is

$$a_t \leq S_t + f_{1t}, \qquad t = 1, 2, 3, 4 \tag{25}$$

where f_{1t} is the preassigned natural flow into the reservoir during time period t.

The third constraint is that the contents of the reservoir at the beginning of any period can not exceed the amount left over from the previous period, or

$$S_t \leq S_{t-1} + f_{1,t-1} - a_{t-1} \qquad t\ 2, 3, 4 \tag{26}$$

Also it is necessary to include a constraint which makes sure that the contents of the reservoir at the end of any period can not exceed the capacity of the reservoir, or

$$S_t + f_{1,t} - a_t \leq Y \qquad t = 1, 2, 3, 4 \tag{27}$$

The last two requirements ensure that the contents of the reservoir at the beginning of any period do not exceed its capacity, hence it is not listed separately.

The last constraint is related to power generation. The flow of water past the power plant must be sufficient to meet the requirements of power generation. As in the previous section, it is assumed, that 9.5×10^6 acre ft of water are required to generate 1×10^9 kw hr of electric energy. The flow available at the power plant is the sum of the flow past the irrigated area, the return flow from the irrigated area, $k_t' I$ (where the return flow coefficients are given in Table 25) and the natural flow from the eastern tributary, f_{3t}. Therefore,

$$a_t + f_{2t} - (k_t - k_t')I + f_{3t} \geq 6.95E_t, \quad t = 1, 2, 3, 4 \tag{28}$$

where E_t is equal to a specified proportion, c_t (shown in Table 25), of the annual energy output.

4.4.2 Objective Function

Since the quantity of irrigation is prespecified, it no longer enters the objective function. The objective function now consists of only two terms: the capital cost of constructing the reservoir B and the present value of the net hydrodectric benefits. The expressions used in previous example are used here. Thus

$$\pi = 229.4E + E^2 - 43Y/(1 + 0.2Y) \tag{29}$$

The purpose is to obtain the optimal values of control variables a_t, S_t, Y and E which satisfy the constraints (24) through (27).

4.4.3 Solution

Again, one approach to solve this nonlinear programming problem involving linear constraints, would be by replacing the nonlinear expressions by linear segments. If a problem would have been linear, a method that drastically reduces the number of constraints that have to be handled at any one time can be used. This is the decomposition principle developed by Dantzig and Wolfe [3]. Such an approach is very useful for a large problem and is not required here.

The gradient projection method was applied to solve this problem. The following values for different limits and tolerances were assumed.

$\varepsilon_1 = .0005$	MXNU = 20	$\gamma_{max} = 0$
$\varepsilon_2 = .005$	MXRN = 3	$\eta_{max} = 0$
$\varepsilon_3 = .0005$	$\beta_{max} = 1$	$\Upsilon_{max} = 10$

The problem being one of maximization, the gradient projection maximization algorithm was applied directly without any changes.

A close look at the details of the problem indicates that it is not necessary to have both S_t and a_t in the optimization. Inequality (26) is, in fact, an equality and hence knowing one of these, other can be determined. Therefore, in actual procedure a_t were not considered but were derived from values of S_t.

Number of runs were tried with different initial starting points. The approach for selecting the starting points was same as illustrated in the previous section. The details of these runs are presented in Table 26. In two of these runs, the value of the projected gradient is not within ε_1, but the value

Table 26. Optimal π obtained with Different Starting Control Variable Values Four-Period Problem

Serial No.	Initial Conditions	π x 10^6($)	$\lVert P_q\ g \rVert$ at Optimal	Total No. of Functional Evaluations	Total No. of Iterations	Execution Time in Seconds
1	S_1 = 1.200 S_2 = 3.200 S_3 = 0.700 S_4 = 0.000 Y = 3.200 E = 1.300	200.6827	0.000009	5	4	8.57
2	S_1 = 1.100 S_2 = 3.100 S_3 = 0.600 S_4 = 0.000 Y = 3.100 E = 1.300	200.6774	0.000010	5	4	9.15
3	S_1 = 1.000 S_2 = 3.000 S_3 = 0.500 S_4 = 0.000 Y = 3.000 E = 1.100	200.6741	0.000010	5	4	9.55

Table 26. (Cont'd)

Serial No.	Initial Conditions	π $\times 10^6$ (\$)	$\|P_q\, g\|$ at Optimal	Total No. of Functional Evaluations	Total No. of Iterations	Execution Time in Seconds
4	$S_1 = 0.50$ $S_2 = 2.90$ $S_3 = 0.40$ $S_4 = 0.00$ $Y = 2.90$ $E = 1.00$	200.6709	0.467178	6	5	11.47
5	$S_1 = 0.85$ $S_2 = 2.85$ $S_3 = 0.35$ $S_4 = 0.00$ $Y = 2.85$ $E = 0.95$	200.6707	0.467178	6	5	11.8[illegible]

Table 27. A Typical Convergence Rate
Four-Period Problem (Table 26, S. No. 1)

Iteration No.	π	$\lVert P_q g \rVert$	Total No. of Functional Evaluations	No. of Constraints in Basis
0	163.1115	1.609266	1	1
1	193.5994	0.958116	2	2
2	194.8405	0.431042	3	3
3	196.8477	0.071369	4	4
4	200.6827	0.000009	5	5

Table 28. Optimal Solution to the Four-Period Problem

Optimal Levels of Decision Variables						
Variable	Value in All Period	Value in Periods 1	2	3	4	Units
S_t		0.675	2.975	0.474	0.000	x 10^6 acre-ft
a_t		1.100	3.001	0.774	0.425	x 10^6 acre-ft
E_t		0.317	0.353	0.292	0.256	x 10^9 kw hr
Y	2.975					x 10^6 acre-ft
E	1.218					x 10^9 kw hr

Maximum Net Benefits = $200,682,700

Table 29. Optimal Solution to the Four-Period Problem by Maass et.al. [20]

Optimal Levels of Decision Variables						
Variable	Value in All Periods	Value in Periods				Unit
		1	2	3	4	
S_t		0.688	2.988	0.471	0	x 10^6 acre-ft.
a_t		1.100	2.817	0.971	0.412	x 10^6 acre-ft
E_t		0.316	0.353	0.292	0.256	x 10^9 kw-hr
Y	2.988					x 10^6 acre-ft
E	1.217					x 10^9 kw-hr

Maximum Net Benefits = $173,400,000

of the objective function is very close to optimal. A typical convergence rate is illustrated in Table 27.

The optimal solution is presented in Table 28. The solution to the same problem as obtained in [20] is shown in Table 29. The value of the objective function can not be compared because different expressions were used. But one can observe the closeness of the values of various decision variables in two solutions.

As can be seen, the gradient projection approach proved quite suitable for solution of this problem.

4.5 DISCUSSION

Two problems solved here relate to the situations far off from those encountered in practice. But the approach provides an insight to the problem in initial exploratory stages. It can be seen that even with simplification, a mathematical programming description of a problem can retain the crucial characteristics of a fairly complicated system.

The first problem introduced an approach that can be used when one or two time periods can be isolated from the rest of a project's life. This method was then extended to a problem with more periods and involving an overyear storage. Both these problems involved only deterministic aspects. A more realistic representation of the system would include the unpredictibily of water flows and other random factors.

With judicious selection of various limits, tolerences and initial starting points, the gradient projection method proved quite efficient. The convergence was quite fast and the solutions were quite accurate. The overall approach can be regarded superior to one adopted in [20], because no approximations are involved.

CHAPTER 5

CONCLUSION

The different test systems presented in this work suggest the usefulness of mathematical programing approach in the planning and management of water resource systems. These experiments demonstrate the fact that the crucial characteristics of a fairly complicated system can be retained in a mathematical-programming description without redering the model unduly difficult.

The dissolved oxygen (DO) model presented for water quality management can be used to determine the minimum total cost associated with any particular set of minimum allowable DO concentrations in a river basin. It can also provide the useful information regarding sensitivity of both the cost and actual minimum DO concentrations in the reaches to changes in minimum allowable DO concentrations in any particular reach.

The example discussed in Chapter 4 were connected with water quantity. Though the problems are simple, the approach provides an insight to the problem of water resources planning in initial exploratory stages. This is the beginning only. If more complexity is desired the benefits from the flood control, recreation, urban water supply etc. can be incorporated.

All models considered were deterministic in nature. A more realistic approach would be to consider the stochastic nature inherent in inflows and water-demands.

All the problems fell into the nonlinear programming class characterized by linear constraints. The gradient projection method developed by Rosen [24] proved quite efficient in solution of these problems. A rapid convergence rate was observed in solutions of all the problems. This implies computational efficiency in terms of computer time for a prescribed accuracy.

No doubt, the success of the method is largely contingent with a judicious selection of various limits and tolerences. Also the results obtained are very dependent on the design of the problem to be solved. Therefore, it is very essential to have basic physical knowledge of the system. The program can be used to solve problems with a large number of variables (about 60) and constraints (about 150) but it is felt, that it will be more efficient for fewer variables and constraints.

REFERENCES

1. Buras, N., "A Three-Dimensional Optimization Problem in Water-Resources Engineering," Operational Research Quarterly, 16, 419 (1965).

2. Camp, T. R., Water and Its Impurities, Reinhold publishing Co., New York (1963).

3. Dantzig, G. B., and P. Wolfe, "Decomposition Principle for Linear Programming", J. Operations Research, 8, 101 (1960).

4. Deininger, R. A., "Water Quality Management: The Planning of Economically Optimal Pollution Control Systems", Ph. D. Thesis, Northwestern University (1965).

5. Dobbins, W. E., "BOD and Oxygen Relationships in Streams", J. Sanitary Eng. Div., Am. Soc. Civil Eng., 90, No. 3 (1964).

6. Dorfman, R., "Formal Models in the Design of Water Resource Systems", Water Resources Research, 1, 329 (1965).

7. Draper, N. R., and H. Smith, Applied Regression Analysis, John Wiley & Sons, Inc., New York (1966).

8. Fair, G. M., J. C. Geyer, D. A. Okun, Water and Wastewater Engineering, Vol. I. Water Supply and Wastewater Removal, John Wiley & Sons, Inc., New York (1966).

9. Goldfarb, D. and L. Lapidus, "Conjugate Gradient Method for Nonlinear Programming Problems with Linear Constraints", I & EC Fundamentals, 7, 142 (1968).

10. Hadley, G., Linear Algebra, Addison-Wesley Publishing Co. Inc. (1961).

11. Hadley, G., Nonlinear and Dynamic Programming, Addison-Wesley Publishing Co. Inc. (1964).

12. Hall, W. A., "Optimum Design of a Multiple-Purpose Reservoir," J. Hydraulics Div., Am. Soc. Civil Eng., 90, No. HY4, 141 (1964).

13. Hall, W. A., and N. Buras, "The Dynamic Programming Approach to Water-Resources Development", J. Geophysical Research, 66, 517 (1961).

14. Hall, W. A., W. S. Butcher, and A. Esogbue, "Optimization of the Operation of a Multiple-Purpose Reservoir by Dynamic Programming", Water Resources Research, 4, 471 (1968).

15. Hillier, F. S., and G. J. Lieberman, Introduction to Operations Research, Holden-Day, Inc. California (1967).

16. Hufschmidt, M. M., and M. B. Fiering, Simulation Techniques for Design of Water-Resource Systems, Harvard University Press. Cambridge, Mass (1966).

17. Kerri, K. D., "A Dynamic Model for Water Quality Control", J. Water Pollution Control Federation, 39, 772 (1967).

18. Liebman, J. C., and W. R. Lynn, "The Optimal Allocation of Stream Dissolved Oxygen," Water Resources Research, 2, 581 (1966).

19. Loucks, D. P., C. S. Revelle, and W. R. Lynn, "Linear Programming Models for Water Pollution Control", Management Science, 14, B-166 (1967).

20. Maass, A., M. Hufschmidt, R. Dorfman, H. A. Thomas, S. Marglin, and G. M. Fair, Design of Water Resource Systems, Harvard University Press, Cambridge, Mass. (1962).

21. Merrill, R. P., "Gradient Projection 7090", SHARE Program Library-Report No. 3430 (1966).

22. Meyer, D. L., "Modeling and Optimization of a Water Resource Control System", Master's Thesis, Kansas State University (1968).

23. Revelle, C. S., D. P. Loucks, and M. R. Lynn, "A Management Model for Water Quality Control", J. Water Pollution Control Federation, 39, 1164 (1967).

24. Rosen, J. B., "The Gradient Projection Method for Nonlinear Programming Part I. Linear Constraints", J. SIAM, 8, 181 (1960).

25. Rosen, J. B., "The Gradient Projection Method for Nonlinear Programming Part II. Nonlinear Constraints", J. SIAM. 9, 514 (1961).

26. Sobel, M. J., "Water Quality Improvement Programming Problems," Water Resources Research, 1, 477 (1965).

27. Streeter, H. W., and E. B. Phelps, A Study of the Pollution and Natural Purification of the Ohio River, Public Health Bulletin, 146, U. S. Public Health Service, Washington, D. C. (1925).

28. Thomann, R. V., "Mathematical Model for Dissolved Oxygen", J. Sanitary Eng. Div., Am.Soc. Civil Eng., 85, SA-5 (1963).

29. Wilde, D. J., Optimum Seeking Methods, Prentice-Hall, Inc. Englewood Cliffs, N.J. (1964).

APPENDIX A

GP NOMENCLATURE AND SUBPROGRAMS

NOMENCLATURE

a_{ij}	coefficient of input constraint
b_i	right hand side of a constraint
d_{ij}	element of the inverse matrix
e	number of equalities
F	value of objective function
g	gradient in the direction of increasing F
H_i	a constraint (hyperplane)
k	total number of constraints
m	number of variables
MXNU	maximum number of steps
MXRN	maximum number of re-inversions
n_i	constraint vector
N_k	constraint matrix
N_q	basis
$(N_q^T N_q)^{-1}$	inverse matrix
Pg	projected gradient
Pn	projected constraint vector
q	constraints in the basis
q^*	constraints added to the initial basis
u	linearly dependent constraints
v	constraints not in the basis with $\lambda = 0$
w	constraints not in the basis with $\lambda > 0$
x	variable vector
Z	unit vector in the direction of step
β	gradient interpolations for $z^T g = 0$

γ	gradient interpolations to increase F
ε_1	gradient tolerance
ε_2	constraints tolerance
ε_3	linear dependence tolerance
η	interior steps
λ_i	normal distance to a constraint
Υ	step counter
σ	temporary flag
Υ	step length

SUBPROGRAMS

NAME	FUNCTION
AMDA	Calculates lambdas ($\lambda(x)$).
CLASS	Classifies constraints not in the basis to V and W.
COMMAT	Carries out necessary matrix computations when a constraint is dropped from the basis.
FUNCT	Computes the value of the objective function, F, and its gradient, g.
MATCOM	Carries out necessary matrix computations when a constraint is added to the basis.

APPENDIX B

COMPUTER FLOW DIAGRAMS

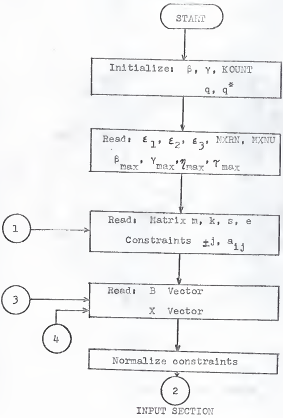
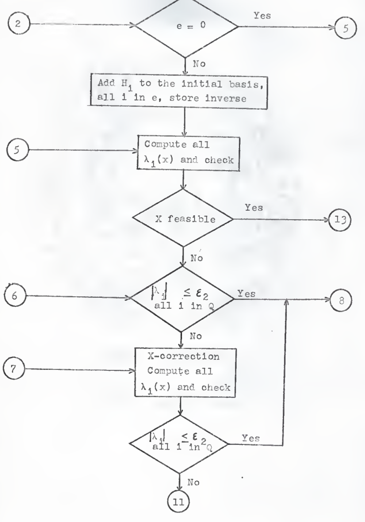
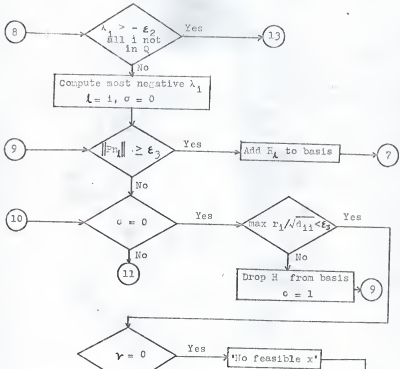


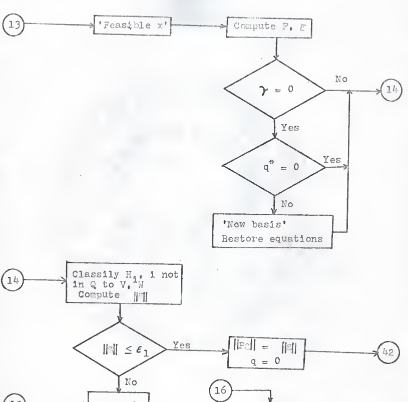
Fig. 10. Flow Chart for the Main Program of the Gradient Projection Method.

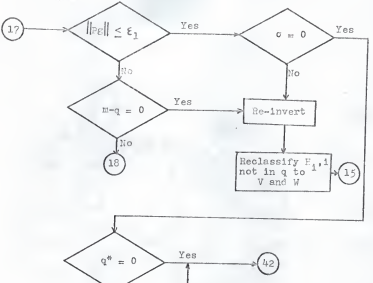


2
e = 0
Yes
5
No
Add H_i to the initial basis, all i in e, store inverse
5
Compute all $\lambda_i(x)$ and check
X feasible
Yes
13
No
6
$|\lambda_i| \leq \varepsilon_2$ all i in Q
Yes
8
No
7
X-correction Compute all $\lambda_i(x)$ and check
$|\lambda_i| \leq \varepsilon_2$ all i in Q
Yes
No
11

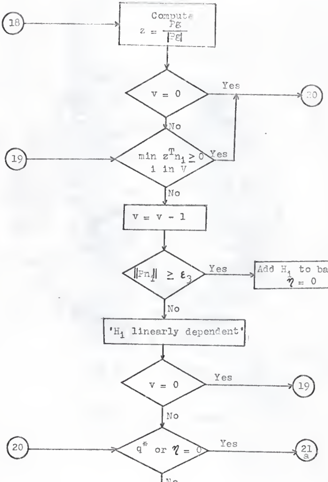


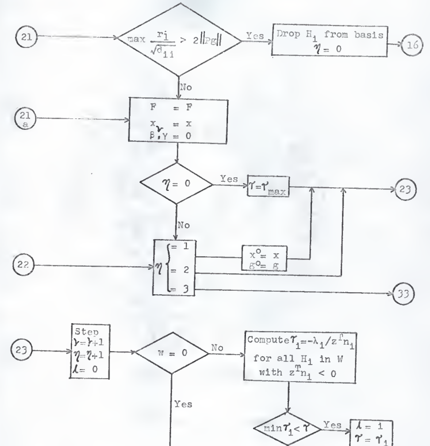
8
$\lambda_i > -\varepsilon_2$ all i not in Q
Yes
13
No
Compute most negative λ_i
$\ell = i$, $\sigma = 0$
9
$\|Pn_\ell\| \geq \varepsilon_3$
Yes
Add H_ℓ to basis
7
No
10
$\sigma = 0$
Yes
$\max r_i/\sqrt{d_{ii}} < \varepsilon_3$
Yes
No
11
No
Drop H from basis
$\sigma = 1$
9
$\gamma = 0$
Yes
'No feasible x'
No
11
'X-correction failed'
Re-invert
12
Re-check $\lambda_i(x)$
6

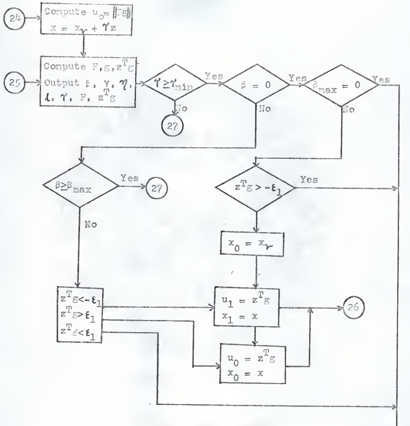

13
'Feasible x'
Compute F, g
γ = 0
No
14
Yes
q* = 0
Yes
No
'New basis'
Restore equations
14
Classify H_i, i not in Q to V, W
Compute ||g||
||g|| ≤ ε_1
Yes
||Pg|| = ||g||
q = 0
42
No
15
σ = 0
16
q = 0
Yes
||Pg|| = ||g||
Pg = g
18
No
Project gradient
17



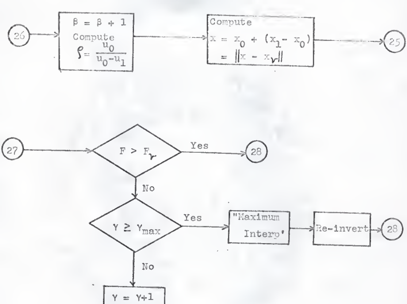
17
$\|Pg\| \leq \varepsilon_1$
Yes
No
$\sigma = 0$
Yes
No
$m-q = 0$
Yes
No
18
Re-invert
Reclassify H_i, i not in q to V and W
15
$q^* = 0$
Yes
No
42
max $\frac{r_i}{\sqrt{d_{ii}}} \leq \varepsilon_1$ i in Q^*
Yes
No
Drop H_i from basis
$\sigma = nz$, $\eta = 0$
16

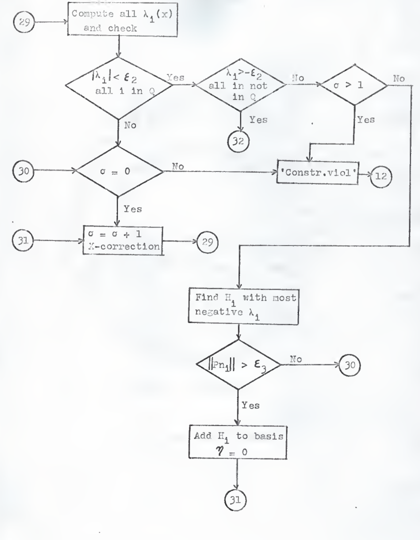

18
Compute
$z = \frac{Pg}{|Pg|}$
$v = 0$
Yes
20
No
19
$\min_{i \text{ in } V} z^T n_i \geq 0$
Yes
No
$v = v - 1$
$\|Pn_i\| \geq \varepsilon_3$
Yes
Add H_i to basis
$\eta = 0$
15
No
'H_i linearly dependent'
$v = 0$
Yes
19
No
20
q^* or $\eta = 0$
Yes
21 a
No
21


21
$\max \frac{r_i}{\sqrt{d_{ii}}} > 2\|Pg\|$
Yes
Drop H_i from basis
$\eta = 0$
16
No
21 a
$F_\gamma = F$
$x_\gamma = x$
$\beta, \gamma = 0$
$\eta = 0$
Yes
$\tau = \tau_{max}$
23
No
22
$\eta = 1$
$= 2$
$= 3$
$x^o = x$
$g^o = g$
33
23
Step
$\nu = \nu + 1$
$\eta = \eta + 1$
$\ell = 0$
$W = 0$
No
Compute $\tau_i = -\lambda_i / z^T n_i$
for all H_i in W
with $z^T n_i < 0$
Yes
$\min \tau_i < \tau$
Yes
$\ell = i$
$\tau = \tau_i$
No
24

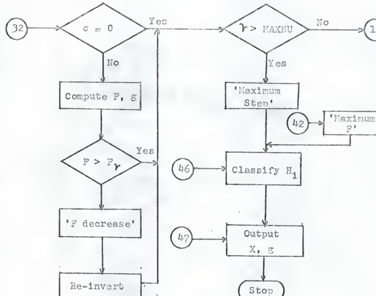


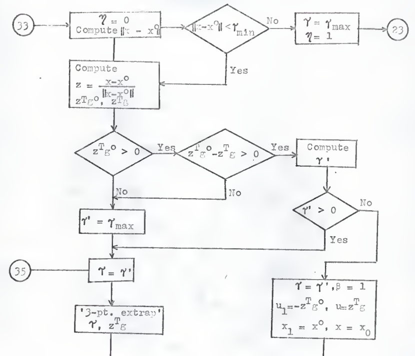
24
Compute $u_0 = \|Fg\|$
$x = x_\gamma + \gamma z$
25
Compute F, g, z^Tg
Output $\beta, \gamma, \eta, \ell, \gamma, F, z^Tg$
$\gamma \geq \gamma_{min}$
Yes
No
27
$\beta = 0$
Yes
No
$\beta_{max} = 0$
Yes
No
$\beta \geq \beta_{max}$
Yes
27
No
$z^Tg > -\varepsilon_1$
Yes
$x_0 = x_\gamma$
$z^Tg < -\varepsilon_1$
$z^Tg > \varepsilon_1$
$z^Tg < \varepsilon_1$
$u_1 = z^Tg$
$x_1 = x$
26
$u_0 = z^Tg$
$x_0 = x$
27


26
β = β + 1
Compute
ρ = u0 / (u0 - u1)
Compute
x = x0 + (x1 - x0)
= ||x - xγ||
25
27
F > Fγ
Yes
28
No
γ ≥ γmax
Yes
"Maximum
Interp'
Re-invert
28
No
γ = γ+1
β = 0
γ = γ/2
28
σ = 0
29



29
Compute all λi(x) and check
|λi| < ε2 all i in Q
Yes
λi > -ε2 all in not in Q
No
σ > 1
No
Yes
32
Yes
No
30
σ = 0
No
'Constr.viol'
12
Yes
31
σ = σ + 1
X-correction
29
Find Hi with most negative λi
||Pni|| > ε3
No
30
Yes
Add Hi to basis
γ = 0
31


32
σ = 0
Yes
No
Compute F, g
F > F_γ
Yes
'F decrease'
Re-invert
γ > MAXNU
No
14
Yes
'Maximum Step'
42
'Maximum F'
46
Classify H_1
47
Output X, g
Stop



3-Point Extrapolation

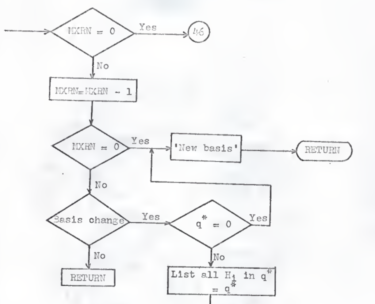
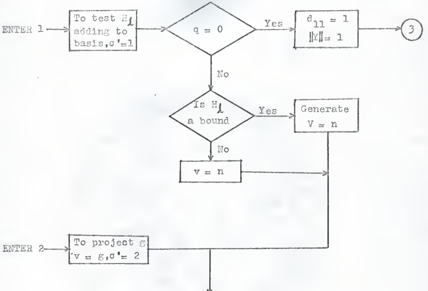
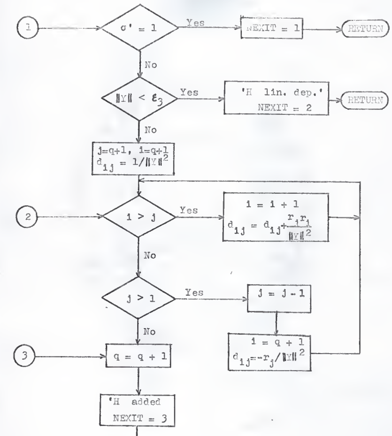


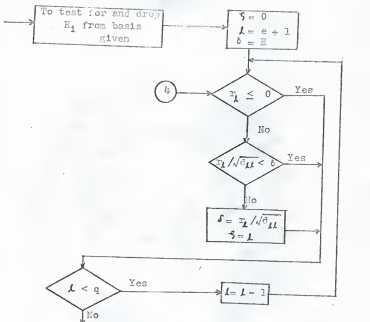
Fig. 11. Flow Chart for Subroutine REINV



MATRIX COMPUTATIONS

Fig. 12. Flow Chart for Subroutine MATCOM


1
σ' = 1
Yes
NEXIT = 1
RETURN
No
‖Y‖ < ε3
Yes
'H lin. dep.'
NEXIT = 2
RETURN
No
j=q+1, i=q+1
d_ij = 1/‖Y‖^2
2
i > j
Yes
i = i + 1
d_ij = d_ij + r_i r_j / ‖Y‖^2
No
j > 1
Yes
j = j - 1
i = q + 1
d_ij = -r_j/‖Y‖^2
No
3
q = q + 1
'H added
NEXIT = 3
RETURN
END



MATRIX COMPUTATIONS

Fig. 13. Flow Chart for Subroutine COMMAT

APPENDIX C

COMPUTER PROGRAM

```
$JOB            DAND,RUN=CHECK,TIME=5,PAGES=50,LINES=60
C
C
C
C         MAIN PROGRAM FOR THE GRADIENT PROJECTION METHOD
C
      DIMENSION X2(10),D(10,10),DN(10,10),A(25,10),G(10),X(10),P(10),
     1IU(10),IW(25),B(25),SD(25),PR(10),Z(10),XO(10),GO(10),X1(10),
     1V(10),JXF(20),IHI(10),R(10),AMBDA(25),Y(10),PG(10),IV(10),
     1DELX(10)
      COMMON Y,PN,AMBDA,M,G,X,F,P,EPSI1,NEXIT,KQ,D,IH,NB,V,A,JXB,IHI,
     1 EPSI3,EPSI2,LD,IU,R,KEQ,MEXIT,IV,KV,B,KMNU,IW,KW,K,MXRN,INV,DN,
     1MU,NETA,INT,LOC
      COMMON/WATER/NFUNC,KOUNT,K1
      EQUIVALENCE (Y,PG),(PN,PGNORM)
C
C         EPSI1=GRADIENT TOLERENCE
C         EPSI2=CONSTRAINT TOLERENCE
C         EPSI3=LINEAR DEPENDENCE TOLERENCE
C         TMAX=TAU MAX
C         MUMAX=BETA MAX
C         INTMAX=GAMMA MAX
C         MAXE=NETA MAX
C         MXRN=MAXIMUM   OF REINVERSIONS
C         MXNU=MAXIMUM   OF STEPS
C
 1000 FORMAT(4(I5))
 1020 FORMAT(10I5)
 1040 FORMAT(6F12.6)
 1050 FORMAT(6F12.6)
 1002 FORMAT ('     # OF VARIABLES'6XI2)
 1003 FORMAT ('     # OF CONSTRAINTS ' 3XI2)
 1004 FORMAT ('     # OF BOUNDS ' 8XI2) .
 1005 FORMAT ('     # OF EQUALITIES' 5XI2)
 1006 FORMAT ('     BOUNDS')
 1007 FORMAT ('     COEFFICIENTS OF CONSTRAINTS')
 1008 FORMAT ('     NORMALISED COEFFICIENTS OF CONSTRAINTS')
 1009 FORMAT ('     NORMALISED LIMITING VALUES OF CONSTRAINTS')
 1010 FORMAT ('     LIMITING VALUES OF CONSTRAINTS')
 1011 FORMAT ('    INITIAL X-VECTOR')
 1012 FORMAT(4F12.6)
 1013 FORMAT(5I4)
 1014 FORMAT ('     EPSI1=',F10.4,'  EPSI2=',F10.4,'  EPSI3=',F10.4,'  TA
     1U MAX=',F5.2)
 1015 FORMAT ('     BETA MAX=',I3,'  GAMMA MAX=',I3,'  NETA MAX=',I3,'  M
     1XRN=',I3,'  MXNU=',I3)
C
 1016 FORMAT ('     **GRADIET PROJECTION METHOD FOR NONLINEAR PROGRAMMING
     1 PROBLEMS WITH LINEAR CONSTRAINTS**')
      PRINT 1016
C         READ AND PRINT LIMITS AND TOLERENCES
C
      READ 1012,EPSI1,EPSI2,EPSI3,TMAX
      READ 1013,MUMAX,INTMAX,MAXE,MXRN,MXNU
      PRINT 1014,EPSI1,EPSI2,EPSI3,TMAX
      PRINT 1015,MUMAX,INTMAX,MAXE,MXRN,MXNU
C
C         INITIALIZE
C
   10 INT=0
```

```
      KQ=0
      KEQ=0
      LOC=0
      INV=1
   11 MU=0
      KOUNT=0
      LLL=0
      KI=0
      TMIN=.0001*TMAX
C
C          ' NEW MATRIX '    READ M,K,NB,NE
C
      READ 1000,M,K,NB,NE
      PRINT 1002,M
      PRINT 1003,K
      PRINT 1004,NB
      PRINT 1005,NE
      NBP1=NB+1
      KMNB=K-NB
      IF(NB)24,24,22
C
C          READ AND PRINT SUBSCRIPTS FOR BOUNDS
C
   22 READ 1020,(JXB(I),I=1,NB)
      PRINT 1006
      PRINT 1020,(JXB(I),I=1,NB)
      IF(KMNB)27,27,24
C
C          READ AND NORMALIZE CONSTRAINTS
C
   24 READ 1040,((A(I,J),J=1,M),I=1,KMNB)
      PRINT 1007
      PRINT 1040,((A(I,J),J=1,M),I=1,KMNB)
      DO 26 I=1,KMNB
      SD(I)=0.
      DO 25 J=1,M
   25 SD(I)=SD(I)+A(I,J)**2
      SD(I)=SQRT(SD(I))
      DO 26 J=1,M
   26 A(I,J)=A(I,J)/SD(I)
      PRINT 1008
      PRINT 1040, ((A(I,J),J=1,M),I=1,KMNB)
C
C          READ SUBSCRIPTS FOR EQUALITIES, ADD H TO INITIAL BASIS FOR ALL
C          I IN E,  STORE INVERSE
C
   27 IF(NE) 30,30,28
   28 KQ=0
      IH=NB+1
      DO 610 I=1,NE
  610 IW(I)=0
      CALL MATCOM(1)
      IHI(1)=IH
  615 IF(NE-KQ)675,675,620
  620 N=1
      INDEX=0
      DELTA=EPSI3
  625 N=N+1
      IF(IW(N)) 630,630,650
  630 DO 635 J=1,M
```

```
  635 V(J)=A(N,J)
      CALL MATCOM(2)
      PN=0.0
      DO 640 J=1,M
  640 PN=PN+Y(J)**2
      PN=SQRT(PN)
      IF(PN-DELTA) 650,650,645
  645 DELTA=PN
      INDEX=N
  650 IF(N-NE) 625,655,655
  655 IF(INDEX) 675,675,660
  660 IH=INDEX+NB
      CALL MATCOM(1)
      IW(INDEX)=777
      IHI(KQ)=IH
      GO TO 615
  675 KEQ=KQ
      DO 680 I=1,KQ
      DO 680 J=1,KQ
  680 DN(I,J)=D(I,J)
C
C         READ B-VECTOR
C
   30 READ 1040, (B(I),I=1,K)
      PRINT 1010
      PRINT 1040,(B(I),I=1,K)
      IF(KMNB) 40,40,31
   31 DO 32 I=1,KMNB
      J=I+NE
   32 B(J)=B(J)/SD(I)
      PRINT 1009
      PRINT 1040,(B(I),I=1,K)
   40 CONTINUE
   41 NFUNC=0
C
C         READ X-VECTOR
C
   45 READ 1040,(X(I),I=1,M)
      PRINT 1011
      PRINT 1040,(X(I),I=1,M)
   51 CONTINUE
C
C         COMPUTE ALL LAMBDA(X) AND CHECK
C
   52 CALL AMDA
      IF(NB) 54,54,64
   64 DO 53 I=1,NB
      IF(AMBDA(I)+EPSI2) 59,53,53
   53 CONTINUE
   54 IF(NBP1-K)65,65,130
   65 DO 57 J=NBP1,K
      IF(J-NBP1-NE) 55,56,56
   55 IF(ABS(AMBDA(J))-EPSI2) 57,57,59
   56 IF(AMBDA(J)+EPSI2) 59,57,57
   57 CONTINUE
      GO TO 130
   59 PRINT 1040,(X(I),I=1,M)
      PRINT 1040,(AMBDA(J),J=1,K)
      IF(KQ) 80,80,60
   60 DO 61 I=1,KQ
```

```
      J=IH1(I)
      IF(ABS(AMBDA(J))-EPSI2) 61,61,70
   61 CONTINUE
      GO TO 80
C
C         X-CORRECTION   COMPUTE ALL LAMBDA(X) AND CHECK
C
   70 DO 71 J=1,M
   71 V(J)=X(J)
      CALL MATCOM(3)
      PRINT 1040,(Y(I),I=1,M)
      DO 72 J=1,M
   72 X(J)=Y(J)
      CALL AMDA
      IF(KO) 80,80,66
   66 DO 62 I=1,KO
      J=IHI(I)
      IF(ABS(AMBDA(J))-EPSI2) 62,62,110
   62 CONTINUE
   80 DO 83 I=1,K
      IF(AMBDA(I)+EPSI2) 85,83,83
   83 CONTINUE
      GO TO 130
C
C         FIND MOST NEGATIVE LAMBDA
C
   85 N=1
      SIGMA=AMBDA(1)
      IF(K-2) 89,86,86
   86 DO 88 I=2,K
      IF(AMBDA(I)-SIGMA) 87,88,88
   87 N=I
      SIGMA=AMBDA(I)
   88 CONTINUE
   89 SIGMA=0.0
      IH=N
   90 CALL MATCOM(1)
      GO TO (100,100,95),NEXIT
   95 KOM1=KO-1
      IF (KOM1) 97,97,94
   94 DO 96 I=1,KOM1
   96 AMBDA(I)=0.0
   97 INV=1
      GO TO 70
  100 IF(SIGMA) 110,101,110
  101 CALL COMAT(EPSI3)
      GO TO (105,103),NEXIT
C
C         DROP H(I) FROM BASIS
C
  103 SIGMA=1.0
      INV=1
      IH=N
      GO TO 90
 `105 IF(KOUNT) 106,106,110
C
C         NO FEASIBLE X
C
  106 PRINT 5001
 5001 FORMAT ('     NOT FEASIBLE X')
```

```
      GO TO 120
C
C         X-CORRECTION FAILED
C
  110 PRINT 5002
 5002 FORMAT ('    X-CORRECTION FAILED')
C
C         RE-INVERT          RE-CHECK LAMBDAS
C
  120 CALL REINV
      IF(NEXIT) 60,60,460
C
C         FEASIBLE X   PRINT X AND LAMBDAS
C
  130 PRINT 5003
 5003 FORMAT(' FEASIBLE X')
      PRINT 1040,(X(J),J=1,M)
      PRINT 5004
 5004 FORMAT ('  LAMDAS')
      PRINT 1040,(AMEDA(J),J=1,K)
      CALL FUNCT(X,F,G,KQ)
      IF(KOUNT) 133,133,140
  133 IF(KQ-KEQ) 9999,138,134
  134 CONTINUE
  138 CONTINUE
      NETA=0
  140 CALL CLASS
      INV=1
C
C         CLASSIFY H, FOR I NOT IN Q TO V,W
C
      GNORM=0
      DO 141 J=1,M
  141 GNORM=GNORM+G(J)**2
      GNORM=SQRT(GNORM)
      PRINT 5006,GNORM,(G(J),J=1,M)
 5006 FORMAT('     GRADIENT',F12.6/(6F12.6))
      IF(GNORM-EPSI1) 142,142,150
  142 PGNORM=GNORM
      KQ=0
      GO TO 420
  150 SIGMA=0.0
  160 IF(KQ) 161,161,170
  161 PGNORM=GNORM
      DO 162 J=1,M
  162 PG(J)=G(J)
      GO TO 180
C
C         PROJECT GRADIENT
C
  170 DO 169 J=1,M
  169 V(J)=G(J)
      CALL MATCOM(2)
      PGNORM=0.0
      DO 171 J=1,M
  171 PGNORM=PGNORM+PG(J)**2
      PGNORM=SQRT(PGNORM)
      PRINT 5007,PGNORM,(PG(J),J=1,M)
 5007 FORMAT('   PROJ GRAD',F12.6/(6F12.6))
      IF(PGNORM-EPSI1) 175,175,172
```

```
  172 IF(M-KC) 9999,173,180
C
C         PROJECTION NOT ZERO AT VERTEX
C
  173 PRINT 5008
 5008 FORMAT(' PROJ NOT ZERO AT VERTEX')
  174 CALL REINV
      IF(NEXIT) 1745,1745,460
C
C         RECLASSIFY H,FOR I NOT IN Q TO V,W
C
 1745 CALL CLASS
      GO TO 150
  175 IF(SIGMA) 176,177,176
C
C         PROJECTION ZERO AFTER DROP
C
  176 PRINT 5009
 5009 FORMAT (' PROJ.ZERO AFTER DROP')
      GO TO 174
  177 IF(KQ-KEQ) 9999,420,178
  178 CALL COMMAT(EPSI1)
      GO TO (420,179) ,MEXIT
C
C         DROP H(I) FROM BASIS
C
  179 NETA=0
      INV=1
      SIGMA=5.0
      GO TO 160
  180 DO 181 J=1,M
  181 Z(J)=PG(J)/PGNORM
      LDC=0
      IF(KV) 9999,188,190
  188 SUM=2.*PGNORM
      GO TO 200
  190 KEQP=KEQ+1
      IF(KEQP-KQ) 1903,1903,1902
 1903 DO 1901 J=KEQP,KQ
 1901 RR(J)=R(J)
 1902 L=0
      SUM=2.*PGNORM
      INDEX=0
      DELTA=0.0
  191 L=L+1
      IF(IV(L)-NB) 1912,1912,1915
 1912 KK=IV(L)
      J=JXB(KK)
      IF(J) 1913,9999,1914
 1913 J=-J
      ZN=-Z(J)
      GO TO 1925
 1914 ZN=Z(J)
      GO TO 1925
 1915 KK=IV(L)-NB
      ZN=0.0
      DO 192 J=1,M
  192 ZN=ZN+Z(J)*A(KK,J)
 1925 IF(ZN-DELTA) 193,1935,1935
  193 INDEX=IV(L)
```

```
      LL=L
      DELTA=7N
 1935 IF(KV-L) 194,194,191
  194 IF(INDEX) 9999,1945,195
  195 KV=KV-1
      IF(KV-LL) 1995,198,198
  198 DO 199 I=LL,KV
  199 IV(I)=IV(I+1)
 1995 IH=INDEX
      CALL MATCOM(1)
      GO TO (197,197,196),NEXIT
  196 NETA=0
      INV=1
      LLL=0
      GO TO 150
  197 IF(KV) 9999,1997,190
 1997 IF(KEQP-KQ) 1998,1998,1945
 1998 DO 1999 J=KEQP,KQ
 1999 R(J)=RR(J)
 1945 IF(LLL-230)200,1946,200
 1946 LLL=0
      GO TO 230
  200 IF(NETA) 201,204,201
  201 IF(KQ-KEQ)202,204,202
  202 CALL COMMAT(SUM)
      GO TO (204,203),MEXIT
  203 NETA=0
      INV=1
      GO TO 160
  204 FY=F
      MU=0
      INT=0
C
C         PRINT SUMMARY
C
      KQS=KQ-KEQ
      PRINT 5010,KQ,NE,KQS,LDC,KV,KW
 5010 FORMAT(4X2H Q,I2,3H F ,I2,4H Q* ,I2,3H U ,I2,3H V ,I2,3H W ,I2)
      IF(KQ)212,212,211
  211 PRINT 9211,(IHI(J),J=1,KQ)
  212 IF(LDC)214,214,213
  213 PRINT 9213,(IU(J),J=1,LDC)
  214 IF(KV)216,216,215
  215 PRINT 9215,(IV(J),J=1,KV)
  216 IF(KW)218,218,217
  217 PRINT 9217,(IW(J),J=1,KW)
 9211 FORMAT('     Q'/10I3)
 9213 FORMAT('     U'/10I3)
 9215 FORMAT('     V'/10I3)
 9217 FORMAT('     W'/10I3)
  218 DO 205 I=1,M
  205 Y(I)=X(I)
      IF(NETA)9999,206,207
  206 T=TMAX
      GO TO 230
  207 CONTINUE
  210 CONTINUE
  220 GO TO (222,230,330),NETA
  222 DO 223 J=1,M
      XO(J)=X(J)
```

```
 223 G0(J)=G(J)
 230 NETA=NETA+1
     KOUNT=KOUNT+1
     L=0
     IF(KW)9999,240,231
 231 DO 239 I=1,KW
     J=IW(I)
     IF(J-NB)232,232,235
 232 KK=JXB(J)
     IF(KK) 233,9999,234
 233 KK=-KK
     ZN=-Z(KK)
     GO TO 2365
 234 ZN=Z(KK)
     GO TO 2365
 235 KK=J-NB
     ZN=0.0
     DO 236 N=1,M
 236 ZN=ZN+Z(N)*A(KK,N)
2365 IF(ZN) 237,239,239
 237 TI=-AMBDA(J)/ZN
     IF(TI-T)238,239,239
 238 T=TI
     L=J
 239 CONTINUE
     PRINT 5011,KOUNT
5011 FORMAT (' STEP'I4)
 240 DO 241 J=1,M
 241 X(J)=Y(J)+T*Z(J)
     XMU0=PCNORM
 250 CALL FUNCT(X,F,G,KQ)
     ZG=0.0
     DO 251 J=1,M
 251 ZG=ZG+Z(J)*G(J)
     PRINT 5012,MU,INT,NETA,L,T,F,ZG
5012 FORMAT(4X5HBETA=,3XI3,4X6HGAMMA=,3XI3,4X5HNETA=,3XI3,3X2HH=,3XI3,4
    1X2HT=,F12.6,4X2HF=,F12.6,4X3HZG=,F12.6)
     IF(T-TMIN)270,252,252
 252 IF(MU)9999,253,257
 253 IF(MUMAX)9999,270,254
 254 IF(ZG+EPSI1)255,255,270
 255 DO 256 J=1,M
 256 X2(J)=Y(J)
     GO TO 8255
 257 IF(MU-MUMAX)258,270,270
 258 IF(ABS(ZG)-EPSI1)8255,8255,8250
8250 XMU0=ZG
     DO 8251 J=1,M
8251 X2(J)=X(J)
     GO TO 260
8255 XMU1=ZG
     DO 8256 J=1,M
8256 X1(J)=X(J)
 260 MU=MU+1
     RHO=XMU0/(XMU0-XMU1)
     T=0.0
     DO 261 J=1,M
     X(J)=X2(J)+RHO*(X1(J)-X2(J))
 261 T=T+(X(J)-Y(J))**2
     T=SQRT(T)
```

```
      GO TO 250
 270  IF(F-FY)271,271,280
 271  IF((NT-INTMAX)272,273,273
 272  INT=INT+1
      MU=0
      T=T/2.0
      GO TO 240
 273  PRINT 5022
5022  FORMAT ('     MAXIMUM INTERPOLATION')
      CALL REINV
      IF(NEXIT) 280,280,460
 280  SIGMA=0.0
 290  CALL AMOA
      IF(KQ)284,284,283
 283  DO 281 I=1,KQ
      J=IHI(I)
      IF(ABS(AMBOA(J))-EPSI2)281,281,300
 281  CONTINUE
 284  DO 282 I=1,K
      IF(AMBDA(I)+EPSI2)292,282,282
 282  CONTINUE
      GO TO 320
 292  IF(SIGMA-1.0)293,293,301
 293  L=1
      RHO=AMBDA(1)
      IF(K-2)298,299,299
 299  DO 295 I=2,K
      IF(AMBDA(I)-RHO)294,295,295
 294  L=I
      RHO=AMBDA(I)
 295  CONTINUE
 298  IH=L
      CALL MATCOM(1)
      GO TO(300,300,296),NEXIT
 296  NETA=0
      INV=1
      GO TO 310
 300  IF(SIGMA)301,310,301
 301  PRINT 5021
5021  FORMAT('    CONSTRAINT VIOLATION')
      GO TO 120
 310  SIGMA=SIGMA+1.0
      PRINT 5032
5032  FORMAT('    X-CORRECTION')
      DO 311 J=1,M
 311  Y(J)=X(J)
      CALL MATCOM(3)
      DO 312 J=1,M
 312  X(J)=Y(J)
      INV=1
      GO TO 290
 320  IF(SIGMA)321,325,321
 321  CALL FUNCT(X,F,G,KQ)
      IF(F-FY)322,325,325
 322  PRINT 5013
5013  FORMAT('    F  DECREASE')
      CALL REINV
      IF(NEXIT)325,325,460
 325  IF(KOUNT-MXNU)327,326,326
 326  PRINT 5014
```

```
5014 FORMAT('   MAXIMUM STEPS TAKEN')
     GO TO 460
 327 PRINT 5015,(X(J),J=1,M)
5015 FORMAT(6F12.6)
     GO TO 140
C
C         THREE POINT EXTRAPOLATION SCHEME
C
 330 NETA=0
     DELXN=0.0
     DO 331 J=1,M
     DELX(J)=X(J)-XO(J)
 331 DELXN=DELXN+DELX(J)**2
     DELXN=SQRT(DELXN)
     IF(DELXN-TMIN)332,332,333
 332 NETA=1
     T=TMAX
     GO TO 230
 333 ZG=0.0
     ZGO=0.0
     DO 334 J=1,M
     Z(J)=DELX(J)/DELXN
     ZG=ZG+Z(J)*G(J)
 334 ZGO=ZGO+Z(J)*GO(J)
     IF(ZGO)335,335,338
 338 IF(ZGO-ZG)335,335,336
 335 TPRIME=TMAX
     GO TO 350
 336 TPRIME=ZG*DELXN/(ZGO-ZG)
     IF(TPRIME)340,350,350
 340 T=-TPRIME
     MU=1
     XMU1=-ZGO
     XMUO=-ZG
     DO 341 J=1,M
     Z(J)=-Z(J)
     X1(J)=XO(J)
 341 X2(J)=X(J)
      PRINT 5016,T,ZG,(Z(J),J=1,M)
5016 FORMAT('   3 PT. INTERPOLATION'5X2HT=F12.6,
    1 5X3HZG=F12.3/'      Z-VECTOR'/(6F12.6))
     GO TO 351
 350 T=TPRIME
     PRINT 5017,T,ZG,(Z(J),J=1,M)
5017 FORMAT('  3 PT. EXTRAPOLATION',5X2HT=F12.6,
    1 5X3HZG=F12.3/'      Z-VECTOR'/(6F12.6))
 351 IF(T-TMAX)353,353,352
 352 T=TMAX
 353 IF(KV)230,230,354
 354 LLL=230
     GO TO 190
 420 PRINT 5018 ,F
5018 FORMAT('  MAXIMUM F',F12.6)
 460 CALL CLASS
 470 PRINT 5019
5019 FORMAT ('  J     X VECTOR     GRADIENT')
     DO 5678 J=1,M
5678 PRINT 5020,J,X(J),G(J)
5020 FORMAT(I3,2F12.6)
     K1=1
```

```
      CALL FUNCT(X,F,G,KQ)
 9998 CONTINUE
      GO TO 480
 9999 PRINT 5031
 5031 FORMAT ('      ERROR')
  480 CONTINUE
      STOP
      END
```

```
      SUBROUTINE REINV
C
C        THIS SUBROUTINE CARRIES OUT REINVERSION CALCULATIONS NECESSARY
C        FOR THE GP ALGORITHM
C
      DIMENSION X2(10),D(10,10),DN(10,10),A(25,10),G(10),X(10),P(10),
     1IU(10),IW(25),B(25),SD(25),RR(10),Z(10),X0(10),G0(10),X1(10),
     1V(10),JX2(20),IHI(10),R(10),AMBDA(25),Y(10),PG(10),IV(10),
     1DELX(10)
      COMMON Y,PN,AMBDA,M,G,X,F,P,EPSI1,NEXIT,KQ,D,IH,NB,V,A,JXB,IHI,
     I EPSI3,EPSI2,LD,IU,R,KEQ,MEXIT,IV,KV,B,KMNB,IW,KW,K,MXRN,INV,DN,
     1MU,NETA,INT,LOC
      IF(MXRN)10,10,20
   10 NEXIT=1
      RETURN
   20 NEXIT=0
      MXRN=MXRN-1
      IF(MXRN)30,30,40
   30 IF(KEQ)36,36,31
   31 DO 35 I=1,KEQ
      DO 35 J=1,KEQ
   35 D(I,J)=DN(I,J)
   36 MU=0
      NETA=0
      INT=0
      LOC=0
      KQ=KEQ
      PRINT 935
  935 FORMAT('       NEW BASIS')
      INV=0
      RETURN
   40 IF(INV)10,10,50
   50 IF(KQ-KEQ)30,30,51
   51 L=KQ-KEQ
      IF (KEQ) 54,54,52
   52 DO 53 I=1,KEQ
      DO 53 J=I,KEQ
   53 D(I,J)=DN(I,J)
   54 MU=0
      NETA=0
      INT=0
      LOC=0
      KQ=KEQ
      PRINT 935
      INV=0
   55 CONTINUE
   60 IH=IHI(KQ+1)
      CALL MATCOM(1)
   70 IF (L-1) 90,90,80
   80 L=L-1
      GO TO 55
   90 NEXIT=0
      RETURN
      END
```

```
      SUBROUTINE MATCOM(NENTER)
C
C         THIS SUBROUTINE CARRIES OUT NECESSARY MATRIX COMPUTATIONS FOR
C         ENTERING A CONSTRAINT INTO THE BASIS
C
      DIMENSION X2(10),D(10,10),ON(10,10),A(25,10),G(10),X(10),P(10),
     1V(10),JXB(20),IHI(10),R(10),AMBDA(25),Y(10),PG(10),IV(10),
     1IU(10),IW(25),B(25),SD(25),RR(10),Z(10),XO(10),GO(10),X1(10),
     1DELX(10)
      COMMON Y,PN,AMBDA,M,G,X,F,P,EPSI1,NEXIT,KQ,D,IH,NB,V,A,JXB,IHI,
     1 EPSI3,EPSI2,LD,IU,R,KEQ,MEXIT,IV,KV,G,KMNB,IW,KW,K,MXRN,INV,ON,
     1MU,META,INT,LDC
      EQUIVALENCE (Y,PG),(PN,PGNORM)
      GO TO (10, 80, 110),NENTER
   10 IFLAG = 1
      IF (KQ) 20,20,30
   20 D(1,1) = 1.0
      PN=1.0
      GO TO 260
   30 IF(IH-NB) 40,40,35
   35 I = IH-NB
      DO 36 J = 1,M
   36 V(J) = A(I,J)
      GO TO 90
   40 DO 50 I = 1,M
   50 V(I) = 0.0
      JK = JXB(IH)
      IF (JK) 60,60,70
   60 JKM = -JK
      V(JKM) = -1.0
      GO TO 90
   70 V(JK) = 1.0
      GO TO 90
   80 IFLAG = 2
   90 IF (KQ) 85,85,88
   85 DO 86 J = 1,M
   86 Y(J) = V(J)
      RETURN
   88 DO 100 I = 1,KQ
      KK = IHI(I)-NB
      IF (KK) 91,91,95
   91 JBD = IHI(I)
      JK = JXB(JBD)
      IF (JK) 92,92,93
   92 JKM = -JK
      P(I) = -V(JKM)
      GO TO 100
   93 P(I) = V(JK)
      GO TO 100
   95 P(I) = 0.0
      DO 99 J = 1,M
   99 P(I) = P(I)+A(KK,J)*V(J)
  100 CONTINUE
      GO TO 130
  110 IFLAG = 2
      DO 120 I = 1,KQ
      KK = IHI(I)
  120 P(I) =  AMBDA(KK)
  130 DO 140 I = 1,KQ
      R(I) = 0.0
```

```
      DO 140 J = 1,KC
  140 R(I) = R(I)+D(I,J)*P(J)
      DO 160 J = 1,M
      Y(J) = 0.0
      DO 150 I = 1,KQ
      KK = IHI(I)-NB
      IF (KK) 151,151,155
  151 JBD = IHI(I)
      JK = JXB(JBD)
      IF (JK) 152,152,153
  152 IF (J+JK) 150,156,150
  156 Y(J) = Y(J)-R(I)
      GO TO 150
  153 IF (J-JK) 150,154,150
  154 Y(J) = Y(J)+R(I)
      GO TO 150
  155 Y(J) = Y(J)+A(KK,J)*R(I)
  150 CONTINUE
  160 Y(J) = V(J)-Y(J)
      GO TO (180,170), IFLAG
  170 NEXIT = 1
      RETURN
  180 PN = 0.0
      DO 190 J = 1,M
  190 PN = PN + Y(J)**2
      YB2 = PN
      PN=SQRT(PN)
      IF (PN-EPSI3) 200,210,210
  200 NEXIT = 2
      LDC = LDC+1
      IU(LDC) = IH
C
C        H(L) LINEARLY DEPENDENT
      PRINT 9200, IH, PN
 9200 FORMAT('     H  ',I2,' LINEARLY DEPENDENT    PN=',F12.6)
      RETURN
  210 J = KC + 1
      I = J
      D(I,J) = 1.0/YB2
  220 IF (I-J) 240,240,230
  230 I = I-1
      D(I,J) = D(I,J)+R(J)*R(I)/YB2
      D(J,I) = D(I,J)
      GO TO 220
  240 IF (J-I) 260,260,250
  250 J = J-1
      I = KC + 1
      D(I,J) = -R(J)/YB2
      D(J,I) = D(I,J)
      GO TO 220
  260 KQ = KC + 1
      IHI(KC) = IH
C
C        H(L) ADDED
C
      PRINT 9260, IH, PN
 9260 FORMAT('     H',2XI2,'     ADDED    PN',F12.6)
      NEXIT = 3
      RETURN
      END
```

```
      SUBROUTINE COMMAT(DEL)
C
C        THIS SUBROUTINE CARRIES OUT NECESSARY MATRIX COMPUTATIONS FOR
C        DROPPING A CONSTRAINT FROM THE BASIS
C
      DIMENSION X2(10),D(10,10),DN(10,10),A(25,10),G(10),X(10),P(10),
     1IU(10),IW(25),B(25),SD(25),RR(10),Z(10),XO(10),GO(10),X1(10),
     1V(10),JXB(20),IHI(10),R(10),AMBDA(25),Y(10),PG(10),IV(10),
     1DELX(10)
      COMMON Y,PN,AMBDA,M,G,X,F,P,EPSI1,NEXIT,KQ,D,IH,NB,V,A,JXB,IHI,
     1 EPSI3,EPSI2,LD,IU,R,KEC,MEXIT,IV,KV,B,KMNB,IW,KW,K,MXRN,INV,DN,
     1MU,NETA,INT,LDC
      EQUIVALENCE (Y,PG),(PN,PGNORM)
      DELTA = DEL
      NZ = 0
      L = KEC + 1
   40 IF (R(L)) 43,43,41
   41 RD = R(L)/SQRT (D(L,L))
      IF (RD-DELTA) 43,42,42
   42 DELTA = RD
      NZ = L
   43 IF (L-KQ) 44,45,45
   44 L = L+1
      GO TO 40
   45 IF (NZ) 46,46,50
   46 MEXIT = 1
      RETURN
   50 DO 60 I = 1,KQ
   60 V(I) = D(I,NZ)
      IH = IHI(NZ)
      NZM = NZ -1
      KQ = KQ - 1
      IF (NZ-KQ) 63,63,62
   62 VNZ = V(NZ)
      GO TO 105
   63 DO 80 I = NZ, KQ
      IHI(I) = IHI(I+1)
      IF (NZM) 75,75,65
   65 DO 70 J = 1,NZM
      D(I,J) = D(I+1,J)
   70 D(J,I) = D(I,J)
   75 DO 80 J = NZ, KQ
   80 D(I,J) = D(I+1,J+1)
      VNZ = V(NZ)
      DO 100 I = NZ,KQ
  100 V(I) = V(I+1)
  105 DO 130 I = 1,KQ
      DO 130 J = 1,KQ
      IF (I-J) 110,110,120
  110 D(I,J) = D(I,J)-V(I)*V(J)/VNZ
      GO TO 130
  120 D(I,J) = D(J,I)
  130 CONTINUE
      IF (LDC) 150,150,140
  140 DO 142 I = 1,LDC
      J = KV + I
  142 IV(J) = IU(I)
      KV = KV + LDC
      LDC = 0
  150 MEXIT = 2
```

```
      SUBROUTINE AMDA
C
C         THIS SUBROUTINE CARRIES OUT CALCULATION OF LAMBDAS
C
      DIMENSION X2(10),D(10,10),DN(10,10),A(25,10),G(10),X(10),P(10),
     1IU(10),IV(25),B(25),SD(25),RR(10),Z(10),X0(10),G0(10),XI(10),
     1V(10),JXB(25),IHI(10),R(10),AMBDA(25),Y(10),PG(10),IV(10),
     1DELX(10)
      COMMON Y,PN,AMBDA,M,G,X,F,P,EPSI1,NEXIT,KQ,D,IH,NB,V,A,JXB,IH1,
     1 EPSI3,EPSI2,LD,IU,R,KEQ,MEXIT,IV,KV,B,KMNB,IW,KW,K,MXRN,INV,DN,
     1MU,NETA,INT,LDC
      IF(NB .EQ. 0)GO TO 40
   10 DO 30 I=1,NB
      J=JXB(I)
      IF(J .LT. 0)GO TO 20
      AMBDA(I)=X(J)-B(I)
      GO TO 30
   20 AMBDA(I)=-X(-J)-B(I)
   30 CONTINUE
   40 IF(KMNB)80,80,50
   50 DO 70 I=1,KMNB
      TOTA=0.
      KK=NB+I
      DO 60 J=1,M
   60 TOTA=TOTA+A(I,J)*X(J)
   70 AMBDA(KK)=TOTA-B(KK)
   80 RETURN
      END
```

```
C
C       THIS SUBROUTINE CLASSIFIES THE CONSTRAINTS
C
      DIMENSION X2(10),D(10,10),DN(10,10),A(25,10),G(10),X(10),P(10),
     1IU(10),IW(25),B(25),SD(25),RR(10),Z(10),X0(10),G0(10),X1(10),
     1V(10),JXB(20),IHI(10),R(10),AMBDA(25),Y(10),PG(10),IV(10),
     1DELX(10)
      COMMON Y,PN,AMBDA,M,G,X,F,P,EPSI1,NEXIT,KO,D,IH,NB,V,A,JXB,IHI,
     1 EPSI3,EPSI2,LD,IU,R,KEC,MEXIT,IV,KV,B,KMNB,IW,KW,K,MXRN,INV,DN,
     1MU,NETA,INT,LCC
      KW=0
      KV=0
      DO 60 I=1,K
      IF(KO .EQ. 0)GO TO 30
   10 DO 20 J=1,KO
      IF(I-IHI(J))20,60,20
   20 CONTINUE
   30 IF(ABS(AMBDA(I))-EPSI2)40,40,50
   40 KV=KV+1
      IV(KV)=I
      GO TO 60
   50 KW=KW+1
      IW(KW)=I
   60 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FUNCT(X,F,G,KQ)
C
C        WATER QUALITY MANAGEMENT MODEL
C
C
C        THIS SUBROUTINE CALCULATES THE FUNCTION VALUE AND THE GRADIENT
C
      COMMON/WATER/NFUNC,KOUNT,K1
      DIMENSION X(10),G(10),C1(10),C2(10),C3(10),C4(10),D1(10),D2(10),B1
     1(10),B2(10),C11(10),X3(1 )
  100 FORMAT('      ITERATION # = ',I3,'      # OF FUNCTIONAL EVALUATIONS
     1= ',I3,'      # OF CONSTRAINTS IN BASIS = ',I3,/)
  101 FORMAT(6F12.6/'      F-VALUE',F12.6)
  102 FORMAT (1H1,'     COST COEFFICIENTS FOR PLANTS')
  103 FORMAT (1H1,'                   **COST,BOD AND DO INFORMATION**')
  104 FORMAT ('  REACH #    TREATMENT       COST        MAX.BOD    MINIMUM DO
     1     ALLO. DO')
  105 FORMAT(7XI2,5F12.6)
  106 FORMAT ('      MINIMUM TOTAL COST',F12.6)
 1000 FORMAT(3F15.7)
 1001 FORMAT(I4,3F15.7)
 1002 FORMAT(7F8.3)
      M=6
      IF(NFUNC)10,10,20
C
C        READ AND PRINT COST COEFFICIENTS
C
   10 READ 1000,(C1(I),C2(I),C3(I),I=1,M)
      PRINT 102
      DO 11 I=1,M
   11 PRINT 1001,I,C1(I),C2(I),C3(I)
   20 IF(K1)30,30,22
   30 NFUNC=NFUNC+1
C
C        CALCULATE THE FUNCTION VALUE AND THE GRADIENT VECTOR
C
      F=0.0
      DO 21 I=1,M
      F=F-C2(I)*X(I)-C3(I)*X(I)**2
   21 G(I)=-C2(I)-C3(I)*X(I)*2.
      PRINT 100,KOUNT,NFUNC,KQ
      PRINT 101,(X(I),I=1,M),F
      GO TO 40
C
C        READ BOD AND DO INFORMATION
C
   22 READ 1002,(B1(I),I=1,7)
      READ 1002,(D2(I),I=1,7)
C
C        STORE X-VECTOR
C
      DO 23 I=1,4
   23 X3(I)=X(I)
      X3(5)=0.0
      X3(6)=X(5)
      X3(7)=X(6)
C
C        CALCULATION OF MAXIMUM BOD
C
      DO 24 I=1,7
```

```
   24 D2(I)=(1.-X3(I))*D1(I)
C
C        CALCULATE ACTUAL [illegible]
C
      D1(1)=9.464529+0.056879*X(1)
      D1(2)=5.622 27+3.177510*X(2)
      D1(3)=7.[illegible]401+.112564*X(1)+1.533369*X(2)+0.163363*X(3)
      D1(4)=-21.314639+30.367590*X(4)
      D1(5)=4.501614+0.103951*X(1)+1.277556*X(2)+.155779*X(3)+2.788927*X
     1(4)
      D1(6)=-1.988039+0.154961*X(1)+1.594329*X(2)+.243230*X(3)+3.214846*
     1X(4)+5.324249*X(5)
      D1(7)=-16.716300+0.258366*X(1)+2.326023*X(2)+.417329*X(3)+4.356409
     1*X(4)+14.580280*X(5)+2.895020*X(6)
C
C        CALCULATION OF INDIVIDUAL PLANT COSTS
C
      DO 27 I=1,4
   27 C11(I)=C1(I)+C2(I)*X(I)+C3(I)*X(I)**2
      C11(5)=0.0
      DO 28 I=5,6
   28 C11(I+1)=C1(I)+C2(I)*X(I)+C3(I)*X(I)**2
C
C        CALCULATION OF TOTAL COST
C
   25 FA=0.0
      DO 26 I=1,7
   26 FA=FA+C11(I)
C        OUTPUT COST,EDD,DO INFORMATION
      PRINT 103
      PRINT 104
      DO 31 I=1,7
   31 PRINT 105,I,X3(I),C11(I),B2(I),D1(I),D2(I)
      PRINT 106,FA
   40 RETURN
      END
```

```
      SUBROUTINE FUNCT(X,F,G,KQ)
C
C         TWC PERIOD PROBLEM FOR WATER RESOURCES
C
C
C         THIS SUBROUTINE CALCULATES THE FUNCTION VALUE AND THE GRADIENT
C
      DIMENSION X(20),G(20)
      COMMON/WATER/NFUNC,KOUNT,K1
  100 FORMAT('      ITERATION # = ',I3,'      # OF FUNCTIONAL EVALUATIONS
     1= ',I3,'      # OF CONSTRAINTS IN BASIS = ',I3,/)
  101 FORMAT(5F12.6/'      F-VALUE',F12.6)
 1001 FORMAT(1H-,'      BENEFIT FROM ENERGY      ',F12.6)
 1002 FORMAT(1H-,'      BENEFIT FROM IRRIGATION',F12.6)
 1003 FORMAT(1H-,'      COST OF RESERVOIR B      ',F12.6)
 1004 FORMAT(1H-,'      COST OF RESERVOIR C      ',F12.6)
 1010 FORMAT(1H-,'      MAXIMUM NET BENEFITS     ',F12.6)
      IF(K1)10,10,20
   10 NFUNC=NFUNC+1
      M=5
C
C         CALCULATE THE FUNCTION VALUE AND THE GRADIENT VECTOR
C
      F=(229.4*X(1)+X(1)**2)+(1.4*X(2))-(48.7*X(3))+(453.7*ALOG(1.+0.2*(
     1X(2)+X(3))))-(43.0*X(4)/(1.+0.2*X(4)))-(47.*X(5)/(1.+0.3*X(5)))-4.
     15
      G(1)=2.*X(1)+229.4
      G(2)=1.4+(453.7*0.2/(1.+0.2*(X(2)+X(3))))
      G(3)=-48.7+(453.7*0.2/(1.+0.2*(X(2)+X(3))))
      G(4)=-43.*(1./(1.+.2*X(4)))+43.*(.2*X(4)/((1.+.2*X(4))**2))
      G(5)=-47.*(1./(1.+.3*X(5)))+47.*(.3*X(5)/((1.+.3*X(5))**2))
      PRINT 100,KOUNT,NFUNC,KQ
      PRINT 101,(X(I),I=1,M),F
      GO TO 40
C
C         CALULATE BENEFIT AND COST FUNCTIONS
C
   20 X(6)=229.4*X(1)+X(1)**2
      X(7)=1.4*X(2)-48.7*X(3)+453.7*ALOG(1.+0.2*(X(2)+X(3)))-4.5
      X(8)=43.*X(4)/(1.+.2*X(4))
      X(9)=47.*X(5)/(1.+.3*X(5))
      PRINT 1001,X(6)
      PRINT 1002,X(7)
      PRINT 1003,X(8)
      PRINT 1004,X(9)
      PRINT 1010,F
   40 RETURN
      END
```

```
      SUBROUTINE FUNCT(X,F,G,KQ)
C
C        FOUR PERIOD PROBLEM FOR WATER RESOURCES
C
C
C        THIS SUBROUTINE CALCULATES THE FUNCTION VALUE AND THE GRADIENT
C
      DIMENSION X(20),G(20),X11(20),F1(10),F2(10)
      COMMON/WATER/NFUNC,KOUNT,K1
  100 FORMAT('     ITERATION # = ',I3,'     # OF FUNCTIONAL EVALUATIONS
     1= ',I3,'     # OF CONSTRAINTS IN BASIS = ',I3,/)
  101 FORMAT(6F10.3/'      F-VALUE',F12.6)
 1001 FORMAT(4F8.3)
 1002 FORMAT(1H-,'     PERIODS     1          2          3          4')
 1003 FORMAT(1H-,'    STORAGES',4F8.3)
 1004 FORMAT(1H-,'    OUTFLOWS',4F8.3)
 1005 FORMAT(1H-,'      ENERGY',4F8.3)
 1006 FORMAT(1H-,'    TOTAL ENERGY',F8.3)
 1007 FORMAT(1H-,'    CAPACITY OF RESERVOIR',F8.3)
 1008 FORMAT(1H-,'    BENEFIT FROM ENERGY',F8.3)
 1009 FORMAT(1H-,'    COST OF RESERVOIR',F8.3)
 1010 FORMAT(1H-,'    MAXIMUM NET BENEFITS',F12.6)
      IF(K1)10,10,20
   10 NFUNC=NFUNC+1
      M=6
C
C        CALCULATE THE FUNCTION VALUE AND THE GRADIENT VECTOR
C
      F=2.294*X(6 )+.01*(X(6 )**2)-.43*X(5)/(1.+0.2*X(5))
      G(1)=0.0
      G(2)=0.0
      G(3)=0.0
      G(4)=0.0
      G(5)=-.43*(1./(1.+.2*X(5)))+.43*(.2*X(5)/((1.+.2*X(5))**2))
      G(6)=2.294+.02*X(6)
      PRINT 100,KOUNT,NFUNC,KQ
      PRINT 101,(X(I),I=1,M),F
      GO TO 40
C
C        READ INFLOWS TO THE RESERVOIR
C
   20 READ 1001,(F1(I),I=1,4)
C
C        CALCULATE OUTFLOWS FROM THE RESERVOIR
C
      DO 21 I=1,4
   21 X11(I)=X(I)
      DO 22 I=1,3
   22 X11(I+4)=X11(I)+F1(I)-X11(I+1)
      X11(8)=X11(4)+F1(4)-X11(1)
      X11(9)=X(5)
      X11(10)=X(6)
C
C        READ ENERGY COEFFICIENTS
C
      READ 1001,(F2(I),I=1,4)
      DO 31 I=11,14
      X11(I)=F2(I-10)*X11(10)
   31 CONTINUE
C
```

```
C         CALULATE BENEFIT AND COST FUNCTIONS
C
      X11(15)=2.294*X11(10)+.01*X11(10)**2
      X11(16)=.43*X11(9)/(1.+0.2*X11(9))
      PRINT 1002
      PRINT 1003,(X11(I),I=1,4)
      PRINT 1004,(X11(I),I=5,8)
      PRINT 1005,(X11(I),I=11,14)
      PRINT 1006,X11(10)
      PRINT 1007,X11(9)
      PRINT 1008,X11(15)
      PRINT 1009,X11(16)
      PRINT 1010,F
   40 RETURN
      END
```

OPTIMIZATION
OF
WATER RESOURCES

by

PRATAP DHANJI DAND

B.Tech., Indian Institute of Technology
Bombay, India, 1967

AN ABSTRACT OF A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Deaprtment of Industrial Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1969

In recent years attempts have been made to apply the various mathematical programming techniques to the planning, design and operation of water resource systems. Two general types of models have been fruitful in the field of water resource development: the simulation model and analytic model. In this thesis, two deterministic models-one connected with water quality and other connected with water quantity-are proposed and solved within the framework of an analytical approach.

For each system that is considered, the procedure involves the understanding of the basic physical systems; the development of systems equations or mathematical models and the solution of the proposed models. Data used in various models was drawn from the literature wherever possible. This practice was adopted to insure the realistic response behavior.

Like most of the models used to describe the water resource systems, the models presented here are characterized by a nonlinear objective function and linear constraints.

Rosen's gradient projection method appears to be a powerful tool for this class of nonlinear programming problems characterized by linear constraints. The specific purpose of this thesis is to apply this technique to the various water resource models and analyze the results to derive optimal design and operation policies.

The method is initially described in some detail with an emphasis on computational aspects. Then the method is successfully applied to solve the various models describing water resource systems.

The construction and solution of mathematical model of dissolved oxygen for a simplified river basin indicates how a mathematical model can generate useful water quality control information.

Next a simplified river basin configuration is selected to illustrate the mathematical programming approach to water quantity aspects. Such an approach is useful in initial exploratory stages of water resources planning.